

University of Nottingham
UK | CHINA | MALAYSIA

NUMERICAL AND EXPERIMENTAL INVESTIGATION OF TSUNAMIS GENERATED BY ICEBERG CALVING

FAN CHEN

STUDENT ID: 14268871

Thesis submitted to the University of Nottingham
for the degree of Doctor of Philosophy

January 2021

Abstract

With climate change as an increasingly important issue, the Greenland and Antarctic ice sheets have suffered rapid ice mass losses contributing to sea level rise. Iceberg calving is one of the main factors for ice mass loss and may also generate large tsunamis when icebergs calve into a water body, threatening the fishing and shipping industries and coastal communities. One of the most impressive tsunamis generated by iceberg calving with an amplitude of 45 to 50 m was observed at the Eqip Sermia glacier in Greenland in 2014. Herein, such tsunamis are called iceberg-tsunamis (IBTs).

This study aims to investigate the generation and propagation of large IBTs. A novel numerical model is developed based on the Immersed Boundary Method (IBM) provided in a toolbox in the open source code Foam-extend 4.0. A new multiphase flow solver is implemented to solve the Reynolds-Averaged Navier-Stokes equations with the support of handling moving immersed boundaries, and it is then coupled with a motion solver to simulate Fluid-Structure Interaction (FSI) to determine the iceberg motion. Further, unique large-scale laboratory experiments were conducted in a 50 m $\times$ 50 m large basin. Two rigid blocks with densities of $\approx$920 kg/m^3 weighting up to 187 kg were used to mimic icebergs and to model the five idealised calving mechanisms: (A) capsizing, (B) gravity-dominated fall, (C) buoyancy-dominated fall, (D) gravity-dominated overturning and (E) buoyancy-dominated overturning.

The analytical solution of a floating heaving sphere case is used to validate the newly implemented flow solver and the numerical radiated wave amplitudes show a maximum deviation of only 10.3% from the corresponding analytical solution. Further, the numerical model is used to simulate the two laboratory experiments from mechanisms B and D generating the largest and most dangerous IBTs. A laminar flow model is selected as the consideration of turbulence does not improve the accuracy. For IBT generation, the maximum vertical displacement in the fall case and the pitch angle in the overturning case are 24.3 and 19.9% larger than the corresponding laboratory measurements. During this stage of the IBT, the pressure force is the dominant force while the viscosity force can be neglected in the overall force balance. For IBT propagation, the leading

IBT amplitude and wave height decay are captured with a maximum relative error of -41.2 and -29.3%, respectively. This large deviation is mainly due to the water splash involved in the laboratory experiments, which can not fully be modelled in the numerical simulations. If this splash is artificially removed, then the corresponding relative errors become -17.2 and -9.2%, respectively. Additionally, the numerical IBT train energy shows a good agreement with the laboratory results with a relative error of 1.0% in mechanism B, while the difference in mechanism D is much larger due to the oscillation of the iceberg.

Further, the capability of this numerical model of simulating FSI is confirmed by the agreement of the contact forces of the icebergs acting on the glacier terminus with that in other studies. Smaller and less important IBTs generated in the remaining mechanisms A, C and E are also simulated. However, due to the nature of the IBM making it challenging to fully satisfy the continuity equation and the no-slip boundary condition around the immersed boundary when multiphase flows are involved, more work is required to also simulate these small waves, in the order of 10^{-3} m at laboratory scale, well. Nevertheless, the numerical model is capable of simulating large and dangerous IBTs as further demonstrated by the successfully reproduced 2014 Eqip Sermia case. Overall, this work demonstrates that the proposed numerical model has the potential to support IBT hazard assessment. Future work will focus on the improvement of the immersed boundary representation technique to numerically investigate IBTs involving additional and more complex calving mechanisms and scenarios.

Acknowledgements

Firstly, I would like to express my gratitude to my supervisors Dr Valentin Heller and Dr Riccardo Briganti for their patient guidance, beneficial comments and suggestions and engagement through the learning process of my PhD study. I really appreciate their invested time in teaching me the basic skills of doing research.

I would like to thank my external assessor Prof Nils Goseberg from the Technical University of Braunschweig for his time and useful comments. I would also like to thank my internal assessor Dr David Hargreaves for useful comments and suggestions for my PhD study. Finally, I would like to acknowledge Dr Niels Gjøl Jacobsen for recommending to use the Immersed Boundary Method.

I felt lucky and honoured to be invited as a team member for the large-scale measurement campaign in Deltares, which was supported by the European Community's Horizon 2020 Research and Innovation Programme through the grant to HYDRALAB+, Contract no. 654110.

Most of the numerical simulations in this study were conducted on Athena at HPC Midlands+, which was funded by EPSRC on grant EP/P020232/1, as part of the HPC Midlands+ consortium. There were also some numerical simulations running on the University of Nottingham HPC clusters Minerva and Augusta.

China Scholarship Council (CSC) is acknowledged for the financial support for my PhD study.

Finally, I would like to express my deepest gratitude to my parents for their support and my wife Wenshi for her support and encouragement throughout my PhD study.

Table of Content

1 INTRODUCTION **1**

1.1 Overview 1

1.2 Aims and Objectives 5

1.3 Thesis Outline 5

2 LITERATURE REVIEW **7**

2.1 Introduction 7

2.2 Iceberg Calving into a Water Body 7

2.2.1 Iceberg calving mechanisms 7

2.2.1.1 Fall 10

2.2.1.2 Overturning 11

2.2.1.3 Capsizing 12

2.2.2 Field observations of IBTs 13

2.2.3 Laboratory investigations of IBTs 17

2.2.4 Numerical simulations of IBTs 19

2.2.5 Analytical modelling of IBTs 21

2.3 Wave Theories 23

2.3.1 Linear wave theory 23

2.3.2 Nonlinear wave theories 25

2.4 Landslide-tsunamis 25

2.4.1 Introduction 25

2.4.2 Landslide-tsunami generation 26

2.4.2.1 Subaerial landslides 26
2.4.2.2 Effect of slide types 28
2.4.3 Landslide-tsunami propagation 30
2.4.3.1 Wave types 30
2.4.3.2 Effect of the water body geometry 32
2.4.3.3 Effect of the bathymetry 35
2.5 Numerical Methods to Model Floating Bodies 37
2.5.1 Introduction 37
2.5.2 Methods to simulate Fluid-Structure Interaction 38
2.5.2.1 Arbitrary Lagrangian Eulerian (ALE) 38
2.5.2.2 Immersed Boundary Method (IBM) 39
2.5.2.3 Meshless methods 40
2.5.2.4 Lattice-Boltzmann Method (LBM) 42
2.5.3 Numerical methods to model IBTs 43
2.5.3.1 Direct Numerical Simulation (DNS) 44
2.5.3.2 Large Eddy Simulation (LES) 44
2.5.3.3 Reynolds-Averaged Navier-Stokes (RANS) equations 45
2.5.3.4 Higher-order Boussinesq equations 46
2.5.3.5 Selection of numerical methods to model IBTs . 47
2.5.4 OpenFOAM 48
2.6 Summary 49

3 NUMERICAL MODEL **51**

3.1 Introduction 51

3.2 IBM Toolbox in Foam-extend 52
3.2.1 Introduction 52
3.2.2 IB boundary type 52
3.2.3 Moving IB support 54
3.2.4 IB force calculation 55
3.2.5 IB wall functions 56
3.3 Flow Solver 56
3.3.1 Introduction 56
3.3.2 Governing equations 57
3.3.3 Solving process 58
3.3.3.1 Equation discretisation 58
3.3.3.2 Solving algorithm 61
3.3.4 Solver structure 62
3.4 Motion Solver 64
3.4.1 Introduction 64
3.4.2 Motion equations 65
3.4.3 Motion state update 66
3.4.3.1 Position update 66
3.4.3.2 Force update 68
3.5 Coupling Method 69
3.5.1 Introduction 69
3.5.2 Force calculation 69
3.5.3 IB movement 71
3.6 Numerical Model Workflow 73

3.7 Summary 75

4 LABORATORY MODEL 77

4.1 Introduction 77

4.2 Experimental Setup 77

4.2.1 Coordinate systems 78

4.2.2 Iceberg model and support frame 78

4.2.3 Triggering methods 80

4.2.4 Experimental parameters 82

4.3 Data Acquisition and Processing 82

4.3.1 Motion sensor 83

4.3.2 Wave probes 85

4.3.3 Instrument synchronisation 87

4.3.4 Post-processing 87

4.4 Model Similarity 89

4.5 Summary 92

5 RESULTS 93

5.1 Introduction 93

5.2 Validation with a Theoretical Heaving Sphere Case 93

5.2.1 Mathematical description of the heaving sphere case . . . 94

5.2.2 Numerical setup and results 95

5.3 Simulation of Laboratory IBTs 98

5.3.1 Overview of the laboratory experiments 98

5.3.2 Convergence tests 99

5.3.3 Numerical setup . . . 102
5.3.4 IBT generation . . . 104
5.3.4.1 Numerically reproduced iceberg motion . . . 106
5.3.4.2 Forces acting on the iceberg . . . 108
5.3.5 IBT propagation . . . 109
5.3.5.1 IBT wave parameters . . . 109
5.3.5.2 Wave train energy . . . 115
5.3.5.3 Wave decay . . . 116
5.4 Summary . . . 119

6 DISCUSSION OF THE RESULTS 121

6.1 Introduction . . . 121
6.2 Comparison of the Contact Force with Other Studies . . . 122
6.3 Effect of Turbulence Modelling . . . 123
6.4 Simulation of the Remaining Mechanisms . . . 126
6.5 Simulation of the 2014 Eqip Sermia Case . . . 131
6.5.1 Numerical setup . . . 132
6.5.2 Results . . . 135
6.6 Comparison with Other CFD Codes . . . 137
6.7 Summary . . . 140

7 CONCLUSIONS AND OUTLOOK 141

7.1 Conclusions . . . 141
7.2 Limitations and Outlook . . . 143

References 145

A Implementation of the new flow solver **165**

B Trajectory inference using the motion sensor **169**

C Implementation of the solution of the theoretical floating heaving sphere case **171**

D Turbulence model in the Foam-extend IBM toolkit **173**

Nomenclature

$\mathbf{a}$	acceleration vector	[m·s^{-2}]
$\mathbf{a}_a$	global acceleration vector	[m·s^{-2}]
$\mathbf{a}_l$	local acceleration vector	[m·s^{-2}]
a	wave amplitude	[m]
a_b	transformed wave amplitude after a bathymetry feature	[m]
a_M	maximum iceberg-tsunami amplitude	[m]
a_N	coefficients regarding the cell N	[-]
a_P	coefficients regarding the cell P	[-]
a_r	sphere radius	[m]
a_0	incident wave amplitude	[m]
a_1	first wave amplitude	[m]
A	oscillation amplitude; flow law parameter	[m; -]
A_b	main cross section perpendicular to the direction of velocity	[m^2]
b	iceberg or slide width; channel width	[m]
b_s	block width	[m]
b_1	distance between the two wavefronts passing location 1	[m]
b_2	distance between the two wavefronts passing location 2	[m]
b'	source width at the coupling location	[m]
B	relative iceberg or slide thickness	[m]
c	wave celerity; wave decay exponent	[m·s^{-1}; -]
c_{i}	arc length of the i$^{\mathrm{th}}$ sector	[m]
c_1	the first wave celerity	[m·s^{-1}]
C	complex constant	[-]
C_o	Courant-Friedrichs-Lewy number	[-]
C_d	drag force coefficient	[-]
C_m	added mass coefficient	[-]
C_μ, $C_{1\epsilon}$, $C_{2\epsilon}$	constants in the turbulence model	[-]
C_0, C_1, C_2, C_3, C_4	coefficients in a polynomial fitting equation	[-]

$\mathbf{d}$	position vector pointing from the point P to point N	[m]
$\mathbf{d}_{CoR}$	vector pointing from the immersed boundary cell to the centre of rotation	[m]
d	fracture depth	[m]
d_c	constant related to iceberg ductility	[-]
d_{i}	complex constant series	[-]
D	iceberg or slide density relative to the water density	[-]
e	mathematical constant; $e = 2.71828$	[-]
E_r	relative released energy	[-]
E_w	IBT train energy	[kg·m^2·s^{-2}]
$\overline{E_k}$	mean wave kinetic energy per surface area over a wave period	[kg·s^{-2}]
$\overline{E_p}$	mean wave potential energy per surface area over a wave period	[kg·s^{-2}]
$\overline{E_w}$	mean wave energy per surface area over a wave period	[kg·s^{-2}]
$\mathbf{F}$	total force vector acting on the iceberg	[kg·m·s^{-2}]
$\mathbf{F}_c$	contact force	[kg·m·s^{-2}]
$\mathbf{F}_d$	drag force vector	[kg·m·s^{-2}]
$\mathbf{F}_p$	pressure force vector	[kg·m·s^{-2}]
$\mathbf{F}_v$	viscosity force vector	[kg·m·s^{-2}]
F_x, F_y, F_z	force component along x-, y- and z-axis	[kg·m·s^{-2}]
Fr	Froude number	[-]
$\mathbf{g}$	gravitational acceleration vector	[m·s^{-2}]
g	gravitational acceleration	[m·s^{-2}]
$\mathbf{G}$	gravity force vector	[kg·m·s^{-2}]
h	water depth	[m]
h_f	water depth at a bathymetry feature	[m]
h_0	initial water depth	[m]
h_1	wave height at the location 1	[m]
h_2	wave height at the location 2	[m]
H	wave height; coefficients	[m; -]
H_b	wave height of the transformed downwave over a bathymetry feature	[m]

H_0	incident wave height	[m]
H_1	first wave height	[m]
i	imaginary unit	[-]
$\mathbf{I}$	moment of inertia	[kg·m^2]
I	integral of Legendre polynomials	[-]
I_x, I_y, I_z	component of moment of inertia along x-, y- and z-axis	[kg·m^2]
J	polynomials	[-]
k	turbulent kinetic energy; permeability parameter; wave number	[m^2·s^{-2}; -; m^{-1}]
K_b	wave height transformation coefficient	[-]
$K_{a,b}$	wave amplitude transformation coefficient	[-]
l	block or iceberg length	[m]
l_w	wavefront length	[m]
l_0	glacier front or iceberg shelf elevation	[m]
$\mathbf{L}$	angular momentum vector	[kg·m^2·s^{-1}]
L'	local angular momentum vector	[kg·m^2·s^{-1}]
L	wave length	[m]
L_x, L_y, L_z	angular momentum component along x-, y- and z-axis	[kg·m^2·s^{-1}]
L_1	the first wave length	[m]
m_a	added mass	[kg]
m_s	iceberg mass	[kg]
$\mathbf{M}$	total torque in relation to the centre of rotation	[kg·m^2·s^{-2}]
M'	local total torque in relation to the centre of rotation	[kg·m^2·s^{-2}]
M_{ij}	complex constant series	[-]
$\mathbf{n}_{IB}$	normal vector at the IB point	[-]
n	flow law parameter	[-]
n	number of waves	[-]
N	truncated number of infinite linear system of equations	[-]
N_τ	truncated number of infinite integral upper bound	[-]
p	fluid pressure; wave parameter	[kg·m^{-1}·s^{-2}; -]

p^*	corrected pressure	[kg·m^{-1}·s^{-2}]
p_i	complex constant series	[-]
p_{IB}	pressure acting on the immersed boundary	[kg·m^{-1}·s^{-2}]
P	impulse product parameter; Legendre polynomial	[-]
$\overline{P_I}$	mean ice pressure	[kg·m^{-1}·s^{-2}]
$\overline{P_W}$	mean horizontal water pressure	[kg·m^{-1}·s^{-2}]
$\mathbf{q}$	quaternion	[-]
$\mathbf{q}_r$	rotation quaternion	[-]
$\mathbf{q}_x$, $\mathbf{q}_y$, $\mathbf{q}_z$	rotation quaternion component relative to the x-, y-, z-axis	[-]
$\mathbf{Q}$	orientation matrix	[-]
r	radial distance	[m]
r'	radial distance from the coupling location	[m]
$\mathbf{R}$	rotation matrix	[-]
R	distance between the motion sensor and the centre of rotation	[m]
R_P	source term	[-]
R_x, R_y, R_z	rotation matrix relative to the x-, y-, z-axis	[-]
R	Reynolds number	[-]
$\mathbf{s}$	septernion	[-]
$\mathbf{s}_c$	septernion responsible for rotation	[-]
$\mathbf{s}_r$	septernion responsible for rotation and translation	[-]
s	block thickness; scalar	[m; -]
$\mathbf{S}$	surface vector	[m^2]
$\mathbf{S}_{ib}$	vector of the immersed boundary cell area	[m^2]
S	relative iceberg thickness	[-]
S_P	source term at the cell P	[-]
S_ϕ	source term	[-]
t	time after start of block movement; moment in time	[s]
t_a	time relative to measurement system a	[s]
t_b	time relative to measurement system b	[s]
T	wave period	[s]
T_s	relative time of slide motion	[-]

T_1	the first wave period	[s]
$\mathbf{u}$	fluid velocity vector	[m·s^{-1}]
$\mathbf{u}_N$	velocity of the cell N	[m·s^{-1}]
$\mathbf{u}_P$	velocity of the cell P	[m·s^{-1}]
$\mathbf{u}^*$	corrected velocity	[m·s^{-1}]
$\mathbf{u}^{**}$	final velocity at each time step	[m·s^{-1}]
u_x, u_y, u_z	fluid velocity component along x-, y-, z-axis	[m·s^{-1}]
U	Ursell number	[-]
$\mathbf{v}$	vector	[-]
$\mathbf{v}_{CoM}$	velocity vector of the centre of mass of the iceberg	[m·s^{-1}]
$\mathbf{V}$	velocity vector of the centre of mass of a rigid body	[m·s^{-1}]
V	relative iceberg or slide volume	[-]
V_P	control volume	[m^3]
V_s	iceberg or slide impact velocity	[m·s^{-1}]
V_x, V_y, V_z	velocity component along x-, y-, z-axis	[m·s^{-1}]
$\forall_s$	block, (calving) iceberg or slide volume	[m^3]
W	Weber number	[-]
x	horizontal coordinate	[m]
x_{IB}	horizontal coordinate of the immersed boundary point	[m]
x_P	horizontal coordinate of the centre of cell P	[m]
x'	local horizontal coordinate	[m]
x'_P	local horizontal coordinate of the centre of cell P	[m]
$\mathbf{X}$	displacement vector of the centre of mass of a rigid body	[m]
$\mathbf{X}_{CoM}$	position vector of the centre of mass of the iceberg	[m]
$\mathbf{X}_{CoR}$	position vector of the centre of rotation of the iceberg	[m]
y	coordinate orthogonal to object plane	[m]
y_{IB}	vertical coordinate of the immersed boundary point	[m]
y_P	vertical coordinate of the centre of cell P	[m]

y'	local vertical coordinate	[m]
y'_P	local vertical coordinate of the centre of cell P	[m]
y^+	dimensionless wall distance	[-]
z	vertical coordinate	[m]
$z^{'}$	local vertical coordinate	[m]
α	phase fraction; slope angle	[°; -]
β'	pre-factor	[-]
Δa_M	measurement uncertainty of maximum wave amplitude	[m]
Δb	measurement uncertainty of block width	[m]
ΔE	the difference of the work required by lifting the iceberg before and after calving to the same reference level	[kg·m^2·s^{-2}]
Δh	measurement uncertainty of water depth	[m]
ΔH_M	measurement uncertainty of maximum wave height	[m]
Δl	measurement uncertainty of block length	[m]
Δm_s	measurement uncertainty of block mass	[kg]
Δs	measurement uncertainty of block thickness	[m]
Δt	time step	[s]
ΔT_M	measurement uncertainty of maximum wave period	[s]
ΔV_s	measurement uncertainty of block velocity	[m·s^{-1}]
Δx, Δy, Δz	cell dimension along x-, y-, z-axis	[m]
$\Delta \rho_w$	measurement uncertainty of water density	[m]
$\Delta\theta_x$, $\Delta\theta_x$, $\Delta\theta_x$	Increment of rotation angle component within a time step along x-, y-, z-axis	[°]
ϵ	dissipation of turbulent kinetic energy; iceberg aspect ratio	[m^2·s^{-3}; -]
ϵ_f	threshold fracture strain rate	[m·s^{-3}]
η	water surface elevation	[m]
γ	wave propagation angle; general variable in a transport equation	[°; -]
γ'	wave propagation angle from the coupling location	[°]

γ_ϕ	diffusion coefficient	[-]
Γ	Gamma function	[-]
λ	length scale; Euler-Mascheroni constant	[-]
μ	dynamic viscosity	[kg·m^{-1}·s^{-1}]
μ_t	turbulent viscosity	[kg·m^{-1}·s^{-1}]
∇	differential operator	[-]
ω	angular velocity	[s^{-1}]
ω_x, ω_y, ω_z	angular velocity component along x-, y-, z-axis	[s^{-1}]
$\boldsymbol{\omega}$	angular velocity vector	[s^{-1}]
$\boldsymbol{\omega}_{CoR}$	angular velocity vector of centre of rotation of the iceberg	[s^{-1}]
ϕ	general variable in a transport equation	[-]
ϕ_i	wave-free potential	[m^2·s^{-1}]
ϕ_{IB}	value at the immersed boundary point	[-]
ϕ_N	value of the cell N	[-]
ϕ_P	value of the (immersed boundary) cell P	[-]
ϕ_0	wave source potential	[m^2·s^{-1}]
ϕ'_P	gradient of the immersed boundary cell P	[-]
Φ	velocity potential	[m^2·s^{-1}]
$\boldsymbol{\phi}$	slide front angle	[°]
π	mathematical constant; $\pi = 3.14159$	[-]
ψ	general variable in a transport equation; azimuthal angle	[-; °]
ρ	fluid density	[kg·m^{-3}]
ρ_s	iceberg density	[kg·m^{-3}]
σ	angular frequency	[s^{-1}]
σ_n	net tensile stress perpendicular to the plane of the crack	[kg·m^{-1}·s^{-2}]
σ_ϵ	constant	[-]
σ_k	constant	[-]
σ_w	water surface tension	[kg·s^{-2}]
τ_{IB}	shear stress	[kg·m^{-1}·s^{-2}]
θ	polar angle	[°]
θ_{max}	maximum rotation angle	[°]
$\boldsymbol{\theta}$	iceberg rotation angle vector	[°]

θ_x, θ_y, θ_z	yaw, roll, pitch angle	$[^\circ]$
$\boldsymbol{\xi}$	angular acceleration vector	$[\mathrm{s}^{-2}]$

Subscripts

a	air; global
CoM	Centre of Mass
CoR	Centre of Rotation
d	drag
f	face; interface
i	indexing number
IB	immersed boundary
j	indexing number
l	laboratory; local
LL	lower-left
LR	lower-right
m	m-order
M	maximum
max	maximum
min	minimum
n	n-order
n	numerical
N	maximum indexing number
N	neighbouring cell
p	pressure
P	cell P
s	slide or iceberg
t	turbulent
UL	upper-left
UR	upper-right
v	viscosity
w	water
x, y, z	x-, y-, z-axis
τ	indexing number
1	first; location 1
2	location 2

Superscript

n	n^{th} moment in time
$\frac{1}{2}$	half time step

Abbreviations

ALE	Arbitrary Lagrangian Eulerian
CFD	Computational Fluid Dynamics
CFL	Courant-Friedrichs-Lewy
CoM	Centre of Mass
CoR	Centre of Rotation
DEM	Discrete Element Method
DNS	Direct Numerical Simulation
DoF	Degree of Freedom
FAVOR	Fractional Area-Volume Obstacle Representation
FDM	Finite Difference Method
FEM	Finite Element Method
FSI	Fluid-Structure Interaction
FVM	Finite Volume Method
GEBCO	GEneral Bathymetric Chart of the Oceans
HPC	High Performance Computing
IB	Immersed Boundary
IBM	Immersed Boundary Method
IBT	Iceberg-tsunami
LBM	Lattice-Boltzmann Method
LES	Large Eddy Simulation
nRMSE	normalised Root Mean Square Error
NSE	Navier-Stokes Equation
OGM	Overset Grid Method
OpenFOAM	Open source Field Operation and Manipulation
PDE	Partial Differential Equation
PIMPLE	Pressure Implicit splitting operator and semi-implicit Method for Pressure-Linked Equations (SIMPLE)
PISO	Pressure Implicit Splitting Operator

RANS	Reynolds-Averaged Navier-Stokes
RNG	ReNormalization Group
SIMPLE	Semi-Implicit Method for Pressure-Linked Equations
SPH	Smoothed Particle Hydrodynamics
STL	STereoLithography
VoF	Volume of Fluid
2D	Two Dimensional
3D	Three Dimensional

List of Figures

Figure 1.1 (a) Falling iceberg at Neko Harbour, Antarctica (courtesy of Alek Komarnitsky) and (b) overturning iceberg at Perito Moreno Glacier in Argentina (courtesy of Victor Qixiang Chen) 2

Figure 1.2 Sketch of the five idealised iceberg calving mechanisms investigated in this study (Heller et al., 2019c) 3

Figure 1.3 (a) Satellite image of the Eqip Sermia glacier and (b) a video still filmed from a tour boat. The tsunami reached an amplitude of 45 to 50 m (Lüthi and Vieli, 2016) 3

Figure 1.4 A floating iceberg near the village Innaarsuit in Greenland posing an IBT threat (The Guardian, 2018) 4

Figure 2.1 Three models of fracture propagation (adapted from Van der Veen, 1998) 8

Figure 2.2 A calving event at the Perito Moreno glacier involving the fall mechanism: (a) prior and (b) during calving (snapshots from Missy Andrea, 2016) 10

Figure 2.3 Sketch of forces on a floating iceberg shelf (adapted from Benn et al., 2007) 11

Figure 2.4 A calving event at the Perito Moreno glacier involving the overturning mechanism: (a) prior to calving and (b) during overturning (snapshots from Missy Andrea, 2016) 12

Figure 2.5 A calving event involving the capsizing mechanism: (a) a floating iceberg prior to calving and (b) during capsizing including the gene- rated tsunamis (snapshots from Top Fives, 2018) . 13

Figure 2.6 A satellite image of the Tasman glacier and Tasman lake (Dykes et al., 2017) 14

Figure 2.7 Sketch of the common calving mechanisms at Perito Moreno glacier (Minowa et al., 2018) 14

Figure 2.8 A map of the Sermilik fjord in Greenland including the Helheim glacier. The pressure metres are marked with black stars (Vaňková and Holland, 2016) 15

Figure 2.9 A photo taken before triggering a gravity-dominated fall test in the laboratory (Heller et al., 2019c) 17

Figure 2.10 Numerical setup for waves generated by snow avalanches (adapted from Naaim, 2013) 20

Figure 2.11 Numerical results of the normalised power spectral density at the first sensor in Fig. 2.8 under different resonant periods (adapted from Vaňková and Holland, 2016) 20

Figure 2.12 Illustration of the iceberg capsizing energy analysis (adapted from MacAyeal et al., 2011) 21

Figure 2.13 Main views of four different calving cases of an ice column with a height l and width b. The parameter l_0 is the glacier front elevation above the still water level (adapted from Massel and Przyborska, 2013) 22

Figure 2.14 Sketch of a periodic water wave 23

Figure 2.15 Landslide-tsunami classification in function of the initial slide position 26

Figure 2.16 A schematic of the different stages of subaerial landslide-tsunamis (adapted from Yavari-Ramshe and Ataie-Ashtiani, 2016) 27

Figure 2.17 Wave amplitude ratio in function of the impulse product parameter P (adapted from Heller and Spinneken, 2013) 29

Figure 2.18 Theoretical wave profiles of (a) Stokes wave, (b) cnoidal wave, (c) solitary wave and (d) bore wave together with the relevant wave parameters (adapted from Heller and Hager, 2011) . 31

Figure 2.19 Landslide-tsunami tests in (a) a wave flume (2D, Heller and Spinneken, 2013) and (b) a wave basin (3D, Heller and Spinneken, 2015) 33

Figure 2.20 Numerical setup with the locations of the wave probes: (a) side view and (b) to (g) various water body geometries from 2D to 3D (Heller et al., 2016) 34

Figure 2.21 Sketch of waves transformed over a bathymetry feature with the relevant parameters (adapted from Ruffini et al., 2021) 36

Figure 2.22 Classification of the CFD methods based on the solved equations (eqs.) 38

Figure 2.23 Crashing simulation of a falling object in pre-tests of this study due to skewed cells if mesh motion is applied: (a) general view of the 3D mesh and (b) cross-sectional view when the simulation crashes 39

Figure 2.24 Block intruding into the water boundary 39

Figure 2.25 The boundary of the solid body (green lines) is represented by the background mesh (brown cells) when using IBM (Chen et al., 2020) 40

Figure 2.26 Sketch of the coupling strategy in a SPH-DEM hybrid model (Tan and Chen, 2017) 42

Figure 2.27 Numerical (LES) and laboratory subaerial landslide-tsunamis at different wave probes (adapted from Liu et al., 2005) 45

Figure 3.1 Illustration of the main concepts in the implementation of the IB type (adapted from Jasak et al., 2014) 53

Figure 3.2 Source code files in the moving IB handling class 54

Figure 3.3 Flowchart of the newly implemented solver *interDyMIbFoam* 63

Figure 3.4 Illustration of the 6 Degree of Freedoms of a rigid body . 65

Figure 3.5 Sketch of the global and local coordinates used in solving Eq. (3.26). The local coordinates move with the block with the origin at the centre of rotation 66

Figure 3.6 Sketch of vectors used in Eq. (3.44). The brown cells denote the IB cells (adapted from Chen et al., 2020) 71

Figure 4.1 The 50 m × 50 m wave basin used for the experiments (Chen and Heller, 2020) 78

Figure 4.2 Illustration of the blocks in the laboratory tests (adapted from Heller, 2019) 79

Figure 4.3 Sketch of the support frame: (a) front view and (b) side view in the fall cases (with some of the components also used in the overturning cases) and (c) side view of capsizing case (adapted from Heller, 2019) 81

Figure 4.4 (a) Photo and (b) sketch of the experimental setup of mechanism E (adapted from Heller, 2019) 81

Figure 4.5 Sketch of the measured components of the motion sensor 84

Figure 4.6 Photo of the enclosure fixed to the block surface including the motion sensor in the mechanism A test (adapted from Heller, 2019) 84

Figure 4.7 Plan views illustrating the wave probe layouts in (a) the mechanism A and (b) the mechanism B tests. (c) Side view of the mechanism B test. Details about the wave probe locations are given in Table 4.4 (adapted from Heller et al., 2019c) 86

Figure 4.8 Sketch of the synchronisation procedure of the instruments 87

Figure 4.9 Example of the water surface elevation data shifted by the offset in the mechanism E case 88

Figure 5.1 Sketch illustrating the key parameters for the heaving sphere case (adapted from Hulme, 1982) 94

Figure 5.2 Sketch of the numerical basin used for the theoretical case 96

Figure 5.3 Comparison of the numerical water surface elevation η with the analytical solution at (a) $x = 1.0$ m and (b) $x = 3.0$ m 97

Figure 5.4 Snapshots of the IBTs generated in mechanism B in the laboratory tests at (a) $t = 0.00$ s, (b) $t = 1.33$ s, (c) $t = 2.67$ s and (d) $t = 4.00$ s (Chen et al., 2020) 98

Figure 5.5 Snapshots of the IBTs generated in mechanism D in the laboratory tests at (a) $t = 0.00$ s, (b) $t = 1.33$ s, (c) $t = 2.67$ s and (d) $t = 4.00$ s (Chen et al., 2020) 99

Figure 5.6 Computational domain used in the convergence tests. The brown dashed lines denote the IBT generation zone and the rest of the domain the IBT propagation zone 100

Figure 5.7 Time series of displacements obtained from the motion sensor data of (a) mechanism B along the z- and (b) mechanism D along the r- and z-directions (Chen et al., 2020) 101

Figure 5.8 IBTs generated with prescribed motion in the convergence tests of mechanisms (a) B and (b) D (for the legend see a) . . . 102

Figure 5.9 Sketch of the computational domain used in the main tests and the locations of the wave probes 103

Figure 5.10 Snapshot series of mechanism B at (a) $t = 0.00$ s, (b) $t = 1.33$ s, (c) $t = 2.67$ s and (d) $t = 4.00$ s 105

Figure 5.11 Snapshot series of mechanism D simulated with $\omega = 0.02$ rad/s at (a) $t = 0.00$ s, (b) $t = 1.33$ s, (c) $t = 2.67$ s and (d) $t = 4.00$ s 105

Figure 5.12 Comparison of the numerically reproduced iceberg motion in (a,b) mechanism B and (c,d,e,f) mechanism D with the corresponding measurements in the laboratory experiments; for the legends see (a) and (c), respectively 107

Figure 5.13 Time series of (a) the pressure force and (b) the viscosity force acting on the iceberg in mechanism B 108

Figure 5.14 Time series of (a) the pressure force and (b) the viscosity force acting on the iceberg in mechanism D 109

Figure 5.15 IBTs measured at $r = 2.0$ m involving varying C_d, C_m and ω for (a) mechanism B and (b) mechanism D 110

Figure 5.16 IBTs measured at wave probes B1 to B18 in mechanism B with the solid lines denoting the laboratory results and the dashed lines the numerical IBTs. The wave profiles in each column are measured at the same radial distance r and in each row at the same propagation angle γ 112

Figure 5.17 IBTs measured at wave probes B1 to B18 in mechanism D with the solid lines denoting the laboratory results and the dashed lines the numerical IBTs. The wave profiles in each column are measured at the same radial distance r and in each row at the same propagation angle γ 113

Figure 5.18 a_1 measured at wave probes B1 to B18 in (a) mechanism B and (b) mechanism D (for the legend see a) 114

Figure 5.19 H_1 measured at wave probes B1 to B18 in (a) mechanism B and (b) mechanism D (for the legend see a) 114

Figure 5.20 Numerical and laboratory IBTs measured at wave probes B1, B7 and B13 in (a) mechanism B and (b) mechanism D (for the legend see a) 117

Figure 6.1 Time history of the contact force in mechanism D 122

Figure 6.2 Comparison of the wave profiles $\eta(t)$ measured at wave probe B1 in mechanisms (a) B and (b) D based on the laboratory experiments and the laminar and turbulence models 124

Figure 6.3 Comparison of IBT decays with distance generated in the laboratory experiments and the turbulence model for mechanisms (a) B and (b) D 125

Figure 6.4 Sketch of the numerical basin for mechanism A 126

Figure 6.5 Comparison of the (a) displacement and (b) velocity of the iceberg in the laboratory and numerical models involving mechanism A (for the legend see a) 128

Figure 6.6 Comparison of the (a) displacement and (b) velocity of the iceberg in the laboratory and numerical models involving mechanism C (for the legend see a) 128

Figure 6.7 Comparison of the (a) displacement and (b) velocity of the iceberg in the laboratory and numerical models involving mechanism E (for the legend see a) 129

Figure 6.8 Time series of water surface elevation $\eta(t)$ at wave probe A1 in mechanism A 130

Figure 6.9 Time series of water surface elevation $\eta(t)$ at wave probe B1 in mechanism C 131

Figure 6.10 Time series of water surface elevation $\eta(t)$ at wave probe B1 in mechanism E 131

Figure 6.11 Illustration of the bilinear interpolation applied in the bathyme- tric data 132

Figure 6.12 Comparison of the outlines of the water body geometry at the Eqip Sermia glacier based on (a) the satellite image (adapted from Lüthi and Vieli, 2016) and (b) the processed bathymetry data. The red lines mark the glacier front and the orange squares denote the calving location 133

Figure 6.13 Computation domain for the Eqip Sermia case with the calving iceberg represented by the IBM (Chen et al., 2020) . . . 134

Figure 6.14 Comparison of the measured iceberg geometry (adapted from Lüthi and Vieli, 2016) and the iceberg represented by the IBM 135

Figure 6.15 Snapshots of the reproduced IBTs in the Eqip Sermia case at a scale 1:100 at (a) $t = 0.0$ s, (b) $t = 0.4$ s, (c) $t = 0.8$ s, (d) $t = 1.2$ s, (e) $t = 1.6$ s and (f) $t = 2.0$ s. The white circle is the wave probe located at 3.80 m from the glacier front (Chen et al., 2020) 136

Figure 6.16 Reproduced IBT at scale 1:100 measured 3.80 m from the glacier terminus 137

Figure D.1 Steps in the turbulence correction function 174

List of Tables

Table 2.1 Key parameters of IBTs in field observations 16

Table 2.2 Key parameters of IBTs in laboratory investigations . . . 19

Table 2.3 Linear wave parameters in different water regimes 24

Table 3.1 Parameters and their functions in the least square polynominal fitting function 54

Table 3.2 Commonly used terms in the discretisation of PDEs in Foam-extend (adapted from Greenshields, 2015) 60

Table 3.3 Boundary conditions for each patch in the domain for a laminar flow case 73

Table 3.4 Files used for parameters initialisations 74

Table 4.1 Block dimensions in the five selected tests 79

Table 4.2 Experimental parameters in the selected tests 82

Table 4.3 Initial locations of the motion sensor in the selected tests 84

Table 4.4 Locations of the wave probes and cameras in the laboratory tests. Coordinates marked with * were also used in the numerical simulation (adapted from Heller, 2019) 85

Table 4.5 Quantification of the measurement uncertainty (adapted from Heller et al., 2019c) 89

Table 4.6 Froude scaling factors for some parameters 90

Table 5.1 Key parameters used in the numerical setup for mechanisms B and D 104

Table 5.2 IBT wave parameters obtained based on wave probe B1 in mechanisms B and D. The relative error is defined as $(p_{numerical} - p_{laboratory})/p_{laboratory} \times 100\%$ with p denoting any wave parameter 111

Table 5.3 Comparison of the wave train energy based on the laboratory tests and numerical simulations 116

Table 5.4 Comparison of the laboratory and numerical wave decay: a_1/h and H_1/h measured along different directions and laboratory c_l and numerical c_n wave decay exponents 118

Table 6.1 Boundary conditions for each patch in the domain for a turbulent flow case 124

Table 6.2 Key parameters used in the numerical setups for the mechanism A, C and E cases 127

Table 6.3 Summary of comparison of different CFD codes for modelling IBTs 139

Chapter 1

INTRODUCTION

1.1 Overview

Climate change is of increasing concern. It has a significant effect on the ice sheet mass balance (Steig et al., 2013; Zwally et al., 2011) and is responsible for the rapid mass loss of ice, e.g. in Greenland (Zwally et al., 2011) and Antarctica (Cook et al., 2005; Liu et al., 2015). The annual ice mass loss in the Greenland and Antarctic ice sheets were approximately -214 ± 58 Gt/yr from 2002 to 2011 (Van den Broeke et al., 2017) and -180 ± 10 Gt/yr between 2003 and 2013 (Velicogna et al., 2014), respectively. While for the remaining glaciers and ice sheets around the world, the mass loss was estimated as -199 ± 32 Gt/yr between 2002 and 2016 (Wouters et al., 2019). Such ice mass loss results in global sea-level rise of 2.26 ± 0.52 mm/yr (Feng et al., 2013), with the Greenland ice sheet contributing 0.59 mm/yr and the Antarctic ice sheet 0.4 mm/yr between 2002 and 2011 (Velicogna et al., 2014).

There are several factors contributing to ice mass loss such as basal melt (Liu et al., 2015), iceberg calving (Cook et al., 2012) and ice discharge to oceans (Nick et al., 2013). Iceberg calving, which is the sudden breaking off of an iceberg from a glacier or an ice shelf, accounts for nearly half of the ice mass loss in Greenland and the majority at Antarctica (Benn et al., 2007; Levermann, 2011), especially for marine-terminating outlet glaciers (Duddu et al., 2013). In the context of a warming climate, the iceberg calving rate is also increasing, because the accumulated melt water may stimulate crevasse propagation (Nick et al., 2013). The volume of calving icebergs vary in geometries and dimensions reaching a height of up to several hundred metres (Levermann, 2011).

In addition to contributing to sea level rise, iceberg calving into a water body can also generate impulse waves of tens of metres in height. Impulse waves are also generated by landslides, snow avalanches and even asteroids impacting into

a water body. The water body can be a reservoir, a lake or an ocean. Impulse waves occurring in oceans are referred to as tsunamis and they propagate as surface gravity water waves. In this thesis, the term "tsunami" is used to denote impulse waves generated in any water body. Further, such tsunamis are named iceberg-tsunamis (IBTs) herein, short for iceberg-generated tsunamis (Heller et al., 2019c; 2020).

IBTs can be generated through different calving mechanisms depending on the iceberg geometry, fracture propagation within the iceberg and the detachment elevation relative to the water surface (Bassis and Jacobs, 2013; Benn et al., 2007). Examples of falling icebergs and a overturning iceberg in nature are shown in Fig. 1.1. Heller et al. (2019c) investigated five different idealised iceberg calving mechanisms (Fig. 1.2), namely (A) capsizing, (B) gravity-dominated fall, (C) buoyancy-dominated fall, (D) gravity-dominated overturning and (E) buoyancy-dominated overturning. After leaving the generation zone, IBT propagation is affected by the varying bathymetry and usually involves frequency dispersion, IBT energy loss as well as changing of wave celerities and amplitudes. When IBTs reach the nearshore area, wave run-up or overtopping may occur, posing threats to the fishing and shipping industries and coastal structures.

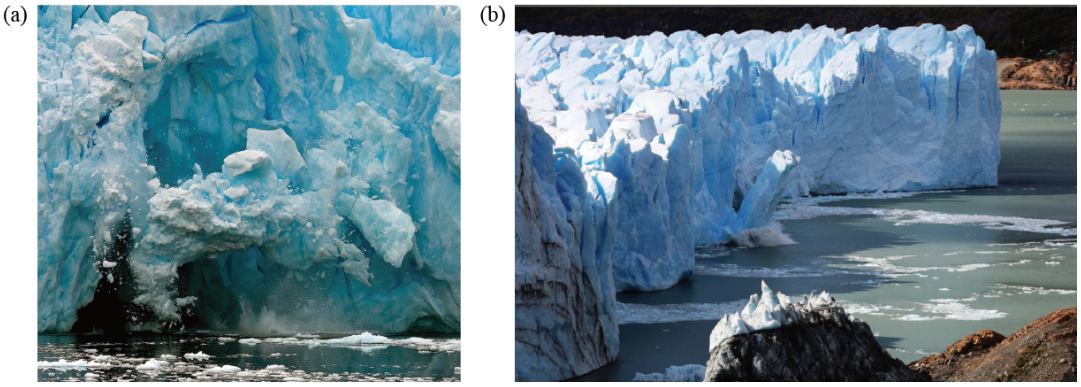


Figure 1.1 (a) Falling iceberg at Neko Harbour, Antarctica (courtesy of Alek Komarnitsky) and (b) overturning iceberg at Perito Moreno Glacier in Argentina (courtesy of Victor Qixiang Chen)

In Greenland, iceberg calving may generate waves with long periods, affecting shores far away from the calving location (Reeh, 1985). In the Sermilik fjord in east Greenland, the maximum observed IBT was still 24 cm 30 km away from the Helheim Glacier terminus for an event measured in 2014 (Vaňková and Hol-

land, 2016). An extreme case was the IBT observed at the Eqip Sermia glacier in Greenland in 2014 with an amplitude of 45 to 50 m (Lüthi and Vieli, 2016). The IBTs reached then a run-up height of 10-15 m on a shore 4 km from the generation location. Further, a harbour and some fishing boats were severely damaged by tsunamis generated by capsizing icebergs, and people were threatened in Greenland in 1995 (Mendsonboaz, 2009). A last example in Greenland is the evacuation of some of the 170 inhabitants in the village Innaarsuit in July 2018 due to IBT hazards from the floating iceberg shown in Fig. 1.4 (The Guardian, 2018).

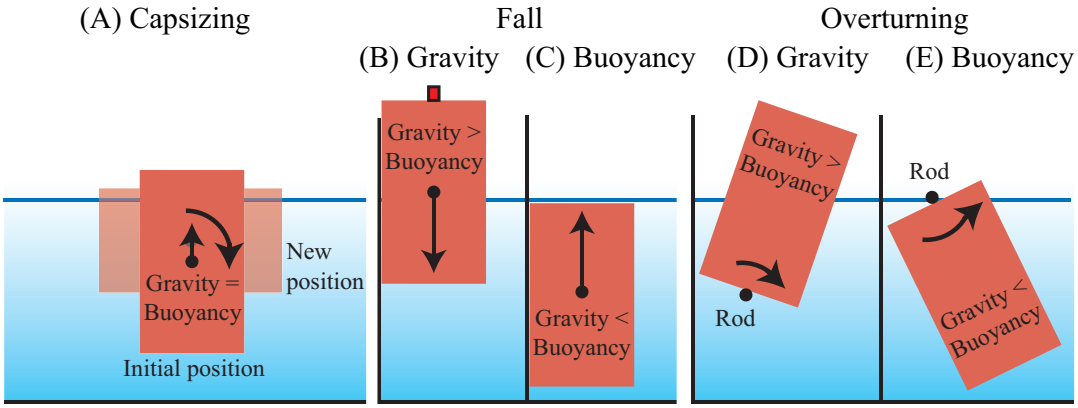


Figure 1.2 Sketch of the five idealised iceberg calving mechanisms investigated in this study (Heller et al., 2019c)

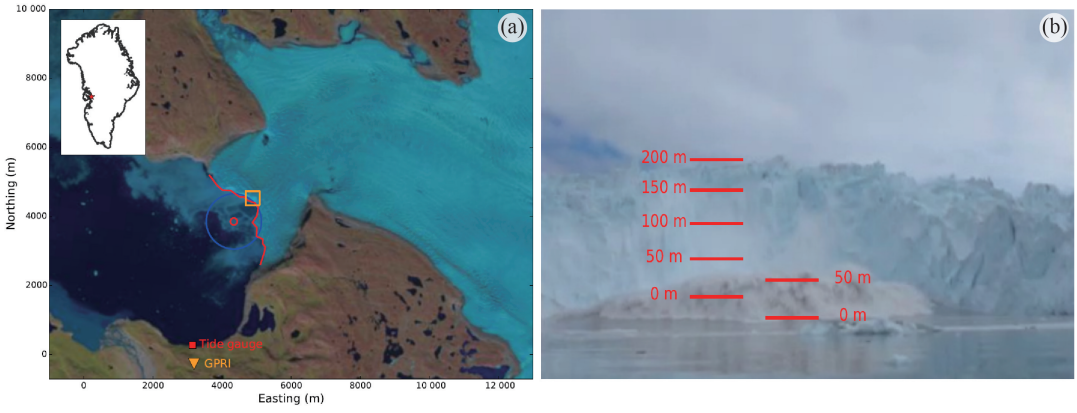


Figure 1.3 (a) Satellite image of the Eqip Sermia glacier and (b) a video still filmed from a tour boat. The tsunami reached an amplitude of 45 to 50 m (Lüthi and Vieli, 2016)



Figure 1.4 A floating iceberg near the village Innaarsuit in Greenland posing an IBT threat (The Guardian, 2018)

IBT hazards were highlighted in a number of recent studies (e.g. Benn et al., 2007; Gilbert et al., 2015; Heller et al., 2019a,b,c; 2020; Lüthi and Vieli, 2016; MacAyeal et al., 2011). However, IBTs were rarely investigated. Thus far, there is a lack of knowledge and quantifications of IBT generation and propagation for IBT hazard assessment in complex water body geometries such as fjords. Before our large-scale three-dimensional laboratory tests (Heller et al., 2019c; 2020), investigations were limited to small-scale two-dimensional laboratory experiments for the capsizing mechanism (Burton et al., 2012). Further, the numerical modelling involving case-specific bathymetries and geometries is still lacking.

Furthermore, there are currently no time-domain numerical models available for IBT generation, considering the capabilities to handle large displacements of rigid bodies in the computational domain for a wide range of iceberg calving mechanisms. As it may be time-consuming and hazardous to conduct field observations, in particular for large and dangerous IBT events, the research field of IBTs is likely to be advanced by numerical simulations and laboratory experiments.

1.2 Aims and Objectives

The aims of this study are to numerically and experimentally investigate the generation and propagation of tsunamis generated by iceberg calving, involving various iceberg calving mechanisms to support IBT hazard assessment. These aims will be achieved with the following objectives:

- Modify an open source Computational Fluid Dynamics (CFD) code to develop a numerical model to simulate large IBTs involving different iceberg calving mechanisms, allowing for multiphase flows and large displacements of rigid bodies in the computational domain
- Support a large-scale laboratory test measurement campaign conducted in a 50 m $\times$ 50 m wave basin to investigate the generation and propagation of IBTs, including the measurement of the movement of the calving iceberg and IBTs at different locations
- Validate the numerical model with a theoretical case and the large-scale laboratory tests from the previous objective
- Apply this numerical model to reproduce the 2014 Eqip Sermia case

1.3 Thesis Outline

This PhD thesis is structured as follows. Chapter 1 contains the research background, aims and objectives together with the outline of the thesis.

A detailed literature review is provided in Chapter 2. This involves experimental, numerical and theoretical studies on IBTs. The literature on landslide-tsunamis is also reviewed, given that landslide-tsunamis are closely related to IBTs, followed by a review of numerical methods to model floating bodies.

Chapter 3 presents details about the numerical model based on the open source CFD code Foam-extend. Newly implemented functionalities are introduced, including an expanded multiphase flow solver, a tailored (iceberg) motion solver and the coupling method between these two solvers.

The experimental set-up of the large-scale IBT tests conducted in a 50 m $\times$ 50 m wave basin is described in Chapter 4, where the data acquisition and processing and model similitude are also addressed.

In Chapter 5, the newly implemented flow solver is validated with a theoretical case. This is followed by convergence tests. Large IBTs of selected laboratory tests are numerically simulated. IBT generation and propagation in the numerical simulations and laboratory tests are compared, including the iceberg motion, forces acting on the iceberg, IBT parameters, wave train energy and wave decay.

Chapter 6 includes a discussion of the results of the numerical simulations. This includes the comparison of contact forces with other studies, the effect of turbulence modelling on IBTs and simulations of the remaining calving mechanisms. As an application of this numerical model, the 2014 Eqip Sermia event was reproduced. Further, a brief comparison of the capabilities and features of this new numerical model and the ones from alternative CFD codes are provided.

Finally, the most important conclusions of this study are presented in Chapter 7, together with the limitations and an outlook.

The appendices include the implementation details of the new flow solver (Appendix A), the method to obtain the block's trajectory (Appendix B), details of obtaining the analytical solution of the floating heaving sphere case (Appendix C) and the modification of the turbulence model in the toolbox of the Immersed Boundary Method (Appendix D).

Chapter 2

LITERATURE REVIEW

2.1 Introduction

This chapter starts with a review of the studies on iceberg-tsunamis (IBTs) (Section 2.2). Several iceberg calving mechanisms are described in Section 2.2.1, followed by field observations, laboratory tests and numerical simulations of IBTs. Finally, analytical studies of IBTs involving idealised calving scenarios are addressed. The basic wave theories are presented in Section 2.3 to provide the theoretical aspects of IBTs. As a physically similar process to IBTs, some aspects of the generation and propagation of landslide-tsunamis are reviewed in Section 2.4. The effects of the slide type and the water body geometry on landslide-tsunamis are discussed, together with the different tsunami wave types.

Regarding the numerical simulation of both the generation and propagation of IBTs, the general methods available to model floating bodies are reviewed. The approaches of modelling Fluid-Structure Interaction (FSI), which play an important role in floating body simulations, are classified and presented in Section 2.5.2. Further, focusing on the simulation of IBTs (Section 2.5.3), the requirements for numerical modelling of IBTs are given and the potential methods are compared. Finally, the herein applied open source code Foam-extend and its application in coastal and offshore engineering is reviewed.

2.2 Iceberg Calving into a Water Body

2.2.1 Iceberg calving mechanisms

Iceberg calving is the consequence of complex fracture propagation due to internal stress distribution. There are four key features relevant for fracture propagation: (a) failure model, (b) rates of fracture propagation, (c) fracture depths

and (d) fracture orientation (Benn et al., 2007).

a. Failure model

Van der Veen (1998) summarises three common models based on linear elastic fracture mechanics for fracturing as shown in Fig. 2.1, namely opening, sliding and tearing. For Model 1, fractures develop perpendicular to the plane of normal stresses. In Model 2, fracture grows along a shear plane and propagates in the direction of shearing. The Model 3 fracture is similar to Model 2, however, the fracture propagation is now perpendicular to the direction of the shear stresses. Most studies focus on Model 1 failures, however, combinations of different models are also observed which are called mixed model of fracture failures.

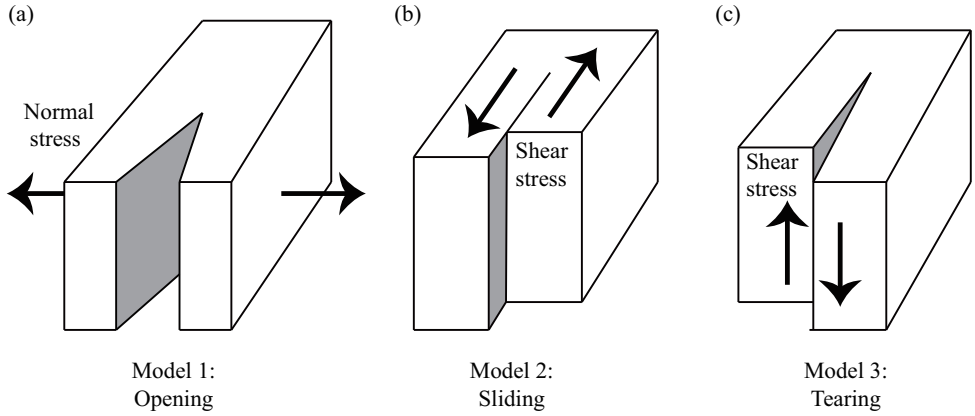


Figure 2.1 Three models of fracture propagation (adapted from Van der Veen, 1998)

b. Rates of fracture propagation

Calving may also be affected by the rates of fracture propagation (Kenneally and Hughes, 2002; Pralong and Funk, 2005). A formula for the fracture propagation rate is obtained using creep fracture theory by Kenneally and Hughes (2002):

$$\frac{\partial d}{\partial t} = \frac{1}{\epsilon_f}[\frac{\sigma_n n}{(n+1)A}]^n (s-d)^{\frac{n}{n+1}} d_c{}^{\frac{1}{n+1}}, \tag{2.1}$$

where d is the fracture depth, ϵ_f the threshold fracture strain rate, σ_n the tensile stress perpendicular to the crack plane, s the thickness of the iceberg, A and n are the ductile constants and d_c is a parameter related to the iceberg ductility. However, this method focuses on individual cracks and is only effec-

tive for certain glacier geometries. Other studies (Pralong et al., 2003; Pralong and Funk, 2005) applied continuum damage mechanics to simulate the iceberg blocks failure of hanging Alpine glaciers and showed a good agreement with the observed geometric change and iceberg motion.

c. Fracture depths

Fractures develop to a certain depth where the forces responsible for fracture propagation are balanced by the opposing forces. Nye (1957) applied this concept to propose a formula to obtain the fracture depth in Mode 1 fracture:

$$d = \frac{2}{\rho_s g} \left(\frac{\dot{\epsilon}}{A} \right)^{\frac{1}{n}}. \tag{2.2}$$

In Eq. (2.2), ρ_s denotes the iceberg density, g the gravitational acceleration and $\dot{\epsilon}$ represents the tensile strain rate. The aim of a large number of studies was to improve the model given by Eq. (2.2) by applying linear elastic fracture mechanics to consider the stress concentrations at the crack tip (Weiss, 2004). However, relatively few comparisons with field data have been made. More recent studies (e.g. Cook et al., 2012; Duddu et al., 2013) investigate the effect of water on the depth of surface fractures and the formulas proposed by Nye (1957) have been strengthened, but this still requires further field data to be validated.

d. Fracture orientation

Some theories such as Nye (1952) and Van der Veen (1999) proposed a method to determine the direction of maximum extension for Mode 1 fracture, but did generally not obtain a good agreement with field data, and this disagreement probably reflects an opening-sliding mixed failure mode (Van der Veen, 1999). Further, it is challenging to obtain the exact relationship between stress fields and fracture propagation patterns, as fracture orientation is still not well understood.

Despite of the fact that no universal theory has been derived for fracture orientation, different types of calving events may still be identified based on the range of positions where fractures can develop (Benn et al., 2007). O'Keel et al. (2003) summarised the 6 calving mechanisms: avalanches, block slumps, column drops, column rotations, internal and submarine. However, this clas-

sification only includes 2 types of motion: translation and rotation. MacAyeal et al. (2003) focused an additional calving mechanism: capsizing, where the ice-shelf fragment rotated by 90°. Based on the motion type of the calving icebergs, Heller et al. (2019a,b,c) investigated tsunamis generated by 5 idealised calving mechanisms, namely (A) capsizing, (B) gravity-dominated fall, (C) gravity-dominated buoyancy, (D) gravity-dominated overturning and (E) buoyancy-dominated overturning. In the following sections, the three most common calving mechanisms are discussed namely fall, overturning and capsizing.

2.2.1.1 Fall

The fall mechanism is closely related to subaerial landslides impacting into a water body (Section 2.4). Further, also the waves generated by the fall mechanism are similar to landslide-tsunamis (Heller et al., 2019c; MacAyeal et al., 2009). A fall calving event is shown in Fig. 2.2. Iceberg blocks detach from the glacier front and the whole chunk of ice falls vertically without further disintegration.

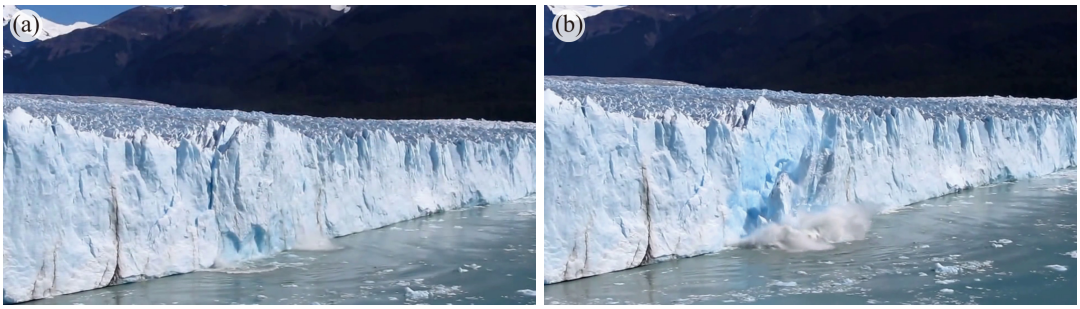


Figure 2.2 A calving event at the Perito Moreno glacier involving the fall mechanism: (a) prior and (b) during calving (snapshots from Missy Andrea, 2016)

The fall mechanism can be triggered by an imbalance of the forces acting on the terminal iceberg cliff (Benn et al., 2007), as illustrated in Reeh (1968) with the model showing a force imbalance in a floating iceberg shelf. At the shelf terminal, the cryostatic pressure is always larger than the atmospheric pressure above the water level and the water pressure underwater (Fig. 2.3). Paterson (1994) provided an equation for the horizontal force equilibrium:

$$\overline{P_I} l - \overline{P_W} h = \frac{1}{2} \rho_s g l s. \tag{2.3}$$

In Eq. (2.3), l is the height of the iceberg, h the water depth and $\overline{P_I}$ and $\overline{P_W}$ indicate the mean pressures of the ice and water in the horizontal direction, respectively.

Therefore, the resultant force is pointing outward from the shelf terminal. It is unevenly distributed along the iceberg shelf front and reaches the maximum at the water surface, which may result in the fracture propagation downward from the elevation of the iceberg shelf surface. Once the fracture causes iceberg instability, a calving event under the fall mechanism occurs.

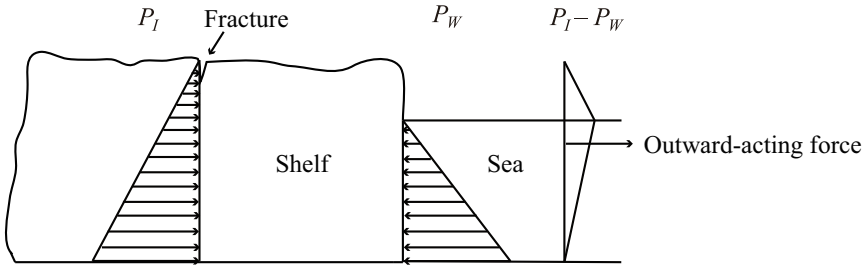


Figure 2.3 Sketch of forces on a floating iceberg shelf (adapted from Benn et al., 2007)

Note that the buoyancy force can also contribute to fracture propagation, resulting in a buoyancy-dominated calving. This mechanism may be caused by the surface melting or the calving mass loss of a subaerial section of the iceberg, breaking down the force equilibrium of the submerged iceberg when the buoyancy force exceeds a certain threshold.

2.2.1.2 Overturning

A calving event involving the overturning mechanism is shown in Fig. 2.4. In the overturning case, an iceberg column tends to gradually break off the glacier front with the fracture propagating deeper to the base (Benn et al., 2007). Then the detached iceberg rotates around its base and impacts into the water body.

When an eccentrically-applied stress causes a torque or bending moment tending to rotate the iceberg front, the overturning mechanism is triggered. Other than the overturning event shown in Fig. 2.4 dominated by gravity, the overturning mechanism may also be dominated by the buoyancy force, i.e. the iceberg rotates around its top towards the water surface. A recent major

buoyancy-dominated overturning event was observed in the Tasman glacier in New Zealand (Dykes et al., 2017), which is revisited in detail in Section 2.2.2. Compared with other calving cases, fall appears to occur more frequently but involves relatively small masses. For example, 84% of the 420 calving events recorded by Minowa et al. (2018) at the Perito Moreno glacier involved the fall mechanism.

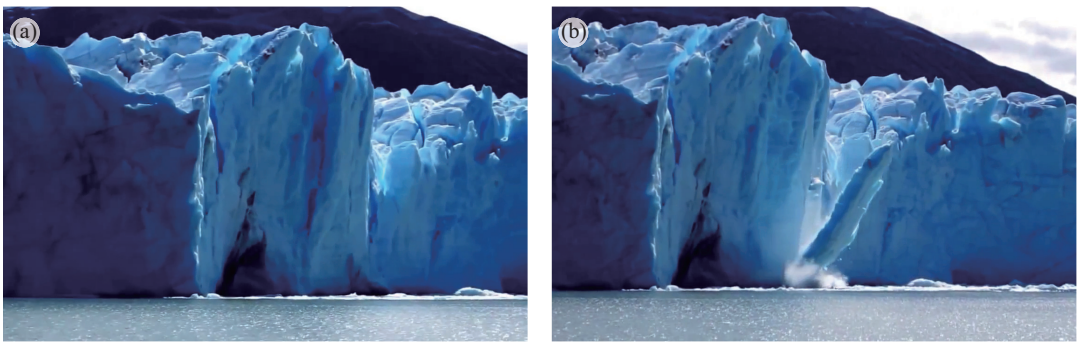


Figure 2.4 A calving event at the Perito Moreno glacier involving the overturning mechanism: (a) prior to calving and (b) during overturning (snapshots from Missy Andrea, 2016)

2.2.1.3 Capsizing

This calving mechanism involves floating tabular icebergs after isolating from the main ice shelves (MacAyeal et al., 2003). These detached floating icebergs may have different dimensions and shapes due to the complex fracture propagation prior to penetrating entire ice shelves. Therefore, these floating icebergs can be unstable due to their geometries and tend to capsize, e.g. when their aspect ratios violate certain equilibrium criteria (Levermann, 2011). The equilibrium of floating icebergs can also be violated by melting or a detachment of small iceberg segments (MacAyeal et al., 2003), resulting in capsizing. A capsizing event can be seen in Fig. 2.5, where an iceberg rolls over and generates tsunamis.

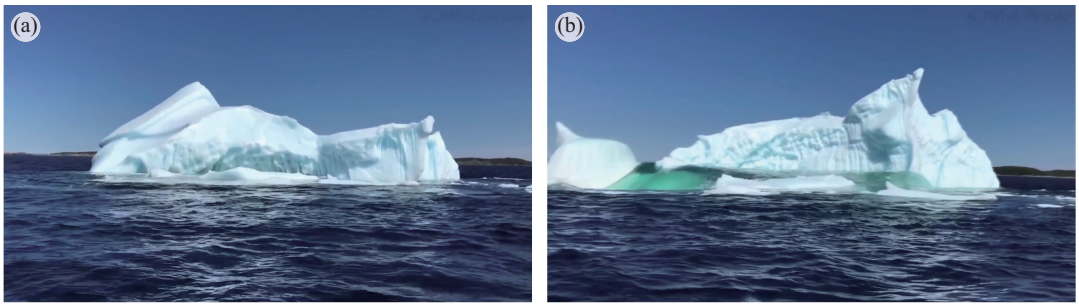


Figure 2.5 A calving event involving the capsizing mechanism: (a) a floating iceberg prior to calving and (b) during capsizing including the generated tsunamis (snapshots from Top Fives, 2018)

2.2.2 Field observations of IBTs

There are a number of studies which recorded IBTs in nature thus far, and some of the observed IBTs were threatening and destructive. Lüthi and Vieli (2016) recorded an iceberg calving event in 2014 in Eqip Sermia (an marine-terminating glacier in the west of Greenland shown in Fig. 1.3a) based on the fall mechanism. They used a multi-method observation consisting of a terrestrial radar interferometer, a tide gauge as well as a video filmed from a boat, contributing to the best documented IBT event thus far. The observed wave amplitude reached 45 to 50 m in proximity of the glacier terminus (Fig. 1.3b), and the tsunami reached a 10-15 m run-up on the opposite shore at a distance of 4 km from the calving location. The waves were successfully predicted with the empirical equations from Heller and Hager (2009) and Heller and Spinneken (2015), commonly applied to predict landslide-tsunamis (Section 2.4).

A large iceberg calving event at the Tasman glacier in New Zealand in 2011 was recorded by Dykes et al. (2017). According to the video recording, the iceberg with an estimated volume of 4.5×10^6 m^3 calved into the Tasman Lake (Fig. 2.6), followed by the fracture penetrating from its top surface to the bottom. The detached iceberg rotated then driven by buoyancy, i.e. as a buoyancy-dominated overturning case, and generated waves. The IBT with 3.1 m in amplitude was detected by a water level data logger on the southern shoreline of the Tasman lake.

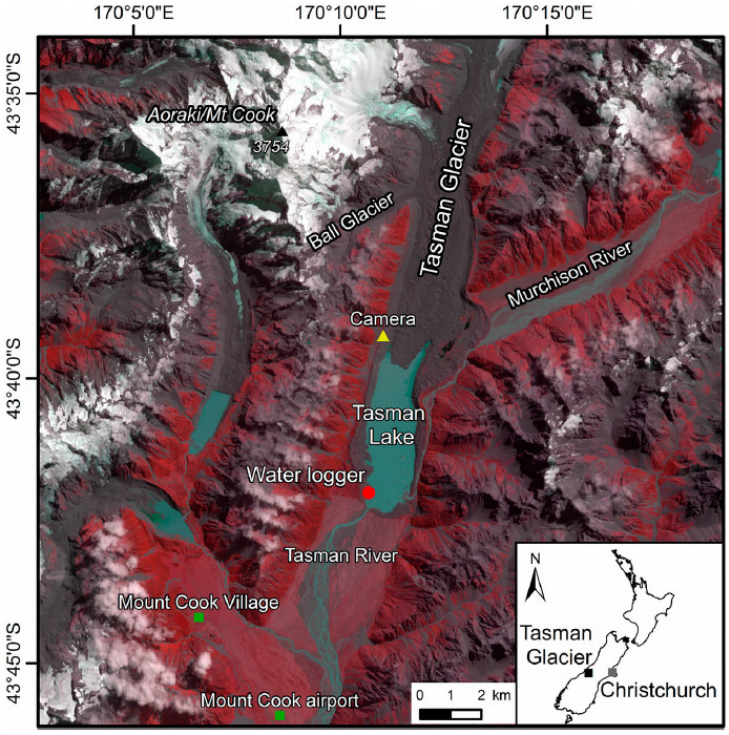


Figure 2.6 A satellite image of the Tasman glacier and Tasman lake (Dykes et al., 2017)

MacAyeal et al. (2009) observed 368 glacial wave generation events in the Ross Sea and the Ross Ice Shelf in Antarctica, parts of which were caused by capsizing icebergs. They found the dominant frequency of the IBTs between 0.05-0.20 Hz. The wave features, such as amplitudes and heights, were not further provided in this study. But in a later study (MacAyeal et al., 2011), the wave height involving the capsizing mechanism was estimated as 1% of the iceberg height.

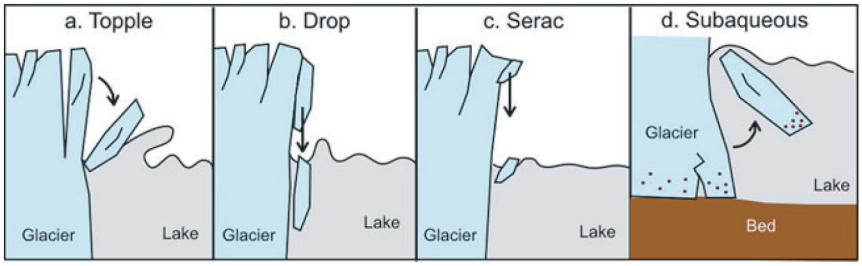


Figure 2.7 Sketch of the common calving mechanisms at Perito Moreno glacier (Minowa et al., 2018)

Minowa et al. (2018) recorded 420 calving events at the Perito Moreno

glacier, Patagonia, with time-lapse photography to observe calving iceberg movement and a water pressure sensor for measuring IBTs. They found that the wave spectra were similar although they were generated by different calving mechanisms, namely topple, drop, serac and subaqueous (Fig. 2.7). Further, they found that the amplitudes of IBTs increased with the volumes of the calving icebergs. Another field study conducted by Minowa et al. (2019) focused on the calving events at the Bowdoin glacier in the north-west of Greenland. A relation between the maximum IBT amplitude a_M and the volume of the calved iceberg $\not{V}_s$ was derived as

$$\not{V}_s = (1.3 \times 10^5) a_M - 9.3 \times 10^3. \tag{2.4}$$

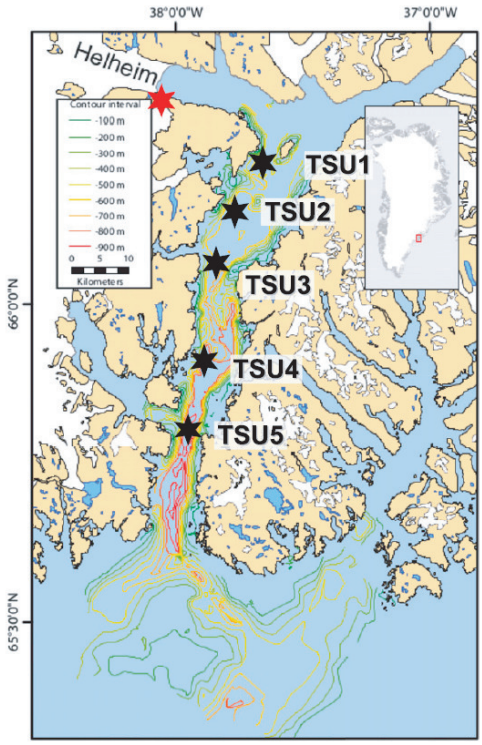


Figure 2.8 A map of the Sermilik fjord in Greenland including the Helheim glacier. The pressure metres are marked with black stars (Vaňková and Holland, 2016)

The IBTs in the above field observation studies were generated and propagated in relative open water bodies. Vaňková and Holland (2016) firstly investigated IBT propagation along a fjord, namely the Sermilik fjord in the east of Greenland, based on five high-frequency pressure metres marked in Fig. 2.8. The height of the largest observed IBT was 24 cm in amplitude at a location

approximately 30 km away from the glacier terminus, and the largest IBT amplitude travelled along the measurement sites at a speed of 58 to 87 m/s. This study provides a good understanding of wave features in fjords due to calving glaciers.

The key parameters of IBTs shown in field observations are listed in Table 2.1, including the calving mechanism, iceberg parameters (iceberg height l, volume $\forall_s$ and density ρ_s), water depth at the calving location, the Froude number Fr and the relative iceberg volume V. The Froude number Fr is defined as

$$\mathrm{Fr} = \frac{V_s}{\sqrt{gh}}, \tag{2.5}$$

where V_s denotes the iceberg impact velocity. The relative iceberg volume V is

$$V = \frac{\forall_s}{h^3}. \tag{2.6}$$

Table 2.1 Key parameters of IBTs in field observations

	Dykes et al. (2017)	**Lüthi and Vieli (2016)**	**MacAyeal et al. (2011)**	**Minowa et al. (2018)**	**Minowa et al. (2019)**
Calving mechanism	Overturning	Fall	Capsizing	NA	Fall
Iceberg height l (m)	NA	180	75-250	55-75	NA
Iceberg volume $\forall_s$ (m^3)	4.50×10^6	9.00×10^5	6.75×10^6	1.04×10^3-2.60×10^5	3.50×10^2-4.90×10^4
Iceberg density ρ_s (kg/m^3)	917	900	910	NA	NA
Water depth at calving location (m)	153	20-40	500-1000	60-100	100-250
Froude number Fr	NA	1.97-3.00	0.2	NA	NA
Relative iceberg volume V	1.26	14.06-112.50	0.25	1.00×10^{-3}-1.20	2.24×10^{-5}-0.05

2.2.3 Laboratory investigations of IBTs

Only a limited number of studies on IBTs have been conducted in the laboratory thus far. Burton et al. (2012) firstly conducted two dimensional (2D) small-scale laboratory tests in a wave flume to quantify the energy transformation from the capsizing iceberg mass to the generated IBTs. The dimension of the wave flume was 2.44 m $\times$ 0.30 m $\times$ 0.30 m. A series of box-shaped blocks with a density $\rho_s = 920$ kg/m^3 were used to model the icebergs, with dimensions of 0.267 m $\times$ 0.103 m $\times$ 0.025-0.102 m. The water depth was varied from 0.114 to 0.243 m. The results indicate that the maximum IBT amplitude involving the capsizing mechanism was 1% of the block height. Further, the energy of the radiated surface wave and the maximum kinetic energy of the icebergs accounted for 1% and 15% of the total iceberg released energy, respectively. The majority of the remaining energy was rapidly dissipated via viscous drag and turbulence.

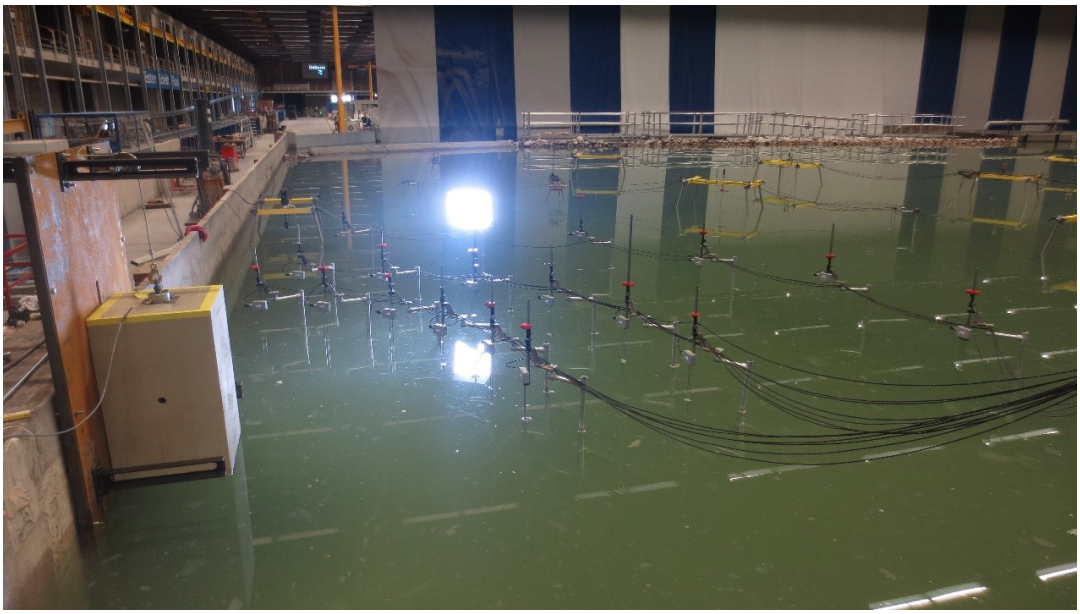


Figure 2.9 A photo taken before triggering a gravity-dominated fall test in the laboratory (Heller et al., 2019c)

A pilot study for the present work was conducted by Yang (2016). He carried out a series of small-scale laboratory tests to investigate the generation and propagation of IBTs under different calving mechanisms, and compared the experimental data with predictions from the empirical landslide-tsunami equations of Heller (2007). However, only the IBTs involving the fall mecha-

nism agree well with results obtained by the existing empirical equations and the requirement for more research was identified to investigate the tsunamigenic potential of all other calving mechanisms.

Heller et al. (2019a,b,c; 2020) carried out three dimensional (3D) large-scale tests in a 50 m × 50 m large wave basin (Fig. 2.9). Five different calving mechanisms were considered, namely (A) capsizing, (B) gravity-dominated fall, (C) buoyancy-dominated fall, (D) gravity-dominated overturning and (E) buoyancy-dominated overturning. The IBT energy analysis method of MacAyeal et al. (2011) for (A) was expanded to (B) to (E). The results show that the IBT train energy involving all calving mechanisms accounts for 0.6-59.6% of the corresponding theoretical released iceberg energy, which is defined as the difference in work required to lift the iceberg to the same reference level prior to and after calving. The IBTs generated by gravity-dominated mechanisms (B) and (D) are approximately an order of magnitude larger than those generated by the remaining mechanisms. Further, empirical equations for a_M were derived by Heller et al. (2020):

Capsizing:

$$\frac{a_M}{h} = 0.93(E_r^{0.71}\mathrm{Fr}^{1.32}S^{0.31}B^{0.67}V^{0.36}D^{-2.20})^{0.68} \tag{2.7}$$

Gravity-dominated fall:

$$\frac{a_M}{h} = 0.13(E_r^{0.20}\mathrm{Fr}^{1.66}S^{0.21}B^{1.46}V^{0.20}D^{0.20})^{0.50} \tag{2.8}$$

Buoyancy-dominated fall:

$$\frac{a_M}{h} = 0.38(E_r^{0.43}\mathrm{Fr}^{2.10}S^{0.15}B^{1.73}V^{0.27}D^{-2.20})^{0.52} \tag{2.9}$$

Gravity-dominated overturning:

$$\frac{a_M}{h} = 0.20(E_r^{0.20}\mathrm{Fr}^{1.12}S^{0.20}B^{1.20}V^{0.43}D^{0.15})^{0.50} \tag{2.10}$$

Buoyancy-dominated overturning:

$$\frac{a_M}{h} = 0.10(E_r^{0.24}\mathrm{Fr}^{1.68}S^{1.27}B^{1.40}V^{0.23}D^{-0.96})^{0.50} \tag{2.11}$$

All mechanisms combined:

$$\frac{a_M}{h} = 0.14(E_r^{0.20}\mathrm{Fr}^{2.07}S^{0.10}B^{1.20}V^{0.20}D^{0.40})^{0.50} \tag{2.12}$$

E_r denotes the relative released energy, Fr the Froude number and S, B and V the relative iceberg thickness, width and volume, respectively. D is the iceberg density relative to the water density. Some of the Eqs. (2.7) to (2.12) were then applied to the Eqip Sermia event resulting in a good agreement. Furthermore, there are also equations available in Heller et al. (2020) to predict the IBT features in function of the relative radial distance. Table 2.2 shows the key parameters of IBTs in laboratory experiments.

Table 2.2 Key parameters of IBTs in laboratory investigations

	Burton et al. (2017)	**Heller et al. (2019c; 2020)**	**Yang (2016)**
Calving mechanism	Capsizing	Fall, overturning and capsizing	Fall and overturning
Test facility	Flume (2D)	Basin (3D)	Flume (2D)
Iceberg height l (m)	0.025-0.102	0.50 and 0.80	0.200 and 0.400
Iceberg width b (m)	0.267	0.50 and 0.80	0.242
Iceberg thickness s (m)	0.103	0.25 and 0.50	0.030 and 0.060
Iceberg volume $\forall_s$ (m^3)	6.9-28.0 $\times 10^{-4}$	0.1 and 0.2	6.5×10^{-3} and 2.9×10^{-3}
Iceberg density ρ_s (kg/m^3)	920	≈ 920	920
Water depth h (m)	0.114-0.243	0.75 and 1.00	0.20-0.30
Froude number Fr	≈ 1.00	0.10-0.47	NA
Relative iceberg volume V	0.05-1.89	0.09-1.33	0.11-0.81

2.2.4 Numerical simulations of IBTs

An numerical model to simulate IBTs was developed by Naaim (2013) who investigated waves generated by snow avalanches using a 2D multiphase numerical computational fluid dynamics code. After investigating a range of slide densities ρ_s from 0.1 to 0.3 kg/m^3 based on the numerical setup illustrated in Fig. 2.10, he found that the wave amplitude increases with the avalanche density

and proposed an empirical equation for a_M:

$$\frac{a_M}{h} = (0.18e^{0.33D} - 0.30e^{-14.28D})\mathrm{Fr}^{0.63}[0.745 + 0.47\ln(V)], \tag{2.13}$$

where $V = sl/h^2$ is the relative slide volume and s and l denote the snow avalanche thickness and length, respectively.

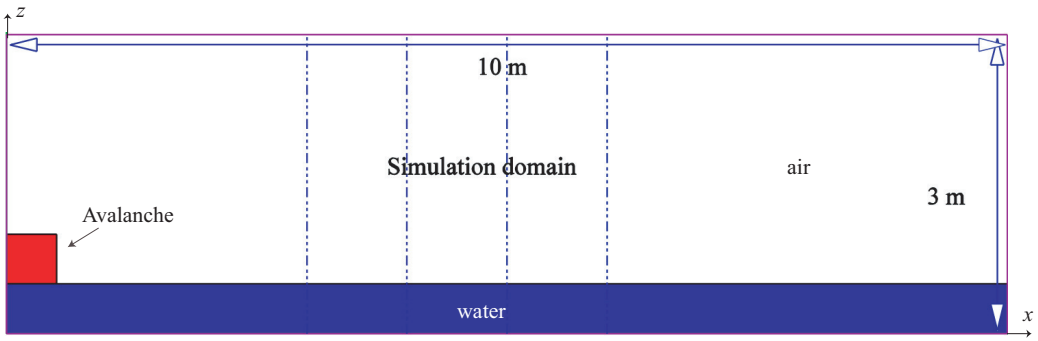


Figure 2.10 Numerical setup for waves generated by snow avalanches (adapted from Naaim, 2013)

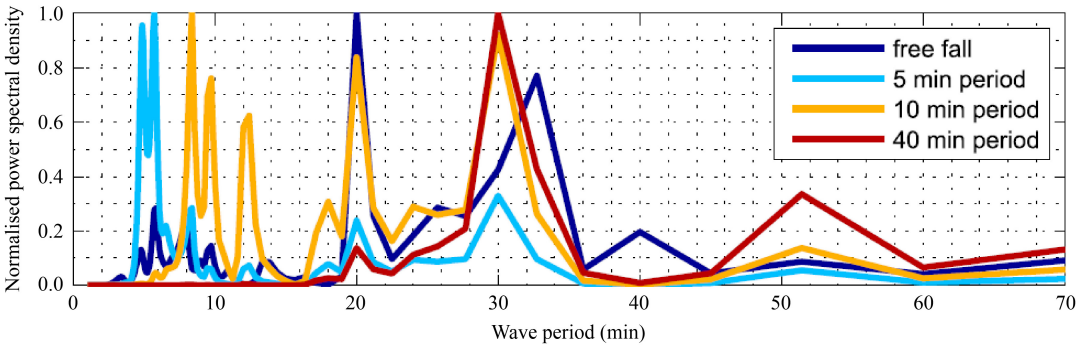


Figure 2.11 Numerical results of the normalised power spectral density at the first sensor in Fig. 2.8 under different resonant periods (adapted from Vaňková and Holland, 2016)

Vaňková and Holland (2016) used a numerical method to obtain the resonant modes of the fjord and to simulate IBTs based on the MITgcm model, a finite-volume method developed by Marshall et al. (1997). The hydrostatic primitive equations, consisting of the Navier-Stokes equations and a thermal energy equation, were thereby solved. In order to mimic the calving process, prescribed forces on the glacier boundaries were imposed. They compared the results (Fig. 2.11) with their own observations in the Sermilik fjord resulting in a good agreement, where the dominant wave periods were 2-10, 21 and 30

min. This simulation appears to be the only numerical investigation of IBTs to date. However, it lacks of a time-domain method to help quantify the IBT parameters.

2.2.5 Analytical modelling of IBTs

The analytical modelling of IBTs includes both the IBT generation and propagation. For the IBT generation, MacAyeal et al. (2011) conducted an energy analysis for a capsizing iceberg. They calculated the theoretical released energy ΔE (Section 2.2.3) with the principle shown in Fig. 2.12:

$$\Delta E = \frac{1}{2}\rho_s g l^3 \epsilon (1-\epsilon)(1-\frac{\rho_s}{\rho_w}). \tag{2.14}$$

In Eq. (2.14), $\epsilon = b/l$ denotes the iceberg aspect ratio and l and b are the iceberg length and width, respectively. The obtained ΔE is the maximum available work to generate IBTs. Further, a dimensional analysis was applied to help quantify the generated IBT amplitude. This energy analysis method was then extended by Heller et al. (2019c) for the additional calving mechanisms.

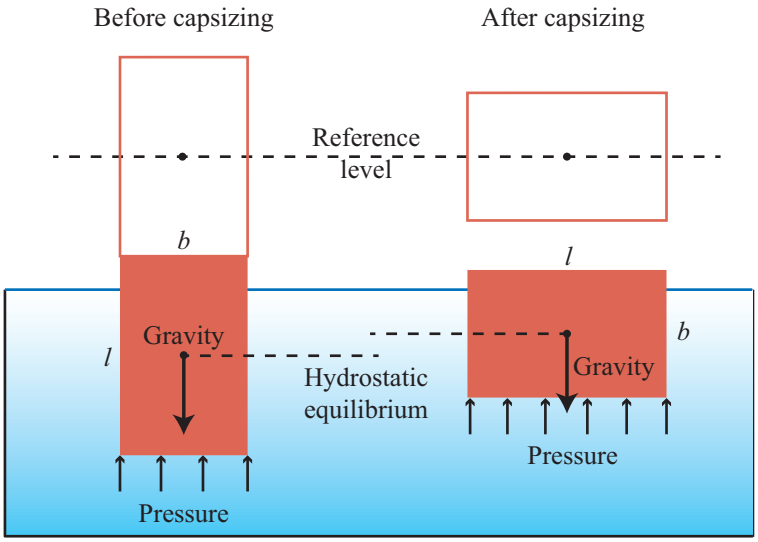


Figure 2.12 Illustration of the iceberg capsizing energy analysis (adapted from MacAyeal et al., 2011)

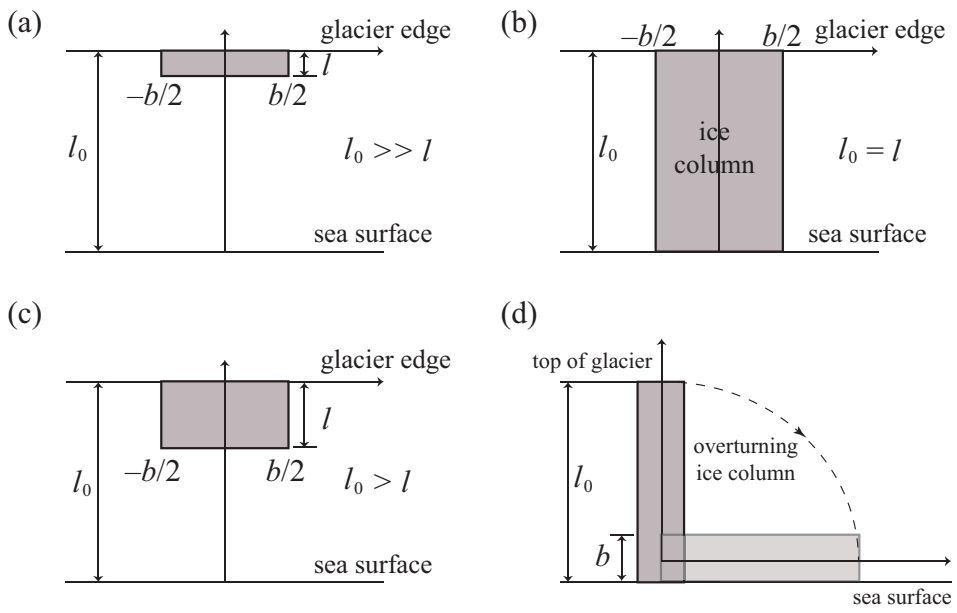


Figure 2.13 Main views of four different calving cases of an ice column with a height l and width b. The parameter l_0 is the glacier front elevation above the still water level (adapted from Massel and Przyborska, 2013)

Massel and Przyborska (2013) present a theoretical model for IBT propagation. Four cases have been investigated: (1) a cylindrical ice block with small height l relative to the glacier front height l_0 ($l \ll l_0$) impacting on water (Fig. 2.13a), (2) an ice column with $l = l_0$ located on the still water level sliding into the water body (Fig. 2.13b), (3) a large ice block ($l < l_0$) vertically impacting into water under a pressure impulse (Fig. 2.13c) and (4) an ice column tipping away the glacier front and overturning onto the water body (Fig. 2.13d). The obtained wave profiles in these four cases show a large wave amplitude decay with distance. Massel and Przyborska (2013) argued that their theory may be helpful for some fall cases to estimate the wave amplitudes. However, when validated with the laboratory IBTs in Heller et al. (2019c; 2020), the agreement was unsatisfactory (Attili, 2019) such that further work is still required to analytically model IBTs well.

2.3 Wave Theories

Water wave theories play a significant role in coastal and offshore engineering, providing the theoretical foundation for landslide-tsunami research. Figure 2.14 illustrates a sketch of a periodic water wave. The wave height H is defined as the elevation difference from the wave crest to the wave trough. a denotes the wave amplitude, which refers to the distance from the wave crest to the still water surface. The wave length L is the distance from two neighbouring zero crossings (or the distance between the two neighbouring wave crests or wave troughs). The wave propagates along the x-axis with the celerity c. Finally, $\eta(x,t)$ denotes the free water surface profile.

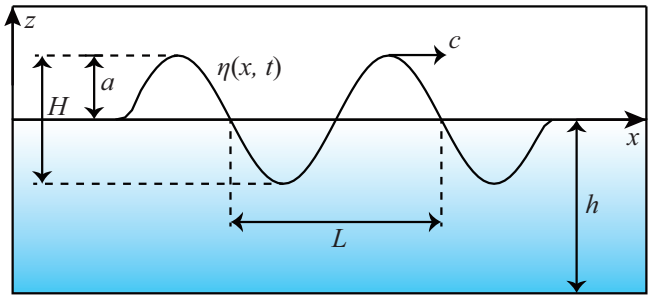


Figure 2.14 Sketch of a periodic water wave

Given that several researchers have presented different wave theories for specific wave types based on different assumptions and boundary conditions (e.g. Dean, 1970; Le Méhauté, 2013; Mei, 1989), a wave theory classification is necessary. In addition to the commonly used parameters such as the wave steepness H/L, the relative wave length L/h and the relative wave height H/h, Ursell (1953) proposed the Ursell number U for evaluating the nonlinearity of water waves

$$\mathrm{U} = \frac{H}{h}\left(\frac{L}{h}\right)^2 = \frac{HL^2}{h^3}. \tag{2.15}$$

Wave theories can be categorised into linear and nonlinear wave theories based on H/h and U.

2.3.1 Linear wave theory

The linear wave theory was developed by George Biddell Airy (Craik, 2004), who applied the potential flow method and assumed the water is irrotational,

inviscid and incompressible. The linear wave theory is appropriate when H/h and U are sufficient small, i.e. $H/h \ll 1$ and U $\ll 1$, such that the higher order terms such as $(H/h)^2$ can be neglected in the derivation of the velocity potential and $\eta(x,t)$. According to the linear wave theory, the solution of $\eta(x,t)$ is a sinusoidal function given as

$$\eta(x,t) = \frac{H}{2}\cos(kx - \omega t), \tag{2.16}$$

where $k = 2\pi/L$ denotes the wave number, $\omega = 2\pi/T$ and T is the wave period. Further, the dispersion relation can be obtained as a function of ω and k

$$\omega^2 = gk\tanh(kh). \tag{2.17}$$

L can be calculated from

$$L = \frac{gT^2}{2\pi}\tanh(kh). \tag{2.18}$$

Thus, the wave celerity c can be obtained

$$c = \frac{L}{T} = \frac{gT}{2\pi}\tanh(kh). \tag{2.19}$$

Considering the approximation of the function $\tanh(kh)$, the water body can be divided into three regimes based on L/h: deep water, intermediate water and shallow water (Dean and Dalrymple, 1991) as shown in Table 2.3. For the deep-water and shallow-water regimes, wave parameters have different approximations. Note that because c relates to T in intermediate and deep water, frequency dispersion occurs, e.g. waves with larger T (lower frequency) propagate faster than waves with smaller T within a wave group. The IBTs of this study are expected to be affected by frequency dispersion.

Table 2.3 Linear wave parameters in different water regimes

Water regime	Deep water	Intermediate water	Shallow water
Relative wave length L/h	$L/h < 2$	$2 \leq L/h \leq 20$	$L/h > 20$
Wave length L (m)	$gT^2/(2\pi)$	$gT^2\tanh(2\pi\mathrm{h/L})/(2\pi)$	$T\sqrt{gh}$
Wave celerity c (m/s)	$gT/(2\pi)$	$gT\tanh(kh)/(2\pi)$	$\sqrt{gh}$

The velocity components along the x- and z-axes, u_x and u_z, show an elliptic trajectory of water particles. These closed trajectories mean that no mass

transport for linear waves occurs. However, the energy still progresses from water particle to water particle. The wave energy for a linear wave $\overline{E_w}$ is the sum of the kinetic energy $\overline{E_k}$ and the potential energy $\overline{E_p}$. For a unit surface area, $\overline{E_k}$ and $\overline{E_p}$ are given by Dean and Dalrymple (1991) as

$$\overline{E_k} = \frac{1}{L}\frac{1}{2}\int_x^{x+L}\int_{-h}^{\eta}\rho_w\frac{u_x^2+u_w^2}{2}\mathrm{d}z\mathrm{d}x = \frac{1}{16}\rho_w g H^2, \tag{2.20}$$

$$\overline{E_p} = \frac{1}{L}\int_x^{x+L}\rho_w g\frac{(h+\eta)^2}{2}\mathrm{d}x = \frac{1}{16}\rho_w g H^2. \tag{2.21}$$

Therefore, $\overline{E_w} = \overline{E_k} + \overline{E_p} = 1/8\rho_s g H^2$ is the mean wave energy per surface area over a wave period.

2.3.2 Nonlinear wave theories

For large H/L the nonlinear terms cannot be neglected and the linear wave theory is not applicable. A number of nonlinear wave theories have been proposed, such as Stokes wave (Stokes, 1847), cnoidal wave (Korteweg and de Vries, 1895), solitary wave (Miles, 1980) and bore wave theory (Le Méhauté, 1962). These nonlinear waves are relevant for landslide-tsunami propagation and the corresponding wave features will be further reviewed in Section 2.4.3.1.

2.4 Landslide-tsunamis

2.4.1 Introduction

Landslide-tsunamis are impulse waves generated by masses interacting with a water body (Section 1.1), belonging to the class of surface gravity water waves. These masses may be landslides, rockfalls, snow avalanches, etc. One of the most destructive impulse wave generated by a landslide occurred in Vaiont, North Italy in 1963, where the landslide volume exceeded twice the storage water volume of the artificial reservoir. This resulted in a wave of 70 m in height after overtopping a dam and causing approximately 2000 casualties (Genevois and Ghirotti, 2005).

Landslide-tsunamis are among natural hazards that have caught attention and concerns of researchers and engineers. A large number of landslide-tsunami

studies have been conducted over several decades providing a reasonably good understanding of landslide-tsunamis' generation and propagation, in contrast to the relatively new field of IBTs. As mentioned by a number of researchers (e.g. Benn et al., 2007; Heller et al., 2019c; MacAyeal et al., 2009), IBTs show some similarities to landslide-tsunamis. It is thus useful to review some selected key-concepts and findings of landslide-tsunami research to help enhance the limited physical understanding of IBTs.

2.4.2 Landslide-tsunami generation

The landslide-tsunami generation depends on the slide's initial position relative to the still water level. Three groups are commonly considered as shown in Fig. 2.15, namely (1) subaerial landslide-tsunamis, (2) partially submerged landslide-tsunamis and (3) submarine landslide-tsunamis. However, taking the iceberg calving processes and mechanisms into consideration, only subaerial landslide-tsunamis are directly relevant, as partially submerged and submarine calving icebergs will move to the water surface and float, which is quite different from the process of partially submerged and submarine landslide propagation into the water body. Therefore, studies addressing categories (2) and (3) are not further considered herein.

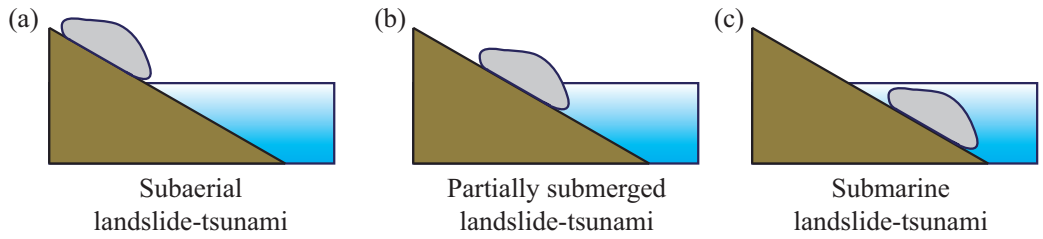


Figure 2.15 Landslide-tsunami classification in function of the initial slide position

2.4.2.1 Subaerial landslides

As the name suggests, subaerial landslides start above the water surface and may generate subaerial landslide-tsunamis if they impact into a water body. A schematic of the different stages of typical subaerial landslide-tsunamis is shown in Fig. 2.16. The impact of a subaerial landslide results in a violent multiphase flow consisting of air, water and solid, specified as the splash zone

(Fig. 2.16). The wave train front starts with a water surface elevation, and in some extreme cases an impact crater is created where the water separates from the slide surface (Heller, 2007). The generated tsunamis propagate then in the water body (e.g. ocean, lake or reservoir) and finally run up on the shoreline or a dam and potentially result in damaged infrastructure and human losses.

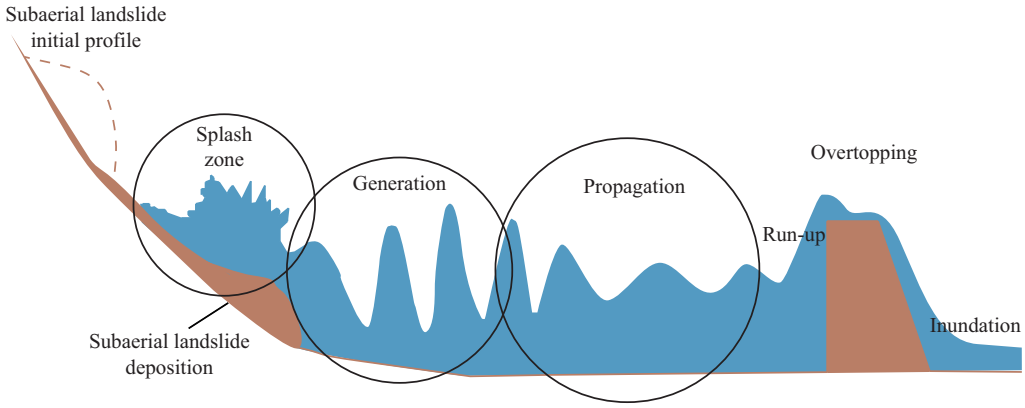


Figure 2.16 A schematic of the different stages of subaerial landslide-tsunamis (adapted from Yavari-Ramshe and Ataie-Ashtiani, 2016)

Scott Russell can be regarded as the first researcher to experimentally investigated subaerial landslide-tsunamis (Monaghan and Kos, 2000). He released a box vertically into a long water tank, a mechanism later called the Scott Russell Wave Generator, resulting in a solitary wave. The relationship between the wave celerity c and its wave amplitude a was obtained as

$$c = \sqrt{g(h+a)}. \tag{2.22}$$

Following Russell's pioneer work, a number of studies focused on similar tests to investigate the effect of the basic slide parameters on tsunamis (Bukreev and Gusev 1996; Panizzo et al., 2002). These parameters involve the water depth h, the slope angle α, the slide impact velocity V_s and the geometry, types, density and dimensions of the slide (Fritz et al., 2003; Panizzo et al., 2005; Zweifel et al., 2006), which control the main tsunami parameters such as a, L, T and c.

2.4.2.2 Effect of slide types

The slide type is an important parameter that affects the tsunami properties. It is also relevant for IBTs because the iceberg mass may fall as one piece or fracture into many smaller pieces during the calving process before impacting into the water body. Studies focusing on the slide type in landslide-tsunamis essentially investigated two extreme types: (a) block models and (b) granular slide models. Type (a) typically represents rockfalls in nature whilst type (b) typically approximates mass movements such as rock slides.

a. Block models

In block model studies, the slide is represented by a rigid mass. A large number of studies with block models were conducted under more realistic conditions than in Russell (1837) where a square box impinged a water body perpendicular to the water surface. These studies typically varied the water depth h, the slide dimensions l, b and s, the slide impact velocity V_s and the slide density ρ_s. However, nearly all studies neglected the blockage ratio $B = b_s/b$ (the ratio of the block width b_s to the channel (or water body) width b) on tsunamigenesis (e.g. Wiegel et al., 1970; Noda, 1970). Only a few studies (e.g. Kamphuis and Bowering, 1972; Walder et al., 2003) investigated the slide front angle ϕ, the slope angle α and the transition type simultaneously. Heller and Spinneken (2013) investigated the effect of these three block parameters on tsunamigenesis and compared the wave profiles with the granular experiments of Heller and Hager (2010) to resolved two shortcomings: the effect of the block parameters and of the slide type (block against granular) on landslide-tsunami characteristics. Heller and Spinneken (2013) obtained the relationship between the wave amplitude ratio $a_{M,block}/a_{M,granular}$ in function of the impulse product parameter P, which includes all relevant slide parameters and the water depth.

$$\frac{a_{M,block}}{a_{M,granular}} = \frac{27}{16}\mathrm{P}^{\frac{1}{10}}(B\phi T_s^{\frac{1}{2}})^{\frac{9}{10}}. \tag{2.23}$$

In Eq. (2.23), T_s is a parameter related to the transition type. The maximum $a_{M,block}/a_{M,granular}$ was 1.66 (Fig. 2.17). Further, Heller and Spinneken (2013) found that waves generated by block slides are not always larger than by granular slides, e.g. when the block slide runs out at the slope toe, a granular slide stops further and converts therefore more of its energy into the tsunami.

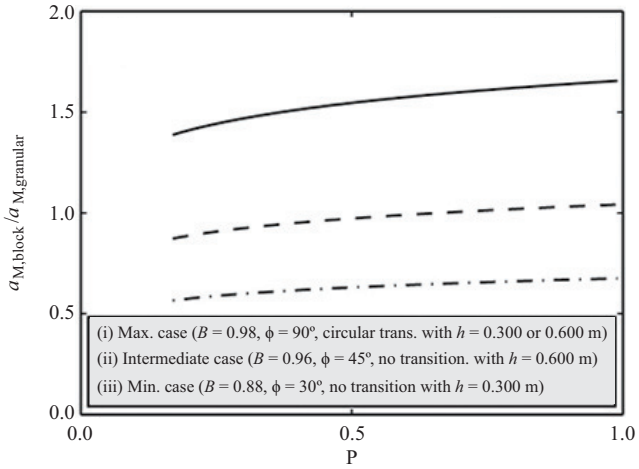


Figure 2.17 Wave amplitude ratio in function of the impulse product parameter P (adapted from Heller and Spinneken, 2013)

b. Granular slide models

To predict landslide events in nature, the slide may also be represented by granular material. It is argued in most studies that granular slides generate smaller waves than blocks. Zweifel (2004) conducted experimental tests and observed that block slides generated larger waves than the corresponding granular slides (a maximum of 7.4 times in wave amplitude). Furthermore, Ataie-Ashtiani and Nik-Khah (2008) concluded that the amplitudes of waves generated by granular slides were 35% smaller and the wave periods were 30% larger than waves generated by block models. A number of aspects may contribute to the difference between block and granular slide generated waves such as the different slide rigidity, porosity, front angle, blockage ratio and the transition type at the slope toe (Heller and Kinnear, 2010).

Lindstrøm (2016) conducted a series of 2D tests and presented maximum wave heights generated by granular slides with various porosities and block slides. She parameterised the effect of the slide type on tsunamigenesis with the permeability parameter k such that conditions between the two extreme case block models and granular sides may be understood. The calculation of k was based on both the Hazen's formula and the Kozeny-Carman equations. The experimental results show that when $k > V_s$ the water may penetrate the slides resulting in smaller waves, and when $k < V_s$ larger waves are expected due to less water entering into the slide. However, this study does not consider other

slide parameters, e.g. h, the slide volume V_s, ϕ or slide deformation. Further investigations are required to fully parameterise the effect of the slide type with the permeability parameter.

2.4.3 Landslide-tsunami propagation

As shown in Fig. 2.16, landslide-tsunamis propagate into the propagation zone and may decay due to lateral energy spread, frequency dispersion and wave energy dissipation before they increase again due to shoaling effects in shallower water. A number of researchers derived empirical equations to predict landslide-tsunami propagation (e.g. Heller and Hager, 2010; Ruffini et al., 2019). In order to better understand landslide-tsunami propagation, it is necessary to review the different wave types and how the water body geometry and bathymetry affect landslide-tsunamis. Both the wave type and the water body geometry are also relevant for IBTs because they may be physically related to landslide-tsunamis and can be generated in very different water body geometries (Fig. 2.8).

2.4.3.1 Wave types

Noda (1970) experimentally studied subaerial landslide-tsunamis and categorised them with four wave types: (a) nonlinear oscillatory waves, (b) nonlinear transition waves, (c) solitary-like waves and (d) dissipative transient bore waves. In nearly all later studies such as Huber (1980) and Fritz et. al. (2004), the identical four wave types have been observed, although slightly different terms have been used for their classification. This also involves Heller and Hager (2011) who investigated the wave types of subaerial landslide-tsunamis based on granular slides. According to the 211 newly conducted and 223 laboratory experiments from Fritz et al. (2004) and Zweifel (2004), the observed wave types were classified into Stokes-like, cnoidal-like, solitary-like and bore-like waves. Most of these wave types are based on theoretical wave profiles (Fig. 2.18), which are briefly presented hereafter. In Fig. 2.18, the wave trough a_t is described as the elevation difference between the still water level and the lowest level of the trough.

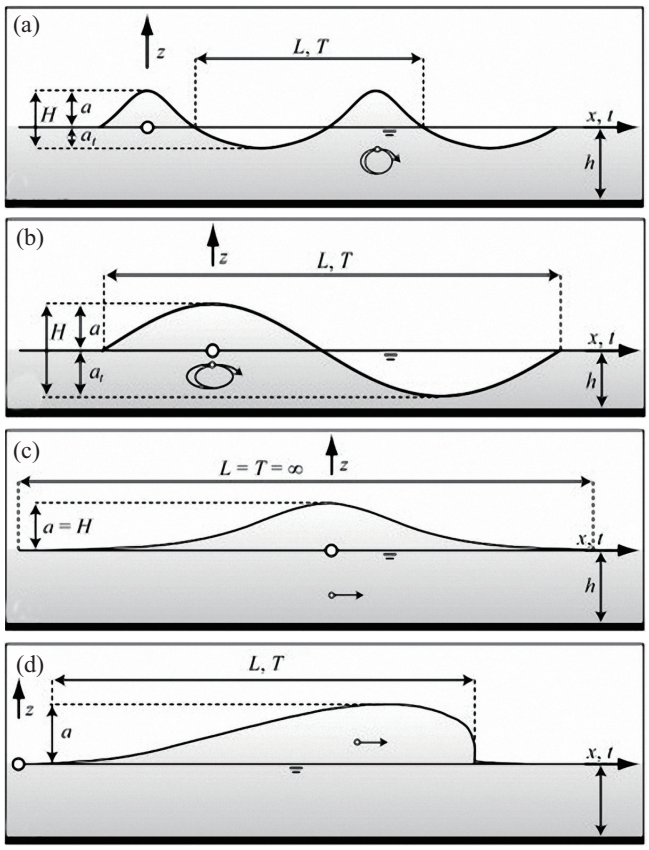


Figure 2.18 Theoretical wave profiles of (a) Stokes wave, (b) cnoidal wave, (c) solitary wave and (d) bore wave together with the relevant wave parameters (adapted from Heller and Hager, 2011)

a. Stokes waves

Stokes waves were derived based on the theory of potential flow and a non-hydrostatic pressure distribution (Heller and Hager, 2011). Compared with the wave profile of a sinusoidal wave, the Stokes wave crest and trough are steeper and flatter, respectively (Fig. 2.18a). It is mainly of oscillatory character and results only in small transport of fluid masses. According to Le Méhauté (1976), i.e. Stokes wave theory applies to intermediate water waves, i.e. $2 \leq L/h \leq 20$.

b. Cnoidal waves

Korteweg and de Vries (1895) proposed the theory for cnoidal waves, belonging to the shallow-water wave regime (Dean and Dalrymple, 2002). The profile of a cnoidal wave has a crest amplitude exceeding the trough while the trough is longer than its crest (Fig. 2.18b). Similar as Stokes waves, cnoidal waves re-

sult in a small fluid mass transport as their water particles follow essentially a drifting elliptical trajectory.

c. Solitary waves

Scott Russel firstly investigated solitary waves and the corresponding theory was derived by Boussinesq in 1871 (Munk, 1949) and then by Rayleigh (1876). The solitary wave is essentially a limiting case of the cnoidal wave with $T \to \infty$ (Laitone, 1960), and thus belongs to the shallow-water wave regime. The solitary wave profile consists only of one wave crest (Fig. 2.18c). The solitary wave is translative such that it involves considerable fluid mass transport.

d. Bore waves

Wiegel (1964) first used tidal bores to describe wave breaking in the near shore zone. Bore waves are highly turbulent. A bore wave normally involves a steep front with a large amount of air and a flat section following its wave crest (Fig. 2.18d) and it can be considered as a shallow-water wave with strong fluid mass transport (Heller and Hager, 2011).

2.4.3.2 Effect of the water body geometry

Different water body geometries, e.g. lakes, fjords, reservoirs and oceans, result in different tsunami magnitudes by identical initial slide conditions. Two extreme geometries are commonly investigated. For the two-dimensional (2D) case the slide impacts longitudinally along the x-axis into a long and narrow water body, with a similar shape as a wave flume. The tsunamis are then characterised as 2D waves without radial spreading other than in the x-direction. 3D landslide-tsunamis are investigated in wave basins such that they represent slides impacting into a larger water body where the tsunamis propagate on semi-circles in different directions from the impact zone (Heller and Spinneken, 2015). In this case, a cylindrical coordinate system can be used to describe the wave propagation, with the radial distance r and the wave propagation angle γ. The effect of the water body geometry is very important as tsunami heights in 2D and 3D can be orders of magnitudes different. Examples of tsunamis generation in 2D and 3D are shown in Fig. 2.19.

Kranzer and Keller (1959) presented an analytical model for the wave am-

plitude decay in 2D and 3D based on the assumption that the initial impulse at any point on the surface is determined by the impulse in the incident shock wave at that point. They concluded that in 2D, the wave amplitudes decay with $x^{-1/3}$ in contrast with r^{-1} in 3D. A systematic comparison of 2D and 3D subaerial landslide-tsunamis was made for the first time by Huber (1980). Based on a large number of experiments, he suggested that impulse waves generated in a 2D are always decaying slower than in 3D. Following this study, a number of researchers established transformation methods of wave parameters from 2D to 3D. Huber and Hager (1997) proposed a formula for 3D wave prediction based on Huber (1980). Further, in the manual of Heller et al. (2009) an empirical approach based on 2D wave parameters (Heller and Hager, 2010) to predict the wave run-up and overtopping volumes at dams in 3D cases has been developed. Some case studies such as Fuchs and Boes (2010) showed that this prediction may result in slightly too large waves for real situations such that Evers et al. (2019) presented a new edition of this manual taking new findings into account.

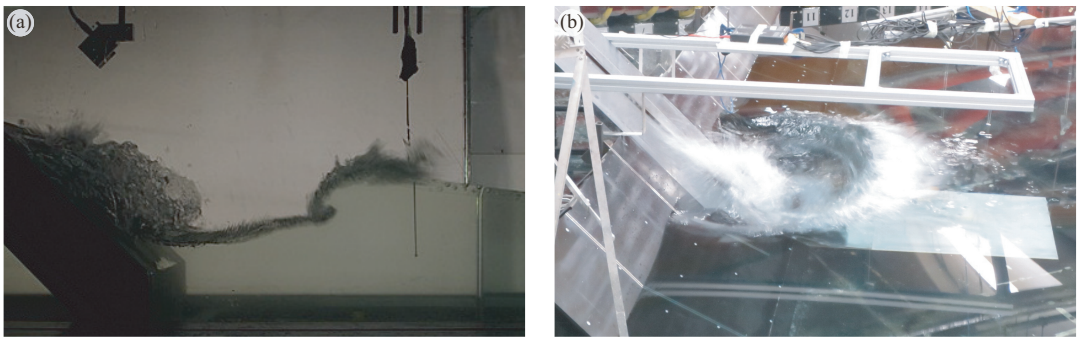


Figure 2.19 Landslide-tsunami tests in (a) a wave flume (2D, Heller and Spinneken, 2013) and (b) a wave basin (3D, Heller and Spinneken, 2015)

Other than focusing on 2D and/or 3D water body geometries only, a series of laboratory experiments were conducted to study the effect of tsunami propagation in intermediate geometries (e.g. Heller et al., 2012; Heller and Spinneken, 2015). The wave features from 2D to 3D were linked with a novel transformation method by Heller and Spinneken (2015). Further, Heller et al. (2016) numerically simulated the waves in five different intermediate geometries (Fig. 2.20). However, due to the large computational power required, the far field wave propagation ($r/h \geq 7.5$) could not be investigated.

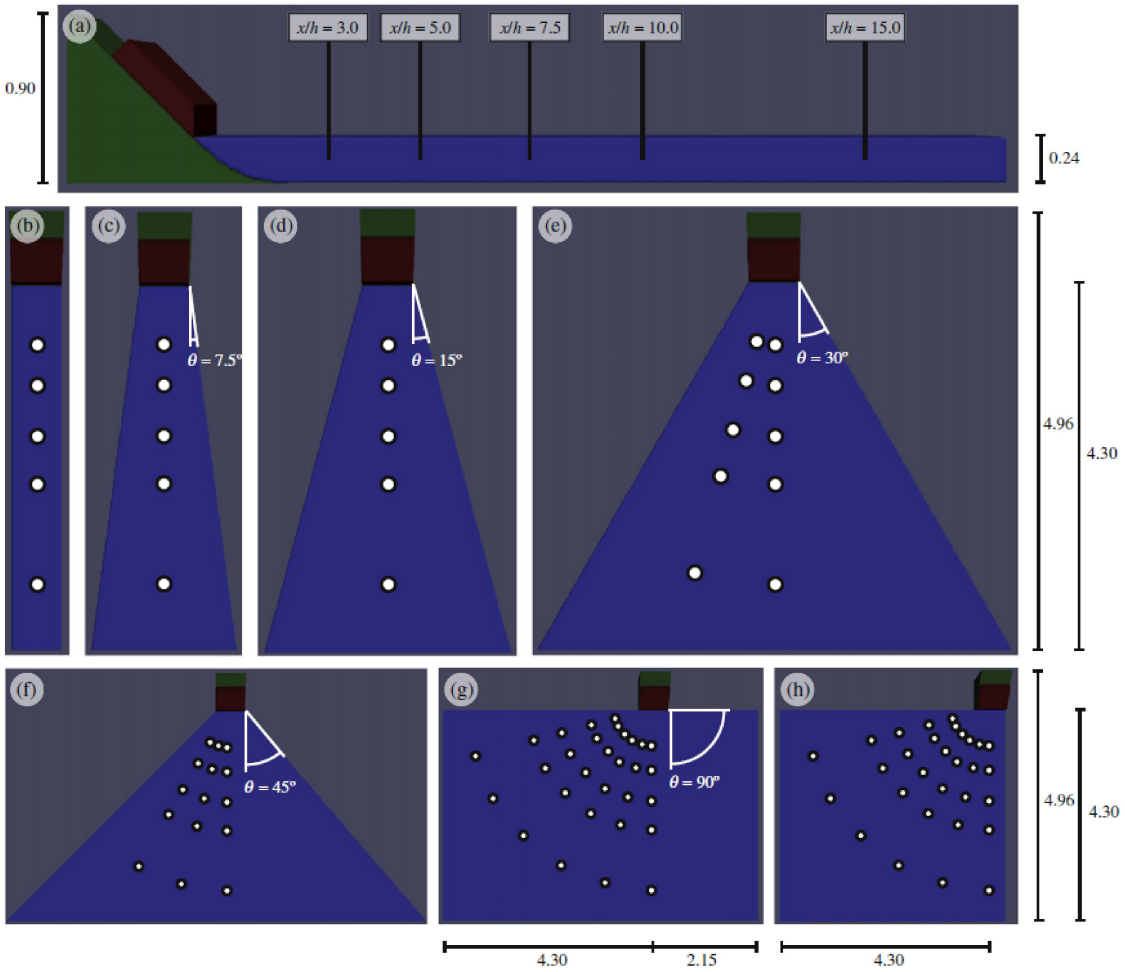


Figure 2.20 Numerical setup with the locations of the wave probes: (a) side view and (b) to (g) various water body geometries from 2D to 3D (Heller et al., 2016)

Ruffini et al. (2019) used the less computational expensive code SWASH to simulate the far field tsunami propagation ($r/h \leq 35$). After validation with Green's law and a diffraction theory, a semi-theoretical equation to predict tsunami amplitudes in any idealised water body geometry was proposed

$$\frac{a(r',\gamma',\theta)}{h} / \left(\frac{b'}{l_w(r',\ \theta)}\right)^{\frac{1}{2}} = \beta \frac{a(r'=0,\gamma'=0°,\theta=0°)}{h} \cos^2\left(\frac{\gamma'}{3}\right). \tag{2.24}$$

where r' and γ' are the radial distance and the wave propagation angle from the coupling location, respectively. θ denotes the water body side angle (e.g. $\theta = 0$ for 2D, $\theta = \pi/2$ for 3D, etc.), b' the source width at the coupling location, $l_w = b' + 2r'\theta$ (θ should be in radians) the wavefront length and β the pre-factor

depending on the wave type (Ruffini et al., 2019). Eq. (2.24) may be regarded as the most complete equation so far to account for the effect of the water body geometry for preliminary landslide-tsunami hazard assessment.

2.4.3.3 Effect of the bathymetry

The tsunami wave characteristics may be changed after propagating over a varying bathymetry. The effect of the bathymetry has been described theoretically by Green's law, which is based on the linear shallow water wave theory and the assumption of a constant energy flux along the wavefront. According to Dean and Dalrymple (1991), the wave transformation based on Green's law is expressed as

$$H_2 = H_1 \sqrt{\frac{b_1}{b_2}} \sqrt[4]{\frac{h_1}{h_2}}, \tag{2.25}$$

where the subscripts 1 and 2 denote two locations of interest and b_1 and b_2 are the widths at location 1 and 2, respectively. However, Eq. (2.25) has limitations: it does not consider energy dissipation and is only applicable for waves without reflection.

Nevertheless, Eq. (2.25) is still powerful in estimating tsunami amplitude changes over changing bathymetries. A number of researchers have proposed methods based on Eq. (2.25) to investigate the propagation of landslide-tsunamis. Synolakis (1991) theoretically extended Eq. (2.25) to account for solitary waves and with reflection under a idealised bathymetry, where h is constant and then linearly decreases to the shoreline at a slope angle α.

$$H_M = h \sqrt[4]{\frac{\cot\alpha}{x}}, \tag{2.26}$$

where H_M is the maximum wave height with the wave propagating towards the beach and x is the corresponding location of H_M. This expanded formula was then employed by Weiss et al. (2013), who numerically investigated the hazard of landslide-tsunamis at coasts. Moreover, Synolakis and Skjelbreia (1993) experimentally investigated solitary wave shoaling at a beach and the measured wave transformation agreed well with Green's law. However, these studies only focus on one specific bathymetry feature.

Ruffini et al. (2021) investigated landslide-tsunamis over a selection of idea-

lised bathymetry features (step shaped, Gaussian shaped and linear beach) and proposed the following formulas to calculate the wave transformation coefficients K_b and $K_{a,b}$

$$K_b = \frac{H_b}{H_0} = 0.23 + 0.77\tanh^{1-(h_f/h_0)(a_0/H_0)}\{4[(h_f/h_0)(a_0/H_0)]^{3/2}\}, \tag{2.27}$$

$$K_{a,b} = \frac{a_b}{a_0} = 0.34 + 0.66\{\frac{1.35[(h_f/h_0)(a_0/H_0)]}{0.59 + (h_f/H_0)^{1.1}}\}^2. \tag{2.28}$$

In Eqs. (2.27) and (2.28) H_b denotes the transformed wave height, H_0 the incident wave height, h_f the water depth at the bathymetric feature, h_0 the initial water depth, a_b the transformed wave amplitude and a_0 is the initial wave amplitude. The parameters in Eqs. (2.27) and (2.28) are illustrated in Fig. 2.21.

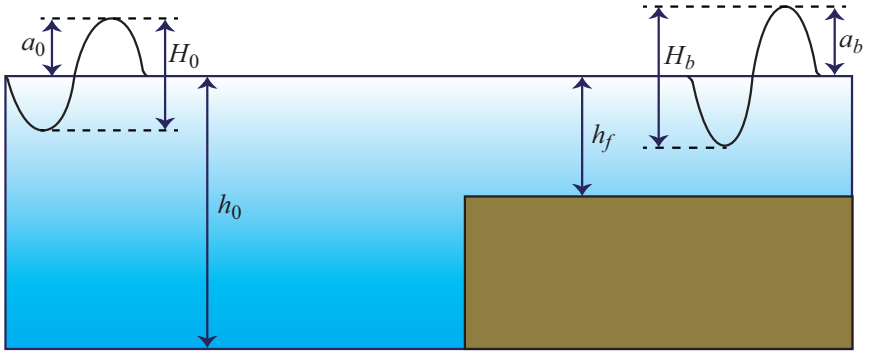


Figure 2.21 Sketch of waves transformed over a bathymetry feature with the relevant parameters (adapted from Ruffini et al., 2021)

Although a number of studies related to the generation and propagation of landslide-tsunamis have been conducted (Section 2.4), the knowledge transferred to IBTs is still limited as there are relatively few studies focusing on IBTs thus far. According to Heller et al. (2020), empirical equations obtained for predicting landslide-tsunamis can in general not well be applied to IBTs, e.g. to wave decay and far-field wave amplitudes.

2.5 Numerical Methods to Model Floating Bodies

2.5.1 Introduction

Computational Fluid Dynamics (CFD) plays an important role in both academia and industry with the rapid development of computational power. CFD has become a robust method in aerodynamics, biological engineering, coastal and offshore engineering, heat transfer problems and many other fields. Different methods have been proposed and developed over the last decades.

As shown in Fig. 2.22, traditional CFD methods can be classified into Eulerian methods and Lagrangian methods according to the description of fluid motion. More recently a novel approach named Lattice-Boltzmann Method (LBM) has raised attention, which focuses on fluid microscopic behaviour by solving the Boltzmann equation in a Lagrangian-Eulerian framework. Based on the discretisation processes, Lagrangian approaches involve methods such as Smoothed Particle Hydrodynamics (SPH) and Eulerian approaches can be classified into Finite Difference Method (FDM), Finite Element Method (FEM), Finite Volume Method (FVM), etc. Based on the solved theoretical equations, CFD methods can be divided into Shallow water equations models, Boussinesq-type equations (e.g. depth-averaged Boussinesq equations and higher-order Boussinesq equations) models and Navier-Stokes equations (NSEs) models such as Direct Numerical Simulation (DNS), Large Eddy Simulation (LES) and Reynolds-Averaged Navier-Stokes (RANS) equation simulations.

Simulations of floating bodies involve Fluid-Structure Interaction (FSI). A wide variety of methods for FSI has been proposed (Fig. 2.22), including the Arbitrary Lagrangian Eulerian (ALE) method (Donea et al., 1982; Zhang and Hisada, 2001), the Immersed Boundary Method (IBM), meshless methods (Antoci et al., 2007; Rabczuk et al., 2010) and the relatively new LBM (Benamour et al., 2015; Sui et al., 2007). These methods are reviewed in detail in Section 2.5.2, followed by a review of the potential methods to simulate IBTs.

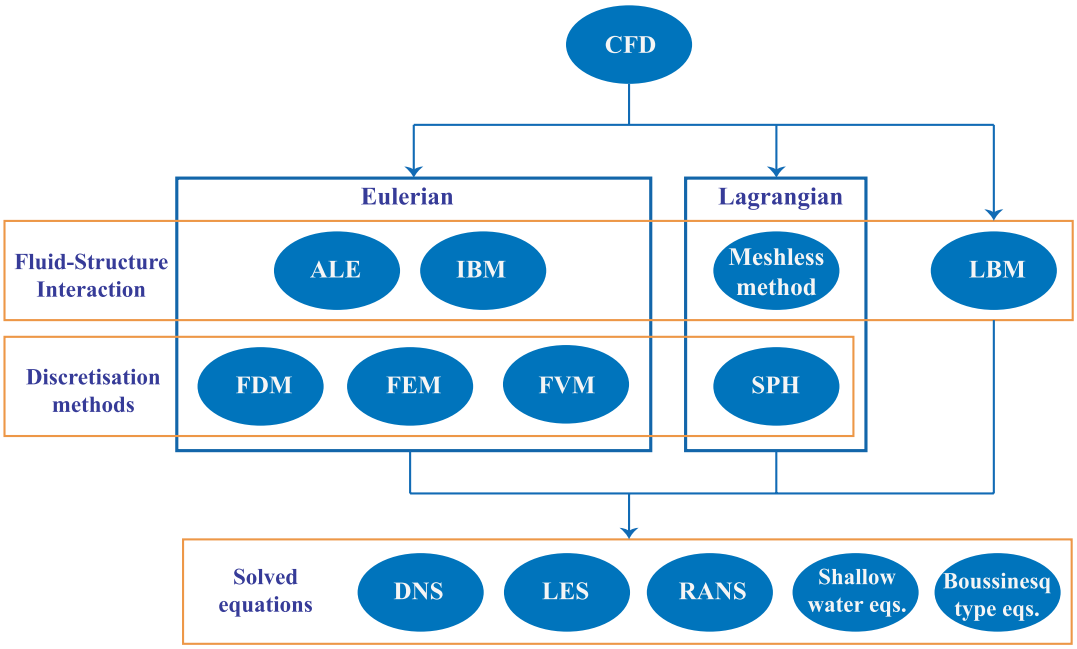


Figure 2.22 Classification of the CFD methods based on the solved equations (eqs.)

2.5.2 Methods to simulate Fluid-Structure Interaction

2.5.2.1 Arbitrary Lagrangian Eulerian (ALE)

The ALE approach is essentially an Eulerian mesh based method with Lagrangian markers to denote the solid body boundary (Donea et al., 1982). The mesh in the computational domain may be changed to fit the movement of the solid body. This body-fitted mesh does not allow a large displacement of a moving body as the mesh quality may not be guaranteed, which will cause the simulation to diverge. Figure 2.23 shows an example of a falling object impacting into water involving severely skewed and/or squeezed cells, which results in simulation crashing. Although some mesh handling techniques, such as sliding interface or specific topological changes (Jasak, 2009), have been proposed to solve the above limitation, e.g. with the simulation of a 2D capsizing case using a sliding interface approach (Fig. 2.24), the remeshing for an arbitrary solid body movement is still challenging.

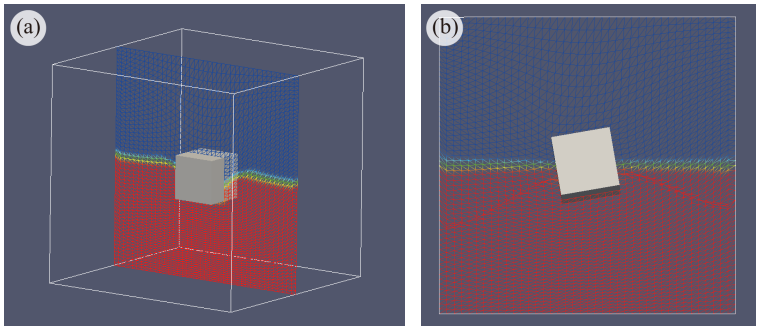


Figure 2.23 Crashing simulation of a falling object in pre-tests of this study due to skewed cells if mesh motion is applied: (a) general view of the 3D mesh and (b) cross-sectional view when the simulation crashes

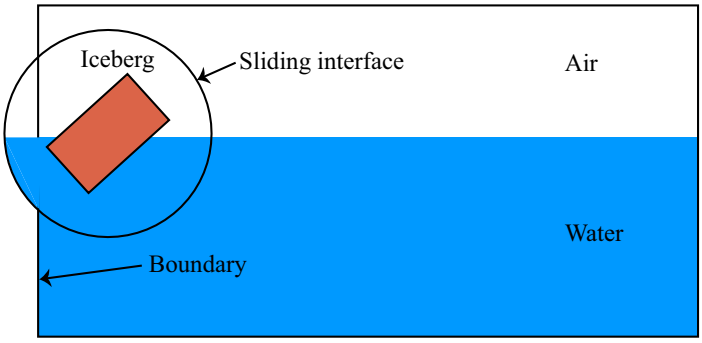


Figure 2.24 Block intruding into the water boundary

2.5.2.2 Immersed Boundary Method (IBM)

In order to avoid the remeshing problem in ALE, the IBM has been proposed and developed during the last few decades (Hieber et al., 2008; Mittal et al., 2008). This method will be applied within this study. IBM was originally proposed by Peskin (1972), who simulated blood flow in a cardiovascular system with a complex geometry. This method considers the solid body as a boundary immersed in the background mesh (Fig. 2.25), meaning the boundary of the solid body is represented by the domain cells. The new boundary of the solid body may not always be conformal to its original shape, but there is no need to perform the associate remeshing when the solid body is moving. Further, this method can also maintain good quality of a mesh when dealing with complex boundaries compared with the body-fitted ALE approach.

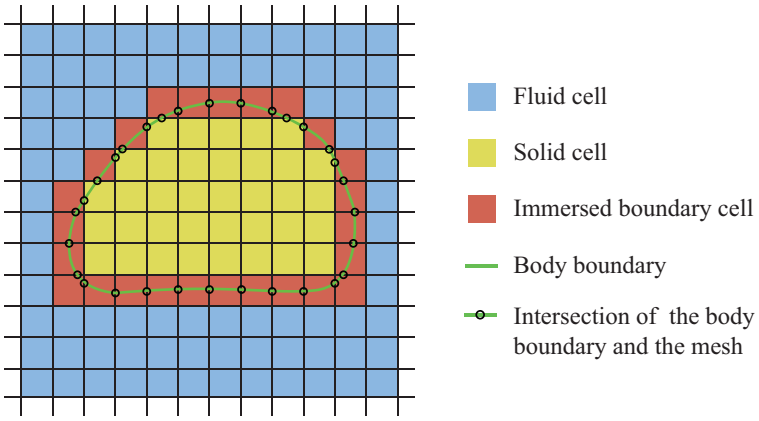


Figure 2.25 The boundary of the solid body (green lines) is represented by the background mesh (brown cells) when using IBM (Chen et al., 2020)

There were a number of attempts to improve IBM. Depending on whether the force (used to model FSI) is added before or after the discretisation of the governing equations, these methods can be categorised into two groups: continuous and discrete forcing IBM (Mittal and Iaccarino, 2005). The IBM proposed by Peskin (1972) is an example of the continuous forcing IBM. Another representative example is the feedback forcing method proposed by Goldstein et al. (1993), who applied feedback forcing to represent a solid body. However, these methods could result in spurious oscillations and require strict Courant-Friedrichs-Lewy (CFL) conditions. In order to improve the continuous forcing IBM, Mohd-Yusof (1997) developed a discrete forcing IBM, with the forcing being introduced at the discretisation stage and obtained by a spectral method, thus allowing higher CFL values in computations. Additionally, this method can also lead to a more stable and efficient algorithm (Mittal and Iaccarino, 2005). Based on previous studies, Jasak et al. (2014) implemented a discrete forcing with a direct boundary condition imposition IBM toolbox in Foam-extend (one of the OpenFOAM forks), and some validation tests show that they are in a good agreement with the body-fitted mesh cases.

2.5.2.3 Meshless methods

Meshless approaches to model FSI have also been proposed. Meshless means that no mesh is required in the simulation and both the fluid and structure in the domain are regarded as a series of nodes or particles. The FSI is represented

by the interaction among these nodes or particles. The physical properties of the original continuum system can be obtained by using interpolation functions on these discrete particles. This meshless Lagrangian feature enables the simulation of large motions of fluids and large deformation or displacement of structures (Rabczuk et al., 2010).

There are a selection of meshless methods which were successfully applied to model FSI, such as SPH (Antoci et al., 2007; Sun et al., 2018), the fast multipole method (Wilkes et al., 2017), Galerkin methods (Wang and Belytschko, 2009) and the immersed particle method (Rabczuk et al., 2010). Among these methods, SPH is a widely used and effective approach in hydraulics, coastal and offshore engineering.

SPH was originally developed for astrophysical gas dynamics problems (Monaghan, 1992). The idea of SPH is to decompose the fluid into a number of discrete elements, nodal points or particles. With the help of a kernel function, the physical properties can be smoothed over a set of neighbouring particles within a certain spatial distance. Based on this method, the fluid simulation can then become an multi-body problem with the interaction among these discrete particles (Monaghan and Kos, 2000).

This method was later successfully applied to a wider range of simulations involving incompressible fluids (Monaghan and Kos, 1999; 2000) and FSI. Both the solid and fluid phases can be simulated by SPH. Monaghan and Kos (2000) used SPH to model the Scott Russell′s wave generator (Section 2.4.2.1) and found a good agreement with their experimental results. Antoci et al. (2007) modelled FSI using the pressure force calculated by the integral of pressure among the interface particles. SPH can also be coupled with another method to model FSI such as the Discrete Element Method (DEM) (Wu et al., 2016; Tan and Chen, 2017) and FEM (Yang et al., 2012), where SPH is usually used for the fluid phases. Fig. 2.26 shows an example of FSI simulated by a SPH-DEM coupling method. An iterative process is performed to obtain the drag force and buoyancy (solved by SPH) to update the position of the solid body (governed by DEM).

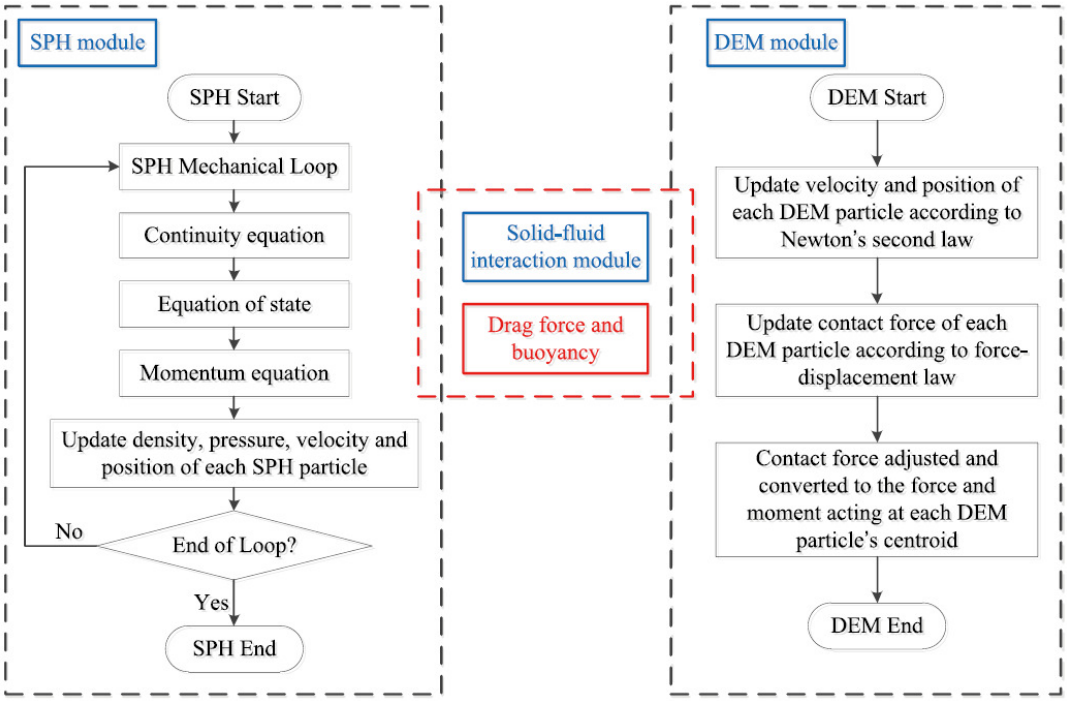


Figure 2.26 Sketch of the coupling strategy in a SPH-DEM hybrid model (Tan and Chen, 2017)

Although SPH has been successfully applied to simulate FSI, SPH is computational expensive due to its particle-based Lagrangian nature. Therefore, it is mainly restricted to small-scale problems. A further weakness of SPH is its numerical energy dissipation, which often results in an overestimation of the wave decay in wave propagation (Violeau and Rogers, 2016).

2.5.2.4 Lattice-Boltzmann Method (LBM)

The LBM was developed as an alternative method in CFD in the 1990s (Benzi et al., 1992). It is based on particles over a discrete Lattice mesh and solves the microscopic kinetic Boltzmann equations.

$$f_\alpha(x + c_\alpha \Delta t, t + \Delta t) - f_\alpha(x,t) = -\frac{1}{\tau}(f_\alpha(x,t) - f_\alpha^{eq}(x,t)) + F_\alpha \Delta t, \qquad (2.29)$$

where f_α indicates the single-particle distribution function, c_α the discrete velocity, Δt the time step, τ the dimensionless relaxation time and F_a a forcing term. The variables such as velocity and density in a macroscopic scale can then be obtained using f_α. LBM has been successful in dealing with flows around

complex boundaries and interfacial dynamics (Chen and Doolen, 1998). Further, this numerical scheme is relatively easy to program and can make good use of parallel computing (He et al., 1996), especially using Graphic Processor Units (GPUs).

There are two groups of LBM simulating FSI: Bounce-Back methods and IB-LBM, short for IBM for LBM (Liberge and Béghein, 2019). The first group treats solid bodies as a bouncing back boundary condition. A number of studies focus on the improvement of the Bounce-Back methods such as Krüger et al. (2017), Ladd and Verberg (2001) and Noble et al. (1998). However, the disadvantages, such as the high computational cost due to detecting boundary points and the challenge to capture the solid volume fraction, have not been fully resolved (Liberge and Béghein, 2019). The second group (IB-LBM) combines the LBM for flow simulation with IBM for handling moving bodies, considering advantages from both methods. It has been demonstrated to be an effective approach to model FSI, e.g. for sediment processes (Feng et al., 2004), flapping wing (Kimura et al., 2014) and cells flow in a microvessel (Xu et al., 2013), although some drawbacks in computational efficiency still exist (Liberge and Béghein, 2019).

2.5.3 Numerical methods to model IBTs

This Section 2.5.3 includes a discussion of potential numerical methods for simulating IBTs as well as subaerial landslide-tsunamis. As this work aims to investigate both the generation and propagation of IBTs, the potential numerical methods should have the ability to model large complex displacements of icebergs and long-distance propagation of tsunamis. The IBM, SPH and LBM may be good options for modelling FSI given their natural advantages in handling moving boundary problems. Note that IBM has an advantage in maintaining a good mesh quality but the represented iceberg is not fully identical to the original geometry of the iceberg. SPH can reliably be applied for handling the large displacement of the iceberg due to its particle-based nature, and thus may simulate IBT generation well. However, its numerical dissipation will often result in a strong overestimation of the decay for wave propagation (Violeau and Rogers, 2016). LBM can deal with complex flows, but it is computationally expensive when facing large-scale simulations (Qiu et al., 2019). In this study,

IBM is selected for simulation of FSI.

Only models based on the NSEs (e.g. DNS, LES and RANS) and higher-order Boussinesq equations are discussed. Methods such as the depth-averaged Boussinesq equations (e.g. Madsen et al., 1991) and shallow-water equations (e.g. Lynett and Liu, 2002) are suitable to simulate tsunami propagation only such that they will not be further reviewed herein.

2.5.3.1 Direct Numerical Simulation (DNS)

The Direct Numerical Simulation (DNS), also known as Direct Navier-Stokes, is the method in CFD which solves the entire NSEs, without simplified assumptions or a turbulence model, as in other CFD approaches. The NSEs read

$$\frac{\partial u_i}{\partial t} + u_j \frac{\partial u_i}{\partial x_j} = -\frac{1}{\rho}\frac{\partial p}{\partial x_i} + \frac{\mu}{\rho}\frac{\partial^2 u_i}{\partial x_j^2} + f_i, \tag{2.30}$$

where u_i and u_j denote the velocity component, p the fluid pressure, f_i represents body forces per unit mass acting in each direction and μ the fluid dynamic viscosity.

However, the memory requirement grows fast with increasing Reynolds number in a DNS. In addition, the advance of the solution in time should be small enough to obtain accurate results (Xu, 2007). Therefore, the DNS method is computationally expensive. Due to these reasons, not many studies used a DNS to simulate landslide-tsunamis thus far. An exception is Abadie et al. (2010). They proposed a model named Thetis based on DNS, where the slide, water and air were treated as fluids and the Volume of Fluid (VoF) method was adopted to track the air-water, slide-air and slide-water interfaces. Block slides were accurately reproduced in the model and showed a good agreement with previous studies (e.g. Monaghan and Kos, 2000; Liu et al., 2005).

2.5.3.2 Large Eddy Simulation (LES)

The Large Eddy Simulation (LES) is a mathematical model in CFD where only the large scale turbulent motions are resolved via the NSEs, and the smaller ones are modelled with a turbulence model (Mason, 1994). LES is a less computational expensive alternative to the DNS. It was initially proposed by Smagorin-

sky (1963) and improved by Deardorff (1970) who applied a spatial filter to the NSEs. Liu et al. (2005) proposed a LES model to study the impulse waves and run-up and run-down generated by a sliding wedge to simulate their 3D laboratory tests. The wave probes were placed at different locations along the seaside and lateral side of the slope, e.g. the seaside distances of wave probes from the slope is $0.20d$ to $0.51d$ in Fig. 2.27. A satisfactory agreement between the numerical and laboratory results for the time histories of subaerial landslide-tsunamis was achieved (Fig. 2.27).

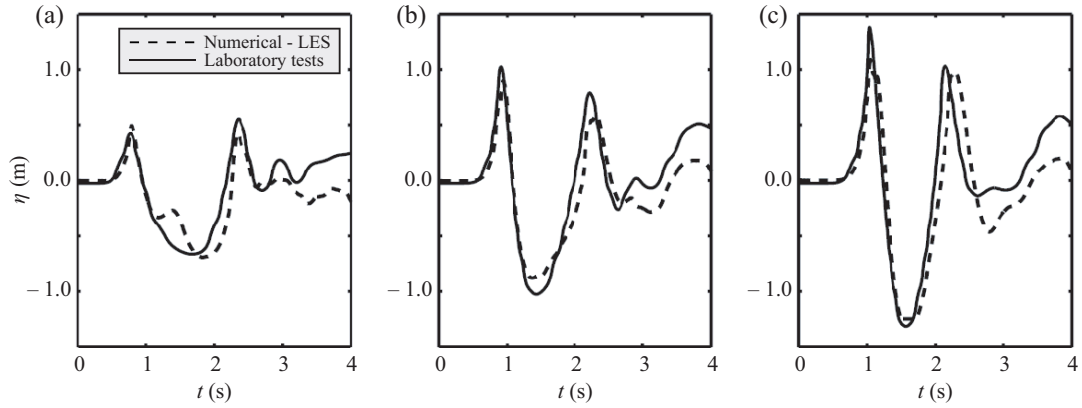


Figure 2.27 Numerical (LES) and laboratory subaerial landslide-tsunamis at different wave probes (adapted from Liu et al., 2005)

2.5.3.3 Reynolds-Averaged Navier-Stokes (RANS) equations

As the name suggests, this method is based on the Reynolds-Averaged Navier-Stokes (RANS) equations with a turbulence model to close the scheme such as the standard or ReNormalization Group (RNG) k-ϵ turbulence models. The idea behind these equations is the Reynolds decomposition, i.e. the velocity is decomposed into a time averaged mean and its fluctuation and most of the fluctuating velocity terms cancel out in the RANS equations. The RANS equations give then approximate time-averaged solutions to the NSEs (Section 2.5.3.1). For an incompressible Newtonian fluid governed by the NSEs as applied herein, the RANS equations can be written in Einstein notation as

$$\frac{\partial \overline{u_i}}{\partial x_j} = 0, \tag{2.31}$$

$$\overline{u_j}\frac{\partial \overline{u_i}}{\partial x_j} = \overline{f_i} + \frac{1}{\rho}\frac{\partial}{x_j}[-\overline{p}\delta_{ij} + \mu(\frac{\partial \overline{u_i}}{\partial x_j} + \frac{\partial \overline{u_j}}{\partial x_i}) - \rho\overline{u_i' u_j'}]. \tag{2.32}$$

Eq. (2.31) is the continuity equation for incompressible flows. In Eq. (2.32) $^-$ is the time-average operator and $\rho\overline{u_i' u_j'}$ is the Reynolds stress. The RANS equations can be solved with many standard CFD codes and software such as FLOW-3D, COBRAS, Fluent and OpenFOAM. Turbulence in tsunami generation and propagation is then modelled via a turbulence model (Das et al., 2009).

A number of researchers applied the RANS equations to simulate landslide-tsunamis, e.g. Das et al. (2009) simulated subaerial and submarine landslide-tsunamis. They investigated the time series of fluid motion, water surface elevation and wave run-up and found that this method can model the wave profiles well both for tsunami generation and propagation. Biscarini (2010) proposed a model based on the RANS with the RNG k-ϵ turbulence closure scheme, which is able to maintain the water surface better as a sharp front than the standard k-ϵ turbulence model. The model was then used to simulate the 1958 Lituya Bay tsunami event in a simplified 2D geometry resulting in a good agreement with observations.

2.5.3.4 Higher-order Boussinesq equations

The Boussinesq-type equations are restricted to describe fairly nonlinear water waves, and are as such useful to model landslide-tsunamis, which are typically highly-nonlinear. To simulate this nonlinear wave propagation through a rapidly varying bathymetry, Madsen et al. (2006) first proposed a higher-order Boussinesq method with high-accuracy, and this method contained the kinematic and dynamic free surface conditions as

$$\frac{\partial \eta}{\partial t} = \mathbf{W}(1 + \nabla\eta \cdot \nabla\eta) - \mathbf{u} \cdot \nabla\eta, \tag{2.33}$$

$$\frac{\partial \mathbf{u}}{\partial t} = -g\nabla\eta - \nabla(\frac{\mathbf{u} \cdot \mathbf{u}}{2} - \frac{\mathbf{u}^2}{2}(1 + \nabla\eta \cdot \nabla\eta)). \tag{2.34}$$

In Eqs. (2.33) and (2.34) $\mathbf{u} = (u_x, u_z)$, u_x and u_z denote the horizontal and vertical velocities on the free water surface η and $\mathbf{g}$ is the gravitational acceleration vector.

Fuhrman and Madsen (2009) extended this method to model the generation, propagation and run-up of landslide-tsunamis. 2D positive and negative

impulsive bottom movements were used to generate tsunamis. Further, they added an additional term to consider bottom fluctuations in Eq. (2.34), hence, this method may retain a high accuracy of nonlinear waves over various water depths. Regarding tsunami run-up, an algorithm was developed by Fuhrman and Madsen (2008), which was an extrapolating boundary type (wet and dry region) method. The results of this model were compared with findings of previous studies (e.g. Lynett and Liu, 2002; Hammack, 1973) and showed an excellent agreement.

2.5.3.5 Selection of numerical methods to model IBTs

A DNS simulation is too computational expensive for both wave generation and propagation. The LES as a simplification of the DNS still seems unsuitable for large-scale flow fields due to the requirements of excessive computational resources. Further, the higher-order Boussinesq equations model is commonly used for tsunami propagation over a large domain with varying bathymetry, but is less well suited for tsunami generation. Therefore, the RANS model is selected for the modelling of IBTs. The RANS equations can be discretised with FDM, FEM and FVM within an Eulerian mesh. The FDM directly approximates terms in partial differential equations (PDEs) using Taylor series expansion and then converts the PDEs into a series of algebra equations (Smith, 1985). In FEM a basis function is applied to discretise the PDEs. The variables are then expressed by the basis function such that the PDEs are reconstructed into its weak form (Babuška et al., 2004). Similarly, the PDEs are finally converted into a system of algebra equations. In contrast to FDM, the discretisation based on FVM requires the integration of the solved PDEs over each control volume prior to each term approximated by using Taylor series (Moukalled et al., 2016). Since FVM applies volume integrals over the conservative equations, the inherent conservation characteristics makes FVM the preferred method in CFD (Moukalled et al., 2016).

In conclusion, the RANS solved with the FVM based on IBM for modelling FSI will be used in this work to simulate IBTs. This appears to be a suitable method capable of dealing with both tsunami generation and propagation with sufficient accuracy. Therefore, the commonly used FVM based on the open source code OpenFOAM (Foam-extend) is used for the simulation of IBTs. Its

wide application and advantages will be discussed in Section 2.5.4.

2.5.4 OpenFOAM

The OpenFOAM (short for Open Source Field Operation and Manipulation) software is a C++ object oriented library designed for computational continuum mechanics. The OpenFOAM distribution contains a selection of solvers and utilities addressing a broad range of CFD problems. With the help of the OpenFOAM library, a number of complex physical models can be implemented efficiently and flexibly (Jasak 2009).

OpenFOAM has three mainstream forks: OpenFOAM (owned by the OpenFOAM Foundation), Foam-extend (maintained by Wikki Ltd.) and OpenFOAM+ (maintained by the ESI Group). These OpenFOAM forks have been applied to a number of problems in fluid dynamics. Jacobsen et al. (2012) implemented a generic wave generation and absorption approach (wave relaxation zones) to model wave propagation and breaking in a widely used solver named waves2Foam in Foam-extend. Higuera et al. (2013a,b) implemented the boundary conditions for the generation and absorption of realistic waves. Further, they presented a robust numerical model for applications in ocean engineering.

The capabilities of OpenFOAM in coastal engineering were confirmed by Morgan et al. (2011), who correctly reproduced laboratory results of the regular waves propagating over a submerged breakwater. Douglas and Nistor (2015) modelled the force of tsunamis acting on a free-standing structure using OpenFOAM and investigated the effect of bed conditions on the structure loading. Regarding the applications in marine renewable energy, Davidson et al. (2015) implemented a numerical wave tank to simulate wave-structure interactions and the hydrodynamic performance of a number of wave energy converters including floating types. Wu and Nguyen (2017) simulated the rotor motion of a flow-induced floating wind turbine platform by using an implemented motion solver which can handle a sliding mesh. Good agreements with other CFD codes were obtained. However, the handling of large displacements of rigid bodies has not yet been fully resolved in OpenFOAM. Considering the criteria of modelling IBTs in Section 2.5.3.5 and the already available IBM toolbox developed by Jasak et al. (2014), Foam-extend will be used in this study.

2.6 Summary

Different iceberg calving mechanisms have been introduced in this chapter, namely (A) capsizing, (B) gravity-dominated fall, (C) buoyancy-dominated fall, (D) gravity-dominated overturning and (E) buoyancy-dominated overturning. Field observations, laboratory tests, numerical simulations and analytical modelling of iceberg-tsunamis (IBTs) have been reviewed in Section 2.2. Despite of iceberg calving events in nature generating IBTs with up to 45 to 50 m in amplitude, posing a threat to coastal communities, not many studies have been conducted into IBTs thus far. Further, no study focusing on the numerical modelling of IBT generation and propagation is available.

The basic wave theories have been reviewed in Section 2.3. Landslide-tsunamis are related to IBTs, such that they have also been reviewed (Section 2.4). This included the effects of the slide type, water body geometry and bathymetry and the generated wave type, which are relevant for the present work into IBTs. The modelling of Fluid-Structure Interaction (FSI) with e.g. the Arbitrary Lagrangian Eulerian (ALE) method, the Immersed Boundary Method (IBM), meshless methods and the Lattice-Boltzmann Method (LBM), which are key to numerically simulate floating bodies, have been discussed. Regarding the fluid governing equations, the Direct Numerical Simulation (DNS), Large Eddy Simulation (LES), Reynolds-Averaged Navier-Stokes (RANS) equations and higher-order Boussinesq equations model have been reviewed and compared. Based on the requirements to numerically model IBTs, the RANS equations based model with IBM for simulating FSI has been selected. Finally, the open source code OpenFOAM has been reviewed and one of its mainstream forks, Foam-extend, was identified as the best option to model IBTs and will be applied in the main part of this work.

Chapter 3

NUMERICAL MODEL

3.1 Introduction

This chapter presents the numerical method based on Foam-extend 4.0 to simulate iceberg-tsunamis (IBTs). Foam-extend provides two categories of applications: solvers and utilities. A solver is designed for solving the governing equations for a specific Computational Fluid Dynamics (CFD) problem, while an utility serves as a general data manipulation task, e.g. mesh generation and post-processing. Each type of application may involve several libraries, consisting of classes and functions. The newly implemented solvers and libraries support are presented in this chapter.

As discussed in Section 2.5, IBM is used to simulate Fluid Structure Interaction (FSI). For the fluids, the multiphase flow (water and air in this study) is resolved on a structured Cartesian background mesh. For the structure, a tailored motion solver is used to handle the movement of the iceberg. The flow and motion solver are then coupled with the help of a new dynamic mesh handling class.

Details of the IBM toolbox (Jasak et al., 2014) provided in Foam-extend 4.0 are described in Section 3.2. In Section 3.3 the governing equations for multiphase flows and their solving processes are presented. The motion solver is introduced in Section 3.4 including the calculation of the acceleration, velocity and position of the iceberg under different calving mechanisms. In Section 3.5, the coupling strategy of these two solvers and the implementation of the new dynamic mesh handling class are focused on, followed by the numerical model workflow in Section 3.6. Finally, the summary of the numerical model is presented in Section 3.7.

3.2 IBM Toolbox in Foam-extend

3.2.1 Introduction

The IBM toolbox in Foam-extend 4.0 has been selected and used in this study to model FSI. This IBM toolbox is a discrete forcing method with direct boundary condition imposition IBM (Section 2.5.2.2). The discrete forcing implies that the existence of the immersed boundary (IB) can be ignored at the stage of the discretisation of governing equations and after that the force of the IB on the flows will be considered (Jasak et al., 2014). The direct boundary condition imposition means that the boundary condition of the IB is directly imposed by evaluating the values of the IB. There are four important modules in this IBM toolbox: IB boundary type, Moving IB support, IB force calculation and IB wall functions.

3.2.2 IB boundary type

The source code of this module is located at */src/immersedBoundary/immersedBoundary*. As a boundary type, IB has been implemented within this module for storing the geometry information of the IB and providing support for solving the governing equations using the Finite Volume Method (FVM). Figure 3.1 illustrates the main concepts used in the implementation of the IB type. The original rigid body boundary assigned as the IB type is represented by the IB cells. The outward faces of the IB cells are the IB faces. Since the solid cells (inside the IB cells) are of no interest and the governing equations are therefore not applied, the solid cells are called dead cells. The fluid cells together with the IB cells (outside the IB cells) are called live cells.

The recognition of the IB cells is achieved by generating a mask that marks live and dead cells based on the geometry information of the rigid body boundary, which is obtained from a STereoLithography (.STL) file. This mask is generated with the help of the function *triSurfaceSearch*, evaluating whether a point is an internal point of the IB or not.

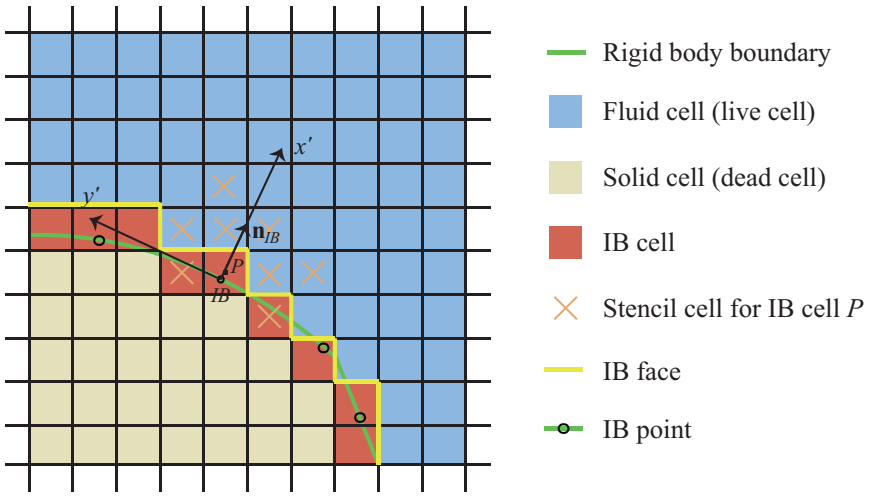


Figure 3.1 Illustration of the main concepts in the implementation of the IB type (adapted from Jasak et al., 2014)

For imposing the boundary conditions after the representation of the rigid body boundary, a cell stencil is needed for a IB cell P to evaluate the values of the IB. The value of the IB cell is obtained by using the interpolation of its stencil cells values and the corresponding boundary value at the IB point. Here, two types of boundary conditions are implemented: Dirichlet and Neumann boundary conditions. For the Dirichlet boundary condition

$$\begin{aligned}\phi_P &= \phi_{IB} + C_0(x_P - x_{IB}) + C_1(y_P - y_{IB}) + C_2(x_P - x_{IB})(y_P - y_{IB}) \\ &\quad + C_3(x_P - x_{IB})^2 + C_4(y_P - y_{IB})^2,\end{aligned} \tag{3.1}$$

and for the Neumann boundary condition

$$\phi_P^{'} = C_0 + [\mathbf{n}_{IB} \cdot (\nabla\phi)_{IB}]x_P^{'} + C_1 y_P^{'} + C_2 x_P^{'} y_P^{'} + C_3(x_P^{'})^2 + C_4(y_P^{'})^2. \tag{3.2}$$

In Eqs. (3.1) and (3.2) ϕ_P and $\phi_P^{'}$ denote the values of the IB cell P under Dirichlet and Neumann boundary conditions, respectively. ϕ_{IB} is the value at the IB point and $x_P^{'}$ and $y_P^{'}$ are the local coordinates with origin at the IB point. The x-axis is directed along the normal vector of the rigid body boundary at the IB point ($\mathbf{n}_{IB}$) and the y-axis the tangential vector. The coefficients C_0, C_1, C_2, C_3 and C_4 are obtained through a least square polynomial fit (Jasak et al., 2014) which is controlled by the following parameters: the radius factor, angle factor, *maxCellCellRows* and the distance factor. The variable names of theses

parameters in this toolbox and its corresponding functions are shown in Table 3.1.

Table 3.1 Parameters and their functions in the least square polynominal fitting function

Parameter	Variable name	Function
Radius factor	*radiusFactor_*	Determines the distance to consider cells in the fitting function
Angle factor	*angleFactor_*	Determines the angle to consider cells in the fitting function
Searching span	*maxCellCellRows_*	Finds the nearest cell within the searching span
Distance factor	*distFactor_*	Collects the IB points

3.2.3 Moving IB support

This section presents the module responsible for the movement of the IB. It serves as a bridge between the IB and some already available prescribed motion handling methods in Foam-extend 4.0, which will be expanded in this study to handle both resolved and additional prescribed moving IB problems (Section 3.5). The source code files in the folder */src/immersedBoundary/immersedBoundaryDynamicMesh/* are shown in Fig. 3.2.

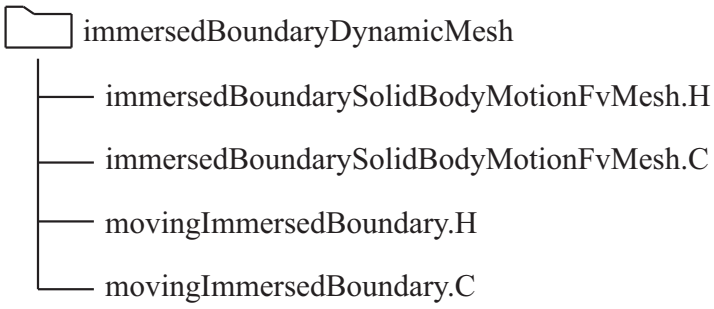


Figure 3.2 Source code files in the moving IB handling class

In Foam-extend the dynamic mesh handling method should be specified in *dynamicMeshDict* under a case folder before running the simulation. The class *immersedBoundarySolidBodyMotionFvMesh* provides a runtime selectable function that can be chosen in *dynamicMeshDict* to handle the moving IB problem, while the specific moving IB handling method is implemented in *movingImmersedBoundary*. Here, the class *movingImmersedBoundary* is the main focus.

In order to apply the already available dynamic mesh handling methods in Foam-extend, the pointer *sbmfPtr_* is used to call the corresponding functions. After specifying the motion type, the current positions of the IB points is stored in the variable *oldIbPoints*. The new positions are then calculated by the called motion functions such as translation, rotation or linear oscillation. Since this IBM toolbox treats the IB as the rigid body boundary, the no-slip boundary condition is applied to update the boundary velocity. This will then be used in the evaluation of IB cells (Section 3.2.2), i.e. the updated boundary velocity is the term ϕ_{IB} in Eq. (3.1).

3.2.4 IB force calculation

As the name suggests, the IB force calculation module is used to obtain the force acting on the IB, which plays a key role in the motion solver to determine the resolved motion. In this IBM toolbox, only the pressure and viscosity forces are considered. There are two methods to calculate these two kind of forces: based on the given force density and based on the pressure and velocity data. The former one is apparently not suitable for resolved motion, as the force density cannot be known or given prior to the simulation. Thus, the latter one is addressed in this subsection.

The IB force calculation is implemented in the *immersedBoundaryForces*, where the output results contain both forces and their corresponding moments in relation to the centre of mass. In analogy with the force calculation in the body-fitted mesh, the pressure force of the IB is obtained by directly integrating the pressure force on all boundary faces of the rigid body. Note that in the IBM the rigid body boundary is represented by the background mesh, by which the volume enclosed by the IB faces is usually different from the rigid body's original volume. This will effect some of the results as further discussed in Chapter 5.

For the calculation of the viscosity force, the velocity gradient at the IB is used to obtain the viscosity force, which can be achieved by directly calling the IB member function *triGrad()*. Further, the density and kinematic viscosity are also required to be specified before the simulation. Note that when using the Volume of Fluid (VoF) method to deal with multiphase flows, the phase fraction α should be used especially to calculate the force around the water-air interface.

3.2.5 IB wall functions

In this IBM toolbox, the Reynolds-Averaged Navier-Stokes (RANS) equations with the standard k-ϵ turbulence model is applied to close Eqs. (2.31) and (2.32) as illustrated in Section 2.5.3.3. Note that in turbulent flows, the boundary layer requires a fine mesh to obtain high accuracy results, e.g. with a cell dimension of 10^{-6} of the domain height (Durbin et al., 2001). The *Wall function* is introduced to reduce the mesh resolution to approximate the results near the boundary layer and to be applicable in high Reynolds number flows (Jasak et al., 2014).

The wall function module is located at */src/immersedBoundary/immersedBoundaryTurbulence/wallFunctions/*, mainly including the classes *immersedBoundaryEpsilonWallFunctionFvPatchScalarField*, *immersedBoundaryVelocityWallFunctionFvPatchVectorField* and *immersedBoundaryWallFunctionFvPatchField*. These three classes correspond to three different boundary conditions in the RANS equations with the k-ϵ turbulence model. *ImmersedBoundaryEpsilonWallFunctionFvPatchScalarField* is implemented for updating the rate of dissipation of turbulent kinetic energy ϵ values at the IB cells, requiring the dimensionless wall distance y plus (y^+), the turbulent viscosity μ_t, the turbulent kinetic energy k, the shear stress and the tangential velocity of the IB values. After obtaining the ϵ values at the IB, the velocity of the IB cells (and the velocity gradient) can be determined by using the wall function and will then be used for force calculation in turbulent flows, i.e. *immersedBoundaryVelocityWallFunctionFvPatchVectorField* is used for the velocity boundary condition of the IB. The remaining boundary variables of the IB, such as μ_t and k in Eqs. (2.31) and (2.32), are updated by the class *immersedBoundaryWallFunctionFvPatchField*.

3.3 Flow Solver

3.3.1 Introduction

A flow solver in Foam-extend is an executable program including the main function and all the steps in a simulation. The multiphase flow solver with IBM support *interIbFoam* provided in Foam-extend 4.0 is extended in this study to enable the handling of moving IB problems, and the new solver is then coupled

with a tailored motion solver. The governing equations for the multiphase flow are given first, followed by the description of the equation discretisation and solving steps. Finally, the structure of the newly implemented flow solver is presented.

3.3.2 Governing equations

The RANS equations are used to describe the multiphase flows (water and air) in this study, which are assumed to be incompressible and viscous Newtonian fluids. Note that within this IBM toolbox framework, the flow solver solves the governing equations in a structured Carthesian coordinate system. Here only the equations solved in a general three dimensional (3D) Carthesian coordinate system are presented. The specific coordinates used in this study can be seen in the numerical setup in the Results chapter. Mass and momentum conservation for fluids are given as

$$\nabla \cdot \mathbf{u} = 0, \tag{3.3}$$

$$\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla)\mathbf{u} = -\frac{1}{\rho}\nabla p + \frac{\mu + \mu_t}{\rho}\nabla^2 \mathbf{u} + \frac{1}{\rho}\mathbf{g}, \tag{3.4}$$

where $\mathbf{u} = (u_x,\ u_y,\ u_z)$ denotes the fluid velocity vector and p the pressure, which are the two unknown variables to be resolved. In Eqs. (3.3) and (3.4) $\nabla = (\frac{\partial}{\partial x},\ \frac{\partial}{\partial y},\ \frac{\partial}{\partial z})$ is the differential operator, $\mathbf{u}\cdot\nabla = u_x\frac{\partial}{\partial x} + u_y\frac{\partial}{\partial y} + u_z\frac{\partial}{\partial z}$ the dot product, $\nabla^2 = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} + \frac{\partial^2}{\partial z^2}$ the Laplace operator, ρ indicates the density, μ the dynamic viscosity, μ_t the turbulent viscosity ($\mu_t = 0$ for laminar flows) and $\mathbf{g}$ is the gravitational acceleration vector with the magnitude $|\mathbf{g}| = 9.81\ \mathrm{m/s^2}$.

In order to distinguish the two fluids and track the water-air interface, the VoF method is applied. The phase fraction α is introduced as

$$\alpha = \begin{cases} 0, & \text{air} \\ \alpha_f\ (0 < \alpha_f < 1), & \text{interface} \\ 1, & \text{water} \end{cases} \tag{3.5}$$

i.e. $\alpha = 0$ denotes that the cell is filled with air, for $\alpha = 1$ the whole volume of the cell is filled with water and $0 < \alpha < 1$ specifies the interface between the two fluids. The physical parameters, such as ρ and μ of the two fluids for a cell,

are then evaluated in function of α as

$$\rho = \rho_w \alpha + \rho_a (1 - \alpha), \tag{3.6}$$

$$\mu = \mu_w \alpha + \mu_a (1 - \alpha), \tag{3.7}$$

where the subscripts a and w indicate air and water, respectively. Since the flow field may be changed during the simulation, α also needs to be changed. Here, a transport equation is applied to update α over time once the velocity field is obtained:

$$\frac{\partial \alpha}{\partial t} + \nabla \cdot (\alpha \mathbf{u}) + \nabla \cdot [\alpha(1 - \alpha)\mathbf{u}] = \mathbf{0}, \tag{3.8}$$

where the term $\nabla \cdot [\alpha(1-\alpha)\mathbf{u}]$ is used to sharpen the air-water interface (Weller et al., 1998).

3.3.3 Solving process

Directly obtaining the analytical solution for the partial differential equations (PDEs) in Section 3.3.2 is challenging. In Foam-extend, the PDEs are solved using the FVM. This method requires first to integrate the equations over time and volume and then to discretise the integral form of governing equations into a series of algebraic equations based on its computational domain.

3.3.3.1 Equation discretisation

The implementation of PDE solvers in Foam-extend considers a general transport equation for a variable ϕ such that the discretisation process can be compatible with many similar equations (Jasak, 1996):

$$\frac{\partial \rho \phi}{\partial t} + \nabla \cdot (\rho \mathbf{u} \phi) - \nabla \cdot (\rho \gamma_\phi \nabla \phi) = S_\phi(\phi), \tag{3.9}$$

where γ_ϕ is the diffusion coefficient. For this study, the mass conservation equation (Eq. 3.3) can be obtained by specifying $\phi = 1$ and $S_\phi = 0$ in Eq. (3.9), and the momentum conservation equation (Eq. 3.4) is represented if $\phi = \mathbf{u}$ and $S_\phi = -\nabla p + \mathbf{g}$. There are 4 terms in Eq. (3.9), which are the temporal term $\frac{\partial \rho \phi}{\partial t}$, convection term $\nabla \cdot (\rho \mathbf{u} \phi)$, diffusion term $\nabla \cdot (\rho \gamma_\phi \nabla \phi)$ and source term $S_\phi(\phi)$, respectively.

The integral form of Eq. (3.9) over a control volume V_P containing a point P (cell P) from time t to $t + \Delta t$ is

$$\int_t^{t+\Delta t} [\frac{\partial \rho\phi}{\partial t} + \nabla \cdot (\rho \mathbf{u}\phi) - \nabla \cdot (\rho\gamma_\phi \nabla\phi)] dV_P dt = \int_t^{t+\Delta t} S_\phi(\phi) dV_P dt. \quad (3.10)$$

The temporal term becomes

$$\int_t^{t+\Delta t} \frac{\partial \rho\phi}{\partial t} dV_P dt = \int_t^{t+\Delta t} \frac{\partial \rho\phi}{\partial t} V_P dt, \quad (3.11)$$

and the source term

$$\int_t^{t+\Delta t} S_\phi(\phi) dV_P dt = \int_t^{t+\Delta t} S_\phi(\phi) V_P dt. \quad (3.12)$$

For the convection and diffusion terms in Eq. (3.10) Gauss' theorem

$$\int_{V_P} \nabla \cdot \phi = \oint_{\partial V_P} \phi d\mathbf{S} = \sum_f \mathbf{S} \cdot \phi_f \quad (3.13)$$

is used. In Eq. (3.13) ∂V_P denotes the surface of V_P, $\mathbf{S}$ the surface vector pointing outward and the subscript f indicates the face value. The convection and diffusion terms are then

$$\int_t^{t+\Delta t} \nabla \cdot (\rho \mathbf{u}\phi) dV_P dt = \int_t^{t+\Delta t} \sum_f \mathbf{S} \cdot (\rho \mathbf{u})_f \phi_f dt = \int_t^{t+\Delta t} \sum_f F \cdot \phi_f dt, \quad (3.14)$$

$$\int_t^{t+\Delta t} \nabla \cdot (\rho\gamma_\phi \nabla\phi) dV_P dt = \int_t^{t+\Delta t} \sum_f \mathbf{S} \cdot (\rho\gamma_\phi \nabla\phi)_f dt, \quad (3.15)$$

where $F = (\rho \mathbf{u})_f$ is the mass flux. Note that in this study, the mesh is orthogonal and thus $(\nabla\phi)_f$ in Eq. (3.15) can be obtained as:

$$(\nabla\phi)_f = \frac{\phi_N - \phi_P}{|\mathbf{d}|}. \quad (3.16)$$

In Eq. (3.16), subscript N indicates the neighbouring control volume (cell N) and $\mathbf{d}$ is the position vector pointing from the point P to point N.

The integral form of the general transport equation (Eq. 3.10) becomes

$$\int_t^{t+\Delta t}[\frac{\partial \rho\phi}{\partial t}V_P + \sum_f \mathbf{S}\cdot(\rho\mathbf{u})_f\phi_f - \sum_f \mathbf{S}(\rho\gamma_\phi)_f\cdot\frac{\phi_N-\phi_P}{|\mathbf{d}|}]dt = \int_t^{t+\Delta t} S_\phi(\phi)V_P dt. \tag{3.17}$$

A further rearrangement of Eq. (3.17) depends on the selected discretisation scheme, such as implicit or explicit for the temporal term and the interpolation method for determining the face value. All the terms in Eq. (3.17) have already been implemented in Foam-extend (Table 3.2) and can be called when specifying the equations. In order to highlight the variable ϕ to be solved, Eq. (3.17) can be represented as a series of algebraic equations on the computational domain

$$a_P\phi_P^{\mathrm{n}} + \sum_N a_N\phi_N^{\mathrm{n}} = S_P^{\mathrm{n}}, \tag{3.18}$$

where a_P and a_N are the coefficients regarding the cell P and its neighbouring cells N and S_P is the source term at the cell P.

Table 3.2 Commonly used terms in the discretisation of PDEs in Foam-extend (adapted from Greenshields, 2015)

Term	**Implicity/Explicit**	**Expression**	**Function name**
Temporal	Implicit/Explicit	$\frac{\partial\phi}{\partial t}$ $\frac{\partial\rho\phi}{\partial t}$ $\frac{\partial}{\partial t}(\rho\frac{\partial\phi}{\partial t})$	ddt(phi) ddt(rho,phi) d2dt2(rho,phi)
Convection	Implicit/Explicit	$\nabla\cdot(\psi)$ $\nabla\cdot(\psi\phi)$	div(psi,scheme) div(psi,phi,word) or div(psi,phi)
Diffusion	Implicit/Explicit	$\nabla^2\phi$ $\nabla\cdot\gamma\nabla\phi$	laplacian(phi) laplacian(gamma,phi)
Divergence	Explicit	$\nabla\cdot\chi$	div(chi)
Gradient	Explicit	$\nabla\chi$ $\nabla\phi$	grad(chi) gGrad(phi) lsGrad(phi) snGrad(phi) snGradCorrection(phi)
Curl	Explicit	$\nabla\times\phi$	curl(phi)
Source	Implicit Implicit/Explicit	$\rho\phi$	Sp(rho,phi) SuSp(rho,phi)

3.3.3.2 Solving algorithm

As the RANS equations involve both $\mathbf{u}$ and p, it is challenging to solve them simultaneously. Therefore, these two unknown variables need to be decoupled. Several solving algorithms are available in Foam-extend: Pressure-Implicit with Splitting of Operators (PISO) by Issa (1986), Semi-Implicit Method for Pressure Linked Equations (SIMPLE) by Patankar and Spalding (1972) and PIMPLE (a combined algorithm of PISO and SIMPLE) by Habchi et al. (2013).

Herein, the PIMPLE algorithm is selected, solving the equations for $\mathbf{u}$ and p, respectively. Further, this PIMPLE algorithm allows for larger time steps when dealing with the moving boundary problem (Habchi et al., 2013). There are two loops in the PIMPLE algorithm: the outer control and the inner corrector. The outer control is responsible for determining the movement at the next moment in time when the tolerances of the iterated equations satisfy the given criteria. The advanced time step Δt is governed by the Courant-Friedrichs-Lewy (CFL) number

$$Co = \Delta t(\frac{u_x}{\Delta x} + \frac{u_y}{\Delta y} + \frac{u_z}{\Delta z}), \tag{3.19}$$

where u_x, u_y and u_z are the components of the velocity vector of a cell and Δx, Δy and Δz the cell dimensions along each axis direction. The CFL number Co should be less than 1 within the entire computational domain to ensure the stability of the solving process (Courant et al., 1967). Further, each loop of the outer control provides a velocity prediction, where the solved equation is the momentum equation (Eq. 3.4). Based on the discretised form of Eq. (3.18), Eq. (3.4) becomes

$$a_P \mathbf{u}_P^{\mathrm{n}} + \sum_N a_N \mathbf{u}_N^{\mathrm{n}} = R_P^{\mathrm{n}} - (\nabla p)_P^{\mathrm{n}}, \tag{3.20}$$

where the source term R_P^{n} contains the gravitational acceleration vector (body force) and coefficients calculated from the velocity $\mathbf{u}$ at the previous moment in time. Thus,

$$a_P \mathbf{u}_P^{\mathrm{n}} = R_P^{\mathrm{n}} - \sum_N a_N \mathbf{u}_N^{\mathrm{n}} - (\nabla p)_P^{\mathrm{n}} = H^{\mathrm{n}} - (\nabla p)_P^{\mathrm{n}}, \tag{3.21}$$

$$\mathbf{u}_P^{\mathrm{n}} = \frac{H^{\mathrm{n}}}{a_P} - \frac{\nabla p_P^{\mathrm{n}}}{a_P}, \tag{3.22}$$

where $H^{\mathrm{n}} = R_P^{\mathrm{n}} - \sum_N a_N \mathbf{u}_N^{\mathrm{n}}$.

The inner corrector mainly solves the pressure p. Note that the predicted velocity $\mathbf{u}$ does not consider the IB or its boundary condition. This predicted $\mathbf{u}$ will then be corrected after imposing the IB Dirichlet boundary condition. Therefore, if the cell P is a IB cell, then $\mathbf{u}_P$ is replaced with the interpolated value obtained from Eq. (3.1). Putting the corrected velocity $\mathbf{u}^*$ into the mass conservation Eq. (3.3) and using the discretised form results in

$$\sum_f \mathbf{S} \cdot (\frac{\nabla p^{\mathrm{n}}}{a_P})_f = \sum_f \mathbf{S} \cdot (\frac{H^{\mathrm{n}}}{a_P})_f. \tag{3.23}$$

Eq. (3.23) is the pressure equation, which is solved in the inner corrector loop. During each loop, the solved p^* is used to correct $\mathbf{u}^*$. The final form of the velocity $\mathbf{u}^{**}$ for a cell P is given as

$$\mathbf{u}^{**\mathrm{n}}_P = \frac{H^{\mathrm{n}}}{a_P} - \frac{\nabla p^{*\mathrm{n}}_P}{a_P}. \tag{3.24}$$

After obtaining the velocity $\mathbf{u}$ in the domain, Eq. (3.8) can be solved and the remaining variables depending on α, such as ρ and μ, are revaluated.

3.3.4 Solver structure

This subsection presents the structure of the newly implemented solver *interDyMIbFoam* and the solver details can be found in Appendix A. First of all, the initialisation is necessary, which includes processes such as reading the mesh file and control parameters, creating fields of solved variables and setting the initial time step. Then the simulation steps are implemented within a loop structure. In addition to specifying the governing equations and applying the solving algorithm presented in Section 3.3.3.2, the application of the IBM toolbox and coupling with a motion solver also play an important role. Finally, the solved variables needs to be output and recorded.

There are three successive modules in the loop structure of *interDyMFoam* (Fig. 3.3): IB recognition, PIMPLE loop and results output. IB recognition is achieved by *createIbMask.H*, including the functions to generate the IB mask field and to evaluate the IB boundary conditions as introduced in Section 3.2.2. Given that the solver deals with the moving IB, the position of the IB needs to be updated each moment in time. Therefore, the motion solver should be

called before the IB mask generation, which is achieved by using the function *mesh.update()* via the motion solver.

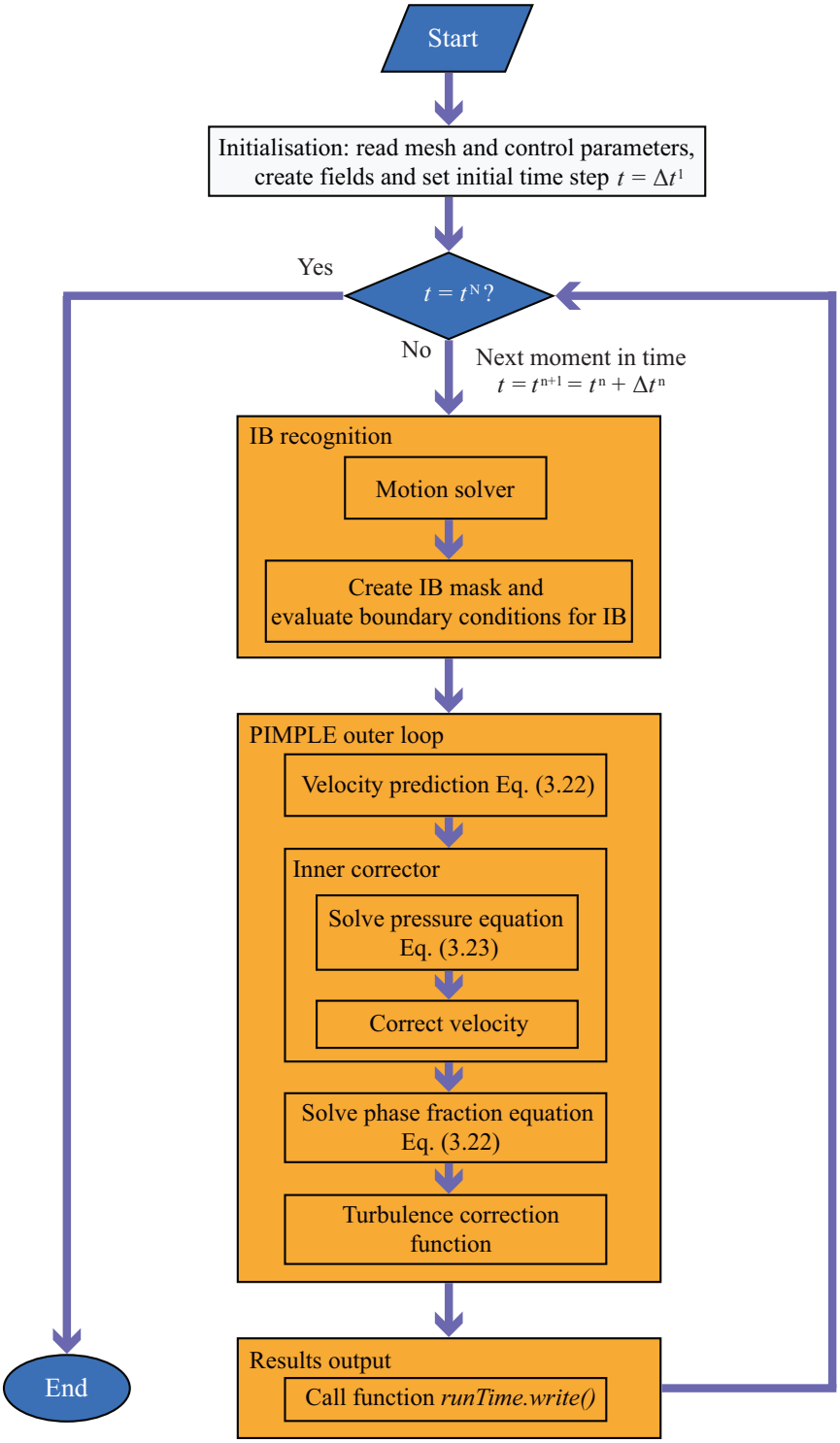


Figure 3.3 Flowchart of the newly implemented solver *interDyMIbFoam*

The PIMPLE loop (Section 3.3.3.2) focuses on the solving process of the governing equations. For the outer loop, the time step is controlled by *Co*. After solving the velocity equation (Eq. 3.22) and pressure equation (Eq. 3.23), a turbulence correction function can be called, if simulating a turbulent flow, using the RANS with the standard k-ϵ turbulence model. The equations for k and ϵ can be solved in this correction function as further discussed in Section 6.3

At the end of the loop at each moment in time, this output module calls the function *runTime.write()* and every type of field data is written into a file. They are located in a folder named after the current moment in time.

3.4 Motion Solver

3.4.1 Introduction

The motion solver responsible for determining the movement of the rigid body relies on other governing equations than the flow solver dealing with the multiphase flow. To be more specific, the motion solver should have the ability to determine the motion involving the acceleration and velocity and update the position over time according to the force acted on the iceberg. For a rigid body, there are 6 Degree of Freedoms (DoFs) in a 3D space as illustrated in Fig. 3.4. These 6 DoFs can be classified into 2 groups: translational and rotational components according to their definitions. The former, including surge, sway and heave, is defined as translation along the *x*-, *y*- and *z*-axes, respectively. The latter, including roll, pitch and yaw, is described as rotation relative to the *x*-, *y*- and *z*-axes, respectively.

Therefore, the motion solver is required to obtain the motions along these 6 components. In this study, the motion solver is modified based on the original *sixDoFRigidBodyMotion*. Given that the 6-DoF motion of a rigid body can be considered as the sum of (a) a translation relative to the centre of mass and (b) a rotation relative to the centre of rotation (CoR), this section firstly presents the motion equations of (a) and (b) in Foam-extend, followed by details about how to update the motion state over time given in Section 3.4.3.

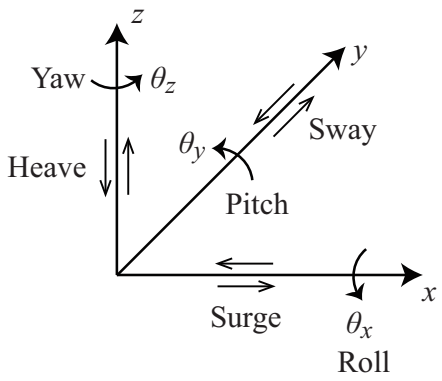


Figure 3.4 Illustration of the 6 Degree of Freedoms of a rigid body

3.4.2 Motion equations

The governing equation for the translation of the rigid body is Newtons Second Law. By applying the same global coordinate as the flow solver, the equation of translation is given as

$$\mathbf{a} = \frac{\mathbf{F}}{m_s}, \tag{3.25}$$

where $\mathbf{a} = (a_x,\, a_y,\, a_z)$ is the acceleration vector, $\mathbf{F} = (F_x,\, F_y,\, F_z)$ the total force vector of the flows acting on the rigid body and m_s is the mass.

The equation for the rotation is

$$\mathbf{M} = \frac{\mathrm{d}(\mathbf{L})}{\mathrm{d}t} = \frac{\mathrm{d}(\mathbf{I}\boldsymbol{\omega})}{\mathrm{d}t}. \tag{3.26}$$

In Eq. (3.26), $\mathbf{M} = (M_x,\, M_y,\, M_z)$ denotes the total torque in relation to the CoR, $\mathbf{L} = (L_x,\, L_y,\, L_z)$ the angular momentum vector, $\mathbf{I} = \mathrm{diag}(I_x,\, I_y,\, I_z)$ the moment of inertia diagonal tensor and $\boldsymbol{\omega} = (\omega_x,\, \omega_y,\, \omega_z)$ the angular velocity vector. The subscripts x, y and z denote the components along each axis. As the components of $\mathbf{I}$ may change relative to the global coordinates during rotation, a local coordinate system (Fig. 3.5) is introduced such that the components of $\mathbf{I}$ in the local coordinate system can be kept consistent as initially specified. Therefore, solving Eq. (3.26) involves a coordinate transformation, which will be explained in Section 3.4.3.

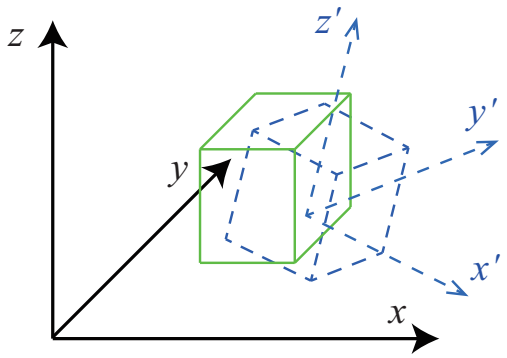


Figure 3.5 Sketch of the global and local coordinates used in solving Eq. (3.26). The local coordinates move with the block with the origin at the centre of rotation

3.4.3 Motion state update

After calculating $\mathbf{a}$ and $\mathbf{M}$, the velocity $\mathbf{V}$ and angular momentum $\mathbf{L}$ are obtained by integrating the vectors $\mathbf{a}$ and $\mathbf{M}$ with respect to time. Similarly, the displacement $\mathbf{X}$ and rotation angle θ can be obtained by integrating the velocity vectors ($\mathbf{V}$ and $\mathbf{L}$) over time. Here, a leapfrog scheme (Dullweber et al., 1997) is applied to advance the motion state ($\mathbf{a}$, $\mathbf{V}$, $\mathbf{L}$, $\mathbf{X}$ and θ) over time. This method involves two main successive processes named position update and force update.

3.4.3.1 Position update

This process is achieved by the function *updatePosition()*. In order to obtain the position from $\mathbf{a}$ and $\mathbf{M}$, two important intermediate variables are required: $\mathbf{V}$ and $\mathbf{L}$. According to the leapfrog scheme, $\mathbf{V}$ and $\mathbf{L}$ are advanced by a half time step and then $\mathbf{X}$ and θ are updated by a full time step using the updated $\mathbf{V}$ and $\mathbf{L}$. Details are given as

$$\mathbf{V}^{\mathrm{n}+\frac{1}{2}} = \mathbf{V}^{\mathrm{n}} + \frac{1}{2}\mathbf{a}^{\mathrm{n}}(t^{\mathrm{n}+1} - t^{\mathrm{n}}). \tag{3.27}$$

In Eq. (3.27), the superscripts n, n+$\frac{1}{2}$ and n+1 indicate the current, half and next moments in time, respectively. The displacement of the centre of mass $\mathbf{X}$ can be obtained by

$$\mathbf{X}^{\mathrm{n}+1} = \mathbf{X}^{\mathrm{n}} + \mathbf{V}^{\mathrm{n}+\frac{1}{2}}(t^{\mathrm{n}+1} - t^{\mathrm{n}}). \tag{3.28}$$

In order to update θ, the local $\mathbf{M}'$ and $\mathbf{L}'$ obtained from the last moment in time are used

$$\mathbf{L}'^{\mathrm{n}+\frac{1}{2}} = \mathbf{L}'^{\mathrm{n}} + \frac{1}{2}\mathbf{M}'^{\mathrm{n}}(t^{\mathrm{n}+1} - t^{\mathrm{n}}). \tag{3.29}$$

The current orientation matrix $\mathbf{Q}$ of the rigid body is required for the coordinate transformation. The following three standard rotation matrices are used, which are rotating vectors by $\Delta\theta$ ($\Delta\theta_x$, $\Delta\theta_y$ and $\Delta\theta_z$) about the *x*-, *y*- and *z*-axes, respectively.

$$\mathbf{R}_x(\Delta\theta_x) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\Delta\theta_x & -\sin\Delta\theta_x \\ 0 & \sin\Delta\theta_x & \cos\Delta\theta_x \end{bmatrix}$$

$$\mathbf{R}_y(\Delta\theta_y) = \begin{bmatrix} \cos\Delta\theta_y & 0 & \sin\Delta\theta_y \\ 0 & 1 & 0 \\ -\sin\Delta\theta_y & 0 & \cos\Delta\theta_y \end{bmatrix} \tag{3.30}$$

$$\mathbf{R}_z(\Delta\theta_z) = \begin{bmatrix} \cos\Delta\theta_z & -\sin\Delta\theta_z & 0 \\ \sin\Delta\theta_z & \cos\Delta\theta_z & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

The rotation matrix $\mathbf{R}$ can be obtained by multiplying the matrices in Eq. (3.30). Herein, a symplectic way is applied to construct $\mathbf{R}$ to maintain the time-reversible property (Dullweber et al., 1997):

$$\mathbf{R} = \mathbf{R}_x(\frac{1}{2}\Delta\theta_x)\mathbf{R}_y(\frac{1}{2}\Delta\theta_y)\mathbf{R}_z(\Delta\theta_z)\mathbf{R}_y(\frac{1}{2}\Delta\theta_y)\mathbf{R}_x(\frac{1}{2}\Delta\theta_x), \tag{3.31}$$

where

$$\Delta\theta_x = \frac{{L'}_x^{\,\mathrm{n}+\frac{1}{2}}(t^{\mathrm{n}+1} - t^{\mathrm{n}})}{I_x}$$

$$\Delta\theta_y = \frac{{L'}_y^{\,\mathrm{n}+\frac{1}{2}}(t^{\mathrm{n}+1} - t^{\mathrm{n}})}{I_y} \tag{3.32}$$

$$\Delta\theta_z = \frac{{L'_z}^{\mathrm{n}+\frac{1}{2}}(t^{\mathrm{n}+1} - t^{\mathrm{n}})}{I_z}$$

In Eq. (3.32), $\Delta\theta_x$, $\Delta\theta_y$ and $\Delta\theta_z$ are the updated rotation angle components. The orientation at the next moment in time $\mathbf{Q}^{\mathrm{n}+1}$ is updated with

$$\mathbf{Q}^{\mathrm{n}+1} = \mathbf{Q}^{\mathrm{n}}\mathbf{R}. \tag{3.33}$$

Finally, $\boldsymbol{\theta}^{\mathrm{n}+1}$ is obtained with

$$\boldsymbol{\theta}^{\mathrm{n}+1} = \boldsymbol{\theta}^{\mathrm{n}} + \Delta\boldsymbol{\theta}. \tag{3.34}$$

3.4.3.2 Force update

The following process is implemented in the function *updateForce()*, mainly used to prepare the calling of *updatePosition()* in the next time step. Firstly, the new acceleration vector $\mathbf{a}^{\mathrm{n}+1}$ is calculated as

$$\mathbf{a}^{\mathrm{n}+1} = \frac{\mathbf{F}^{\mathrm{n}+1}}{m_s}. \tag{3.35}$$

Further, $\mathbf{M}^{\mathrm{n}+1}$ needs to be converted into local coordinates with

$$\mathbf{M}'^{\mathrm{n}+1} = \mathbf{M}^{\mathrm{n}+1}\mathbf{Q}^{\mathrm{n}+1}. \tag{3.36}$$

Note that $\mathbf{V}$ and $\mathbf{L}$ are only advanced by half a time step in the function *updatePosition()*, such that the remaining half time step is considered here with

$$\mathbf{V}^{\mathrm{n}+1} = \mathbf{V}^{\mathrm{n}+\frac{1}{2}} + \frac{1}{2}\mathbf{a}^{\mathrm{n}+1}(t^{\mathrm{n}+1} - t^{\mathrm{n}}), \tag{3.37}$$

$$\mathbf{L}'^{\mathrm{n}+1} = \mathbf{L}'^{\mathrm{n}+\frac{1}{2}} + \frac{1}{2}\mathbf{M}'^{\mathrm{n}+1}(t^{\mathrm{n}+1} - t^{\mathrm{n}}). \tag{3.38}$$

The local $\mathbf{L}'$ can be converted into the global $\mathbf{L}$ for the angular velocity calculation

$$\boldsymbol{\omega} = \mathbf{L}\mathbf{I}^{-1} = \mathbf{Q}\mathbf{L}'\mathbf{I}^{-1}. \tag{3.39}$$

All the variables denoting the motion state are then updated over a full time step.

Given that 5 idealised iceberg calving mechanisms are considered in this study, some restrictions are required in the motion solver to ensure the same numbers of DoF as the laboratory tests when numerically modelling the IBTs. Details of these restrictions will be given in the Results chapter.

3.5 Coupling Method

3.5.1 Introduction

The coupling of the flow and motion solvers is achieved in the herein implemented dynamic mesh handling class *sixDoFImmersedBoundary*. With the help of this class, the main functions *updatePosition()* and *updateForce()* in the motion solver can be called and field data handled by the flow solver can be accessed. As a two-way coupling, the new position of the IB may also affect the flow field. Except for the connection functionality, this class has also modules responsible for the force calculation and moving IB based on the results from the motion solver.

3.5.2 Force calculation

The force calculation is achieved by the module *sixDoFImmersedBoundary*, while the motion solver is only used to determine the motion based on the given force. Given that this is a coupling problem between these two independent systems (the multiphase flow and the rigid body), the force analysis of the rigid body should be conducted to determine the force type used in the calculation. As the rigid body is directly in contact with the multiphase flow, the pressure force $\mathbf{F}_p$, viscosity force $\mathbf{F}_v$ and gravity of the rigid body $\mathbf{G}$ should be included (Hadžić et al., 2005). This is the typical method to calculate forces in CFD when considering FSI and has already been implemented in Foam-extend.

However, given that the current IBM toolbox applies the interpolation method to evaluate the immersed boundary cells' values (i.e. the pressure and velocity), which may underestimate the corresponding pressure and viscosity forces when modelling FSI as shown in the Results chapter, the added mass and drag force are introduced to calibrate the model. Here, the added mass m_a is used to tune the virtual mass force $-m_a\mathbf{a}$ on the rigid body. This added mass rep-

resents the inertia of the surrounding multiphase flow. The drag force $\mathbf{F}_d$ can also be adapted. It considers friction when the submerged moving body has a motion relative to the flows. This calculation of $\mathbf{F}$ involving m_a and $\mathbf{F}_d$ has been demonstrated e.g. by Enet and Grilli (2007) and Grilli and Watts (2005), who numerically simulated tsunamis generated by submerged landslides. In consideration of the close relation between landslide-tsunamis and IBTs, the calculation of m_a and $\mathbf{F}_d$ is also applied here for calibrating the IBT modelling, although the actual added mass and drag force have already been captured by the CFD method.

$\mathbf{F}$ in Eq. (3.25) is thus given as

$$\mathbf{F} = \mathbf{F}_p + \mathbf{F}_v + \mathbf{F}_d - m_a\mathbf{a} + \mathbf{G}, \tag{3.40}$$

where $\mathbf{F}_p$ and $\mathbf{F}_v$ are directly based on field data (velocity gradient and pressure) as illustrated in Section 3.2.4. m_a is given by Enet and Grilli (2007) as

$$m_a = C_m m_s, \tag{3.41}$$

where C_m denotes the added mass coefficient. This formula is also equivalent to the under-relaxation technique used in Hadžić et al. (2005). Regarding the calculation of $\mathbf{F}_d$, although it may be derived from the characteristics of the flow (Mei, 1989), the method in Enet and Grilli (2007) has been applied due to its simplicity for this two-system coupling strategy such that only the velocity of the rigid body is required. $\mathbf{F}_d$ is obtained according to Enet and Grilli (2007)

$$\mathbf{F}_d = -\frac{1}{2} C_d \rho_s A_b \mathbf{V}|\mathbf{V}|. \tag{3.42}$$

In Eq. (3.42) C_d denotes the drag force coefficient, A_b the iceberg's main cross section perpendicular to the direction of the velocity and $|\mathbf{V}|$ is the magnitude of $\mathbf{V}$. Therefore, $\mathbf{F}$ is

$$\begin{aligned} \mathbf{F} &= \mathbf{F}_p + \mathbf{F}_v + \mathbf{F}_d - m_a\mathbf{a} + \mathbf{G} \\ &= \Sigma(p_{IB} \cdot \mathbf{S}_{IB}) + \Sigma(\tau_{IB} \cdot \mathbf{S}_{IB}) - \tfrac{1}{2} C_d \rho_s A_b \mathbf{V}|\mathbf{V}| - C_m m_s \mathbf{a} + m_s \mathbf{g}, \end{aligned} \tag{3.43}$$

where $\mathbf{S}_{IB}$ is the area vector of the IB (subscript *IB*) cell and τ_{IB} denotes the shear stress along the IB cell that is obtained by multiplying the IB cell's μ by its normal velocity gradient.

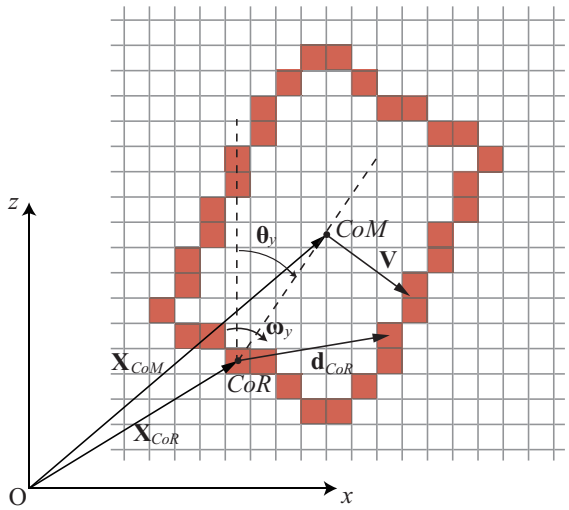


Figure 3.6 Sketch of vectors used in Eq. (3.44). The brown cells denote the IB cells (adapted from Chen et al., 2020)

All the components in $\mathbf{F}$ are also used in the calculation of $\mathbf{M}$.

$$\begin{aligned}\mathbf{M} = {} & \Sigma[\mathbf{d}_{CoR} \times (p_{IB} \cdot \mathbf{S}_{IB})] + \Sigma[\mathbf{d}_{CoR} \times (\tau_{IB} \cdot \mathbf{S}_{IB})] \\ & - (\mathbf{X}_{CoM} - \mathbf{X}_{CoR}) \times \tfrac{1}{2} C_d \rho_s A_b \mathbf{v}_{CoM}{}^2 - (\mathbf{X}_{CoM} - \mathbf{X}_{CoR}) \\ & \times C_m m_s \mathbf{a} + (\mathbf{X}_{CoM} - \mathbf{X}_{CoR}) \times m_s \mathbf{g},\end{aligned} \tag{3.44}$$

The involved vectors in Eq. (3.44) are shown in Fig. 3.6, where $\mathbf{d}_{CoR}$ indicates the vector pointing from the IB cell to the CoR and $\mathbf{X}_{CoR}$ denotes the position vector of the iceberg's CoR.

3.5.3 IB movement

sixDoFImmersedBoundary is also responsible for moving the IB according to the position information returned from the motion solver, including the displacement of its centre of mass $\mathbf{X}$ and rotation angle vector θ (Section 3.4). In Foam-extend, the handling of the moving IB is achieved by using quaternion and septernion ordered pairs. A quaternion consists of 4 numbers (a scalar and a 3D vector), which is mainly responsible for performing rotations in a 3D space. As an extension of a quaternion, a septernion is made up of a 3D vector and a quaternion (including 7 components in total) such that it has the ability to handle both translation and rotation. Details about quaternions and septernions can be found in the source code directory */src/foam/primitives*.

The general form of a quaternion is $\mathbf{q}(s,\, \mathbf{v})$, where s and $\mathbf{v}$ are a scalar and a vector, respectively. In order to support the rotation, the standard form of a rotation quaternion should be constructed as

$$\mathbf{q} = (\cos\frac{1}{2}\theta, \sin\frac{1}{2}\theta\mathbf{v}). \tag{3.45}$$

Eq. (3.45) denotes a quaternion used to rotate a vector by θ relative to the axis $\mathbf{v}$, where $\mathbf{v}$ is supposed to be a unit vector. In analogy with rotation matrices, multiplying rotation quaternions can perform a more complex rotation. In this study, a rotation quaternion $\mathbf{q}_r$ is constructed as the product of three components $\mathbf{q}_x$, $\mathbf{q}_y$ and $\mathbf{q}_z$, each of which rotates a vector by an angle about the x-, y- or z-axes.

$$\begin{aligned}\mathbf{q}_r &= \mathbf{q_x q_y q_z} \\ &= [\cos\tfrac{1}{2}\theta_x, \sin\tfrac{1}{2}\theta_x(1,0,0)] \cdot [\cos\tfrac{1}{2}\theta_y, \sin\tfrac{1}{2}\theta_y(0,1,0)]\cdot \\ &\quad [\cos\tfrac{1}{2}\theta_z, \sin\tfrac{1}{2}\theta_z(0,0,1)].\end{aligned} \tag{3.46}$$

The general form of septernions is given as $\mathbf{s}(\mathbf{v},\, \mathbf{q})$. A septernion $\mathbf{s}_c$ is constructed to perform the rotation first and then the translation in Foam-extend. For a rigid body rotated by $\mathbf{q}_r$ over its CoR the septernion responsible for pure rotation $\mathbf{s}_r$ is given as

$$\mathbf{s}_r = [\mathbf{x}_{CoR}, (1,0,0,0)] \cdot \mathbf{q}_r \cdot [-\mathbf{x}_{CoR}, (1,0,0,0)], \tag{3.47}$$

where $\mathbf{x}_{CoR}$ denotes the position vector of the CoR of the rigid body. Then the displacement of the centre of mass $\mathbf{X}$ is only added to the vector part of $\mathbf{s}_c$.

$$\mathbf{s}_c = \mathbf{s}_r + [\mathbf{X}, (0,0,0,0)]. \tag{3.48}$$

Thus, the new position of the rigid body under the combined motion is achieved by multiplying the coordinates of each point of the IB $\mathbf{x}_{IB}$ (vector) and $\mathbf{s}_r$, i.e.

$$\mathbf{x}_{IB}^{\mathrm{n+1}} = \mathbf{s}_c^{\mathrm{n}} \mathbf{x}_{IB}^{\mathrm{n}}. \tag{3.49}$$

Finally, in order to consider the effect of the new position of the IB on the

flow field, a no-slip boundary condition is applied on the IB, which is used to update the velocity of the IB

$$\mathbf{u}_{IB}^{\mathrm{n+1}} = \frac{\mathbf{x}_{IB}^{\mathrm{n+1}} - \mathbf{x}_{IB}^{\mathrm{n}}}{t^{\mathrm{n+1}} - t^{\mathrm{n}}}. \tag{3.50}$$

3.6 Numerical Model Workflow

The newly implemented solvers (the flow solver *interDyMIbFoam* and the motion solver *sDoFMotion*), the dynamic mesh handling class *sixDoFImmersedBoundary* together with the relevant modified IBM toolbox modules are compiled as an external executable together with some libraries. The new implementation does not affect the original source code of Foam-extend, which has a good transplantability and can be easily used on High Performance Computing (HPC) clusters. For the application of the new executable and libraries, the route in the file *system/controlDict* needs to be specified, which is typically the user directory of Foam-extend.

The following steps illustrate the application of this numerical model for one case. First of all, the computational domain should be generated. This is achieved by the mesh generation utility *blockMesh* in Foam-extend. And the created mesh files are stored in the directory *constant/polyMesh*. The geometry file of the rigid body is also required, which is a .stl file stored in the directory *constant/triSurface*. In order to be readable for this IBM toolbox, this .stl file should be converted into a .ftr file in the same folder by using the command

surfaceConvert constant/triSurface/IB.stl constant/triSurface/IB.ftr

Then the boundary conditions for each patch in the domain need to be specified for each solved variable. Table 3.3 shows the boundary condition setup for a laminar flow case.

Table 3.3 Boundary conditions for each patch in the domain for a laminar flow case

Variable	**Wall**	**IB**	**Atmosphere**
u	*fixedValue*	*immersedBoundary*	*pressureInletOutletVelocity*
p	*zeroGradient*	*immersedBoundary*	*totalPressure*
α	*zeroGradient*	*immersedBoundary*	*inletOutlet*

For parameter initialisations, the files containing the control parameters for the simulation are listed in Table 3.4.

Table 3.4 Files used for parameters initialisations

File name	**Description for the contained parameters**
controlDict	Control parameters such as simulation time, output interval
setFieldsDict	Initialisation of the phase fraction field
decomposeParDict	Decomposition of the domain for parallel computing on HPC
fvScheme	Selection of the discretisation schemes
fvSolution	Specification of the tolerance for each solved variable
dynamicMeshDict	Specification of the parameters of the rigid body
transportProperties	Specification of the parameters of the fluids
turbulenceProperties	Selection of the turbulence model
sampleDict	Sampling field data for post-processing

After specifying the boundary conditions and control parameters, the utility *setFields* is used to initialise the phase fraction field, i.e. to set the water level. Then the main simulation program *interDyMIbFoam* can be executed. If the simulation is run on a HPC, then the number of CPU cores needs to be specified:

mpirun -np number_of_CPU_cores interDyMIbFoam -parallel

Once the simulation has finished, the output field data may need to be extracted for post-processing. This is done by the utility *sample*, where the field data can be collected along a given line or a surface. Note that the whole domain has been decomposed into a series of subdomains if the simulation is run in parallel mode, the data need to be reconstructed before post-processing. This is done by the command

reconstructPar -fields "(names_of_fields)" -time time_range

Finally, the visualisation of the results is achieved in *paraView*.

3.7 Summary

This chapter presented the numerical model for iceberg-tsunamis based on Foam-extend and illustrated the new features and functionalities implemented in this study. The implementation details of the Immersed Boundary Method (IBM) toolbox in Foam-extend has been introduced first. Within this IBM toolbox framework, a new multiphase flow solver with the moving immersed boundary (IB) support *interDyMIbFoam* has been implemented. The solving process of the governing equations has been explained in Section 3.3. Resolved motion of IB is considered in this study and applied for the main simulation, which requires a motion solver. Some modifications have been made in the motion solver in order to simulate the full range of iceberg calving mechanisms in this study. Details of the newly implemented dynamic mesh handling class have been presented in Section 3.5, including the new functionalities, e.g. the force calculation of IB surrounded by the multiphase flow, introducing the drag force and virtual mass force caused by the added mass to calibrate the model and the method to move the IB. Details about how to apply the new solvers and libraries on HPC has also been presented, including the workflow illustrating the use of this numerical model in Foam-extend.

Chapter 4

LABORATORY MODEL

4.1 Introduction

This chapter presents the large-scale laboratory tests to investigate the generation and propagation of iceberg-tsunamis (IBTs). The tests involve the five idealised iceberg calving mechanisms (A) capsizing, (B) gravity-dominated fall, (C) buoyancy-dominated fall, (D) gravity-dominated overturning and (E) buoyancy-dominated overturning. Five typical tests are selected for the numerical validation, one for each calving mechanism. The details of the experimental setup of the selected tests are given in Section 4.2, including the description of the wave basin, coordinate systems, iceberg models, support frame and various triggering methods corresponding to different calving mechanisms. In Section 4.3, the devices used in the laboratory tests and the measurement methods together with the data processing are addressed. The summary of this section about the laboratory model is finally presented in Section 4.5.

4.2 Experimental Setup

The large-scale laboratory tests were conducted in a 50 m $\times$ 50 m large wave basin (Fig. 4.1) at Deltares, Delft, in the Netherlands. Excluding the wavemakers and absorbing beaches, the effective size was 40.3 $\times$ 33.9 m. The basin bottom was horizontal and the water depth of the basin during the selected experiments was 1.0 m (Delta Basin, 2020).

The sketch of the five idealised iceberg calving mechanisms is shown in Fig. 1.2. For mechanism A, the floating block turns over and its centre of mass moves simultaneously upwards. Mechanism A combines two motion types: translation and rotation, i.e. it involves 2 Degree of Freedoms (DoFs). The fall and overturning mechanisms can both be dominated by either gravity or buoyancy. Essentially, blocks dominated by gravity fall into the water body while blocks

dominated by buoyancy rise to the water surface (Heller et al., 2019c). The block performs vertical translations in mechanisms B and C, and rotations in mechanisms D and E.



Figure 4.1 The 50 m × 50 m wave basin used for the experiments (Chen and Heller, 2020)

4.2.1 Coordinate systems

The experimental setup relies on a global cylindrical coordinate system (r, z, γ). The origin is located at the block centre on the water surface in mechanism A (Fig. 4.7a). For the remaining mechanisms, the origin is located at the support frame centre on the still water level (Fig. 4.7b and c). r denotes the radial distance from the origin in the horizontal plane and the z-axis points vertically upwards (Fig. 4.7c). γ is the wave propagation angle, with the positive direction pointing clockwise. A global Cartesian coordinate system (x, y, z) is also applied for data processing, with the same z-axis as in the cylindrical coordinate system. The x-axis points towards $\gamma = 0°$ and the y-axis towards $\gamma = -90°$.

4.2.2 Iceberg model and support frame

Two square blocks made of polypropylene homopolymer were used to model the calving icebergs (Fig. 4.2). The blocks' density were $\rho_s \approx 920$ kg/m^3, which

is similar to the density of ice as in other IBT studies (Tables 2.1 and 2.2). The block lengths l, widths b and thicknesses s are parallel with the z-, y- and x-axes of the global coordinate system. The dimensions of the blocks were (1) 0.800 m × 0.500 m × 0.500 m and (2) 0.800 m × 0.500 m × 0.250 m, respectively. Compared with real iceberg dimensions in field observations (Table 2.2), the geometric scale of these blocks in the laboratory tests may vary from approximately 1:70 to 1:500. Note that the blocks were placed with different orientations in the five mechanisms to vary l, b and s as shown in Table 4.1.

Table 4.1 Block dimensions in the five selected tests

Calving mechanism	**Block type**	**Block length l (m)**	**Block width b (m)**	**Block thickness s (m)**
A	2	0.800	0.500	0.250
B	1	0.500	0.800	0.500
C	1	0.500	0.800	0.500
D	1	0.800	0.500	0.500
E	2	0.800	0.500	0.250

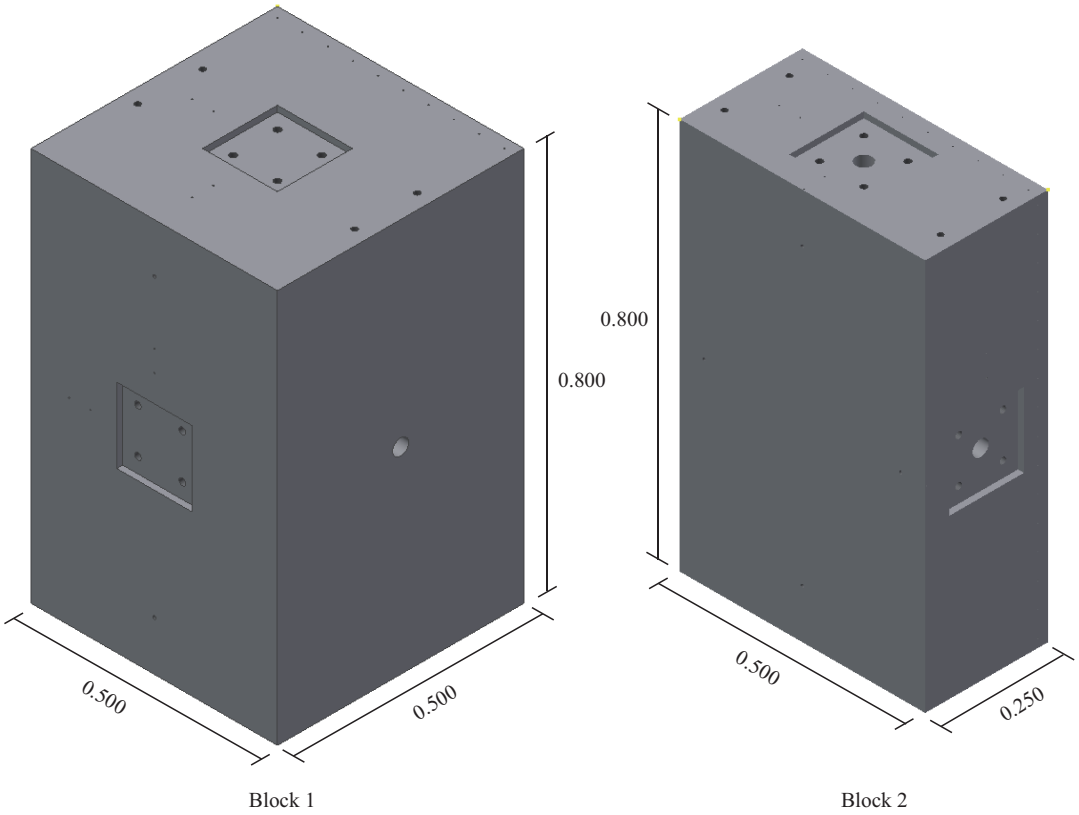


Figure 4.2 Illustration of the blocks in the laboratory tests (adapted from Heller, 2019)

In order to locate and support the blocks in all calving mechanisms, the steel support frame shown in Fig. 4.3 was designed. This support frame mainly consisted of a large steel plate and profiles. Slits on the steel plate allowed the locations of the rigidly installed profiles to be adjusted. In addition to help support the blocks, these profiles were also used to restrict the block motion. The support frame was placed one third of the basin width from the wall (Fig. 4.7b). The steel structure shown in Fig. 4.3(b) was used to move the blocks in place. In the capsizing case, the steel plate was placed offshore on the basin bottom (Fig. 4.3c). Furthermore, a duckboard was used to access the capsizing experiment offshore and to carry some measurement equipment (Fig. 4.7a). This duckboard did not significantly affect the wave field as it was supported by a slim structure in a section of the basin where no measurements were taken.

4.2.3 Triggering methods

This subsection illustrates the triggering methods to initiate the different calving mechanisms. For mechanism A, block 2 was stabilised with a wooden rod, which went through the block centre (Fig. 4.3c). Through slits in the profiles, the rod could move vertically and simultaneously rotate freely. The block motion was restricted to 2 DoFs, namely heave and pitch. The movement of the block in mechanism A was initiated by removing a fitting used to stabilise the block and by simply releasing the block by hand.

For mechanism B, an electromagnet was used to hold the block in position before triggering its motion (Fig. 4.3a). The electromagnet was attached to a small steel plate with a mass of 2.5 kg integrated in the top surface of the block. Further, a rope to lift the electromagnet was connected to a winch system attached to the support profile on the rear side of the basin wall (Fig. 4.3a). By switching off the electromagnet, the block started to heave with 1 DoF.

For mechanism C, a rope connected to the block bottom in the centre was used to pull the block under water and to balance the buoyancy force. The block was also hold in position with the help of a steel beam from above. Once the rope and beam were removed simultaneously, the block moved vertically upwards (1 DoF).

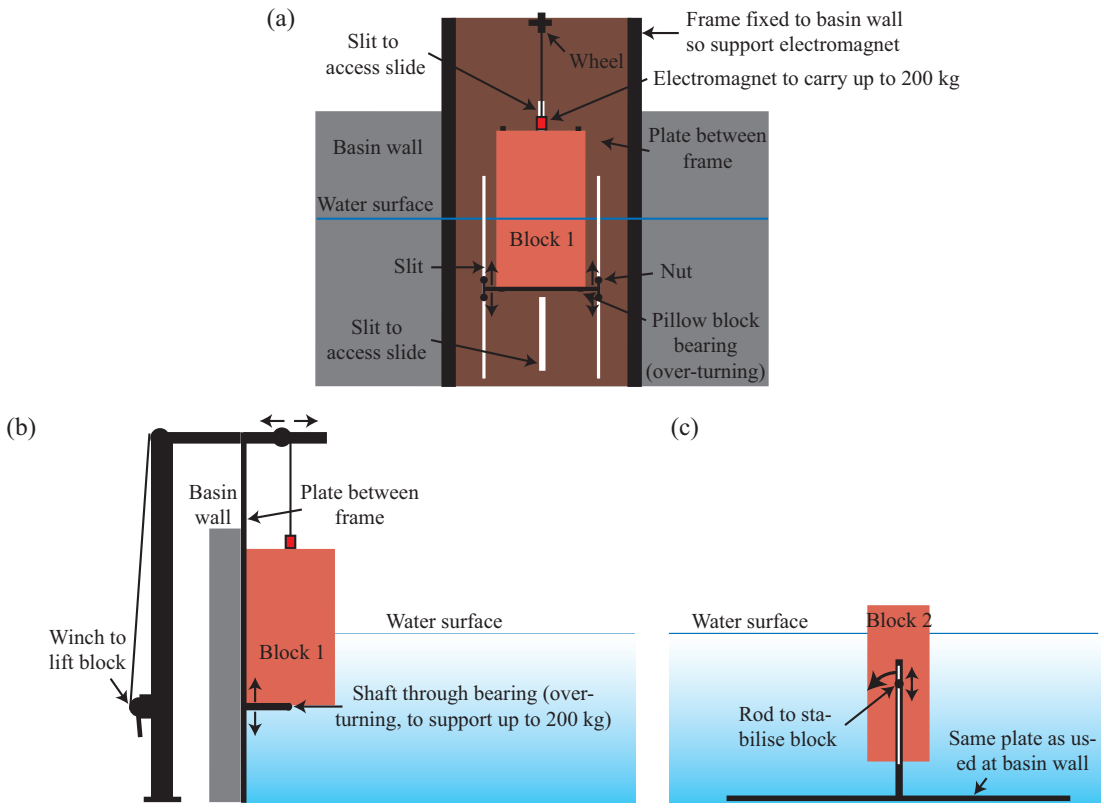


Figure 4.3 Sketch of the support frame: (a) front view and (b) side view in the fall cases (with some of the components also used in the overturning cases) and (c) side view of capsizing case (adapted from Heller, 2019)

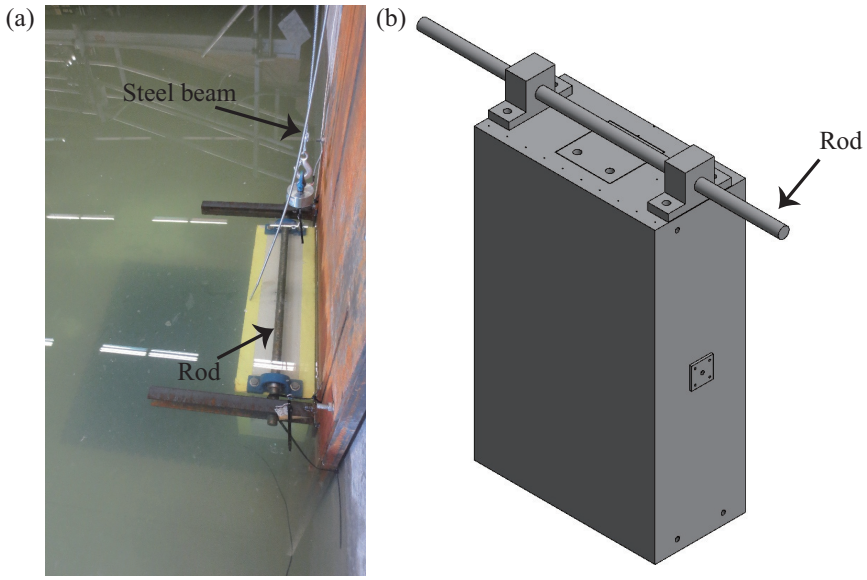


Figure 4.4 (a) Photo and (b) sketch of the experimental setup of mechanism E (adapted from Heller, 2019)

For mechanism D, block 1 was attached to the support frame via a rod with a diameter of 30 mm. The rod went through two roller bearings fixed on each side of the block bottom (Fig. 4.4). This guaranteed a pure rotation of the block, i.e. 1 DoF. Prior to triggering, a steel beam was used to stabilise the block from above. The initiation of the overturning case was achieved by manually detaching the steel beam and by simply releasing the block.

For mechanism E, the triggering method was similar to mechanism D, but the rod was attached to the top surface of the block, as illustrated in Fig. 4.4. Once the steel beam was removed, the block pitched (1 DoF) towards the water surface.

4.2.4 Experimental parameters

The water depth in the five selected laboratory tests was constant $h = 1.0$ m and the water density $\rho_w = 1000$ kg·m^{-3} with a water temperature of 19.3°C ±0.4°C. Further, the distance between the rod centre and the block surface was 0.048 m for both mechanism D and E cases, i.e. the radius of rotation R_l in the overturning cases was 0.848 m. The block parameters and base release locations are important in IBT generation. The block parameters include the block volume $\forall_s$, block density ρ_s, block mass m_s and block dimensions l, b and s. The base release location is defined as the distance between the block bottom and the still water level. These parameters are shown in Table 4.2.

Table 4.2 Experimental parameters in the selected tests

Calving mechanism	A	B	C	D	E
Block volume $\forall_s$ (m^3)	0.100	0.200	0.200	0.200	0.100
Block density ρ_s (kg·m^{-3})	924.3	935.6	923.2	935.6	911.5
Block mass m_s (kg)	92.43	187.11	184.64	187.11	91.15
Base release location (m)	−0.739	0.000	−0.830	−0.300	−0.900

4.3 Data Acquisition and Processing

The measurements taken in the laboratory tests were the motion of the blocks and the water surface elevations at some particular locations. A general observation of the experiments was achieved with two cameras (a 5 MP Point-

Grey ZBR2-PGEHD-50S5C-CS used at 15 Hz and a 2 MP IOIndustries Flare 2M280-CXP used at 100 Hz) recording from different directions (Table 4.4). The post-processing of the raw data mainly involved inferencing the velocity and trajectory via the motion sensor and obtaining the wave features from the wave probe signals.

4.3.1 Motion sensor

The motion sensor was an Adafruit BNO055 used to record the motion of the blocks. It included three sensors: a geomagnetic sensor, a gyroscope and an accelerometer. Given that the geomagnetic sensor cannot be calibrated when an electromagnet is in use, only the data from the gyroscope and accelerometer were used, which was sufficient for obtaining the motion of the blocks in the laboratory tests. Figure 4.5 illustrates the components measured by the motion sensor. The gyroscope recorded rotation angles including the roll, pitch and yaw angles in the global Cartesian coordinate system (Section 4.2.1), with the measured angle ranging from $-180°$ to $+180°$, $-90°$ to $+90°$ and $0°$ to $360°$, respectively. The accelerometer measured three acceleration components along a local coordinate system (x', y', z'), where the origin was located at the geometric centre of the motion sensor with the z'-axis perpendicular to the top surface and the x'- and y'-axes parallel with the two neighbouring edges of the motion sensor (Fig. 4.5).

The motion sensor was fixed in a black waterproven enclosure. Prior to each test, the motion sensor was required to be calibrated such that the offset can be removed by the sensor calibration algorithm. The calibration was achieved by firstly putting the sensor horizontally in a stable position and then rotating it slowly along three axes in the global Cartesian coordinate system. Further, the position of the motion sensor had to be adjusted to ensure that the local and global Cartesian coordinate systems were parallel. After these procedures, the enclosure was screwed on the surface of the blocks as shown in Fig. 4.6. Note that there was a 3 mm distance between the motion sensor and the enclosure bottom. The initial locations of the motion sensor in the global Cartesian coordinate system are given in Table 4.3. The motion sensor worked at the maximum capacity, corresponding to a mean sampling rate of 74 Hz.

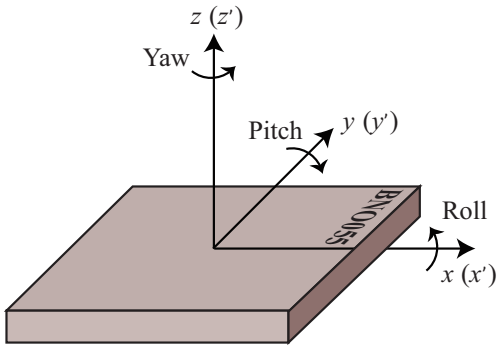


Figure 4.5 Sketch of the measured components of the motion sensor

Table 4.3 Initial locations of the motion sensor in the selected tests

Calving mechanism	x **(m)**	y **(m)**	z **(m)**
A	0.000	0.151	0.064
B	0.250	0.151	0.503
C	0.250	0.151	−0.327
D	0.250	0.151	0.503
E	0.125	0.151	−0.903

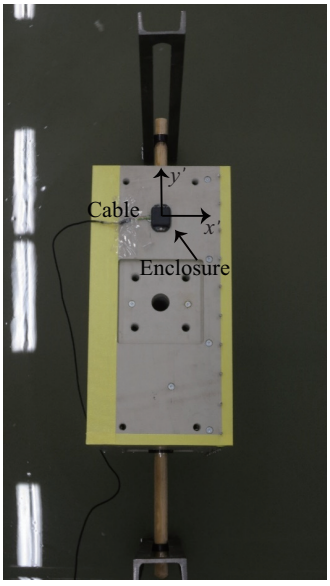


Figure 4.6 Photo of the enclosure fixed to the block surface including the motion sensor in the mechanism A test (adapted from Heller, 2019)

4.3.2 Wave probes

There were up to 35 resistance type wave probes used to record the water surface elevation. Two wave probe layouts were adopted in the selected tests as shown in Fig. 4.7. In mechanism A, 33 wave probes were placed in a semi-circle pattern, with the radial distance r varying from 2 to 15 m (Table 4.4). In the remaining mechanisms, 35 wave probes were located in a quarter-circular pattern given that the wave field was symmetric, with $2 \leq r \leq 35$ m and $-75 \leq \gamma \leq 0°$. The sampling frequency of the wave probes was constant at 100 Hz. The wave probes were calibrated by measuring several different still water levels in the basin. The wave probe calibration took place once a day but less frequently later, given that the calibration parameters only varied by an estimated value of less than 2% (Heller et al., 2020).

Table 4.4 Locations of the wave probes and cameras in the laboratory tests. Coordinates marked with * were also used in the numerical simulation (adapted from Heller, 2019)

Calving mechanism	**Device**	**Locations in function of the radial distance r (m) and wave propagation angle γ (°) (Fig. 4.7)**
A	Wave probes	A1(2, 0)*; A10(3, 0)*; A19(5, 0); A28(10, 0); A32(15, 0); A2(2, –15)*; A11(3, –15)*; A20(5, –15); A29 (10, –15); A3(2, –30)*; A12(3, –30)*; A21(5, –30); A4(2, –60)*; A13(3, –60)*; A22(5, –60); A5(2, –90)*; A14(3, –90)*; A23(5, –90); A6(2, –120)*; A15(3, –120)*; A24(5, –120); A7(2, –150)*; A16(3, –150)*; A25(5, –150); A8(2, –165)*; A17(3, –165)*; A26(5, –165); A30(10, –165); A9(2, –180)*; A18(3, –180)*; A27(5, –180); A31(10, –180); A33(15, –180)
	Cameras	2 MP at 100 Hz: (6, –95); 5 MP at 15 Hz: (6, –45)
B, C, D and E	Wave probes	B1(2, 0)*; B7(3, 0)*; B13(5, 0)*; B19(10, 0); B25(15, 0); B31(22.5, 0); B34(35, 0); B2(2, −15)*; B8(3, −15)*; B14(5, −15)*; B20(10, −15); B26(15, −15); B32(22.5, −15); B35(35, −15); B3(2, −30)*; B9(3, −30)*; B15(5, −30)*; B21(10, −30); B27(15, −30); B33(22.5, −30); B4(2, −45)*; B10(3, −45)*; B16(5, −45)*; B22(10, −45); B28(15, −45); B5 (2, −60)*; B11(3, −60)*; B17(5, −60)*; B23(10, −60); B29(15, −60); B6(2, −75)*; B12(3, −75)*; B18(5, −75)*; B24(10, −75); B30(15, −75)
	Cameras	2 MP at 100 Hz: (6, −85); 5 MP at 15 Hz: (6, 45)

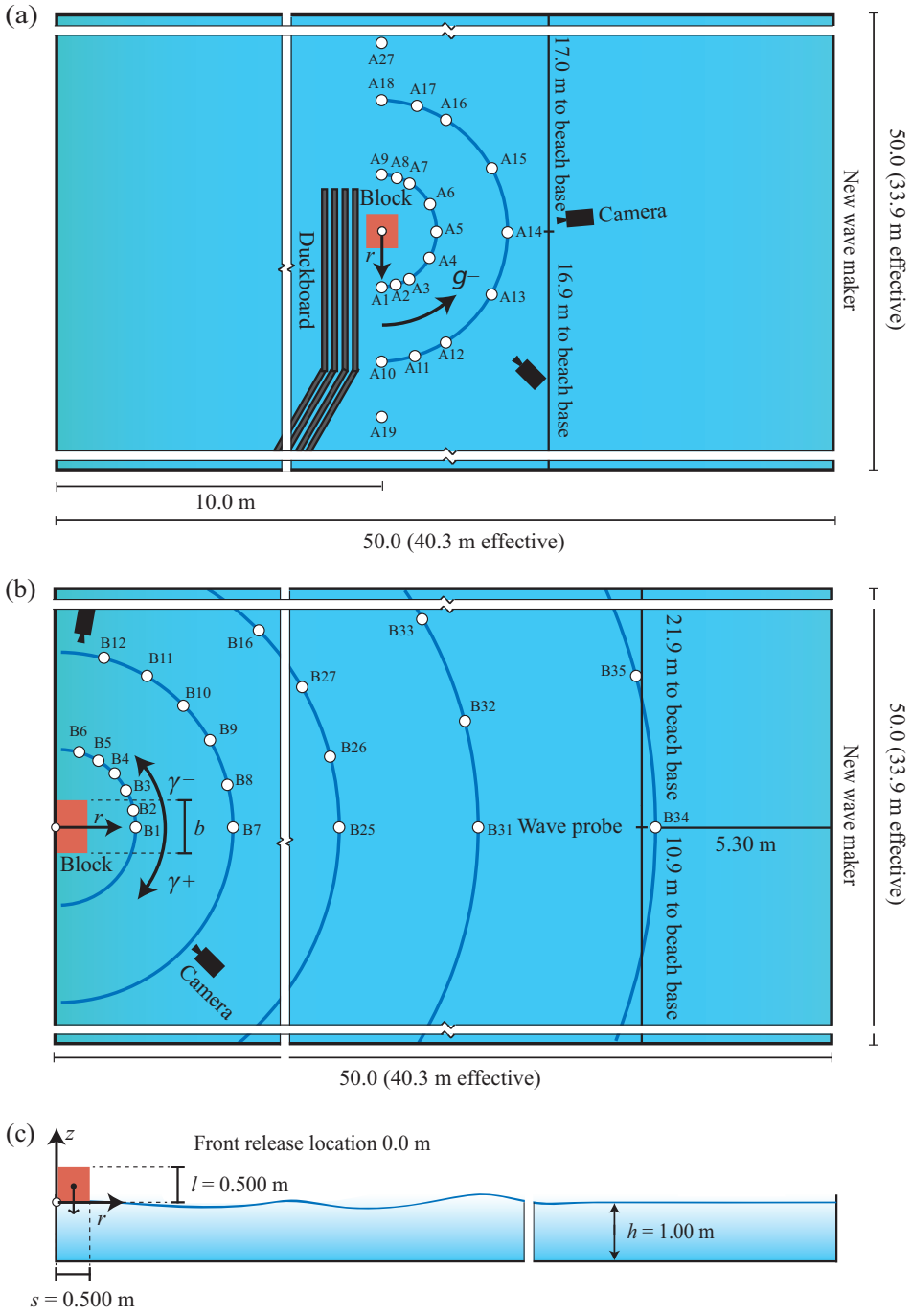


Figure 4.7 Plan views illustrating the wave probe layouts in (a) the mechanism A and (b) the mechanism B tests. (c) Side view of the mechanism B test. Details about the wave probe locations are given in Table 4.4 (adapted from Heller et al., 2019c)

4.3.3 Instrument synchronisation

The instruments were controlled by two independent systems. Figure 4.8 illustrates the synchronisation process of the instruments. The system (a) included the wave probes and both cameras. The system (b) included the motion sensor (and the electromagnet in the mechanism B case), which was recorded via a NI9205 input module from National Instruments. The system (b) was controlled by a purpose written LabView executable. Once the system (a) started, the LabView program recorded a synchronisation pulse generated by system (a) at t_b^1 (relative the system (b)). Thus, the time series from the two different systems could be manually synchronised in a post-processing step.

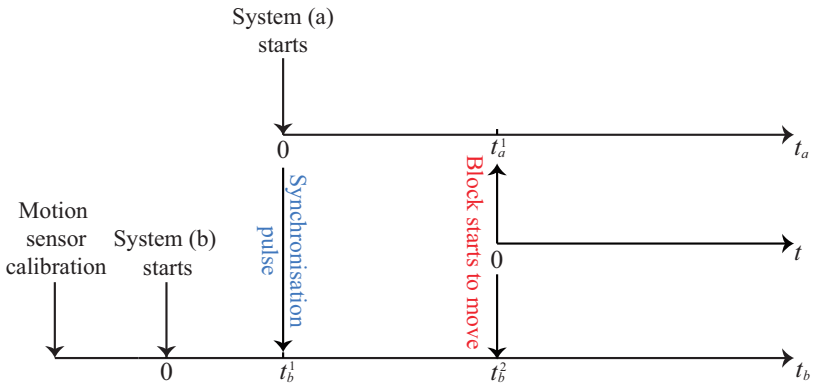


Figure 4.8 Sketch of the synchronisation procedure of the instruments

In the selected laboratory tests, the time origin $t = 0$ s in Fig. 4.8 is defined as the moment in time when the blocks started to move, i.e. when the blocks started interacting with the surrounding water. The time $t = 0$ s was determined via both the cameras (t_a^1) and the data from the motion sensor (t_b^2) in mechanisms A, B and C. The deviation of $t = 0$ s between these two methods was less than 0.1 s ($|t_a^1 - (t_b^2 - t_b^1)| < 0.1$ s). For the remaining mechanisms, where the blocks initially moved underwater, $t = 0$ s was evaluated from the motion sensor only.

4.3.4 Post-processing

The post-processing of the laboratory data mainly included inferencing the velocity and trajectory of the blocks based on the motion sensor data, removing the noise from the wave probe data and the offset in some cases where the initial water surface elevations deviated from 0.

The key point to compute the velocity and trajectory of the motion sensor is to convert the measured acceleration components from local- to global-coordinates with the help of the rotation angles recorded in the global coordinate system. Details about this coordinate transformation approach are shown in Appendix B. Once the global acceleration components are obtained, the velocity can be determined by integrating the acceleration data with respect to time. In the same way, the trajectory can be calculated by integrating the obtained velocity data over time.

Low-pass filters were applied to remove noise in the raw data of the wave probes. This was achieved by firstly applying a Fast Fourier Transform to obtain the data in the frequency domain, and then removing the frequency components which were higher than the given cut off value. In the selected tests, the range of the cut off frequency was between 9.0 and 11.0 Hz. Note that only the first 8.0 s in the time series of wave probe data were used in this study, wave reflection was not involved. In some tests, the initial water surface elevations slightly deviated from 0 at $t = 0$ s, which may be due to a slight change in the water resistance after calibrated. The offset was subtracted or added to the water surface elevation data as shown in Fig. 4.9.

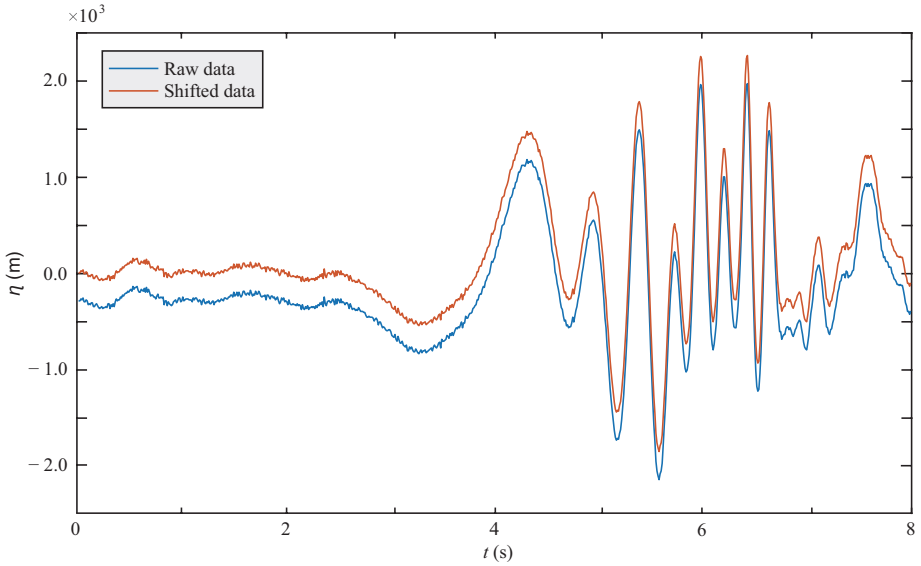


Figure 4.9 Example of the water surface elevation data shifted by the offset in the mechanism E case

Table 4.5 shows the measurement uncertainty in the laboratory tests, including both the directly measured and processed parameters. The measurement

uncertainty for water density is set as $\Delta\rho_w = 0$ kg/m^3.

Table 4.5 Quantification of the measurement uncertainty (adapted from Heller et al., 2019c)

Experimental parameters	**Measurement uncertainty**
Block length l	$\Delta l = \pm 0.001$ m
Block width b	$\Delta b = \pm 0.001$ m
Block thickness s	$\Delta s = \pm 0.001$ m
Block mass m_s	$\Delta m_s = \pm 0.050$ kg
Block velocity V_s	$(\Delta V_s)_\mathrm{A} = \pm 0.050$ m/s, $(\Delta V_s)_\mathrm{B} = \pm 0.08$ m/s, $(\Delta V_s)_\mathrm{C} = \pm 0.03$ m/s, $(\Delta V_s)_\mathrm{D} = \pm 0.05$ m/s, $(\Delta V_s)_\mathrm{E} = \pm 0.03$ m/s,
Water depth h	$\Delta h = \pm 0.003$ m
Maximum wave height H_M and wave amplitude a_M	$\Delta H_M = \Delta a_M = \pm 0.002$ m (for tsunamis affected by air and splash), $\Delta H_M = \Delta a_M = \pm 0.0002$ m (for pure water tsunamis)
Maximum wave period T_M	$\Delta T_M = \pm 0.03$ s

4.4 Model Similarity

A physical model of IBTs at reduced size may result in differences compared to a real IBT due to scale effects. Perfect similarity (mechanical similarity) requires similarity in geometry, kinematics and dynamics. Geometric similarity means the dimensions (e.g. length, width and thickness) of the model should be proportional to their corresponding dimensions of the full scale. Kinematic similarity requires the ratios of velocity remain constant at corresponding points in the model and the full scale, while dynamic similarity focuses on the same force ratios. Note that there is no fluid satisfying all these similarity criteria simultaneously unless the model has the same size as its prototype. There are three important force ratios for dynamic similarity in fluid dynamics which are generally considered: Froude, Reynolds and Weber scaling similarity. For a physical model, the most relevant similarity should be applied.

Under Froude similarity, the force ratio of inertia and gravity remains the same between the model and prototype expressed by the *Froude number* Fr (Heller, 2011)

$$\mathrm{Fr} = \frac{V_s}{\sqrt{gh}}. \tag{4.1}$$

The characteristic velocity scale V_s is the block impact velocity in this physical model and the characteristic length scale is the water depth. For a given length ratio of the prototype and the model λ_l, Table 4.6 shows the scaled factors of some parameters based on Froude similarity.

Table 4.6 Froude scaling factors for some parameters

Parameter	**Dimension**	**Scale factor**
Length	[L]	λ_l
Area	[L^2]	λ_l^2
Volume	[L^3]	λ_l^3
Time	[T]	$\lambda_l^{1/2}$
Velocity	[LT^{-1}]	$\lambda_l^{1/2}$
Acceleration	[LT^{-2}]	1
Force	[MLT^{-2}]	λ_l^3

Similarly, the Reynolds and Weber similarities focus on the ratio of the inertial and viscous forces and the inertial and surface tension forces, respectively. According to Heller (2011), the *Reynolds number* R is defined as

$$\mathrm{R} = \frac{g^{\frac{1}{2}} h^{\frac{3}{2}}}{\nu_w} \tag{4.2}$$

and the *Weber number* W is (Heller 2011)

$$\mathrm{W} = \frac{\rho_w g h^2}{\sigma_w}. \tag{4.3}$$

Here, R and W consider the shallow-water wave speed $\sqrt{gh}$ (Table 2.3) as the characteristic velocity. In Eqs. (4.2) and (4.3) ν_w is the kinematic viscosity and σ_w the surface tension of water.

The presented physical model is based on Froude similarity conducted in a large wave basin. According to Heller et al. (2008), scale effects for the closely related subaerial landslide-tsunamis may be neglected when R $\geq 3{\times}10^5$ and W ≥ 5000. For the five selected laboratory tests, with $g = 9.81$ m/s^2, $h = 1.0$ m, $\rho_w = 1000$ kg/m^3, $\nu_w = 1.01 \times 10^{-6}$ m^2/s and $\sigma_w = 0.073$ N/m, R and W are

$$\mathrm{R} = \frac{g^{\frac{1}{2}} h^{\frac{3}{2}}}{\nu_w} = 3.1 \times 10^6 > 3 \times 10^5, \tag{4.4}$$

$$\mathrm{W} = \frac{\rho_w g h^2}{\sigma_w} = 134384 > 5000. \tag{4.5}$$

Therefore, no significant scale effects for the measured parameters are expected due to the large size of the experiments.

4.5 Summary

Large-scale laboratory tests were designed and conducted to investigate the generation and propagation of iceberg-tsunamis (IBTs). The experimental setup to conduct the five selected tests has been explained. The calving iceberg mo- dels were two rigid blocks with similar densities ($\approx$ 920 kg/m^3) to that of ice and weighted up to 187 kg. The support frame and the triggering methods for different idealised calving mechanisms were described. The water depth was constant in the selected tests. In order to investigate IBT generation, a motion sensor was used to record the motion of the blocks. The propagation of IBTs was measured with up to 35 wave probes. Two cameras were used for general observations. The motion sensor and the wave probes together with the cameras were synchronised via a LabView executable and the synchronisation process has been explained. The post-processing of the raw data has been detailed, mainly involving calculating the velocity and trajectory of the blocks and removing noise and the offsets from the measured data. Wave reflection was not observed due to the short time window of 8.0 s of the wave probe data in this study. The experiments were conducted under Froude similarity which was briefly introduced. Given that the water depth in the wave basin was large and the criteria of Heller et al. (2008) are satisfied, scale effect are neglected in this laboratory tests.

Chapter 5

RESULTS

5.1 Introduction

The results presented in this chapter are mainly based on the numerical simulations. The newly implemented flow solver *interDyMIbFoam* described in Section 3.3 is validated with a theoretical heaving sphere case in Section 5.2, where the numerical results are compared with the analytical solution. Section 5.3 presents the simulation of the laboratory experiments introduced in Chapter 4. The focus is on mechanisms B and D (gravity-dominated fall and overturning) because they generated the largest and most dangerous iceberg-tsunamis (IBTs). Further, each of them involves only heave or pitch, i.e. one degree of freedom (DoF). These two cases are therefore well suitable for the numerical validation. The smaller and less important waves generated by the remaining mechanisms will only be discussed in Section 6.4.

An overview of the laboratory IBT results is provided in Section 5.3.1, followed by the convergence tests in Section 5.3.2. The simulation of the IBT generation and propagation is presented in Sections 5.3.4 and 5.3.5, respectively. The IBT generation contains the comparison of the iceberg kinematics between the laboratory tests and numerical simulations. Further, an analysis of the forces acting on the iceberg during the calving process is also provided based on the numerical results. The IBT propagation focuses on the wave features of the IBTs, including the first (leading) wave amplitude a_1, height H_1, period T_1, length L_1, celerity c_1, wave train energy as well as the wave decay.

5.2 Validation with a Theoretical Heaving Sphere Case

An analytical solution of the radiated waves by a floating hemisphere derived by Hulme (1982) is applied for the validation of *interDyMIbFoam*. The mathematical description of this theoretical case is detailed in Section 5.2.1, followed

by the numerical setup and results.

5.2.1 Mathematical description of the heaving sphere case

The spherical polar coordinate system (r,θ,ψ) shown in Fig. 5.1 is adopted to describe this theoretical case. The origin is located at the centre of the top face of the hemisphere with the z-axis pointing downwards. A floating hemisphere with a radius a_r performs an oscillation along the z-axis. The initial elevation of the top surface of the hemisphere corresponds to the still water level ($z = 0$), oscillates then with a displacement of $z = A\sin(\sigma t)$ where $A = 1$ m denotes the oscillation amplitude and σ the angular frequency. The boundary of the water body is at infinity, i.e. $r \to \infty$, $0 \leq \theta \leq \pi/2$ and $0 \leq \psi \leq 2\pi$. Here, the surrounding water is assumed to be irrotational, inviscid and incompressible.

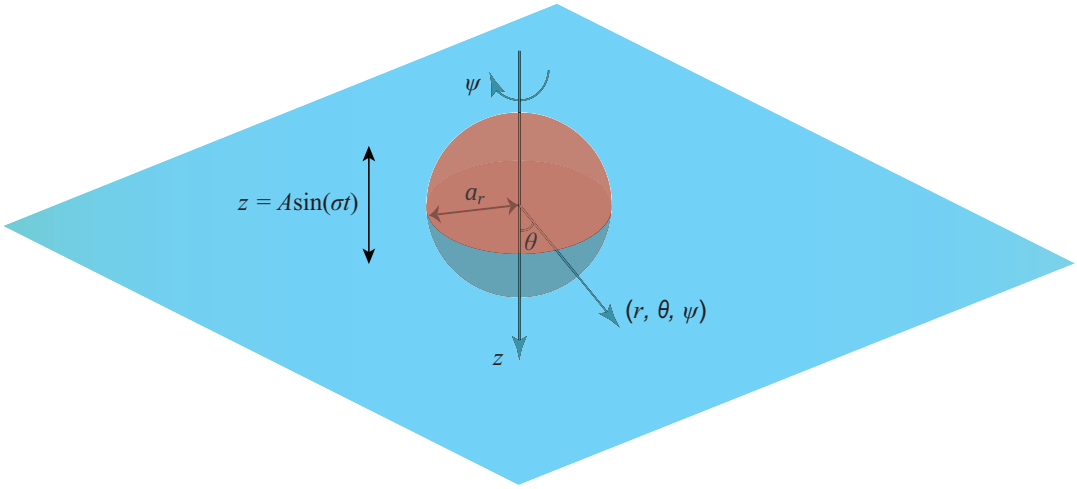


Figure 5.1 Sketch illustrating the key parameters for the heaving sphere case (adapted from Hulme, 1982)

The governing equations and boundary conditions for this theoretical case include the:

continuity equation

$$\Delta\Phi = 0, \tag{5.1}$$

free surface condition

$$(k + \frac{\partial}{\partial z})\Phi|_{z=0} = 0, \tag{5.2}$$

radiation condition

$$r^{\frac{1}{2}}(\frac{\partial}{\partial r} - ik)\Phi|_{r\to\infty} = 0 \tag{5.3}$$

and the body surface condition

$$\frac{\partial\Phi}{\partial r}|_{r=a_r} = \cos\theta, \tag{5.4}$$

where Δ denotes the Laplace operator in the spherical coordinate system, Φ the velocity potential of the surrounding water and $k = \sigma^2/g$ the wave number. It is noteworthy that only Eq. (5.4) is dependent on the geometry of the floating body. Under the assumption that A is sufficient small relative to a_r (Hulme, 1982), Eqs. (5.1) to (5.4) can also be applied to the full sphere case. Therefore, in this work, a full sphere case is considered in this validation. Φ is adopted from Hulme (1982) as

$$\Phi = \mathrm{Re}\{ACa_r^2[\phi_0 + \sum_{\mathrm{i}=1}^{\infty} p_{\mathrm{i}}^0 a_r^{2\mathrm{i}}\phi_{\mathrm{i}}]e^{-i\sigma t}\}, \tag{5.5}$$

where C and p_{i} are the complex constants to be solved and ϕ_0 and ϕ_{i} are the wave source and wave-free potentials, respectively. Appendix C provides details of the solution and values for Φ, C and p_{i}. The water surface elevation η can be obtained based on the linear wave theory

$$\eta = -\frac{1}{g}\frac{\partial\Phi}{\partial t}. \tag{5.6}$$

5.2.2 Numerical setup and results

The numerical basin for this theoretical case is shown in Fig. 5.2, which is established based on a Cartesian coordinate system. The origin is located at the basin centre on the still water level. The *x*-, *y*- and *z*-axes are directed along the length, width and height directions of the basin. The numerical basin is split into two zones: the wave generation and propagation zones. The wave generation zone is 1.0 m × 1.0 m × 3.7 m, while the wave propagation zone is 12.0 m × 12.0 m × 3.2 m. An air layer with a thickness of 0.2 m is located above the still water level. The bottom faces of these two zones are on the same plane (x, y, $z = -3.0$ m). The centre of the sphere with $a_r = 0.25$ m is located at the origin of the coordinate system.

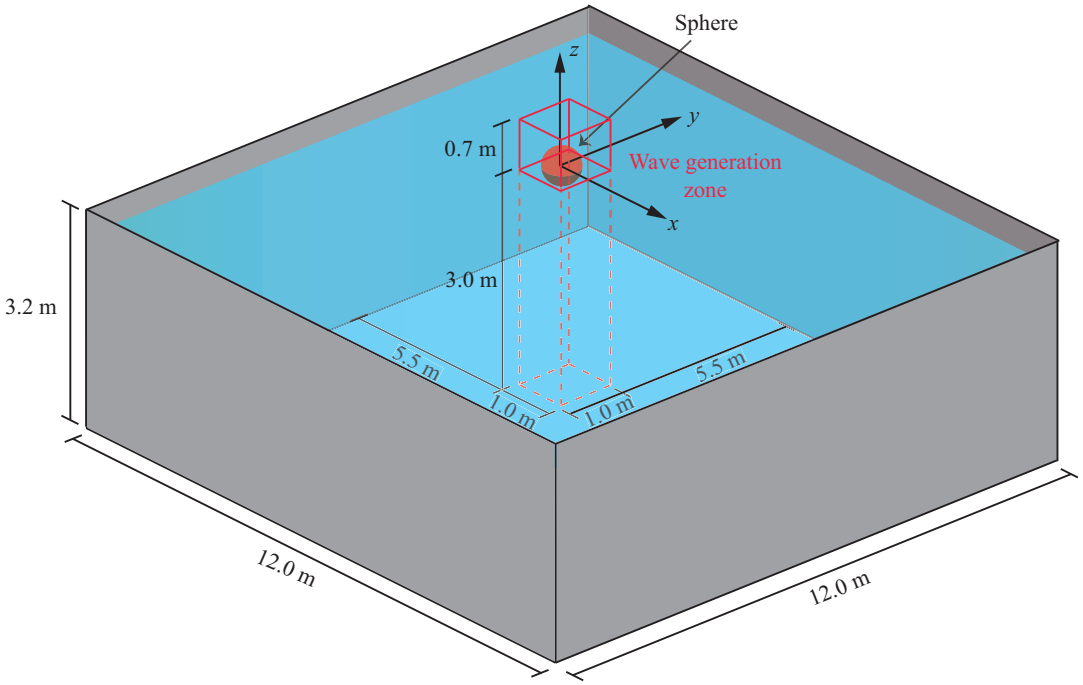


Figure 5.2 Sketch of the numerical basin used for the theoretical case

The same boundary conditions as listed in Table 3.3 were applied in this theoretical case. A small oscillation amplitude $A = 0.06$ m was chosen and the ratio of A to a_r was 0.24. The prescribed displacement $z = 0.06\sin(\sigma t + \pi)$, with $\sigma = 4.72$ rad/s, was specified in *constant/dynamicMeshDict* under the case folder. A resolution of 0.02 m $\times$ 0.02 m $\times$ 0.02 m was selected. The entire domain contained 57,662,500 cells and was decomposed evenly in the length and width directions by 12 $\times$ 12 subdomains. The simulation was run on the HPC (High Performance Computing) Midlands+ with 144 cores, which took approximately 36 h to simulate 8.0 s.

The numerical radiated waves by the heaving sphere at $x = 1.0$ and 3.0 m are shown in Fig. 5.3, together with the corresponding analytical solutions. The analytical solutions are asymptotic and obtained based on the steady state, whereas the numerical solutions are transient as the waves were gradually generated. Note that the numerical radiated waves at $x = 3.0$ m were affected by reflection within the simulation time, as marked in Fig. 5.3(b). The arrival time of the reflection was obtained based on the wave front travelling from the measurement location to the boundary and back and the wave celerity. Compared with the calculated theoretical wave amplitudes 0.0070 m at $x = 1.0$

m and 0.0025 m at $x = 3.0$ m, the relative errors of the mean wave amplitudes of the numerical results are -10.2 and -10.3%. The numerical wave periods are 0.1 and 1.0% larger than the theoretical of 1.33 s at $x = 1.0$ and 3.0 m, respectively.

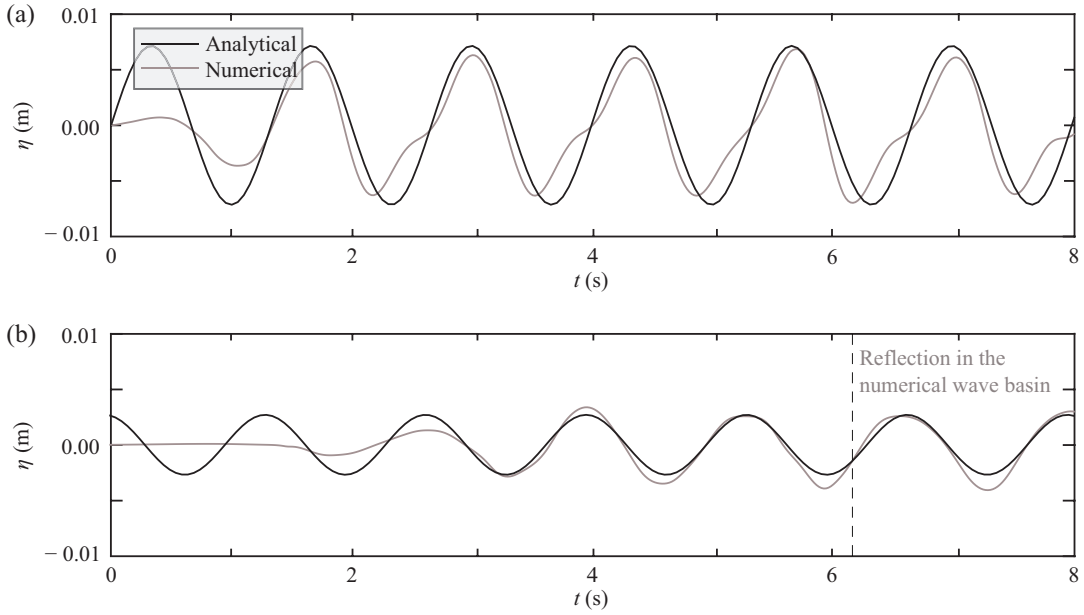


Figure 5.3 Comparison of the numerical water surface elevation η with the analytical solution at (a) $x = 1.0$ m and (b) $x = 3.0$ m

In order to evaluate the similarity of the reproduced wave profiles with the analytical solution, the nRMSE is introduced as

$$\mathrm{nRMSE} = \frac{\sqrt{\frac{\sum_{\mathrm{i}=t_1}^{t_\mathrm{N}} (\eta_{analytical,\mathrm{i}} - \eta_{numerical,\mathrm{i}})^2}{\mathrm{N}}}}{\eta_{analytical,\max} - \eta_{analytical,\min}}, \tag{5.7}$$

where N denotes the maximum indexing number in the wave data series. Here, nRMSE = 0 denotes identical numerical and analytical η. Compared with the analytical solution, the nRMSEs for the numerical η at $x = 1.0$ and 3.0 m in the steady state region are 0.134 and 0.165, respectively. The difference may be due to the geometry of the sphere represented by the IBM (Fig. 3.1) is not fully equal to its original shape, resulting in the irregularity in the reproduced wave profiles as well as the wave amplitudes and wave periods. Nevertheless, this validation with the theoretical case illustrates the potential of the newly implemented flow solver *interDyMIbFoam* for simulating IBT generation and propagation.

5.3 Simulation of Laboratory IBTs

5.3.1 Overview of the laboratory experiments

Snapshots of the IBTs generated in the laboratory tests are presented in this subsection showing the evolution of the IBTs involving mechanisms B and D. These snapshots shown in Figs. 5.4 and 5.5 are obtained from the 5 MP camera used at 15 Hz located at (6 m, 45°) (Table. 4.4). The time intervals between the frames are 1.33 s.

For mechanism B (Fig. 5.4), the block is released at $t = 0.00$ s and falls vertically as illustrated by the black arrow in Fig. 5.4(a). The block is then fully submerged at 1.33 s with a splash generated near the basin wall. Simultaneously, the first two waves can be seen in Fig. 5.4(b). With the block submerged more, the radiated waves progressively propagate in a semi-circular pattern at $t = 2.67$ s. At $t = 4.00$ s, due to the buoyancy force, the block moves upwards towards the water surface, with the block's top face reaching the water surface in Fig. 5.4(d).

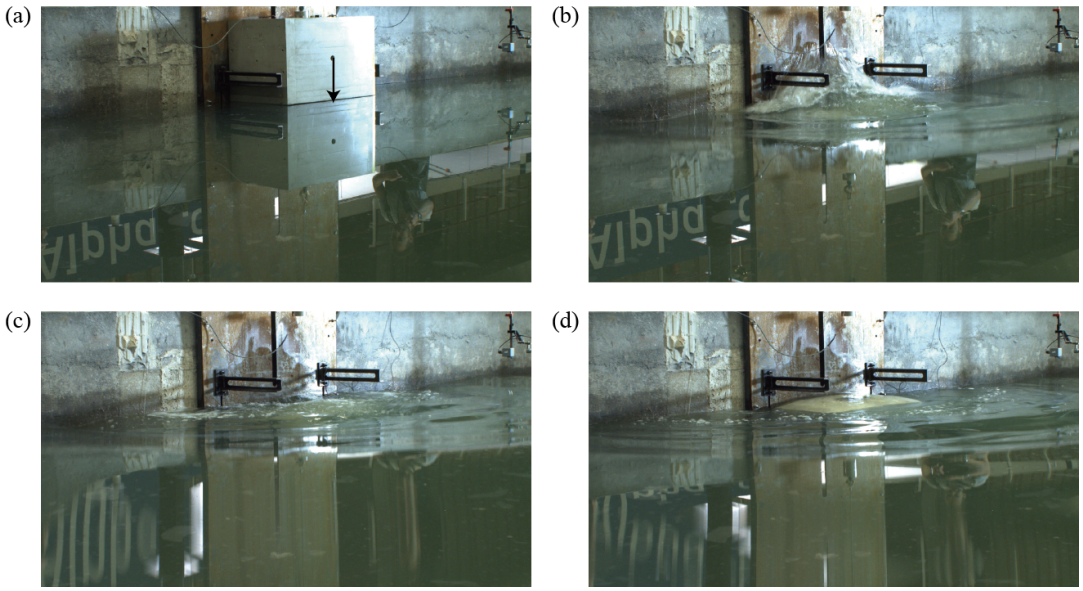


Figure 5.4 Snapshots of the IBTs generated in mechanism B in the laboratory tests at (a) $t = 0.00$ s, (b) $t = 1.33$ s, (c) $t = 2.67$ s and (d) $t = 4.00$ s (Chen et al., 2020)

For mechanism D in Fig. 5.5, the rotation of the block is initiated at $t =$

0.00 s (Fig. 5.5a). The angular velocity of the block increases then slowly. At $t = 1.33$ s, no obvious waves are observed as the block is still moving slowly. The rotation is still ongoing and the block is entirely submerged at $t = 2.67$ s. A large splash is then generated in the main wave propagation direction, i.e. $\gamma = 0°$ (Fig. 5.5c). In contrast with the fall case where the location of the splash is close to the basin wall, it is some distance away in the overturning case. At $t = 4.00$ s, the block edge can be seen in Fig. 5.5(d) as the block moves towards the water surface but has still not reached it.

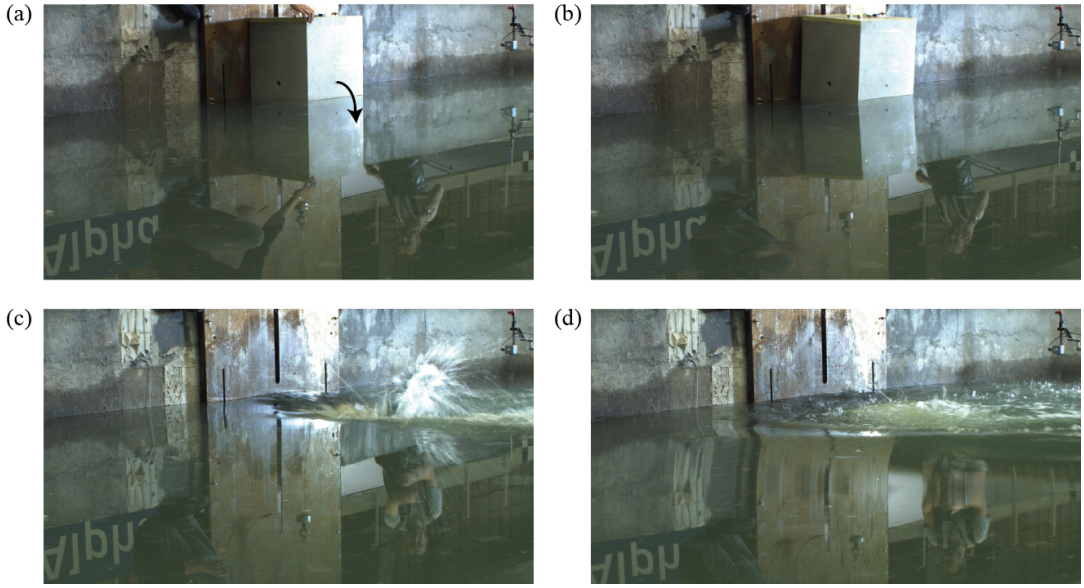


Figure 5.5 Snapshots of the IBTs generated in mechanism D in the laboratory tests at (a) $t = 0.00$ s, (b) $t = 1.33$ s, (c) $t = 2.67$ s and (d) $t = 4.00$ s (Chen et al., 2020)

5.3.2 Convergence tests

In order to investigate the effect of the cell dimensions on the numerical results, convergence tests were carried out. The mechanisms B and D cases are simulated based on the experimental setup (Section 4). The used numerical basin is shown in Fig. 5.6. The coordinate system applied here is the same as for the theoretical case, except that the origin is located at the centre of the block's back face on the water surface. The domain is 6.0 m $\times$ 6.0 m with a height of 1.7 m in the IBT generation zone (zone A) and 1.2 m in the propagation zone (zone B). The water depth $h = 1.0$ m was constant. The block was placed in front of the basin wall in mechanism B. There were two cell widths between the block and the

basin wall in mechanism D for the recognition of the immersed boundary and the accommodation of the rotation. This gap may affect the generated waves, which will be discussed in Section 6.5.

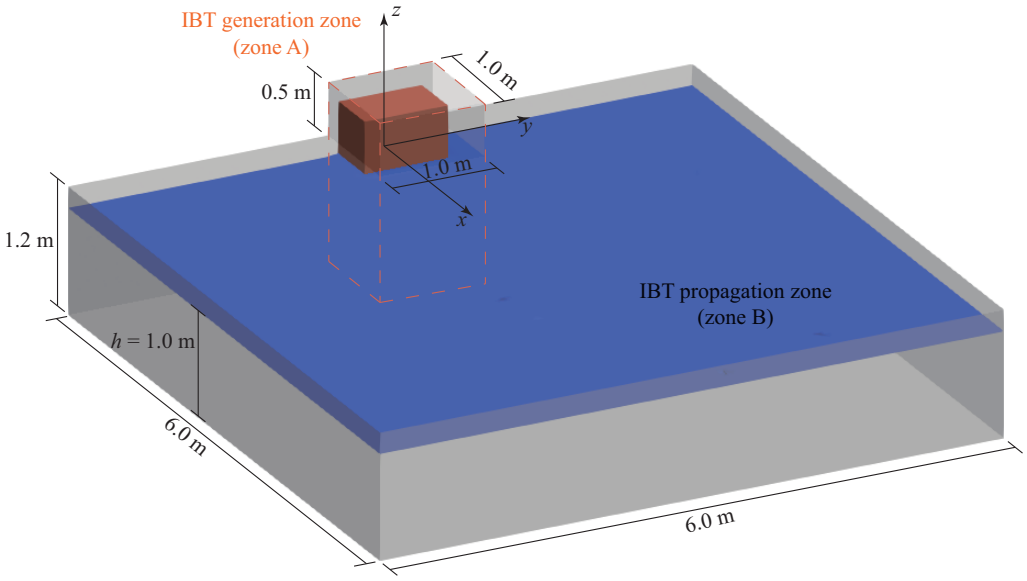


Figure 5.6 Computational domain used in the convergence tests. The brown dashed lines denote the IBT generation zone and the rest of the domain the IBT propagation zone

In zone A, the three resolutions 0.020 m, 0.025 m and 0.050 m in all three directions were selected. Four different cell dimensions were investigated in zone B, namely 0.050 m × 0.050 m × 0.050 m, 0.050 m × 0.025 m × 0.025 m, 0.025 m × 0.025 m × 0.025 m and 0.020 m × 0.020 m × 0.020 m. The cell numbers of the domain varied from 349,000 to 5,462,500. As the resolution may affect the force acting on the block resulting in different motions and generated waves, the motion is preserved for different resolutions in each mechanism. Therefore, the motion is prescribed based on the data from the laboratory tests (Fig. 5.7). The values in the vertical axis in Fig. 5.7 correspond to the displacements of the motion sensor, which was fixed on the top face of the block.

All the convergence tests were run in parallel mode on the HPC Minerva and the range of the requested cores and memory were from 3 cores and 4 GB to 30 cores and 36 GB. The coarsest resolution required 8 h to complete the simulation of 5 s real time, while the finest case took 96 h.

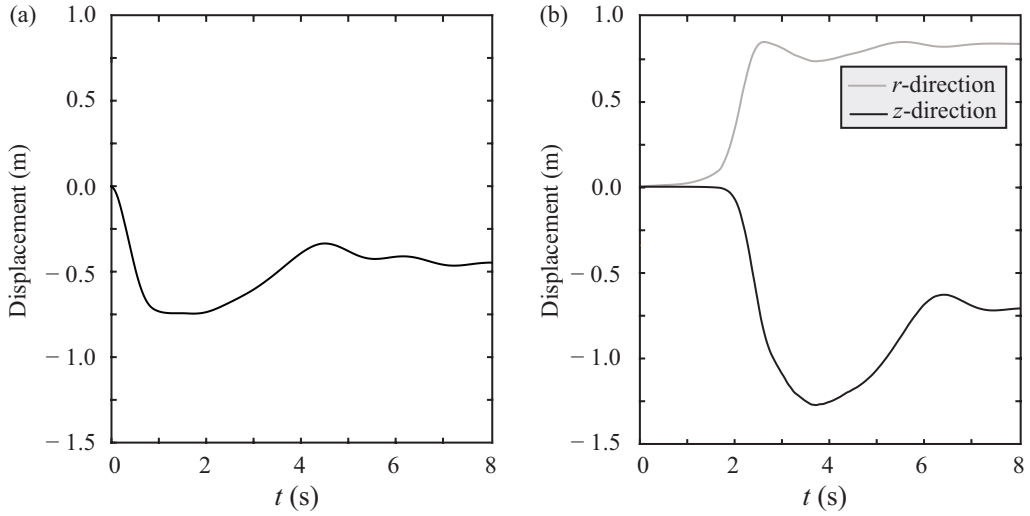


Figure 5.7 Time series of displacements obtained from the motion sensor data of (a) mechanism B along the z- and (b) mechanism D along the r- and z-directions (Chen et al., 2020)

Figure 5.8 shows the comparison of η at $x = 2.0$ m in the convergence tests with the corresponding waves measured in the laboratory tests. Based on the first wave amplitudes a_1, the differences between the two closest wave profiles with resolutions higher than 0.050 m $\times$ 0.025 m $\times$ 0.025 m, are 0.03 cm in mechanism B and 0.21 cm in mechanism D. Therefore, the cell dimension of 0.025 m $\times$ 0.025 m $\times$ 0.025 m in both the IBT generation and propagation zones was selected for the main tests, given that the finer mesh did not result in further benefits, only higher computation expenses.

Further, the IBTs simulated using the prescribed motion are always smaller than those measured in the laboratory tests. This may be due to the interpolation method of the applied IBM toolbox resulting in a slight underestimation of the evaluated velocity and pressure at the immersed boundary cells. This issue may be improved with a more accurate representation technique of the immersed boundary or a more robust interpolation method. However, this shortcoming is a smaller issue when involving the resolved motion in the main tests (Sections 5.3.4 and 5.3.5), where the fluid-structure interaction (FSI) is better simulated.

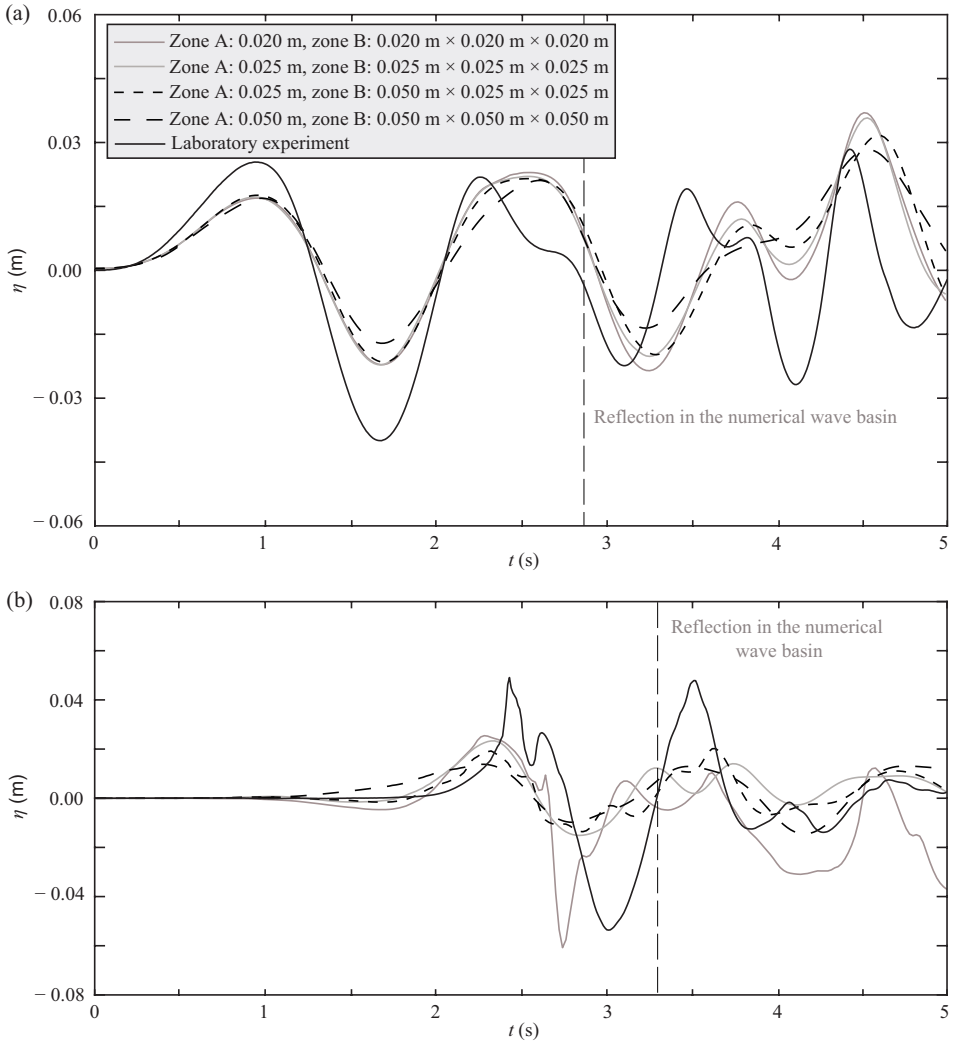


Figure 5.8 IBTs generated with prescribed motion in the convergence tests of mechanisms (a) B and (b) D (for the legend see a)

5.3.3 Numerical setup

The numerical setup for the main tests is presented in this subsection, using resolved iceberg motion (Sections 3.4 and 3.5) and the laminar model. The simulation involving turbulence is discussed in Section 6.3. Compared with the numerical basin applied in the convergence tests, the computational domain for the main tests has the same configuration except the length and width of zone B (IBT propagation zone), which are 15.0 and 18.0 m (Fig. 5.9), respectively. Based on the cell dimension of 2.5 cm in each direction, the numerical basin consists of 20,768,000 cells.

Note that some restrictions for the block's DoFs are required to ensure pure heave and pitch in these two cases. The force and moment components are set to zero at each time step in mechanisms B and D, respectively. Furthermore, only the acceleration along the z-axis and angular acceleration components along the y-axis are considered in solving Eqs. (3.25) and (3.26).

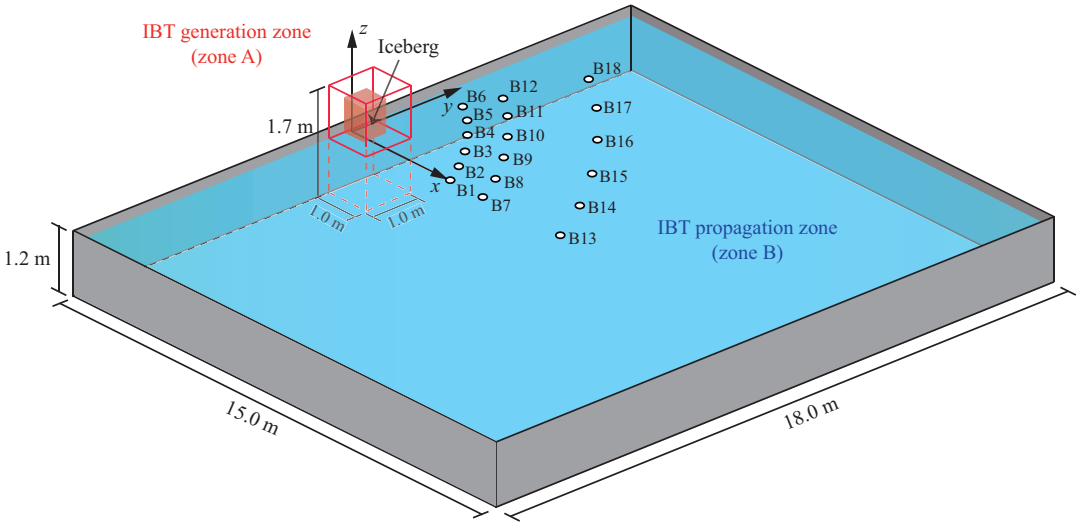


Figure 5.9 Sketch of the computational domain used in the main tests and the locations of the wave probes

An initial angular velocity ω around the y-axis is required to ensure a forward rotation of the iceberg in mechanism D, given that there is a gap between the iceberg and the boundary wall. Therefore, there are two varying calibration coefficients: the drag force coefficient C_d and the added mass coefficient C_m affecting the results in mechanism B, and the additional parameter ω in mechanism D. Even though not the same cases for C_d and C_m have been theoretically analysed, some indications for rectangular structures can be found in Lee (1995), namely $0.0 < C_d < 0.6$ and $0.4 < C_m < 0.8$. Note that $C_d = 0.0$ and $C_m = 0.0$ correspond to the commonly used method to calculate forces in FSI modelling, where the added mass and drag force are captured in the pressure and viscosity forces. The results obtained with $C_d = 0.0$ and $C_m = 0.0$ are therefore focused on in this chapter. C_d, C_m and ω together with other parameters used in these two cases are listed in Table 5.1. The decomposition in Table 5.1 denotes the numbers of the subdomains along the x-, y- and z-axes. These main tests were run on the HPC Midlands+ and took approximately 50 h to simulate 8.0 s of real time.

Table 5.1 Key parameters used in the numerical setup for mechanisms B and D

Calving mechanism	B	D
Iceberg length l (m)	0.500	0.800
Iceberg width b (m)	0.800	0.500
Iceberg thickness s (m)	0.500	0.500
Mass m_s (kg)	187.11	187.11
Centre of mass in initial position (x, y, z) (m)	(0.250, 0.000, 0.250)	(0.250, 0.000, 0.400)
Centre of rotation (x, y, z) (m)	NA	(0.250, 0.000, 0.000)
Moment of inertia I (I_x, I_y, I_z) (kg·m^2)	NA	(43.815, 43.815, 7.800)
Initial angular velocity ω (rad/s)	NA	0.02 and 0.03
Drag force coefficient C_d	0.0, 0.2, 0.4 and 0.6	0.0, 0.2, 0.4 and 0.6
Added mass coefficient C_m	0.0, 0.4, 0.6 and 0.8	0.0, 0.4 and 0.6
Water depth h (m)	1.00	1.00
Number of subdomains	54	54
Decomposition	$6 \times 1 \times 9$	$6 \times 1 \times 9$

5.3.4 IBT generation

The presented results for the IBT generation focus on the iceberg kinematics and the force acting on the iceberg during the calving process. First of all, the evolution of the numerical IBTs is presented. Since ω is introduced in mechanism D, the numerical time series are shifted with a time of 0.8 s to account for the laboratory block reaching a similar ω as in the numerical simulation. Figures 5.10 and 5.11 show snapshot series giving an overview of the mechanisms B and D cases. The evolution of the numerical IBTs generally agrees with the laboratory observations (Figs. 5.4 and 5.5).

In mechanism B, the splash near the basin wall at $t = 1.33$ s (Fig. 5.10b) is reasonably well simulated as well as the location of the iceberg (the top face of the iceberg just reaches above the water surface as shown in Fig. 5.4b) at $t = 4.00$ s. In mechanism D, the represented iceberg boundary during the rotation can be seen in Fig. 5.11(b), which does not fully preserve its original geometry due to the nature of the IBM (Section 2.5.2.2). Further, at $t = 1.33$ s, the iceberg rotates faster than measured in the laboratory experiments (Figs. 5.4b). Only a small part of the splashes are modelled in Figs. 5.10(b) and 5.11(b,c). In analogy

with the laboratory tests, the wave propagation is in a semi-circular pattern in both cases as illustrated in Figs. 5.10(c,d) and 5.11(c,d).

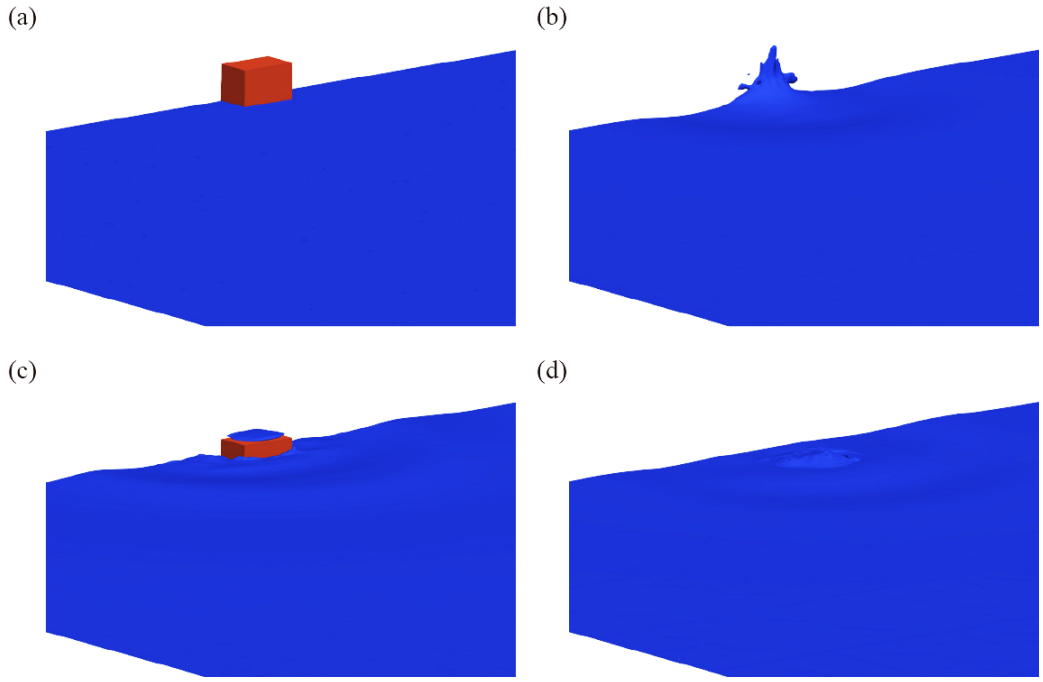


Figure 5.10 Snapshot series of mechanism B at (a) $t = 0.00$ s, (b) $t = 1.33$ s, (c) $t = 2.67$ s and (d) $t = 4.00$ s

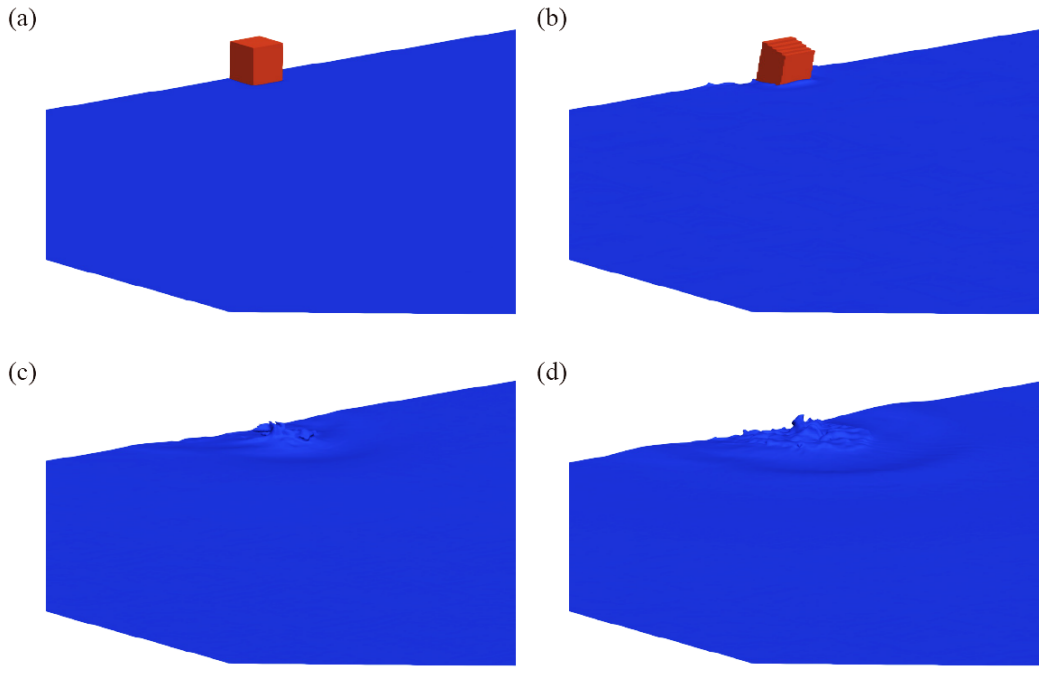


Figure 5.11 Snapshot series of mechanism D simulated with $\omega = 0.02$ rad/s at (a) $t = 0.00$ s, (b) $t = 1.33$ s, (c) $t = 2.67$ s and (d) $t = 4.00$ s

5.3.4.1 Numerically reproduced iceberg motion

The numerically reproduced iceberg motion involving varying C_d, C_m and ω in both mechanisms is shown in Figs. 5.12, together with the motion measured in the laboratory experiments. It can be seen that the iceberg motion in the numerical simulation is faster than in the laboratory tests.

For mechanism B (Fig. 5.12a and b), the top face of the block reached its lowest position $z = -0.24$ m at $t = 1.7$ s in the corresponding laboratory test, while the numerical iceberg's lowest position is $z = -0.42$ m at $t = 1.0$ s. The difference of the maximum displacement between the numerical simulation and the laboratory test is 24.3%. For the numerical results involving calibration coefficients C_d and C_m, the iceberg reached its lowest position $-0.40 < z < -0.33$ m at approximately $1.0 < t < 1.2$ s. Further, the numerical iceberg oscillates at least 3 times within 8 s, a minimum of 2 times more than in the laboratory tests (Fig. 5.12a and b). The top face of the iceberg in most of the simulation reaches around $z = 0.05$ m at $t = 8.0$ s, which agrees well with the laboratory measurement $z = 0.04$ m. The difference of the final position ($t = 8.0$ s) between the numerical and laboratory results accounts for 2% of the iceberg height. The relative difference of the maximum velocity between the numerical and laboratory results is 42.9%. By involving C_d and C_m, the range of the relative difference becomes 4.3 to 22.5%.

For mechanism D (Fig. 5.12c, d, e and f), the iceberg reaches its maximum rotation angle $142° < \theta_{\max} < 143°$ at 2.6 s $< t <$ 2.9 s in the numerical simulation with and without different coefficient combinations, while in the laboratory tests, it was $\theta_{\max} = 119.3°$ at $t = 3.6$ s. The maximum difference of this angle between the numerical and laboratory results is 19.9%. There are three oscillations of the iceberg rotation in the numerical and one in the laboratory after the iceberg reached $\theta_{\max}$. Regarding the final position of the iceberg, the rotation angle θ corresponds to $\theta = 139°$ in the numerical and $\theta = 93°$ in the laboratory tests with a difference of 40.4% relative to the laboratory $\theta_{\max}$. The vertical and horizontal components of the numerically reproduced velocity are much larger with 2.1 - 25.3% and 46.7 - 83.2%, respectively, than the corresponding laboratory measurements.

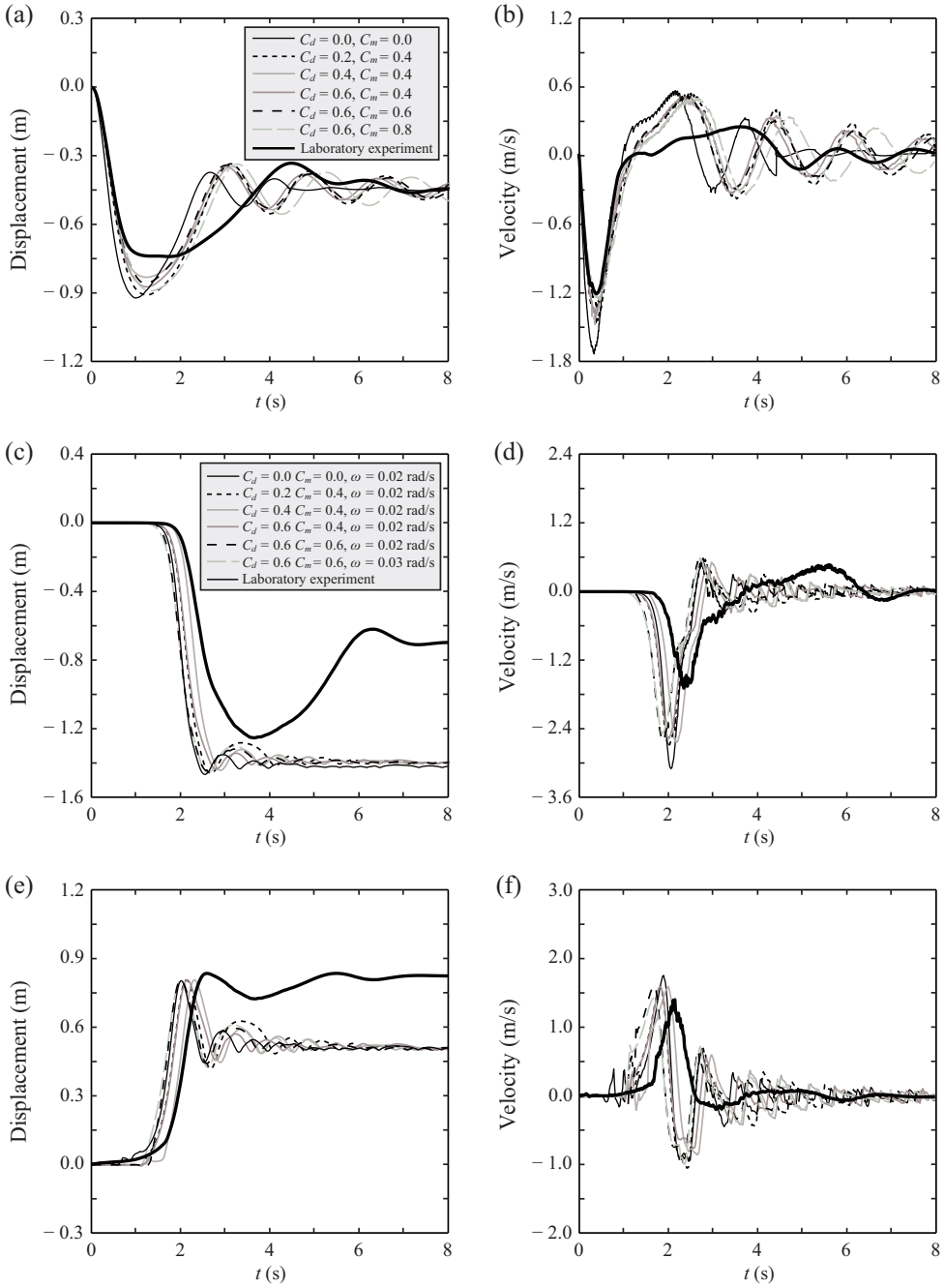


Figure 5.12 Comparison of the numerically reproduced iceberg motion in (a,b) mechanism B and (c,d,e,f) mechanism D with the corresponding measurements in the laboratory experiments; for the legends see (a) and (c), respectively

In general, increasing C_d and C_m slows the iceberg motion and a larger ω results in a faster initiation of the rotation (Fig. 5.12c). The reason for the faster numerical motion may be the overestimated underwater volume of the iceberg

represented by the IBM, causing a larger pressure force, which is the major force as demonstrated in Section 5.3.4.2, and thus the larger numerical acceleration. Again, the numerical simulations without these calibration coefficients result in the worst agreement with laboratory measurements.

5.3.4.2 Forces acting on the iceberg

The results of the forces acting on the iceberg during the IBT generation process from the two main tests shown in Figs. 5.10 and 5.11 are presented hereafter. The pressure force $\mathbf{F}_p$ and viscosity force $\mathbf{F}_v$ in function of time are shown in Figs. 5.13 and 5.14, respectively. $\mathbf{F}_p$ and $\mathbf{F}_v$ are the force components along the z-axis (Fig. 5.9).

In both mechanisms, $\mathbf{F}_p$ is the larger force. Compared with the gravity force $|\mathbf{G}| = 1835.5$ N, the maximum magnitudes of $\mathbf{F}_p$ are $1.71|\mathbf{G}|$ in mechanism B and $1.44|\mathbf{G}|$ in mechanism D, respectively. The viscosity force $\mathbf{F}_v$ is three orders of magnitude smaller than $\mathbf{F}_p$. This may be because the current IBM toolbox cannot well simulate the drag force due to the applied interpolation method, underestimating the immersed boundary's normal velocity gradient.

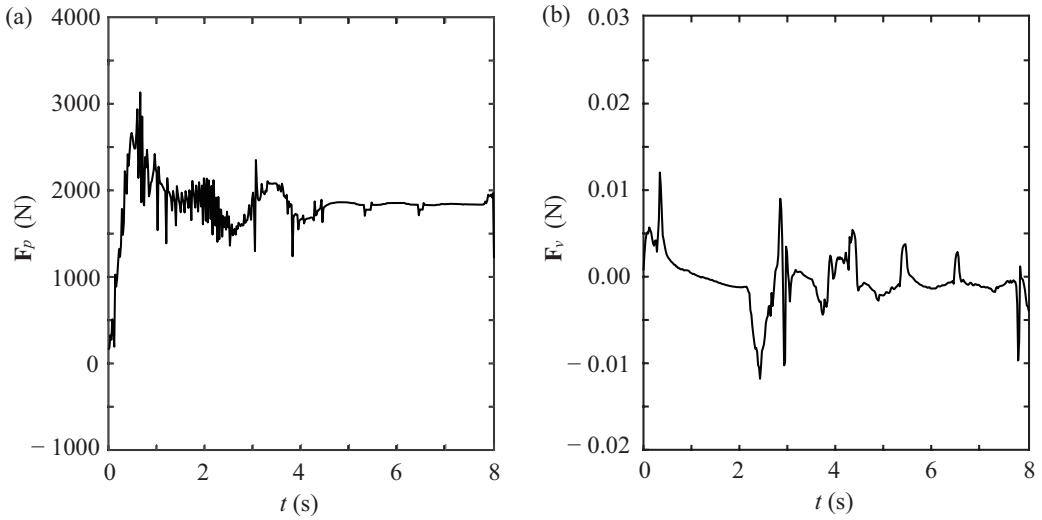


Figure 5.13 Time series of (a) the pressure force and (b) the viscosity force acting on the iceberg in mechanism B

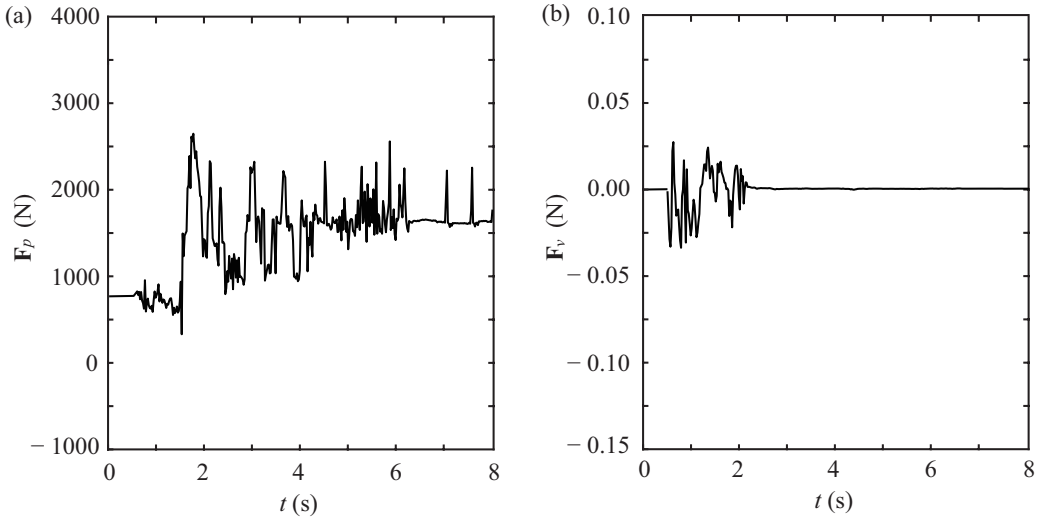


Figure 5.14 Time series of (a) the pressure force and (b) the viscosity force acting on the iceberg in mechanism D

Some numerical oscillations can be seen in Figs. 5.13 and 5.14, which may be caused by the difference of half a time step between the real iceberg velocity and the one used to update the iceberg position in the leapfrog scheme (Section 3.4.3.1). The maximum oscillation amplitudes of $\mathbf{F}_p$ are less than 25% of the corresponding maximum force magnitudes, while for $\mathbf{F}_v$, it shows large oscillations (Figs. 5.13d and 5.14d). However, given that the magnitude of $\mathbf{F}_v$ is very small, the oscillations of $\mathbf{F}_v$ on the overall force balance can be neglected.

5.3.5 IBT propagation

In this subsection, the wave parameters of the numerical IBTs are presented, including the first (leading) wave amplitude a_1, height H_1, period T_1 and celerity c_1. Further, the wave train energy of the numerical IBTs is obtained and compared with the laboratory results. Finally, a comparison of the numerical and laboratory wave decays are provided and discussed.

5.3.5.1 IBT wave parameters

The numerical IBTs together with those obtained with different C_d, C_m and ω are shown in Fig. 5.15. They were measured at wave probe B1 ($r = 2.0$ m, $\gamma = 0°$). The wave parameters in these two cases are listed in Table 5.2, which

are obtained at wave probe B1 based on the first waves in mechanisms B and D. The relative difference between the numerical and laboratory a_1 and H_1 are -17.2 and -6.5% in mechanism B and -41.2 and -29.7% in mechanism D. The large difference in a_1 and H_1 in mechanism D is because the peak of the first wave crest (Fig. 5.15b) is affected by the splash in the laboratory tests (Fig. 5.5c). Due to the selected cell dimension and the applied laminar model, this splash is not fully simulated. The relative error of T_1, L_1 and c_1 between the numerical and laboratory results are -5.9, -8.3 and -2.6% in the fall case and 14.8, 24.6 and 8.7% in the overturning case, respectively.

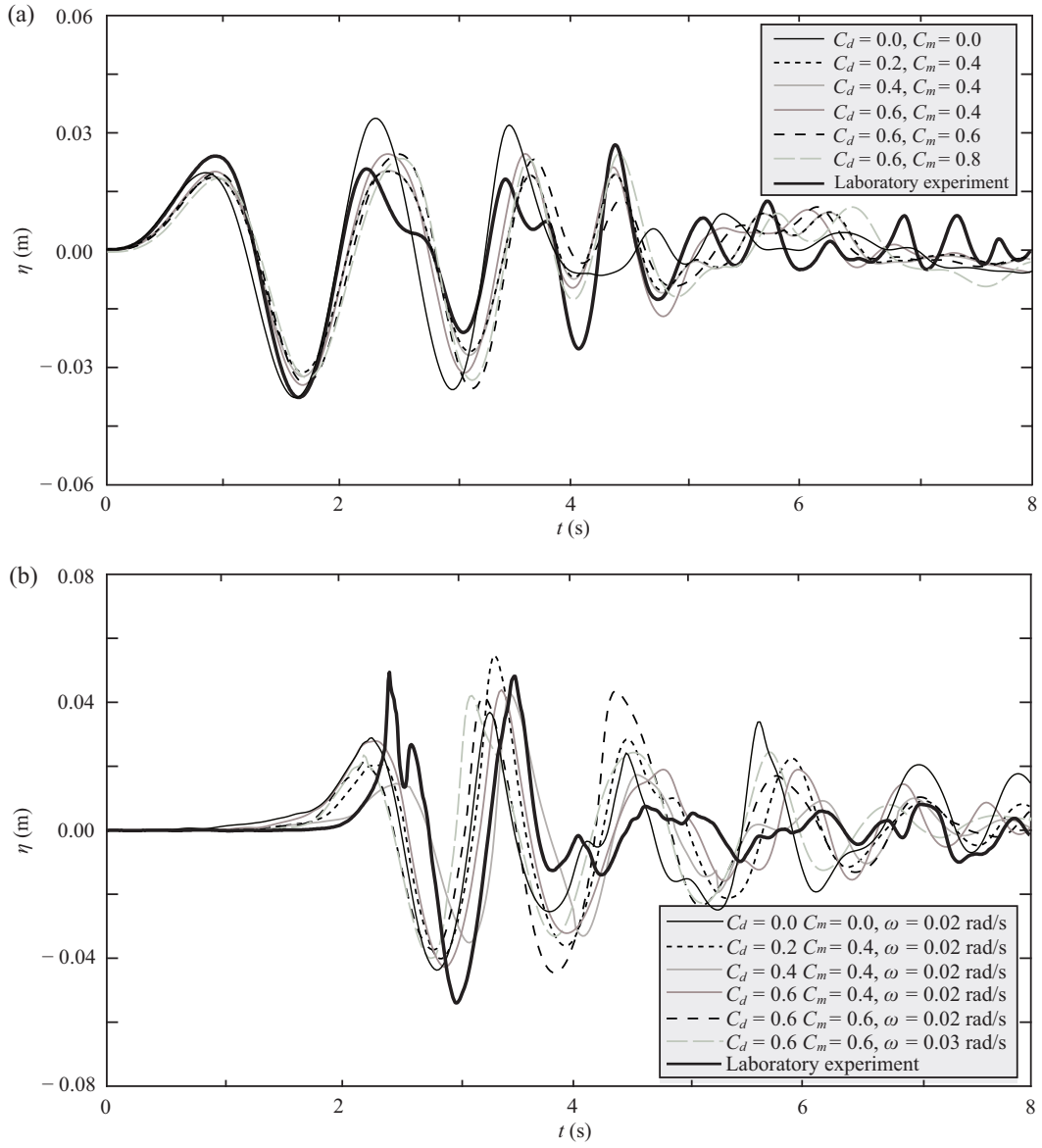


Figure 5.15 IBTs measured at $r = 2.0$ m involving varying C_d, C_m and ω for (a) mechanism B and (b) mechanism D

For the results involving C_d and C_m, all the IBTs in the numerical simulations are also smaller than in the laboratory. The ranges of the relative differences between the simulated and laboratory a_1 are -15.5 to -23.7% in the fall case and -41.2 to -70.7% in the overturning case, respectively. In general, increasing C_d and C_m results in slower iceberg motion, smaller IBT heights and wave celerity. Based on a_1 and H_1 and the corresponding time of a_1, the best agreement between the simulated and laboratory IBTs is achieved for $C_d = 0.6$ and $C_m = 0.4$ in mechanism B and $C_d = 0.6$, $C_m = 0.4$ and $\omega = 0.02$ rad/s in mechanism D. The relative errors between the simulated and laboratory a_1 and H_1 are then -15.5 and -12.6% in the fall case and -44.5 and -33.3% in the overturning case, respectively. In the following comparison with the numerical and laboratory IBTs, the results obtained with C_d and C_m are from the simulation involving this optimised parameter combination $C_d = 0.6$, $C_m = 0.4$ (and $\omega = 0.02$ rad/s).

However, if excluding the splash by using the measured $a_1 = 0.0292$ m at wave probe B2 ($r = 2.0$ m, $\gamma = -15°$ where no splash is observed) and interpolating this value from $\gamma = -15°$ to $\gamma = 0°$ with the term $\cos(\gamma/2)$ in Eq. (B.10) in Heller et al. (2020), $a_1 = 0.0294$ m and $H_1 = 0.0853$ m result and the relative error reduces to 1.0 and -12.3%, respectively. In both cases a_1 and H_1 are then well captured. Further, compared with the laboratory results, the first numerical wave troughs are 0.3 larger and 19.3% smaller, respectively. This underestimation may be because that the aforementioned larger numerical acceleration of the iceberg (Section 5.3.4.1) makes the iceberg move back to the water surface faster, inhibiting the growth of the first wave.

Table 5.2 IBT wave parameters obtained based on wave probe B1 in mechanisms B and D. The relative error is defined as $(p_{numerical} - p_{laboratory})/p_{laboratory} \times 100\%$ with p denoting any wave parameter

Mechanism		a_1 (m)	H_1 (m)	T_1 (s)	L_1 (m)	c_1 (m/s)
B	Laboratory	0.0238	0.0619	2.02	5.29	2.62
	Numerical	0.0197	0.0579	1.90	4.85	2.55
	Relative error (%)	−17.2	−6.5	−5.9	−8.3	−2.6
D	Laboratory	0.0505	0.1064	1.55	3.54	2.28
	Numerical	0.0297	0.0748	1.78	4.41	2.48
	Relative error (%)	−41.2	−29.7	14.8	24.6	8.7

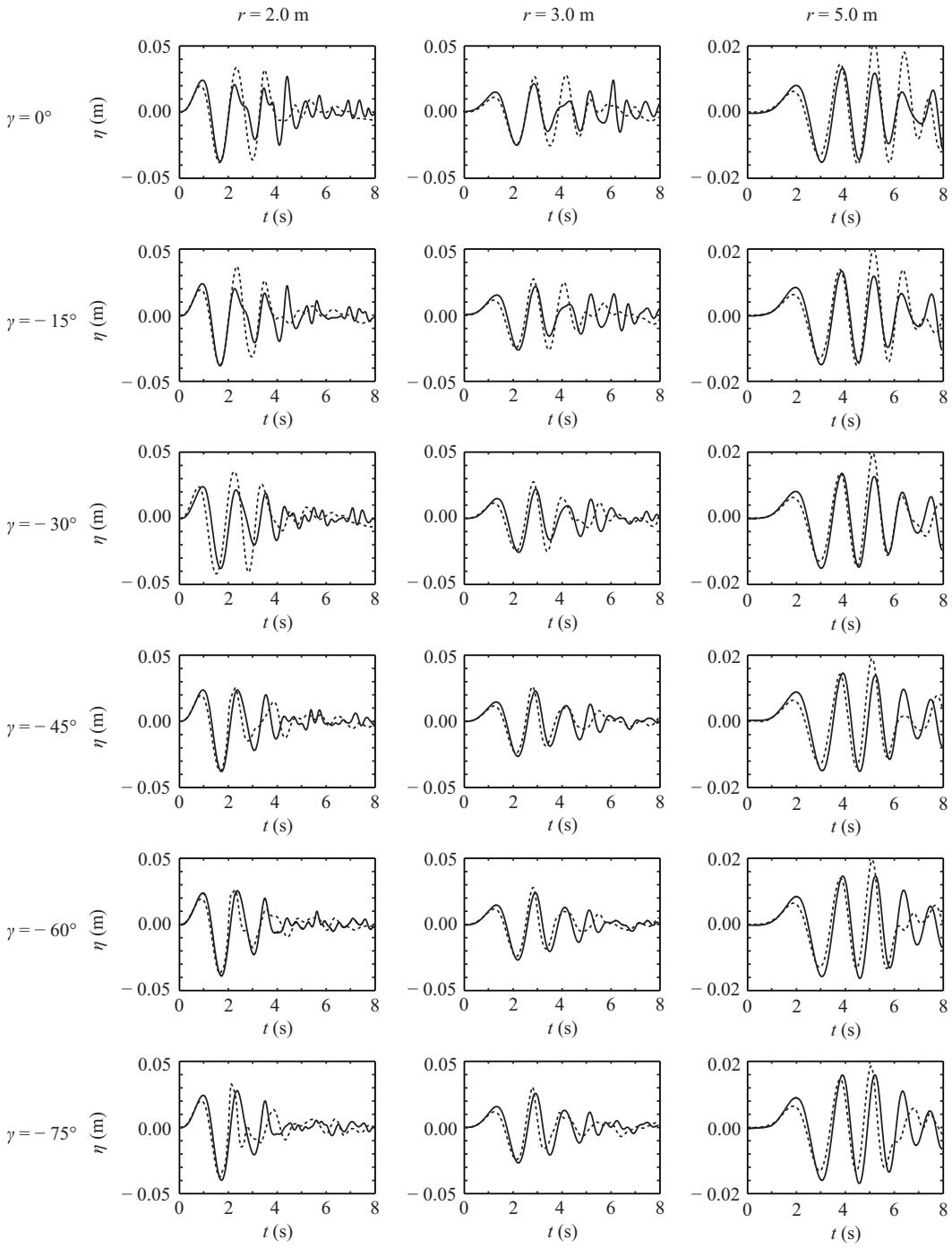


Figure 5.16 IBTs measured at wave probes B1 to B18 in mechanism B with the solid lines denoting the laboratory results and the dashed lines the numerical IBTs. The wave profiles in each column are measured at the same radial distance r and in each row at the same propagation angle γ

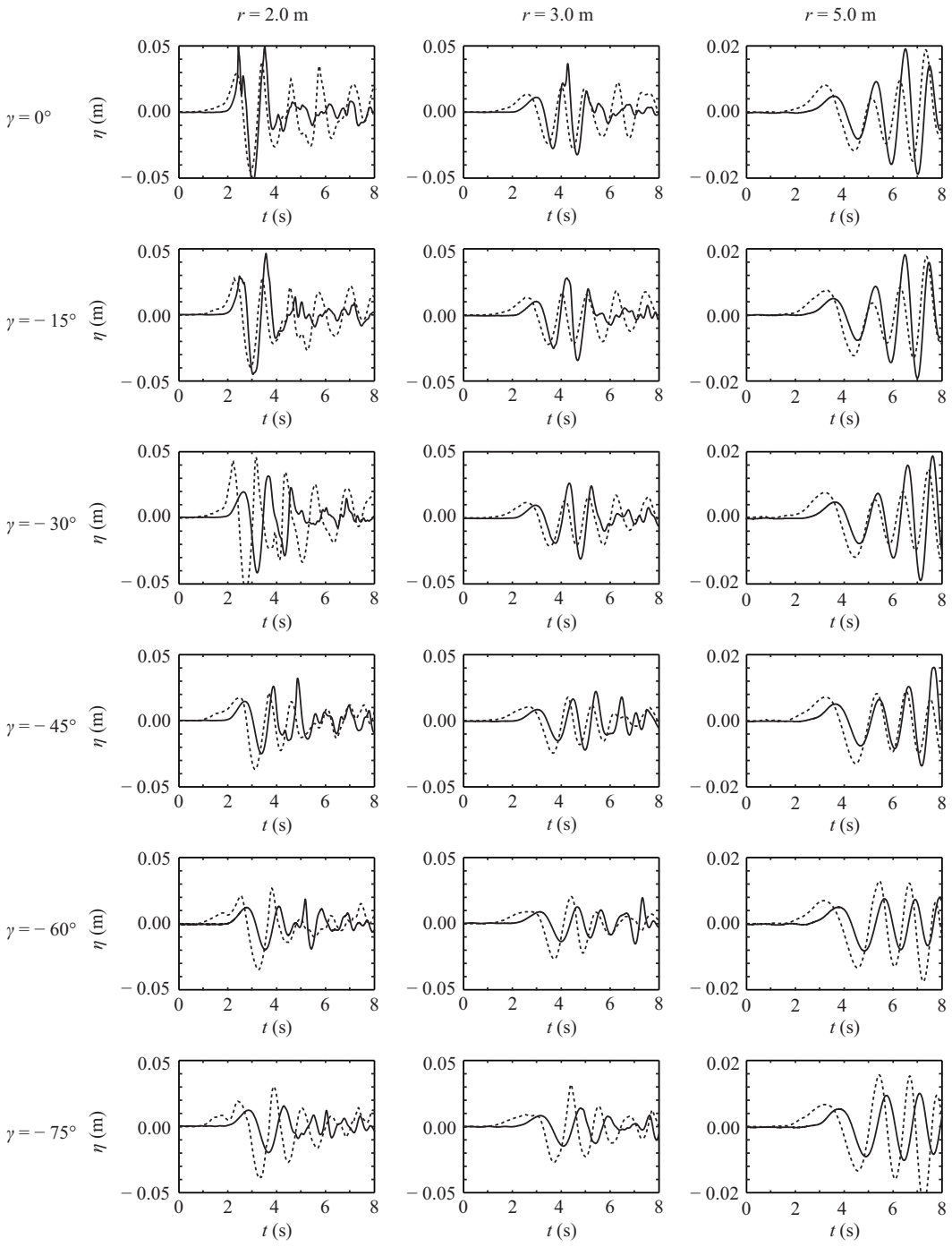


Figure 5.17 IBTs measured at wave probes B1 to B18 in mechanism D with the solid lines denoting the laboratory results and the dashed lines the numerical IBTs. The wave profiles in each column are measured at the same radial distance r and in each row at the same propagation angle γ

Figures 5.16 and 5.17 show the wave profiles measured at different radial distances r and propagation angles γ. In general, the numerical results (wave profiles and wave periods) agree with the laboratory measurements, demonstrating that the numerical model is well able to simulate frequency dispersion. However, there is a discrepancy in the wave celerity in mechanism D where the numerical IBTs arrive earlier. This may be due to the larger underwater acceleration of the iceberg resulting in faster IBT generation.

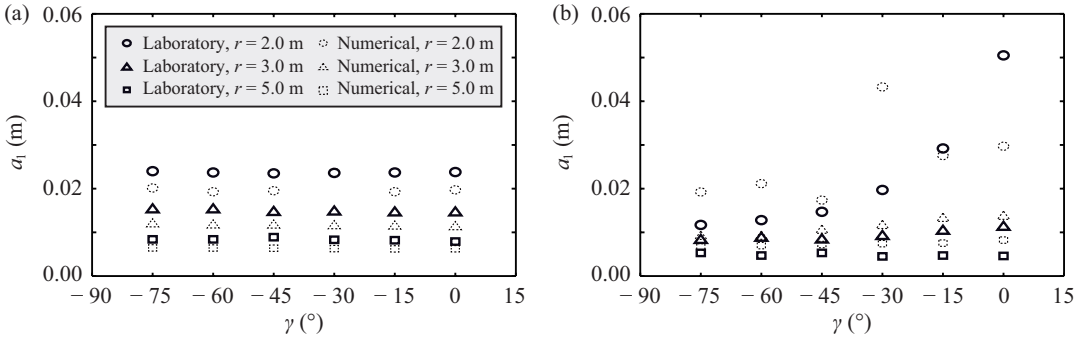


Figure 5.18 a_1 measured at wave probes B1 to B18 in (a) mechanism B and (b) mechanism D (for the legend see a)

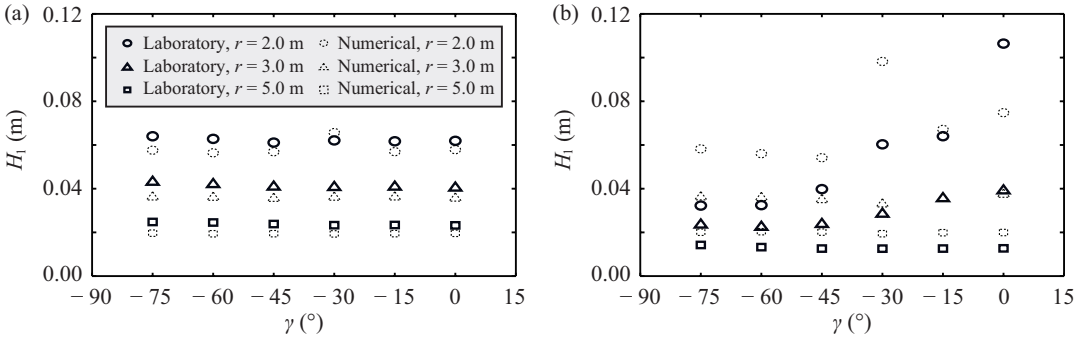


Figure 5.19 H_1 measured at wave probes B1 to B18 in (a) mechanism B and (b) mechanism D (for the legend see a)

For IBTs in different propagation directions, the comparison between the numerical and laboratory maximum a_1 at different wave probes is illustrated in Figs. 5.18 and 5.19. Based on the 8.0 s time window of the simulation and the 18 wave probes, the numerical IBTs propagate evenly along different directions in mechanism B for a given r, which can also be observed in the laboratory experiments. The difference of a_1 is less then 0.005 m in the numerical and 0.0005 m in the laboratory results, or 25.4% and 2.1% of the corresponding a_1

at $r = 2.0$ m and $\gamma = 0°$, respectively. In contrast, a_1 generally increases with increasing γ in mechanism D. The largest a_1 is observed for $\gamma = 0°$, even when the effect of the splash is excluded.

5.3.5.2 Wave train energy

The calculation of the IBT train energy E_w is based on the same method as in Heller et al. (2019c). The wave train energy passing the semi-circle with a radius of $r = 2.0$ m (Fig. 4.7b) computes as

$$E_w = \sum_{\mathrm{i}=1}^{6} \sum_{\mathrm{j}=1}^{\mathrm{n}} \frac{1}{8} \rho_w g H_{\mathrm{ij}}^2 L_{\mathrm{ij}} c_{\mathrm{i}}. \tag{5.8}$$

In Eq. (5.8) i is the index number of the wave train along $\gamma = \gamma_{\mathrm{i}}$ ($\gamma_{\mathrm{i}} = 0,\ -15,\ -30,\ -45,\ -60$ and $-75°$), j the index number of the j^{th} wave in the i^{th} wave train and n is the number of waves in the i^{th} wave train. H_{ij} and L_{ij} denote the wave height and length of the j^{th} wave in the i^{th} wave train. L_{ij} are calculated using the corresponding T_{ij} and linear wave theory (Eq. (2.18)). The term c_{i} in Eq. (5.8) is the arc length of the i^{th} sector with its symmetry axis directed at $\gamma = \gamma_{\mathrm{i}}$. Considering that the field is symmetric about $\gamma_{\mathrm{i}} = 0$ and the sum of all 6 sectors needs to be equal to the semi-circle with a radius of $r = 2.0$ m, the central angle of the i^{th} sector is specified as $\pi/12$ (i = 1), $\pi/6$ (i = 2, 3, 4 and 5) and $\pi/4$ (i = 6).

Table 5.3 shows the calculated numerical (subscript n) IBT train energy $E_{w,n}$ and laboratory (subscript l) $E_{w,l}$. Note that $E_{w,l}$ is calculated for the first 8.0 s and is smaller than the corresponding values in Heller et al. (2019c) computed for 20.0 s. $E_{w,n}$ and $E_{w,l}$ show a good agreement in mechanism B. However, in mechanism D, $E_{w,n}$ is approximately 3 times larger than $E_{w,l}$. This may be due to the underestimation of the numerical drag force in the IBM, which leads to 3 times more oscillations of the iceberg than in the laboratory test. In contrast, based on the results obtained with the optimised C_d and C_m, the calculated $E_{w,n}$ is 198.67 and 71.21 J, which is 1.9 and 13.0% larger than $E_{w,l}$ in mechanisms B and D, respectively.

Table 5.3 Comparison of the wave train energy based on the laboratory tests and numerical simulations

	Mechanism B	Mechanism D
Laboratory wave train energy $E_{w,l}$ (J)	195.06	63.01
Numerical wave train energy $E_{w,n}$ (J)	196.89	253.31
$(E_{w,n} - E_{w,l})/E_{w,l} \times 100$ (%)	1.0	302.0

5.3.5.3 Wave decay

Wave decay is a key aspect for IBT propagation and it plays an important role in hazard assessment for offshore and coastal structures. In the main propagation direction, the numerical and laboratory wave profiles $\eta(t)$ measured at wave probes B1, B7 and B13 (r = 2.0, 3.0 and 5.0 m) are shown in Fig. 5.20. The IBTs decay with increasing r. In order to evaluate the wave decay, it is assumed that IBTs decay with a power function

$$\frac{a_1}{h}(\frac{r}{h},\ \gamma) \sim (\frac{r}{h})^c,\ \frac{H_1}{h}(\frac{r}{h},\ \gamma) \sim (\frac{r}{h})^c. \tag{5.9}$$

In Eq. (5.9), the relative first wave amplitude a_1/h and height H_1/h are adopted with c as the decay exponent. Table 5.4 shows the numerical c_n and laboratory c_l decay exponents for a_1 and H_1. Note that c is calculated using only the first three wave probes for each γ, which is different from $c_l = -1.21$ (for a_1) and -1.20 (for H_1) obtained by Heller et al. (2020) who used all wave probe data.

For mechanism B, c_n and c_l for both wave amplitude and height decay show a good agreement with a maximum deviation of 16.0% apart from the IBTs along $\gamma = -30°$, where the larger numerical a_1 and H_1 result in a faster wave decay. For mechanism D, however, c_n for the wave amplitude and height is up to 32.8% smaller than the corresponding value for c_l. Again, the faster laboratory IBT decay is due to the large splash affecting the first wave crest. If the splash is excluded and a_1 = 0.0294 m is adopted (Section 5.3.5.1), then the relative differences between c_n and c_l for the wave amplitude and height decay are -19.1% and -9.2%, respectively. The splash is of limited relevance for the wave propagation in the far field not only along $\gamma = 0°$ but also along other directions (Fig. 5.17). Therefore, the exclusion of the splash is justified. Further, both the wave amplitude and heigh c_n is larger than c_l in $\gamma = -30°$, which is in contrast with the general decay trend in other directions. This may

be because there is a splash in the numerical IBTs at $r = 2.0$ m and $\gamma = -30°$, where the first crest is sharp and a_1 is approximately twice as large as in the laboratory (Fig. 5.17).

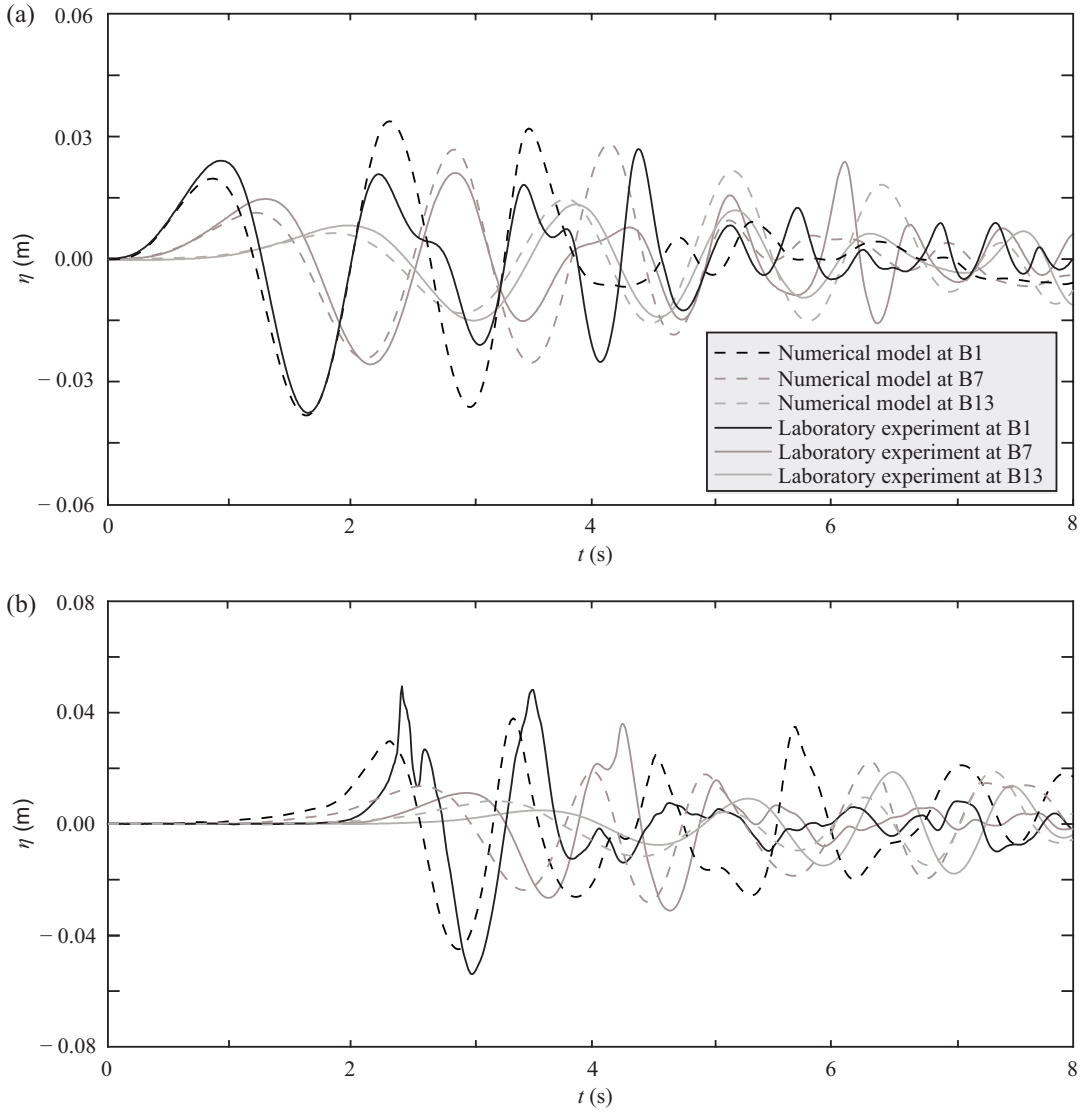


Figure 5.20 Numerical and laboratory IBTs measured at wave probes B1, B7 and B13 in (a) mechanism B and (b) mechanism D (for the legend see a)

Table 5.4 Comparison of the laboratory and numerical wave decay: a_1/h and H_1/h measured along different directions and laboratory c_l and numerical c_n wave decay exponents

Propagation angle γ (°)			0	−15	−30	−45	−60	−75
Mechanism B	Relative wave amplitude a_1/h	Laboratory c_l (-)	−1.214	−1.179	−1.145	−1.071	−1.118	−1.141
		Numerical c_n (-)	−1.301	−1.258	−1.591	−1.243	−1.217	−1.265
		$\frac{(c_n - c_l)}{c_l} \times 100$ (%)	7.2	6.7	39.0	16.0	8.8	10.8
		Mean decay exponent over all directions $c_l = -1.15$						
		Mean decay exponent over all directions $c_n = -1.31$						
	Relative wave height H_1/h	Laboratory c_l (-)	−1.090	−1.051	−1.061	−1.020	−1.021	−1.022
		Numerical c_n (-)	−1.189	−1.158	−1.390	−1.171	−1.149	−1.168
		$\frac{(c_n - c_l)}{c_l} \times 100$ (%)	9.1	10.2	31.0	14.8	12.6	14.2
		Mean decay exponent over all directions $c_l = -1.04$						
		Mean decay exponent over all directions $c_n = -1.20$						
Mechanism D	Relative wave amplitude a_1/h	Laboratory c_l (-)	−2.619	−2.391	−1.762	−1.204	−0.962	−0.869
		Numerical c_n (-)	−1.618	−1.585	−2.064	−1.034	−1.304	−1.231
		$\frac{(c_n - c_l)}{c_l} \times 100$ (%)	−32.8	−33.7	17.1	−14.2	−35.5	−41.7
		Mean decay exponent over all directions $c_l = -1.98$						
		Mean decay exponent over all directions $c_n = -1.66$						
	Relative wave height H_1/h	Laboratory c_l (-)	−2.201	−1.619	−1.815	−1.263	−0.962	−0.869
		Numerical c_n(-)	−1.555	−1.422	−2.298	−1.083	−1.105	−1.167
		$\frac{(c_n - c_l)}{c_l} \times 100$ (%)	−29.3	−12.2	26.6	−14.3	14.9	34.3
		Mean decay exponent over all directions $c_l = -1.61$						
		Mean decay exponent over all directions $c_n = -1.44$						

5.4 Summary

This chapter presented the results of this study. The newly implemented flow solver *interDyMIbFoam* has been validated with a theoretical floating heaving sphere case using prescribed motion. The analytically predicted radiated waves were well simulated including both the wave amplitudes and periods. This demonstrates the potential of the proposed numerical model to simulate both the generation and propagation of large iceberg-tsunamis (IBTs).

An overview of the laboratory tests is illustrated with snapshot series, prior to the simulations of the laboratory experiments (the main tests). Convergence tests revealed a resolution of 0.025 m in all directions as the best compromise between accuracy and computational cost in the main tests. For IBT generation, the numerically reproduced iceberg motion was faster than in the laboratory. This is due to the larger underwater volume presented by the immersed boundary method (IBM) toolbox resulting in a larger pressure force and thus acceleration. Compared with the laboratory tests, the numerical simulations involving C_d and C_m show a better agreement than those without these calibration coefficients. For forces acting on the iceberg, the pressure force was found to be the largest force and the viscosity force is negligibly small.

For IBT propagation, the parameters were well captured apart from the waves affected by splashes and the numerical IBTs were in general smaller than in the laboratory. Because of using a laminar model (Section 3.3.2) and the chosen resolution, the splash was not fully simulated. For the IBT train energy, the numerical and laboratory results show a good agreement in mechanism B and a large difference in mechanism D due to the oscillation of the iceberg in the numerical simulation. This may be because the drag force is not well captured by the current IBM.

Finally, the wave decay has been calculated. In mechanism B, the numerical decay exponents agree with the laboratory results. In mechanism D, however, the laboratory IBTs decay faster than the numerical IBTs, which is again due to the splash. If the splash is excluded then the wave height decay in both mechanisms are well simulated in the main IBT propagation direction with a maximum deviation of 9.2% of the decay exponents. The overall results show

that the numerical model was successful in simulating both IBT generation and propagation.

Chapter 6

DISCUSSION OF THE RESULTS

6.1 Introduction

In this chapter, the results of the main tests are discussed and further applications of the proposed numerical approach are demonstrated, i.e. in calculating the iceberg contact force, in involving the turbulence model and in simulating the remaining calving mechanisms. The presented results related to the main tests are obtained with the calibration coefficients $C_d = 0.6$ and $C_m = 0.4$, as they generally show a better agreement with the laboratory results than $C_d = 0.0$ and $C_m = 0.0$ (Sections 5.3.4 and 5.3.5).

For iceberg-tsunami (IBT) generation, the contact force of the iceberg is compared with forces from other studies in Section 6.2. For IBT propagation, the IBTs simulated with a turbulence model are presented in Section 6.3. Additionally, the wave parameters are compared with the corresponding laminar model and laboratory results, including the first maximum wave amplitude a_1, height H_1 and the wave decay in the main propagation direction $\gamma = 0°$. The capabilities of the new numerical model in dealing with the remaining iceberg calving mechanisms involving small waves (in the order of millimetres) in the laboratory experiments are investigated in Section 6.4, namely for mechanisms A (capsizing), C (buoyancy-dominated fall) and E (buoyancy-dominated overturning). As an application of the proposed numerical model, the 2014 Eqip Sermia case (Lüthi and Vieli, 2016) with the real bathymetry was simulated and presented in Section 6.5. Finally, the capabilities of the new numerical model compared to other Computational Fluid Dynamics (CFD) codes for fluid-structure interaction (FSI) is discussed in Section 6.6.

6.2 Comparison of the Contact Force with Other Studies

The contact force $\mathbf{F}_c$ is the horizontal force of the calving iceberg acting on the glacier terminus, which plays an important role in glacier dynamics and mass balance analysis (Sergeant et al., 2018; Tsai et al., 2008). There are several studies focusing on $\mathbf{F}_c$ generated by rotational icebergs (Amundson et al., 2012; Tsai et al., 2008; Sergeant et al., 2016; Sergeant et al., 2018). The calving mechanism in these studies involve so-called "top-out" and "bottom-out" capsizing events. In contrast with the capsizing mechanism in the present study, these calving mechanisms occur at the glacier front. Therefore, the "top-out" and "bottom-out" capsizing mechanisms can be regarded as gravity-dominated and buoyancy-dominated overturning mechanisms without vertical restrictions before these capsizing icebergs drift away from the glacier front.

Since for mechanism B (gravity-dominated fall) no horizontal forces between the iceberg and the basin back wall (or domain boundary) were considered in both the laboratory experiments and numerical simulations in this study, only $\mathbf{F}_c$ in mechanism D (gravity-dominated overturning) is discussed and compared with other studies herein. Although $\mathbf{F}_c$ does not directly appear in Eq. (3.40), $\mathbf{F}_c$ can be obtained based on the force balance of the iceberg along the x-axis direction (Fig. 5.9). The time series of $\mathbf{F}_c$ in mechanism D is shown in Fig. 6.1. The maximum amplitude of the contact force $\mathbf{F}_{c,max} = 302$ N in this study and the ratios of $\mathbf{F}_{c,max}$ to the iceberg gravity force $|\mathbf{G}|$ is $|\mathbf{F}_c|/|\mathbf{G}| = 0.16$.

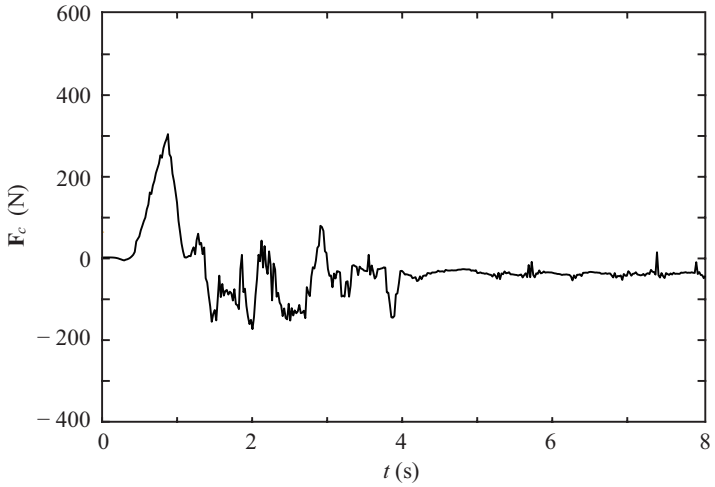


Figure 6.1 Time history of the contact force in mechanism D

For "top-out" and "bottom-out" capsizing (related to mechanisms D and E), Amundson et al. (2012) calculated $\mathbf{F}_c$ based on different models using laboratory experiments and $\mathbf{F}_{c,max} = 0.10$ to 0.47 N was found for an iceberg with dimensions of 0.267 m × 0.100 m × 0.050 m and iceberg density $\rho_s = 920$ kg/m^3. The range of $|\mathbf{F}_c|/|\mathbf{G}|$ is 0.08 to 0.32. Sergeant et al. (2018) numerically reproduced $\mathbf{F}_c$ and obtained $|\mathbf{F}_c|/|\mathbf{G}| = 0.08$ for a "top-out" capsizing case. The ratio $|\mathbf{F}_c|/|\mathbf{G}|$ in the present study is within the range or close to these laboratory experiments and numerical simulations, giving some confidence in the FSI modelling capabilities of the proposed numerical model.

6.3 Effect of Turbulence Modelling

The Reynolds-Averaged Navier-Stokes (RANS) equations based k-ϵ turbulence model is applied in the simulations in this Section 6.3, which is already implemented in the Immersed Boundary Method (IBM) toolbox in Foam-extend 4.0. However, some modifications in the source code of the boundary conditions supporting the immersed boundary (IB) were required. Details about this turbulence model and the necessary modifications in the source code can be found in Appendix D.

The same numerical domain as in the main tests (simulations of mechanisms B and D) is applied here. In contrast to the laminar flow case, three more variables need to be solved in the turbulence correction function (Fig. 3.3), namely the turbulent kinetic energy k, the dissipation of the turbulent kinetic energy ϵ and the turbulent viscosity μ_t. The boundary conditions for each patch of the domain are listed in Table 6.1. The initial values of k and ϵ had to be specified and they were then resolved at each time step. The ranges of k and ϵ in Fig. 6.2 are $[10^{-6}, 0.5]$ m^2/s^2 and $[0.2, 0.8]$ m^2/s^3, respectively. Each of the simulation with the turbulence model required approximately 52 h on the HPC Augusta, which is only 2% longer than the laminar case.

Table 6.1 Boundary conditions for each patch in the domain for a turbulent flow case

Variable	Wall	IB	Atmosphere
u	*fixedValue*	*immersedBoundary*	*pressureInletOutletVelocity*
p	*zeroGradient*	*immersedBoundary*	*totalPressure*
α	*zeroGradient*	*immersedBoundary*	*inletOutlet*
k	*kqRWallFunction*	*immersedBoundary-WallFunction*	*inletOutlet*
ϵ	*epsilonWallFunction*	*iBEpsilonWallFunction*	*inletOutlet*
μ_t	*nutWallFunction*	*immersedBoundary-WallFunction*	*calculated*

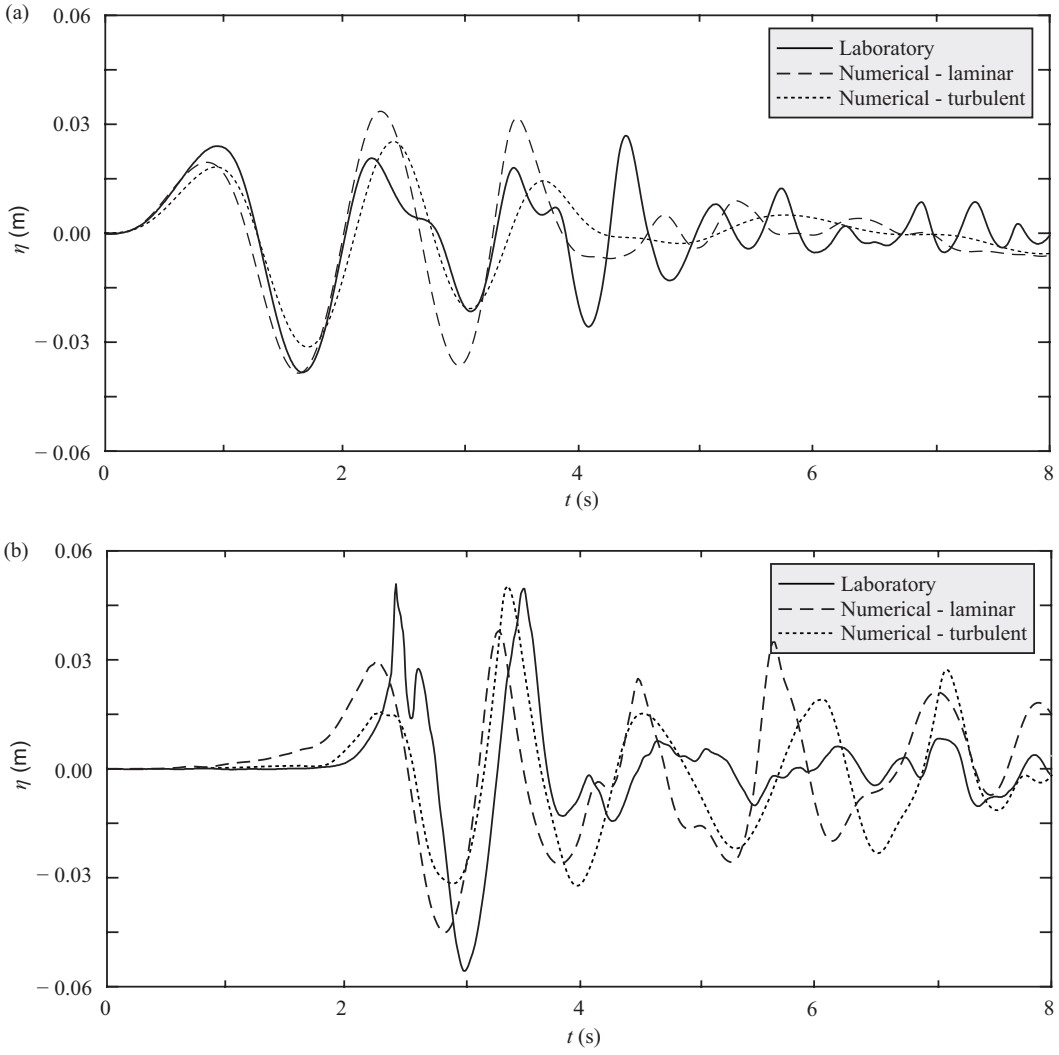


Figure 6.2 Comparison of the wave profiles $\eta(t)$ measured at wave probe B1 in mechanisms (a) B and (b) D based on the laboratory experiments and the laminar and turbulence models

The numerical IBTs involving the turbulence model in mechanisms B and D are shown in Fig. 6.2 with the corresponding laboratory and the laminar results. For IBTs involving turbulence, the first wave amplitude a_1 and height H_1 are 0.0182 and 0.0496 m in the fall case and 0.0155 and 0.472 m in the overturning case, respectively. As expected, the turbulent IBTs are smaller than the laminar ones (Table 5.2). This is because a part of the total released energy ΔE was converted into the turbulent kinetic energy of the IBTs.

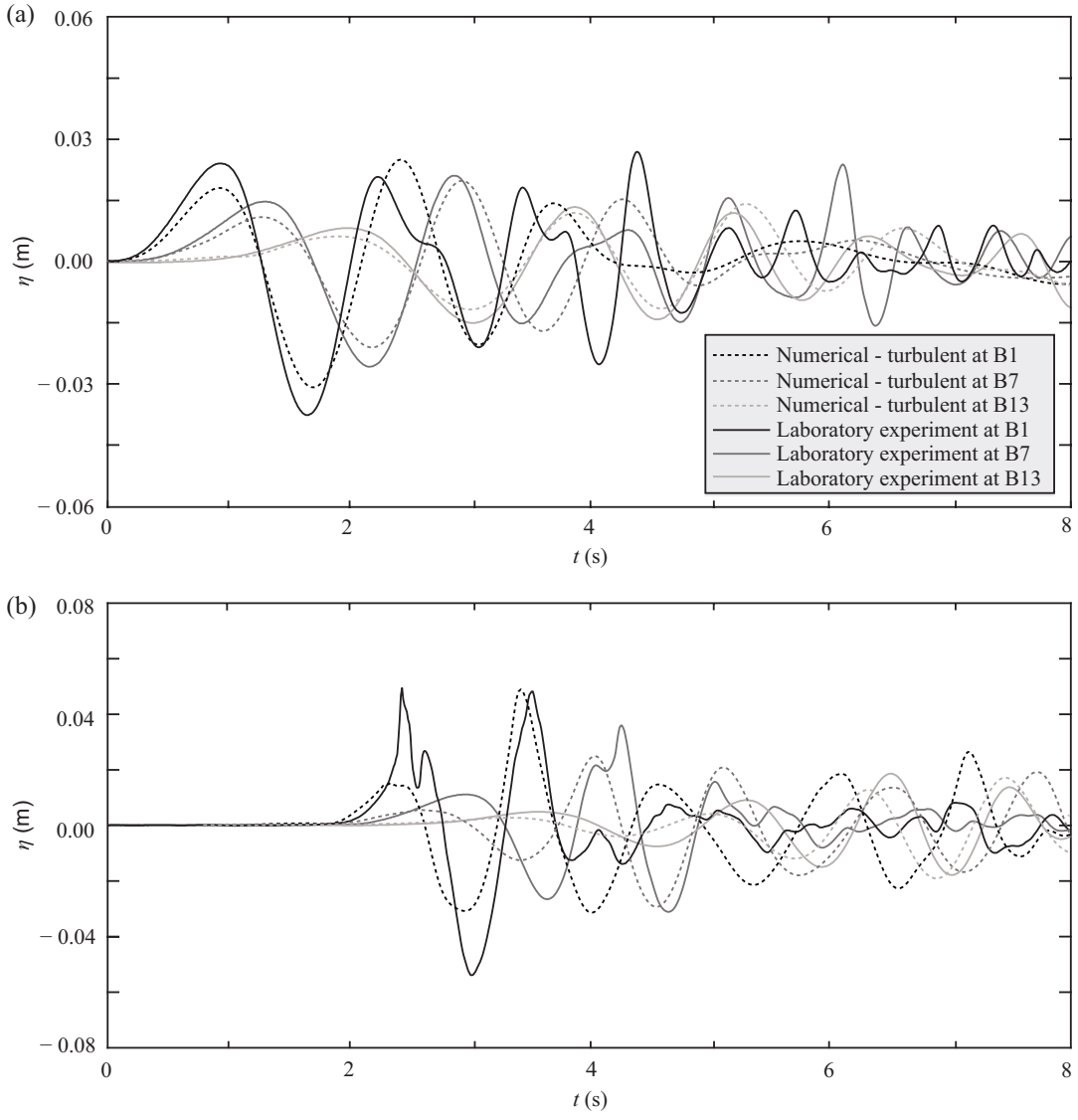


Figure 6.3 Comparison of IBT decays with distance generated in the laboratory experiments and the turbulence model for mechanisms (a) B and (b) D

In the main propagation direction $\gamma = 0°$, the turbulent and laboratory wave profiles $\eta(t)$ measured at wave probes B1, B7 and B13 ($r = 2.0$, 3.0 and 5.0 m, Table 4.4) are shown in Fig. 6.3. In the turbulent cases, the numerical decay exponents c_n for a_1 and H_1 are –1.089 and −1.079 for mechanism B and –2.128 and −2.210 for mechanism D, respectively. In general, the IBTs generated with the turbulence model show a worse fit with the laboratory results than the laminar model. Further, also some computational expenses can be saved in the laminar model, such that the IBTs in the main tests (Sections 5.3.4 and 5.3.5) were simulated without turbulence.

6.4 Simulation of the Remaining Mechanisms

The work in the main part of this thesis focused on the largest and most dangerous IBTs. Here the capabilities of the new model in simulating smaller waves within mechanisms A, C and E (the capsizing, buoyancy-dominated fall and overturning cases) are investigated. In order to reproduce the motions in these cases, restrictions have been implemented in the motion solver. In mechanism A, involving both the heave and pitch motion in the plane (x, $y = 0$, z), only the acceleration ($\mathbf{a}$ and $\boldsymbol{\omega}$) components of interest are considered in the motion solver while other components are imposed as zero at each time step. Similarly, only the components of $\mathbf{a}$ along the z-axis and $\boldsymbol{\omega}$ along the y-axis are considered in mechanisms C and E, respectively.

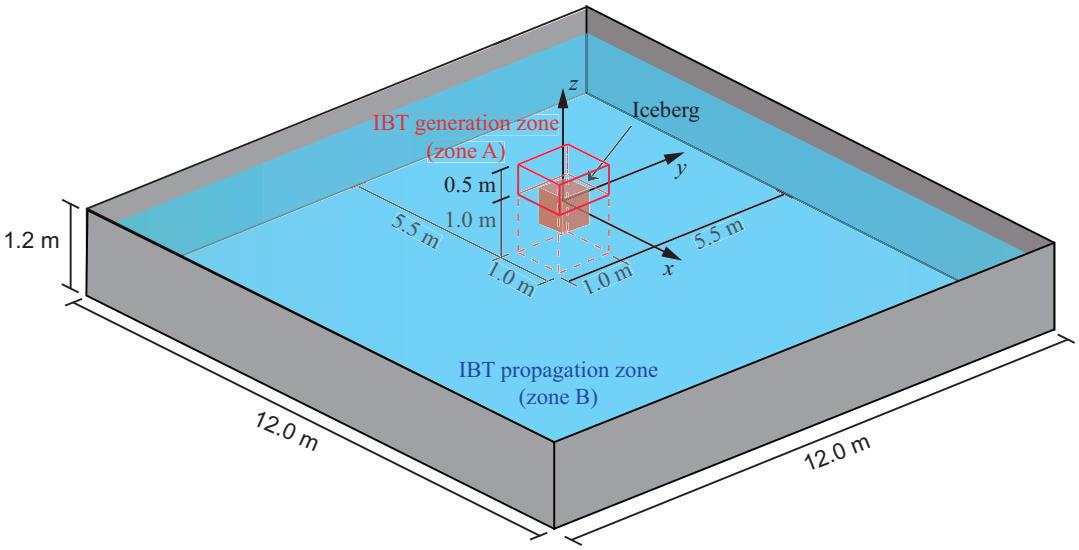


Figure 6.4 Sketch of the numerical basin for mechanism A

The geometries of the numerical basins for mechanisms C and E were the same as for mechanisms B and D (gravity-dominated cases), respectively (Fig. 4.7a). For mechanism A, where the iceberg was located offshore, the numerical domain shown in Fig. 6.4 was adopted. The height of zone A (IBT generation zone) was 1.5 m including 0.5 m above the water surface to allow for iceberg capsizing. In zone B (IBT propagation zone), a 0.2 m thick air layer was also modelled, similar as in the other cases. The same cell dimensions, i.e. 2.5 cm × 2.5 cm × 2.5 cm, were used in all simulations. The simulations of mechanisms A, C and E took 32, 80 and 72 h, respectively, on the HPC Augusta. The key parameters used in these three cases are listed in Table 6.2.

Table 6.2 Key parameters used in the numerical setups for the mechanism A, C and E cases

Calving mechanism	**A**	**C**	**E**
Iceberg length l (m)	0.800	0.500	0.800
Iceberg width b (m)	0.500	0.800	0.500
Iceberg thickness s (m)	0.250	0.500	0.250
Mass m_s (kg)	91.15	187.11	91.15
Centre of mass in initial position (x, y, z) (m)	(0.125, −0.339, 0.000)	(0.125, −0.580, 0.000)	(0.125, −0.500, 0.000)
Centre of rotation (x, y, z) (m)	(0.125, −0.339, 0.000)	NA	(0.125, 0.000, 0.000)
Moment of inertia I (I_x, I_y, I_z) (kg·m^2)	(6.76, 5.37, 2.37)	NA	(21.34, 19.92, 2.37)
Initial angular velocity ω (rad/s)	0.02	NA	0.02
Drag force coefficient C_d	1.0	1.0	0.8
Added mass coefficient C_m	1.2	2.0	1.2
Water depth h (m)	1.00	1.00	1.00
Number of subdomains	144	54	54
Decomposition	12 × 1 × 12	6 × 1 × 9	6 × 1 × 9

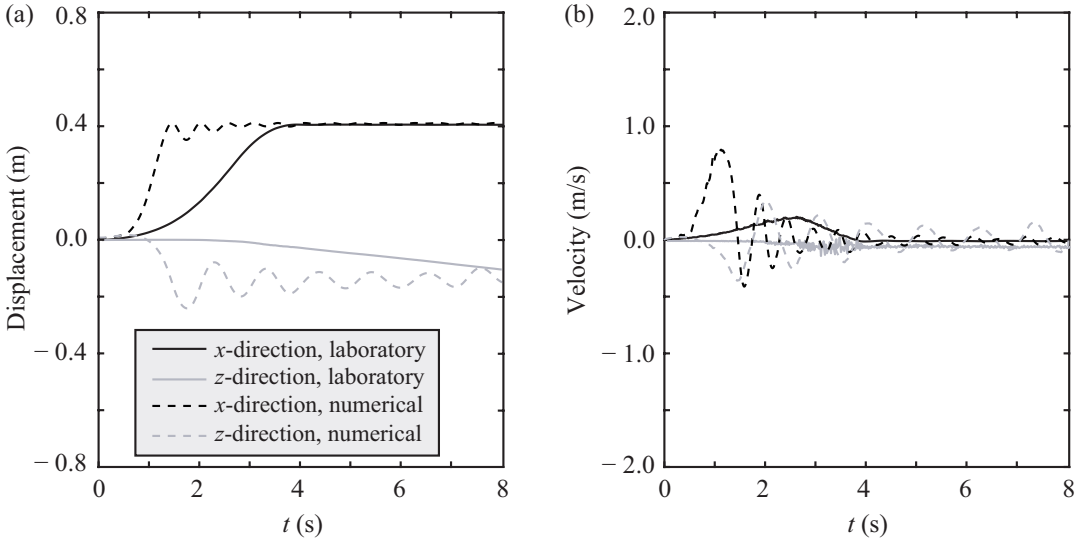


Figure 6.5 Comparison of the (a) displacement and (b) velocity of the iceberg in the laboratory and numerical models involving mechanism A (for the legend see a)

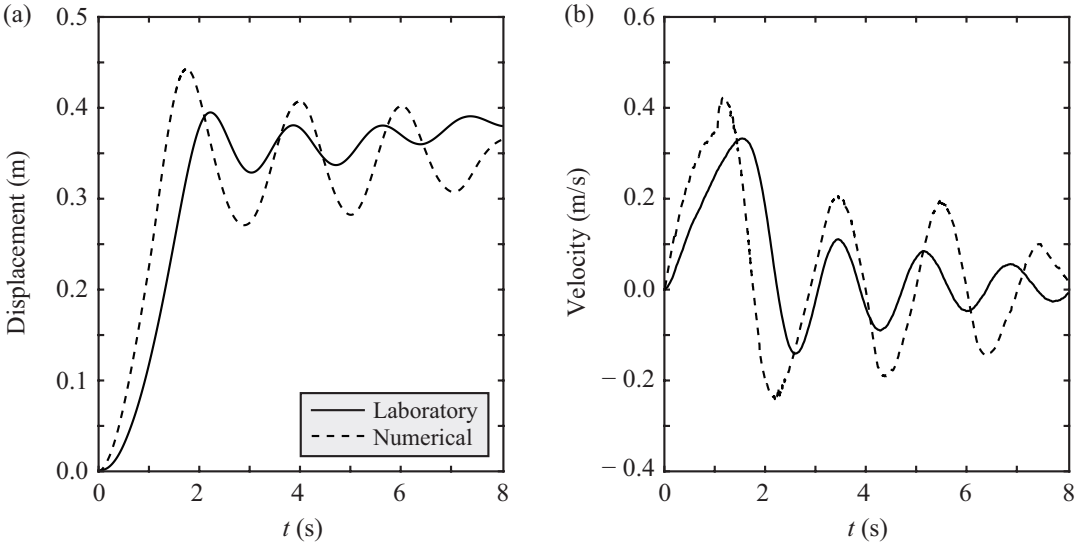


Figure 6.6 Comparison of the (a) displacement and (b) velocity of the iceberg in the laboratory and numerical models involving mechanism C (for the legend see a)

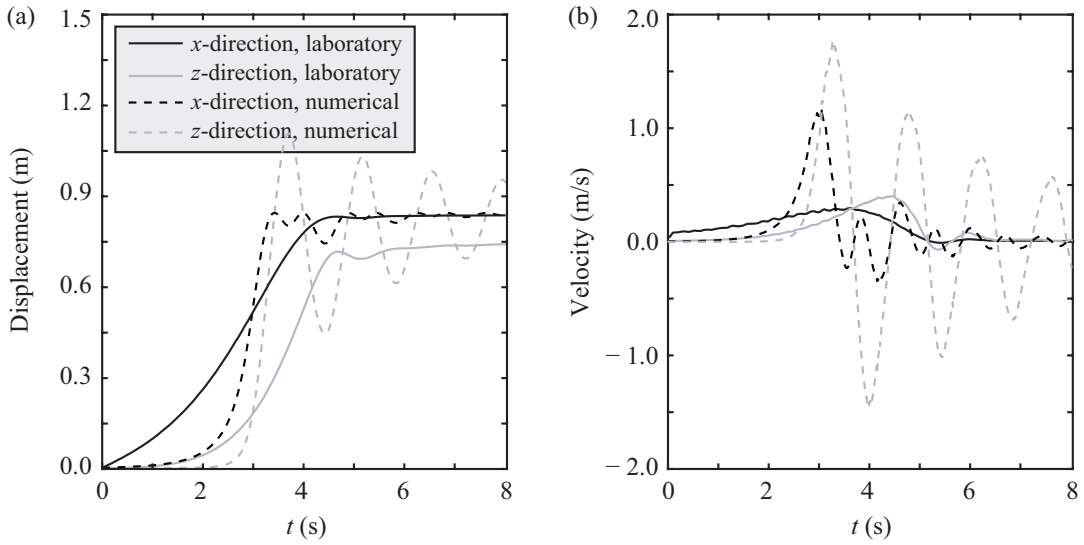


Figure 6.7 Comparison of the (a) displacement and (b) velocity of the iceberg in the laboratory and numerical models involving mechanism E (for the legend see a)

In contrast to mechanisms B and D, the agreements between the numerical simulations and the laboratory tests are unsatisfactory. The numerical reproduced motions are shown in Figs. 6.5, 6.6 and 6.7. All the numerical accelerations are larger than those in the laboratory tests. Especially in mechanisms A and E, the icebergs oscillate around the y-axis faster than in the laboratory tests. The numerical motion agrees best with the laboratory results in mechanism C, however, the numerical oscillation amplitude in the z-direction is still approximately 33.3% larger.

Figures 6.8, 6.9 and 6.10 show the time series of the water surface elevations $\eta(t)$ at wave probe A1 in mechanism A and B1 in mechanisms C and E, respectively (Table 4.4). Compared with the measured waves of the order of millimetres in the laboratory tests, the first numerically reproduced wave amplitudes are a factor of 2.5 to 8.5 larger. Further, the wave fronts arrive at the wave probes earlier in the numerical simulations than in the laboratory tests, particularly in mechanisms A and E where the time discrepancies are 2.85 and 4.65 s, respectively.

There are three reasons for these deviations in the IBTs. Firstly, the fractions of the icebergs underwater account for approximately 92.0% in mechanism A and 100% in mechanisms C and E when the movement is initiated. Since the underwater accelerations (Figs. 6.5, 6.6 and 6.7) were overestimated due to the larger volumes represented by the IBM, the velocities of the icebergs become larger. This may result in more energy transferred into the waves. Secondly, when the iceberg interacts with the multiphase flows (air and water), some disturbances around the IB cells located at the interface with water or air are created. This is due to the current implementation of this IBM in the toolbox, where it is challenging to balance the gradients of the density and pressure of IB cells to fully satisfy the continuity for multiphase flows around the IB (Constant et al., 2017). Although this effect may not significantly affect large IBTs (in the order of centimetres) generated in mechanisms B and D, it plays a more significant role for simulating waves in the order of millimetres. Lastly, the iceberg boundary represented in the IBM is not exactly identical to its original geometry, which may cause some further disturbances affecting the accuracy of the generated IBTs in mechanisms A, C and E. Suggestions to overcome these issues in future work are given in Section 7.2.

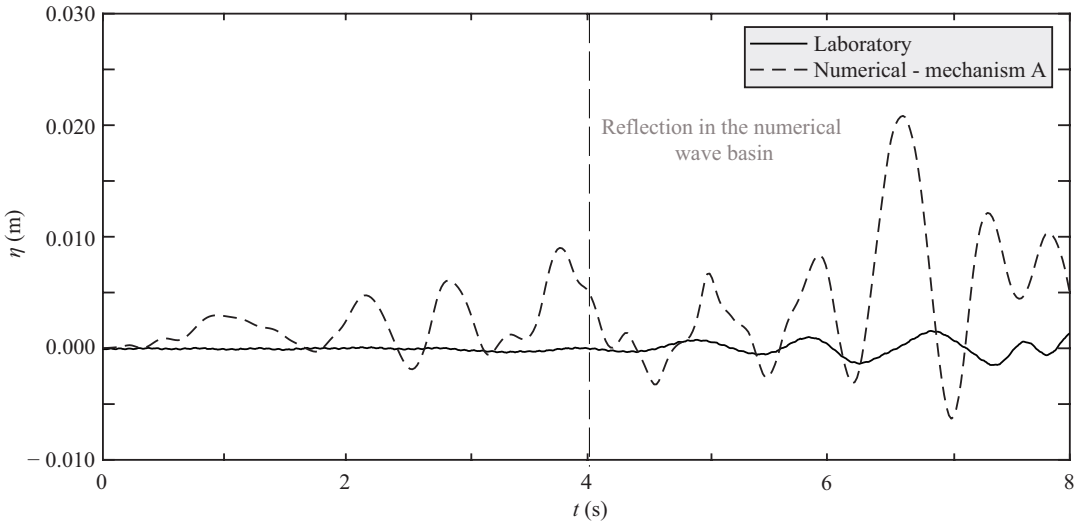


Figure 6.8 Time series of water surface elevation $\eta(t)$ at wave probe A1 in mechanism A

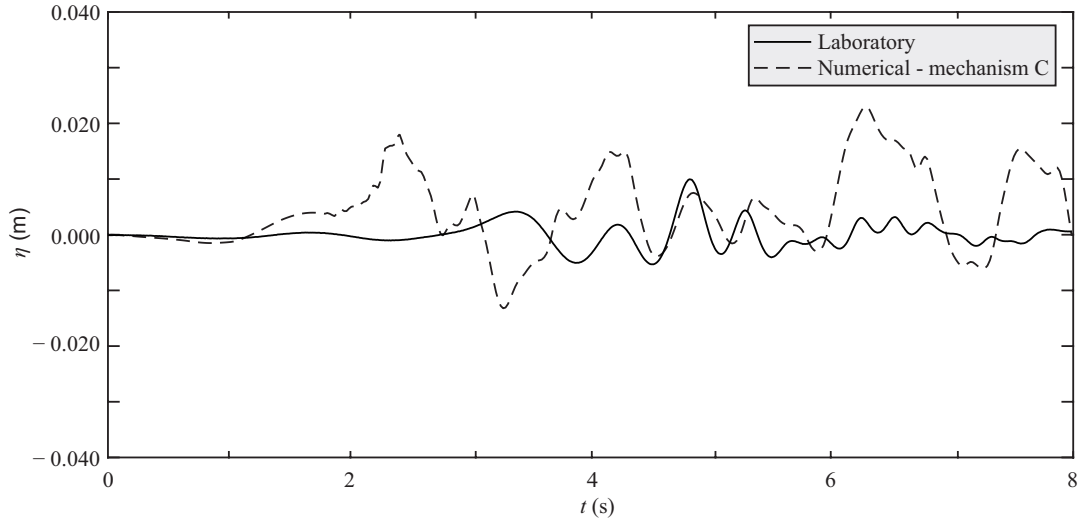


Figure 6.9 Time series of water surface elevation $\eta(t)$ at wave probe B1 in mechanism C

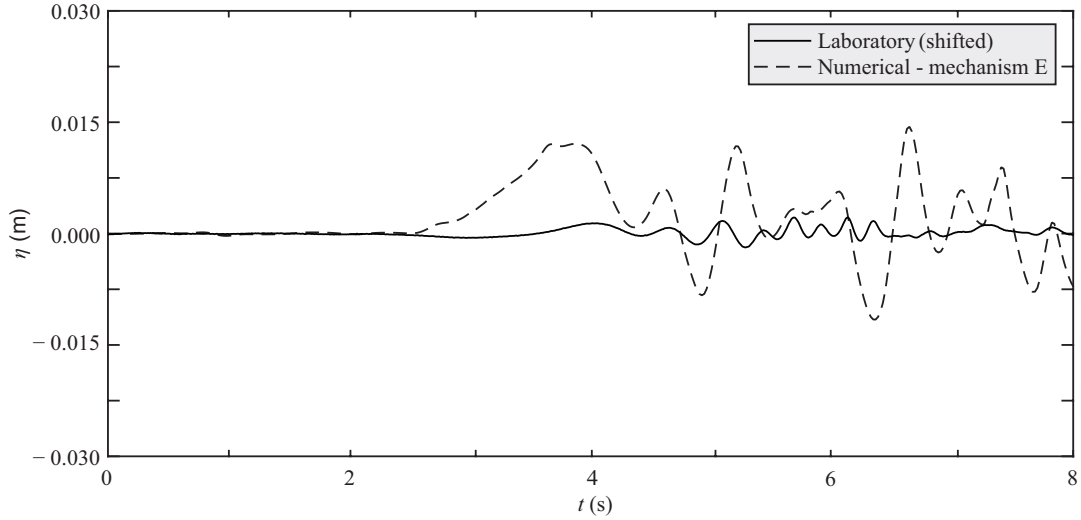


Figure 6.10 Time series of water surface elevation $\eta(t)$ at wave probe B1 in mechanism E

6.5 Simulation of the 2014 Eqip Sermia Case

The wave amplitude in the 2014 Eqip Sermia case reached 45 to 50 m in proximity of the glacier terminus (Lüthi and Vieli, 2016). This IBT is reproduced with the new numerical approach in this Section 6.5.

6.5.1 Numerical setup

The bathymetric data of Eqip Sermia is available from the GEneral Bathymetric Chart of the Oceans (GEBCO) website (GEBCO, 2019). The raw data includes the longitude, latitude and elevation of each point in the grid. After converting the raw data in longitudes and latitudes into a Cartesian coordinate system in x and y representations (Fig. 6.11), the dimensions of each cell were approximately 160 m $\times$ 460 m. In order to obtain a higher resolution of 5 m $\times$ 5 m, a bilinear interpolation was applied. The interpolated elevation z_{ij} is given by

$$z_{\mathrm{ij}} = (1-\frac{\Delta x_{\mathrm{i}}}{\Delta x})(1-\frac{\Delta y_{\mathrm{j}}}{\Delta y})z_{LL}+\frac{\Delta x_{\mathrm{i}}}{\Delta x}(1-\frac{\Delta y_{\mathrm{j}}}{\Delta y})z_{LR}+(1-\frac{\Delta x_{\mathrm{i}}}{\Delta x})\frac{\Delta y_{\mathrm{j}}}{\Delta y}z_{UL}+\frac{\Delta x_{\mathrm{i}}}{\Delta x}\frac{\Delta y_{\mathrm{j}}}{\Delta y}z_{UR}. \tag{6.1}$$

In Eq. (6.1), the subscript ij denotes the index of the interpolated point location in the original cell and UL, UR, LL and LR the upper-left, upper-right, lower-left and lower-right points of a cell (Fig. 6.11). Figure 6.11 shows the relationship between the distances Δx, Δy, Δx_{i} and Δy_{j}. The reproduced outline of the water body geometry at the Eqip Sermia glacier based on the processed bathymetry data compared with the satellite image from Lüthi and Vieli (2016) are shown in Fig. 6.12.

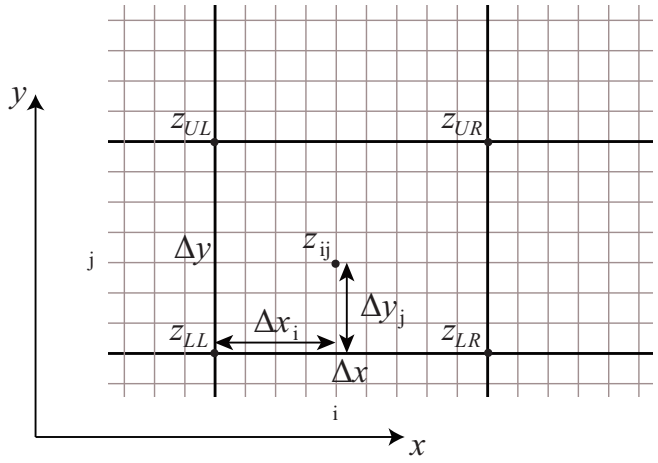


Figure 6.11 Illustration of the bilinear interpolation applied in the bathymetric data

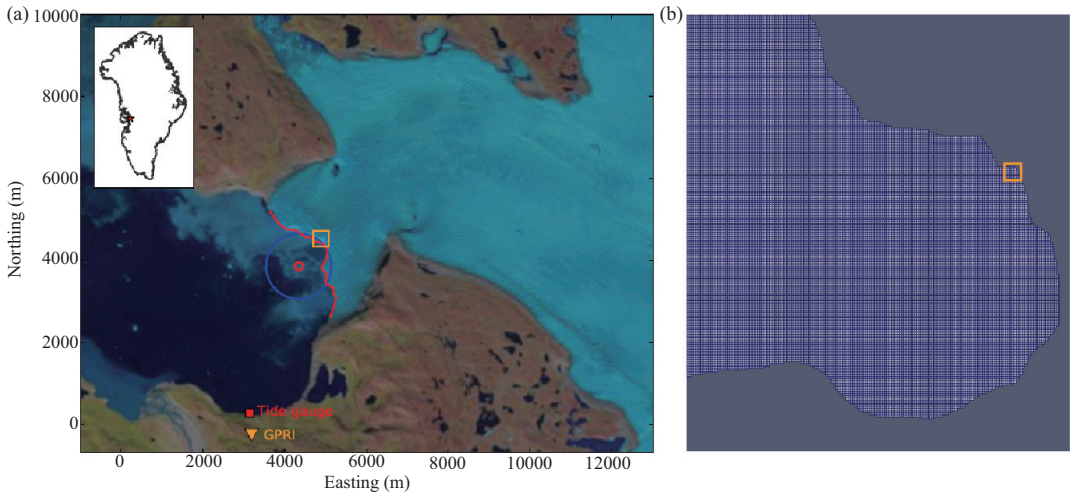


Figure 6.12 Comparison of the outlines of the water body geometry at the Eqip Sermia glacier based on (a) the satellite image (adapted from Lüthi and Vieli, 2016) and (b) the processed bathymetry data. The red lines mark the glacier front and the orange squares denote the calving location

A structured mesh was generated based on the interpolated bathymetric data. A geometric length scale of 1:100 was applied in the mesh in order to be consistent with the resolution selected in the convergence tests and the main IBT tests. The following dimensions and results are all presented at this scale. Figure 6.13 shows the numerical domain for this real case simulation. The origin is located at the corner of the domain with the z-axis pointing upwards and $z = 0.0$ m corresponding to the sea level. The x- and y-axes are along the local latitudinal and longitudinal directions, respectively. The dimension of the domain is 15.0 m $\times$ 15.0 m with heights varying from 2.45 to 3.15 m (Fig. 6.13). A resolution of 0.05 m in each direction was selected. The topography of the glacier terminus and the profile of the calving iceberg were obtained based on Fig. 3 in Lüthi and Vieli (2016). The comparison of the iceberg profiles measured in the real event and represented by the IBM (marked in brown) is shown in Fig. 6.14. The volume of the numerical represented iceberg is $\forall_s = 0.9$ m^3, which is identical to the down-scaled estimated volume in the real case.

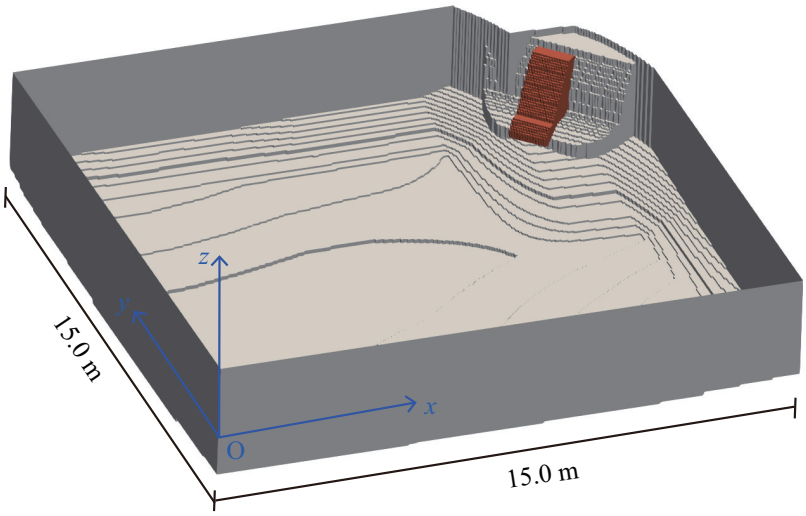


Figure 6.13 Computation domain for the Eqip Sermia case with the calving iceberg represented by the IBM (Chen et al., 2020)

The iceberg motion was performed in the plane ($x = y$, z). Given that the topography of the glacier terminus, the calving iceberg geometry and the water depth $h = 25$ to 45 m in the impact zone, the iceberg motion was assumed to be a combined translation and rotation, and this movement was most likely also observed in nature. Thus, the iceberg moved in 3 degree of freedoms (DoFs). Similar to the main tests, some restrictions were implemented to ensure the assumed movement. The iceberg motion was resolved in this simulation. Since the IBM requires a gap of two-cell width between the iceberg and the domain boundary (Section 5.3.3), a trajectory of the iceberg (including the centre of gravity and the rotation angles) was imposed to ensure that the iceberg stayed clear from the boundary. Note that this gap may affect the generated IBT as discussed in Section 6.5.2. Further, the impact velocity V_s was imposed with a similar observed value as estimated by Lüthi and Vieli (2016). In this simulation, $V_s = 48.2$ m/s was selected after investigating a range of values, which is slightly larger than the value of 42 m/s estimated by Lüthi and Vieli (2016). The numerical simulation was run on the HPC Augusta using 4 cores and 64 GB, requiring 38 h for modelling 5.0 s of real time.

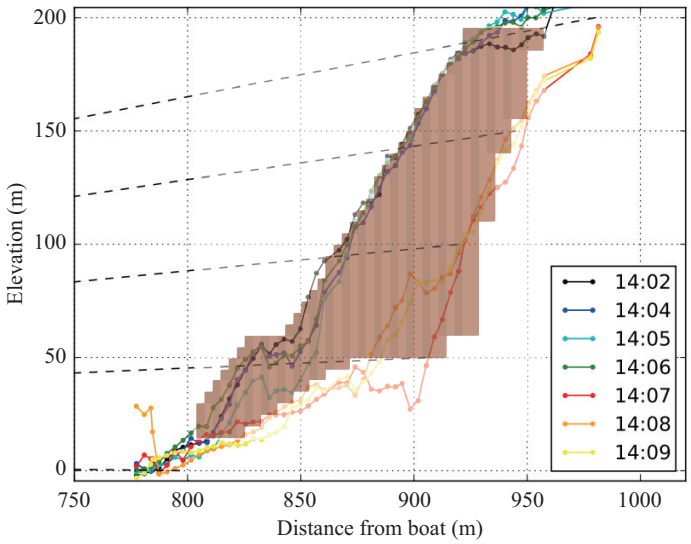


Figure 6.14 Comparison of the measured iceberg geometry (adapted from Lüthi and Vieli, 2016) and the iceberg represented by the IBM

6.5.2 Results

The snapshot series of the numerically reproduced IBT in the domain are shown in Fig. 6.15. The iceberg begins to slide along the glacier terminus and interacts with the water body at $t = 0.4$ s. At $t = 0.8$ s, the iceberg continues to submerge and rotate. Some waves are generated around the iceberg (Fig. 6.15c). The IBT continues to grow at $t = 1.2$ s, and the iceberg rotates towards the water surface (Fig. 6.15d). More waves can be observed at $t = 1.6$ s, especially in the area between the iceberg and the glacier front. At $t = 2.0$ s, some waves run up on the glacier terminus and the iceberg floats backward (Fig. 6.15f). It is noteworthy that no bore or splash is observed in contrast to the real event (Fig. 1.3b). This is because of the selected cell dimensions and the applied laminar model.

A number of wave probes were placed in the impact zone for the identification of the maximum IBT. This maximum was measured at 3.80 m from the glacier front along the direction of the moving iceberg. Compared with the down-scaled observed maximum wave amplitude of 45 to 50 m, the first wave amplitude $a_1 = 0.404$ m is 10.2 to 19.3% smaller. This underestimation may be caused by the aforementioned gap between the glacier terminus and the iceberg, resulting in a decreased generation efficiency for the leading wave.

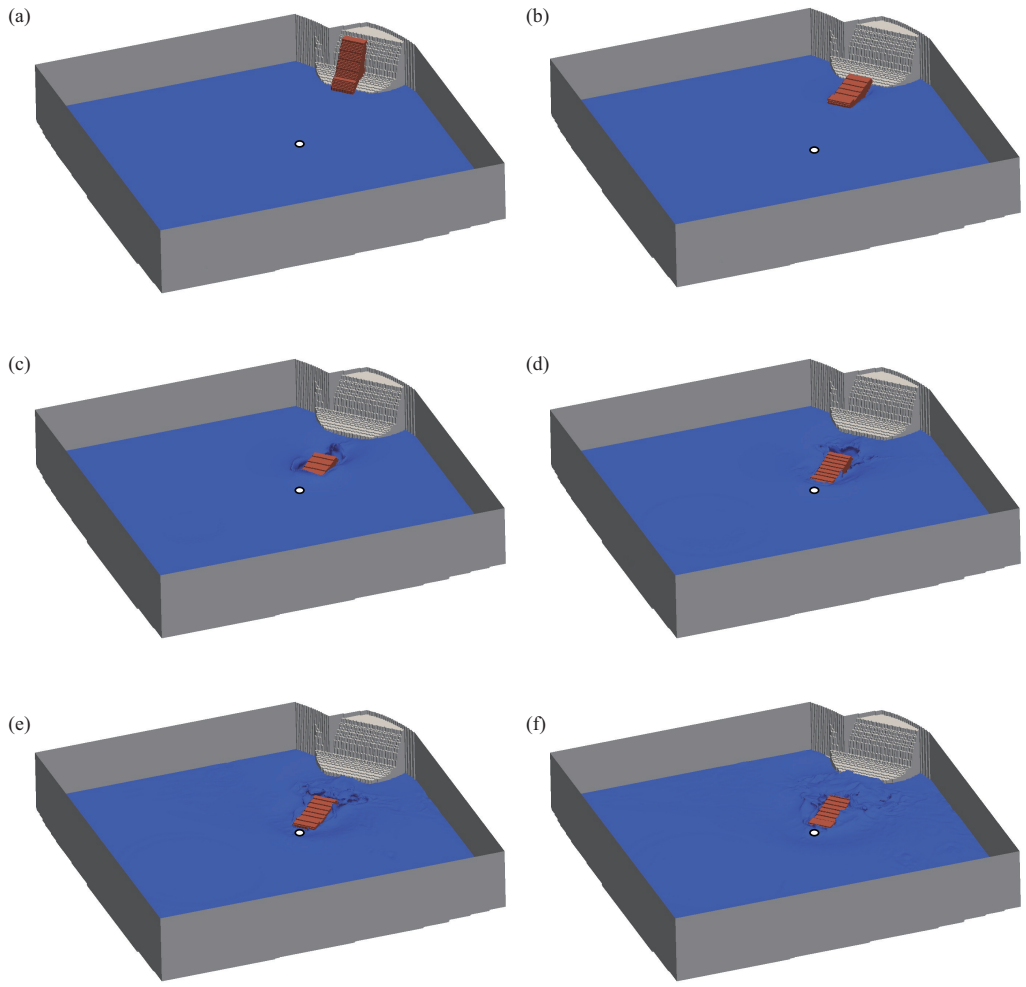


Figure 6.15 Snapshots of the reproduced IBTs in the Eqip Sermia case at a scale 1:100 at (a) $t = 0.0$ s, (b) $t = 0.4$ s, (c) $t = 0.8$ s, (d) $t = 1.2$ s, (e) $t = 1.6$ s and (f) $t = 2.0$ s. The white circle is the wave probe located at 3.80 m from the glacier front (Chen et al., 2020)

It is difficult to estimate the effect of this gap on the IBT. Nevertheless, the phenomenon observed in this simulation is closely related to the one in Heller and Spinneken (2013). In both cases, a rigid mass impacts into a water body and generates a wave. During the wave generation process, the part of the water in the gap between the mass and the boundary remains passive. In Heller and Spinneken (2013), a gap of 12% between a solid slide heavier than water and the side wall in a flume was found to decrease the wave height by a similar percentage. If applying this finding in the present simulation, where the gap is approximately 20% of the iceberg thickness, the wave amplitude may be reduced by ≈20%. $a_1 = 0.404$ m is therefore extrapolated as $1/(1-0.2) \times a_1$

= 0.505 m, which would reach the upper bound of the observed range. On the other hand, the iceberg was regarded as a rigid body in this simulation whilst in nature the iceberg most likely broke into different pieces. A rigid slide may result in a larger wave amplitude compared to a granular slide for masses lighter than water (Heller and Spinneken, 2013). Overall, it may be concluded that this model reasonably well reproduces the 2014 Eqip Sermia IBT, despite of some uncertainties.

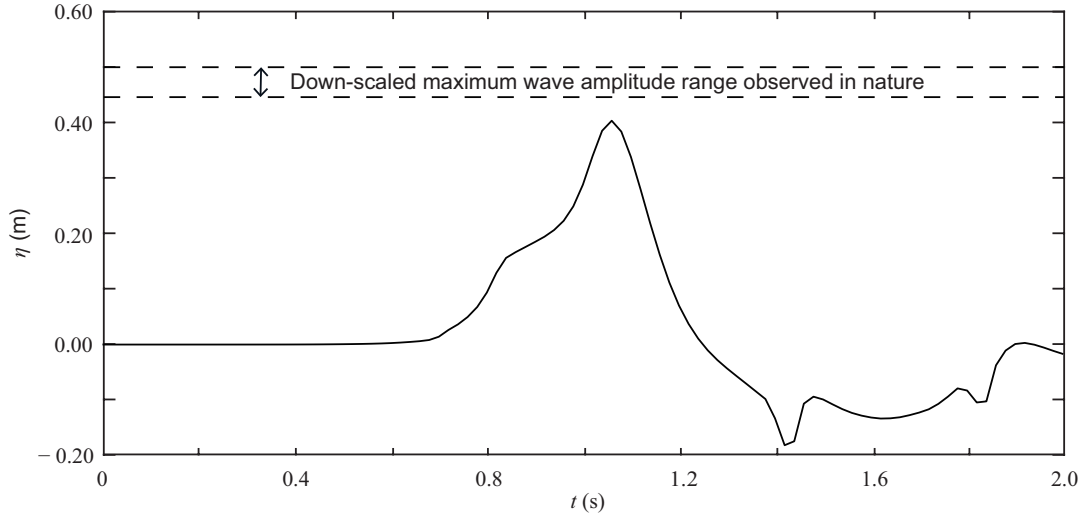


Figure 6.16 Reproduced IBT at scale 1:100 measured 3.80 m from the glacier terminus

6.6 Comparison with Other CFD Codes

In this Section 6.6, the capabilities of the new numerical code are compared with the ones from different CFD codes focusing on numerical modelling and the potential effects of IBTs are discussed. The presented numerical model is established based on the open source code Foam-extend 4.0 and the IBM toolbox developed by Jasak et al. (2014). Open source codes such as Foam-extend enable the users to tailor the code to fit the specific needs in contrast to commercial software where the source code can normally not be modified. Widely used commercial software such as ANSYS Fluent and FLOW-3D have already integrated 6-DoF solvers and remeshing methods that allow for large displacements within the computational domain, e.g. dynamic mesh (unstructured remeshing) in Fluent (ANSYS Fluent User's Guide, 2013) and Fractional Area-Volume Obstacle Representation (FAVOR) in FLOW-3D (Flow Science, 2007). The meshes

in these methods can be well fitted to the rigid body geometry (body-fitted mesh). However, they both involve unstructured mesh handling requiring much more computational expenses and data storages to generate the mesh than a structured one. Note for simulations of the same case using different CFD codes, the larger computation expenses of unstructured mesh based methods may be compensated by varying parallel computing capabilities. In contrast, in the numerical model of the present study only a simple orthogonal Euler mesh is applied. Although the represented iceberg does not fully fit the original shape, this good quality mesh enables a stable and efficient simulation (Constant et al., 2017).

In addition to Foam-extend, there are a range of open source CFD codes used for FSI simulations such as DualSPHysics (Altomare et al., 2015; Tan et al., 2018; Violeau and Rogers, 2016), OpenFOAM (Romano et al., 2020) and REEF3D (Bihs et al., 2015). All of these open source codes can simulate multiphase flows and have 6-DoF solvers for solving the rigid body motion. For DualSPHysics, based on Smoothed Particle Hydrodynamics (SPH), it is often easy to simulate a large displacement of a rigid body due to the nature of Lagrangian approaches. IBT generation in the near field may be well simulated. However, for the IBT propagation, DualSPHysics is prone to overestimate wave decay due to numerical energy dissipation (Violeau and Rogers, 2016). This is a main reason why, DualSPHysics is commonly coupled with other codes suited for wave propagation (Altomare et al., 2015; Narayanaswamy et al., 2010; Tan et al., 2018; Viroulet et al., 2013) making the simulation process more work intensive. In contrast, the proposed numerical model can reasonably well reproduce both IBT generation and propagation.

For OpenFOAM, the mesh handling methods include the mesh motion, topological changes and the Overset Grids Methods (OGM) for version v1706 (the OpenFOAM variant by OpenCFD Ltd) and later versions. The limitations of mesh motion methods have been demonstrated in Section 2.5.2.1, where only small displacements of a rigid body can be simulated. Although topological changes can be applied for the simulation of capsizing cases using sliding interfaces, however, this approach currently is not able to simulate all 5 iceberg calving mechanisms due to extra mesh requirements (Fig. 2.24). The OGM is an alternative option for modelling FSI including the body-fitted mesh. Romano et al. (2020) proposed a numerical model to simulate submarine landslide-tsunamis

based on the OGM. Similar as in the IBM applied in the present work, the landslide mesh in the OGM can not directly be placed at the boundary of the background mesh. Romano et al. (2020) overcame this issue by introducing a structure with sufficient small porosity placed beneath the landslide such that the landslide mesh can be placed in the inner part of the background mesh. However, introducing a porous structure in this model increases the domain size and results in extra computational resources.

For REEF3D, Bihs et al. (2015) simulated a floating ice block in waves, where FSI was simulated by the curvilinear immersed boundary method and the rigid body with complex geometry was represented by a fine triangular mesh. The representation process may preserve the original shape of the body but involves an additional interface tracing algorithm at each time step. The simulation of the ice block in waves demonstrates its ability to simulate the capsizing case. However, placing the rigid body close to the domain boundary to simulate the remaining calving mechanisms would also be associated with the gap issue faced in the present work.

Table 6.3 summarises the comparison of the described CFD codes when applied to model IBTs. The strength of the proposed numerical model, besides that it is open source, is that it can simulate both IBT generation and propagation involving all 5 different calving mechanisms without requiring extra domain sizes or the coupling with other CFD codes. Although the represented iceberg is slightly different from its original shape in some cases, the good mesh quality and simple handling of moving boundaries ensure a good computational efficiency.

Table 6.3 Summary of comparison of different CFD codes for modelling IBTs

	Open source	**Mesh**	**Rigid body**
Proposed numerical model	Yes	Structured	Represented
ANSYS Fluent	No	Unstructured	Body-fitted
DualSPHysics	Yes	Meshless	Represented
FLOW-3D	No	Unstructured	Body-fitted
OpenFOAM	Yes	Structured	Body-fitted
REEF3D	Yes	Structured	Represented

6.7 Summary

In this chapter the results were discussed and the capability of the new numerical model were demonstrated. The contact force of the iceberg was compared with the forces of other studies, demonstrating, in general, a good agreement and the capability of the new numerical model in simulating fluid-structure interaction. Then the effect of turbulence based on the k-ϵ model on iceberg-tsunami (IBT) propagation was investigated. This resulted in a worse fit to the laboratory IBTs than the laminar model, as parts of the total released energy of the iceberg was dissipated by turbulence. Further, smaller and less important IBTs, next to the larger ones from mechanisms B and D, which were the main focus of this work, were also simulated. This involved the calving mechanisms A, C and E (capsizing, buoyancy-dominated fall and overturning). However, for the generated waves in the order of millimeters at laboratory scale, some disturbances around the immersed boundary were observed affecting the magnitude and shape of the simulated IBTs. This is due to the nature of the IBM making it challenging to fully satisfy the continuity equation and the no-slip boundary condition simultaneously when involving multiphase flows. Fortunately, the proposed numerical model worked well for the most hazardous waves.

Meanwhile, the 2014 Eqip Sermia case was simulated in Section 6.5 as a real case application. The well reproduced IBT amplitude demonstrates the potential of the proposed numerical model of predicting IBTs in nature. Finally, the features of the new code were compared with the ones from alternative Computational Fluid Dynamics codes in Section 6.6. The good mesh quality while handling a moving rigid body and the versatility in simulating both the IBT generation and propagation have been identified as particular strengths of the new code. The gap between the iceberg and the domain and the disturbances around the immersed boundary are disadvantages of the model and require further work.

Chapter 7

CONCLUSIONS AND OUTLOOK

7.1 Conclusions

This study presented a novel numerical methodology to model large tsunamis generated by iceberg calving, so-called iceberg-tsunamis (IBTs). Iceberg calving is not only a main contributor to ice mass loss and sea level rise, but also generated waves of up to 45 to 50 m in amplitude (e.g. in Eqip Sermia in 2014), endangering the coastal communities. However, there are relatively few studies about IBTs.

The proposed numerical model was implemented in Foam-extend 4.0 based on the toolbox of the Immersed Boundary Method (IBM). In order to simulate both the generation and propagation of IBTs, the new multiphase flow solver *interDyMIbFoam* with the support of moving immersed boundaries was implemented to solve the Reynolds-Averaged Navier-Stokes (RANS) equations. *interDyMIbFoam* was then coupled with a tailored motion solver, which is able to obtain the solution of the iceberg motion involving a wide range of iceberg calving mechanisms. The coupled two solvers were then used to model Fluid-Structure Interaction (FSI) including calculations of the pressure force, viscosity force, drag force and virtual force caused by the added mass.

Moreover, large-scale laboratory experiments under Froude similarity were conducted in a 50 m $\times$ 50 m wave basin to investigate IBTs. The icebergs were modelled with blocks with a density of approximately 920 kg/m^3, block dimensions of up to 0.800 m $\times$ 0.500 m $\times$ 0.500 m and a still water depth of 1.0 m. The blocks were released in different ways to mimic the five idealised iceberg calving mechanisms, namely (A) capsizing, (B) gravity-dominated fall, (C) buoyancy-dominated fall, (D) gravity-dominated overturning and (E) buoyancy-dominated overturning. The motions of the blocks were recorded by a motion sensor and the generated waves were measured with up to 35 wave probes at distances up to 35 times the water depth from the calving location in different

directions.

The numerical model has been validated with an analytical solution of radiated waves from a floating heaving sphere in still water using prescribed motion, with less than 10.3% deviations in both the wave amplitudes and wave periods. As the main aim of this study is to simulate the largest and most dangerous IBTs to support IBT hazard assessment, one test of the two mechanisms B and D generating the largest waves were selected for validation. These two main tests were simulated with resolved iceberg motion.

The results illustrate that the application of a laminar flow model in the wave simulations leads to better accuracy and agreements with the corresponding laboratory measurements outside the splash zone. Further, the laminar model is less computational expensive than simulations involving a k-ϵ turbulence model. For IBT generation, the numerically reproduced iceberg motion is always faster than the laboratory results due to the IBM overestimating the volume of the submerged part of the iceberg. The deviations of the maximum vertical displacement in mechanism B and the pitch angle in mechanism D are 24.3 and 19.9%, respectively. The results further show that the pressure force acting on the iceberg is the dominating force after the initiation of the motion and the viscosity force can be neglected on the overall force balance. The magnitude of the maximum pressure force is up to 1.71 times the iceberg gravity force. Further, the contact force, which is the horizontal force of the iceberg acting on the glacier terminus, was calculated and generally agreed with other studies. This demonstrated the capability of the proposed numerical model in simulating FSI.

For IBT propagation, the IBT parameters, wave train energy and wave decay were investigated. The numerical model underestimates the first (leading) laboratory IBT amplitude by up to 17.2% (mechanism B) and 41.2% (mechanism D), mainly due to the splash. If this splash is artificially excluded by applying an empirical equation, the deviation reduces to 1.0% in mechanism D. The underestimations in the first wave amplitude and height are likely due to the aforementioned overestimated volume by the IBM, making the iceberg move too fast towards the water surface and prohibiting the growth of waves.

The comparison between the numerical and experimental IBT profiles measured in different directions and distances shows the ability of the numerical

model to simulate frequency dispersion. Additionally, the numerical IBT train energy agrees well with the laboratory results in mechanism B, while it shows a large discrepancy in mechanism D due to the oscillation of the iceberg. For the wave decay, the exponents of the numerical wave height decay power functions are 9.1% larger and 9.2% smaller in mechanisms B and D, respectively, if the effect of the splash is removed for the latter. Although this numerical model is, in principle, capable of simulating all 5 iceberg calving mechanisms, the smaller and less important IBTs in mechanisms A, C and E, in the order of millimetres at laboratory scale, cannot be modelled well. This is due to the nature of the IBM, making it challenging to satisfy the continuity equation and the no-slip boundary condition simultaneously at the immersed boundary when multiphase flows are involved. Nevertheless, the proposed numerical model can simulate the most hazardous IBTs well.

Finally, the numerical model was applied to the 2014 Eqip Sermia IBT event, where a wave amplitude of 45 to 50 m was observed. The real bathymetry and initial iceberg geometry were used. The trajectory of the iceberg was prescribed and the iceberg impact velocity was imposed with a similar value as in nature. The maximum IBT amplitude in the simulations was 0.404 m with a deviation of 10.2 to 19.3% to the field observation. This difference was due to the gap between the iceberg and the domain, which was required by the IBM for immersed boundary evaluations. The good replication of the real event shows further the potential of this model to be used for IBT predictions.

7.2 Limitations and Outlook

The focus of this study was on numerical modelling of IBTs and the proposed technique has demonstrated the capabilities to simulate both IBT generation and propagation. However, there exists some space for improvements.

First of all, there were only five idealised calving mechanisms simulated, additional calving scenarios could be investigated such as iceberg "top-out" and "bottom-out" mechanisms (Section 6.2). Combined iceberg calving mechanisms should also be investigated, e.g. a combination of a fall and overturning mechanism as often observed in nature. These additional iceberg motions could be achieved by further modifying the motion solver to guarantee the corresponding

degrees of freedom.

Secondly, since FSI can be well simulated by this numerical model, as demonstrated by the time history of forces acting on the iceberg (Section 5.3.4.2), the numerical model could be used to simulate glacier dynamics including contact forces. However, this requires more work, e.g. introducing the support of multiple moving immersed boundaries. Further, the representation algorithm of the immersed boundary may also play an important role in the accuracy of the force calculation.

Thirdly, the gap between the basin back wall and iceberg of mechanism D affects IBT generation. This is because the current toolbox of IBM requires such a gap to recognise and evaluate the immersed boundary. This is an issue related to the immersed boundary representation technique, which is actually also the reason of the overestimated iceberg volume by the IBM. The recognition and representation of the immersed boundary could be improved by considering more geometry information at the cells cut by the rigid body (e.g. identifying more immersed boundary points in the background mesh). A more robust immersed boundary representation may also be beneficial to decrease disturbances for multiphase flows, enabling the simulation of much smaller waves.

Lastly, the proposed numerical technique is also useful for many related applications, e.g. for floating body dynamics. By coupling the new model with a wave generation module, the model could be used to simulate floating breakwaters, wave energy converters and vessels in waves.

For IBT research in general, more laboratory experiments should be conducted involving further calving mechanisms and with different iceberg geometries. Since icebergs typically decompose during the calving process, multiple icebergs could also be considered in laboratory tests. Additionally, the water body geometry (e.g. 2D in fjord cases) and the corresponding bathymetry could also be varied to gain a better understanding of IBT propagation. Future research may also focus on theoretical aspects of IBTs, e.g. the maximum wave amplitude or the dominant IBT frequency under some idealised calving mechanisms, however, this requires advanced techniques to solve partial differential equations and the governing equations for IBTs may not readily be available.

References

Abadie, S.M., Harris, J.C., Grilli, S.T. and Fabre, R. (2012). Numerical modeling of tsunami waves generated by the flank collapse of the Cumbre Vieja Volcano (La Palma, Canary Islands): tsunami source and near field effects. *Journal of Geophysical Research: Oceans*, 117(C05030).

Abadie, S., Morichon, D., Grilli, S. and Glockner, S. (2010). Numerical simulation of waves generated by landslides using a multiple-fluid Navier–Stokes model. *Coastal engineering*, 57(9), 779-794.

Altomare, C., Domínguez, J.M., Crespo, A.J.C., Suzuki, T., Caceres, I. and Gómez-Gesteira, M. (2015). Hybridization of the wave propagation model SWASH and the meshfree particle method SPH for real coastal applications. *Coastal Engineering Journal*, 57(4), 1550024-1.

Amundson, J.M., Burton, J.C. and Correa-Legisos, S. (2012). Impact of hydrodynamics on seismic signals generated by iceberg collisions. *Annals of Glaciology*, 53(60), 106-112.

ANSYS Fluent User's Guide (2013). Online under http://www.pmt.usp.br/academic/martoran/notasmodelosgrad/ANSYS%20Fluent%20Users%20Guide.pdf (accessed on 18 January 2021).

Antoci, C., Gallati, M. and Sibilla, S. (2007). Numerical simulation of fluid–structure interaction by SPH. Computers & Structures, 85(11-14), 879-890.

Ataie-Ashtiani, B. and Nik-Khah, A. (2008). Impulsive waves caused by subaerial landslides. *Environmental Fluid Mechanics*, 8(3), 263-280.

Attili, T. (2019). Analysis of iceberg-tsunamis from large-scale experiments. Master thesis, University of Pisa, Pisa.

Babuska, I., Tempone, R. and Zouraris, G.E. (2004). Galerkin finite element approximations of stochastic elliptic partial differential equations. *SIAM Journal on Numerical Analysis*, 42(2), 800-825.

Bassis, J.N. and Jacobs, S. (2013). Diverse calving patterns linked to glacier geometry. *Nature Geoscience*, 6(10), 833-836.

Benamour, M., Liberge, E. and Béghein, C. (2015). Lattice Boltzmann method for fluid flow around bodies using volume penalization. *The International Journal of Multiphysics*, 9(3), 299-316.

Benn, D.I., Cowton, T., Todd, J. and Luckman, A. (2017). Glacier calving in Greenland. *Current Climate Change Reports*, 3(4), 282-290.

Benn, D.I., Warren, C.R. and Mottram, R.H. (2007). Calving processes and the dynamics of calving glaciers. *Earth-Science Reviews*, 82(3-4), 143-179.

Benzi, R., Succi, S. and Vergassola, M. (1992). The lattice Boltzmann equation: Theory and applications. *Physics Reports*, 222(3), 145-197.

Bihs, H., Kamath, A., Alagan Chella, M. and Arntsen, Ø.A., 2015. CFD Simulations of roll motion of a floating ice block in waves using REEF3D. *POAC 2015*, Trondheim, Norway.

Biscarini, C. (2010). Computational fluid dynamics modelling of landslide generated water waves. *Landslides*, 7(2), 117-124.

Bukreev, V.I. and Gusev, A.V. (1996). Gravity waves generated by a body falling onto shallow water. *Journal of Applied Mechanics and Technical Physics*, 37(2), 224-231.

Burton, J.C., Amundson, J.M., Abbot, D.S., Boghosian, A., Cathles, L.M., Correa-Legisos, S., Darnell, K.N., Guttenberg, N., Holland, D.M. and MacAyeal, D.R. (2012). Laboratory investigations of iceberg capsize dynamics, energy dissipation and tsunamigenesis. *Journal of Geophysical Research: Earth Surface*, 117(F01007).

Chen, S. and Doolen, G.D. (1998). Lattice Boltzmann method for fluid flows. *Annual Review of Fluid Mechanics*, 30(1), 329-364.

Chen, F. and Heller, V. (2020). Large-scale iceberg-tsunami benchmark test cases. SPH benchmark test case, SPH European Research Interest Community SPHERIC website (online https://spheric-sph.org/tests/test-15).

Chen, F., Heller, V. and Briganti, R. (2020). Numerical modelling of tsunamis generated by iceberg calving validated with large-scale laboratory experiments. *Advances in Water Resources*, 103647.

Constant, E., Favier, J., Meldi, M., Meliga, P. and Serre, E. (2017). An immersed boundary method in OpenFOAM: Verification and validation. *Computers & Fluids*, 157, 55-72.

Cook, A.J., Fox, A.J., Vaughan, D.G. and Ferrigno, J.G. (2005). Retreating glacier fronts on the Antarctic Peninsula over the past half-century. *Science*, 308(5721), 541-544.

Cook, S., Zwinger, T., Rutt, I.C., O'Neel, S. and Murray, T. (2012). Testing the effect of water in crevasses on a physically based calving model. *Annals of Glaciology*, 53(60), 90-96.

Courant, R., Friedrichs, K. and Lewy, H. (1967). On the partial difference equations of mathematical physics. *IBM Journal of Research and Development*, 11(2), 215-234.

Craik, A.D. (2004). The origins of water wave theory. *Annual Review of Fluid Mechanics*, 36.

Das, K., Janetzke, R., Basu, D., Green, S. and Stamatakos, J. (2009). Numerical simulations of tsunami wave generation by submarine and aerial landslides using RANS and SPH models. In *International Conference on Offshore Mechanics and Arctic Engineering*, 43451, 581-594.

Davidson, J., Cathelain, M., Guillemet, L., Le Huec, T. and Ringwood, J. (2015). Implementation of an OpenFOAM numerical wave tank for wave energy experiments. In *Proceedings of the EWTEC 2015*, Nantes, France.

Dean, R.G. (1970). Relative validities of water wave theories. *Journal of the Waterways, Harbors and Coastal Engineering Division*, 96(1), 105-119.

Dean, R.G. and Dalrymple, R.A. (1991). Water wave mechanics for engineers and scientists. *World Scientific*, Singapore.

Dean, R.G. and Dalrymple, R.A. (2002). Coastal processes with engineering applications. *Cambridge University Press*, Cambridge.

Deardorff, J.W. (1970). A numerical study of three-dimensional turbulent channel flow at large Reynolds numbers. *Journal of Fluid Mechanics*, 41(2), 453-480.

Delta Basin (2020). Available online: https://www.deltares.nl/en/facilities/delta-basin-2/ (accessed on 18 January 2021).

Depoorter, M.A., Bamber, J.L., Griggs, J.A., Lenaerts, J.T.M., Ligtenberg, S.R., Van den Broeke, M.R. and Moholdt, G. (2013). Calving fluxes and basal melt rates of Antarctic ice shelves. *Nature*, 502(7469), 89-92.

Donea, J., Giuliani, S. and Halleux, J.P. (1982). An arbitrary Lagrangian-Eulerian finite element method for transient dynamic fluid-structure interactions. *Computer Methods in Applied Mechanics and Engineering*, 33(1-3), 689-723.

Douglas, S. and Nistor, I. (2015). On the effect of bed condition on the development of tsunami-induced loading on structures using OpenFOAM. *Natural Hazards*, 76(2), 1335-1356.

Duddu, R., Bassis, J.N. and Waisman, H. (2013). A numerical investigation of surface crevasse propagation in glaciers using nonlocal continuum damage mechanics. *Geophysical Research Letters*, 40(12), 3064-3068.

Dullweber, A., Leimkuhler, B. and McLachlan, R. (1997). Symplectic splitting methods for rigid body molecular dynamics. *The Journal of Chemical Physics*, 107(15), 5840-5851.

Durbin, P.A., Medic, G., Seo, J.M., Eaton, J.K. and Song, S. (2001). Rough wall modification of two-layer k-ϵ. *Journal of Fluids Engineering*, 123(1), 16-21.

Dykes, R.C., Brook, M.S. and Lube, G. (2017). A major ice-calving event at Tasman Glacier terminus, Southern Alps, 22 February 2011. *Journal of the Royal Society of New Zealand*, 47(4), 336-343.

Enderlin, E.M., Howat, I.M., Jeong, S., Noh, M.J., Van Angelen, J.H. and Van den Broeke, M.R. (2014). An improved mass budget for the Greenland ice sheet. *Geophysical Research Letters*, 41(3), 866-872.

Enet, F. and Grilli, S.T. (2007). Experimental study of tsunami generation by three-dimensional rigid underwater landslides. *Journal of Waterway, Port, Coastal, and Ocean Engineering*, 133(6), 442-454.

Evers, F.M. and Hager, W.H. (2016). Spatial impulse waves: Wave height decay experiments at laboratory scale. *Landslides*, 13(6), 1395-1403.

Feng, G., Jin, S. and Zhang, T. (2013). Coastal sea level changes in Europe from GPS, tide gauge, satellite altimetry and GRACE, 1993–2011. *Advances in Space Research*, 51(6), 1019-1028.

Feng, Z.G. and Michaelides, E.E. (2004). The immersed boundary-lattice Boltzmann method for solving fluid–particles interaction problems. *Journal of Computational Physics*, 195(2), 602-628.

Flow Science (2007). FLOW-3D User's Manual. Available online: https://www.flow3d.com/resources/cfd-101/modeling-techniques/favor-vs-body-fitted-coordinates/ (accessed on 18 January 2021).

Fritz, H.M., Hager, W.H. and Minor, H.E. (2003). Landslide generated impulse waves. *Experiments in Fluids*, 35(6), 505-519.

Fritz, H.M., Hager, W.H. and Minor, H.E. (2004). Near field characteristics of landslide generated impulse waves. *Journal of Waterway, Port, Coastal, and Ocean Engineering*, 130(6), 287-302.

Fuchs, H., Heller, V. and Hager, W.H. (2010). Impulse wave run-over: Experimental benchmark study for numerical modelling. *Experiments in Fluids*, 49(5), 985-1004.

Fuhrman, D.R. and Madsen, P.A. (2008). Simulation of nonlinear wave run-up with a high-order Boussinesq model. *Coastal Engineering*, 55(2), 139-154.

Fuhrman, D.R. and Madsen, P.A. (2009). Tsunami generation, propagation, and run-up with a high-order Boussinesq model. *Coastal Engineering*, 56(7), 747-758.

GEBCO (2019). GEneral Bathymetric Chart of the Oceans. Available online: https://www.gebco.net/data_and_products/gridded_bathymetry_data/ (accessed on 18 January 2021).

Genevois, R. and Ghirotti, M. (2005). The 1963 Vaiont landslide. *Giornale di Geologia Applicata*, 1(1), 41-52.

Goldstein, D., Handler, R. and Sirovich, L. (1993). Modeling a no-slip flow boundary with an external force field. *Journal of Computational Physics*, 105(2), 354-366.

Greenshields, C.J. (2015). OpenFOAM Programmer's Guide Version 3.0.1. Available online: http://foam.sourceforge.net/docs/Guides-a4/ProgrammersGuide.pdf (accessed on 18 January 2021).

Grilli, S.T. and Watts, P. (2005). Tsunami generation by submarine mass failure. I: Modeling, experimental validation, and sensitivity analyses. *Journal of Waterway, Port, Coastal, and Ocean Engineering*, 131(6), 283-297.

Habchi, C., Russeil, S., Bougeard, D., Harion, J.L., Lemenand, T., Ghanem, A., Della Valle, D. and Peerhossaini, H. (2013). Partitioned solver for strongly coupled fluid–structure interaction. *Computers & Fluids*, 71, 306-319.

Hadžić, I., Hennig, J., Perić, M. and Xing-Kaeding, Y. (2005). Computation of flow-induced motion of floating bodies. *Applied Mathematical Modelling*, 29(12), 1196-1210.

Hammack, J.L. (1973). A note on tsunamis: Their generation and propagation in an ocean of uniform depth. *Journal of Fluid Mechanics*, 60(4), 769-799.

He, X., Luo, L.S. and Dembo, M. (1996). Some progress in lattice Boltzmann method. Part I. Nonuniform mesh grids. *Journal of Computational Physics*, 129(2), 357-363.

Heller, V. (2007). Landslide generated impulse waves: Prediction of near field characteristics. Doctoral dissertation 17531, ETH Zurich, Zurich.

Heller, V. (2011). Scale effects in physical hydraulic engineering models. *Journal of Hydraulic Research*, 49(3), 293-306.

Heller, V. (2019). Tsunamis due to ice masses: Different calving mechanisms and linkage to landslide-tsunamis. Data storage report of HYDRALAB+ test campaign, online under http://doi.org/10.5281/zenodo.2556614.

Heller, V., Attili, T., Chen, F., Brühl, M., Gabl, R., Chen, X., Wolters, G. and Fuchs, H. (2019a). Large-scale experiments of tsunamis generated by iceberg calving. 38$^{\text{th}}$ *IAHR World Congress*, 5628-5639.

Heller, V., Attili, T., Chen, F., Brühl, M., Gabl, R., Chen, X., Wolters, G. and Fuchs, H. (2019b). Large-scale iceberg-tsunami experiments. Proceedings of the HYDRALAB+ Joint User Meeting, Bucharest, Romania, 67-77, Henry, P.-Y., Breteler, M.K. eds.

Heller, V., Atilli, T., Chen, F., Gabl, R. and Wolters, G. (2020). Large-scale investigation into iceberg-tsunamis generated by various iceberg calving mechanisms. *Coastal Engineering*, 103745.

Heller, V., Bruggemann, M., Spinneken, J. and Rogers, B.D. (2016). Composite modelling of subaerial landslide-tsunamis in different water body geometries and novel insight into slide and wave kinematics. *Coastal Engineering*, 109, 20-41.

Heller, V., Chen, F., Brühl, M., Gabl, R., Chen, X., Wolters, G. and Fuchs, H. (2019c). Large-scale experiments into the tsunamigenic potential of different iceberg calving mechanisms. *Scientific Reports*, 9, 861.

Heller, V. and Hager, W.H. (2010). Impulse product parameter in landslide generated impulse waves. *Journal of Waterway, Port, Coastal, and Ocean Engineering*, 136(3), 145-155.

Heller, V. and Hager, W.H. (2011). Wave types of landslide generated impulse waves. *Ocean Engineering*, 38(4), 630-640.

Heller, V., Hager, W.H. and Minor, H.E. (2009). Landslide generated impulse waves in reservoirs - Basics and computation. *VAW-Mitteilung 211, edited by R. Boes*, ETH Zurich, Zurich.

Heller, V., Hager, W.H. and Minor, H.E. (2008). Scale effects in subaerial landslide generated impulse waves. *Experiments in Fluids*, 44(5), 691-703.

Heller, V. and Kinnear, R.D. (2010). Discussion of "Experimental investigation of impact generated tsunami; related to a potential rock slide, Western Norway" by G. Sælevik, A. Jensen, G. Pedersen [Coastal Eng. 56 (2009) 897–906]. *Coastal Engineering*, 57(8), 773-777.

Heller, V., Moalemi, M., Kinnear, R.D. and Adams, R.A. (2012). Geometrical effects on landslide-generated tsunamis. *Journal of Waterway, Port, Coastal, and Ocean Engineering*, 138(4), 286-298.

Heller, V. and Spinneken, J. (2013). Improved landslide-tsunami prediction: Effects of block model parameters and slide model. *Journal of Geophysical Research: Oceans*, 118(3), 1489-1507.

Heller, V. and Spinneken, J. (2015). On the effect of the water body geometry on landslide-tsunamis: Physical insight from laboratory tests and 2D to 3D wave parameter transformation. *Coastal Engineering*, 104(10), 113-134.

Hieber, S.E. and Koumoutsakos, P. (2008). An immersed boundary method for smoothed particle hydrodynamics of self-propelled swimmers. *Journal of Computational Physics*, 227(19), 8636-8654.

Higuera, P., Lara, J.L. and Losada, I.J. (2013a). Realistic wave generation and active wave absorption for Navier–Stokes models: Application to OpenFOAM. *Coastal Engineering*, 71, 102-118.

Higuera, P., Lara, J.L. and Losada, I.J. (2013b). Simulating coastal engineering processes with OpenFOAM. *Coastal Engineering*, 71, 119-134.

Huber, A. (1980). Schwallwellen in Seen als Folge von Bergstürzen, VAW-Mitteilung 47, D. Vischer, ed., Versuchsanstalt für Wasserbau. *Hydrologie und Glaziologie, ETH Zurich*, Zurich.

Hulme, A. (1982). The wave forces acting on a floating hemisphere undergoing forced periodic oscillations. *Journal of Fluid Mechanics*, 121, 443-463.

Huber, A. and Hager, W.H. (1997). Forecasting impulse waves in reservoirs. In *Transactions of the International Congress on Large Dams*, 1, 993-1006.

Issa, R.I. (1986). Solution of the implicitly discretised fluid flow equations by operator-splitting. *Journal of Computational Physics*, 62(1), 40-65.

Jacobsen, N.G., Fuhrman, D.R. and Fredsøe, J. (2012). A wave generation toolbox for the open-source CFD library: OpenFoam. *International Journal for Numerical Methods in Fluids*, 70(9), 1073-1088.

Jasak, H. (1996). Error analysis and estimation for the finite volume method with applications to fluid flows. PhD thesis, Imperial College, University of London, London.

Jasak, H. (2009). OpenFOAM: open source CFD in research and industry. *International Journal of Naval Architecture and Ocean Engineering*, 1(2), 89-94.

Jasak, H., Jemcov, A. and Tuković, Ž. (2007). September. OpenFOAM: a C++ library for complex physics simulations. In *International Workshop on Coupled Methods in Numerical Dynamics*, 1000, 1-20. IUC Dubrovnik, Croatia.

Jasak, H, Rigler, D. and Tuković, Ž. (2014). Design and implementation of Immersed Boundary Method with discrete forcing approach for boundary conditions. In 6^{th} *European Conference on Computational Fluid Dynamics, ECFD 2014. International Center for Numerical Methods in Engineering*, 5319-5332, Barcelona, Spain.

Kamath, A., Chella, M.A., Bihs, H. and Arntsen, Ø.A. (2016). Breaking wave interaction with a vertical cylinder and the effect of breaker location. *Ocean Engineering*, 128, 105-115.

Kamphuis, J.W. and Bowering, R.J. (1970). Impulse waves generated by landslides. In *12*$^{\text{th}}$ *International Conference on Coastal Engineering*, 575-588.

Kenneally, J.P. and Hughes, T.J. (2002). The calving constraints on inception of Quaternary ice sheets. *Quaternary International*, 95, 43-53.

Kimura, Y., Suzuki, K. and Inamuro, T. (2014). Flight simulations of a two-dimensional flapping wing by the IB-LBM. *International Journal of Modern Physics C*, 25(01), 1340020.

Korteweg, D.J. and De Vries, G. (1895). On the change of form of long waves advancing in a rectangular canal, and on a new type of long stationary waves. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 39(240), 422-443.

Kranzer, H.C. and Keller, J.B. (1959). Water waves produced by explosions. *Journal of Applied Physics*, 30(3), 398-407.

Krüger, T., Kusumaatmaja, H., Kuzmin, A., Shardt, O., Silva, G. and Viggen, E.M. (2017). The lattice Boltzmann method. *Springer International Publishing*, 10(978-3), 4-15.

Ladd, A.J.C. and Verberg, R. (2001). Lattice-Boltzmann simulations of particle-fluid suspensions. *Journal of Statistical Physics*, 104(5-6), 1191-1251.

Laitone, E.V. (1960). The second approximation to cnoidal and solitary waves. *Journal of Fluid Mechanics*, 9(3), 430-444.

Lee, J.F. (1995). On the heave radiation of a rectangular structure. *Ocean Engineering*, 22(1), 19-34.

Levermann, A. (2011). When glacial giants roll over. *Nature*, 472(7341), 43-44.

Le Méhauté, B. (2013). An introduction to hydrodynamics and water waves. Springer Science & Business Media.

Le Méhauté, B. (1962). On non-saturated breakers and the wave run-up. *Coastal Engineering Proceedings*, (8), 6-6.

Le Méhauté, B. (1976). Similitude in coastal engineering. *Journal of the Waterways, Harbours and Coastal Engineering Division*, 102(3), 317-335.

Liberge, E. and Béghein, C. (2019). Fluid structure interaction modelling using Lattice Boltzmann and volume penalization method. arXiv preprint arXiv:1905.05110.

Lindstrøm, E.K. (2016). Waves generated by subaerial slides with various porosities. *Coastal Engineering*, 116, 170-179.

Liu, Y., Moore, J.C., Cheng, X., Gladstone, R.M., Bassis, J.N., Liu, H., Wen, J. and Hui, F. (2015). Ocean-driven thinning enhances iceberg calving and retreat of Antarctic ice shelves. *Proceedings of the National Academy of Sciences*, 112(11), 3263-3268.

Liu, P.L., Wu, T.R., Raichlen, F., Synolakis, C.E. and Borrero, J.C. (2005). Runup and rundown generated by three-dimensional sliding masses. *Journal of Fluid Mechanics*, 536, 107.

Lüthi, M.P. and Vieli, A. (2016). Multi-method observation and analysis of a tsunami caused by glacier calving. *The Cryosphere*, 10(3), 995-1002.

Lynett, P.J., Wu, T.R. and Liu, P.L.F. (2002). Modeling wave runup with depth-integrated equations. *Coastal Engineering*, 46(2), 89-107.

MacAyeal, D.R., Abbot, D.S. and Sergienko, O.V. (2011). Iceberg-capsize tsunamigenesis. *Annals of Glaciology*, 52(58), 51-56.

MacAyeal, D.R., Okal, E.A., Aster, R.C. and Bassis, J.N. (2009). Seismic observations of glaciogenic ocean waves (micro-tsunamis) on icebergs and ice shelves. *Journal of Glaciology*, 55(190), 193-206.

MacAyeal, D.R., Scambos, T.A., Hulbe, C.L. and Fahnestock, M.A. (2003). Catastrophic ice-shelf break-up by an ice-shelf-fragment-capsize mechanism. *Journal of Glaciology*, 49(164), 22-36.

Madsen, P.A., Fuhrman, D.R. and Wang, B. (2006). A Boussinesq-type method for fully nonlinear waves interacting with a rapidly varying bathymetry. *Coastal Engineering*, 53(5-6), 487-504.

Marshall, J., Adcroft, A., Hill, C., Perelman, L. and Heisey, C. (1997). A finite-volume, incompressible Navier Stokes model for studies of the ocean on parallel computers. *Journal of Geophysical Research: Oceans*, 102(C3), 5753-5766.

Mason, P.J. (1994). Large-eddy simulation: A critical review of the technique. *Quarterly Journal of the Royal Meteorological Society*, 120(515), 1-26.

Massel, S.R. and Przyborska, A. (2013). Surface wave generation due to glacier calving. *Oceanologia*, 55(1), 101-127.

Mei, C.C. (1989). The applied dynamics of ocean surface waves. *World Scientific*, Singapore.

Mendsonboaz (2009). Tsunami Greenland - Tsunami Groelândia 1995. Online under https:// www.youtube.com/watch?v=z8LWSOPwkn8 (accessed on 18 January 2021, in German).

Miles, J.W. (1980). Solitary waves. *Annual Review of Fluid Mechanics*, 12(1), 11-43.

Missy Andrea (2016). Perito Moreno glacier calving and tsunami wave - Patagonia, Argentina. Online under https://www.youtube.com/watch?v=cF764iqymQk (accessed on 18 January 2021).

Minowa, M., Podolskiy, E., Sugiyama, S., Sakakibara, D. and Skvarca, P. (2018). Glacier calving observed with time-lapse imagery and tsunami waves at Glaciar Perito Moreno, Patagonia. *Journal of Glaciology*, 64(245), 362–376.

Minowa, M., Podolskiy, E.A., Jouvet, G., Weidmann, Y., Sakakibara, D., Tsutaki, S., Genco, R. and Sugiyama, S. (2019). Calving flux estimation from tsunami waves. *Earth and Planetary Science Letters*, 515, 283-290.

Mittal, R., Dong, H., Bozkurttas, M., Najjar, F.M., Vargas, A. and Von Loebbecke, A. (2008). A versatile sharp interface immersed boundary method for incompressible flows with complex boundaries. *Journal of Computational Physics*, 227(10), 4825-4852.

Mittal, R. and Iaccarino, G. (2005). Immersed boundary methods. *Annual Review of Fluid Mechanics*, 37, 239-261.

Mohd-Yusof, J. (1997). Combined immersed-boundary/B-spline methods for simulations of flow in complex geometries. *Center for Turbulence Research Annual Research Briefs*, 161(1), 317-327.

Monaghan, J.J. (1992). Smoothed particle hydrodynamics. *Annual Review of Astronomy and Astrophysics*, 30(1), 543-574.

Monaghan, J.J. and Kos, A. (2000). Scott Russell's wave generator. *Physics of Fluids*, 12(3), 622-630.

Monaghan, J.J. and Kos, A. (1999). Solitary waves on a Cretan beach. *Journal of Waterway, Port, Coastal, and Ocean Engineering*, 125(3), 145-155.

Morgan, G.C.J., Zang, J., Greaves, D., Heath, A., Whitlow, C. and Young, J. (2011). Using the rasInterFoam CFD model for wave transformation and coastal modelling. *Coastal Engineering Proceedings*, 1(32), waves-23.

Moukalled, F., Mangani, L. and Darwish, M. (2016). The finite volume method in computational fluid dynamics. *Springer*, Berlin.

Munk, W.H. (1949). The solitary wave theory and its application to surf problems. *Annals of the New York Academy of Sciences*, 51(3), 376-424.

Naaim, M. (2013). Impulse water wave generation by a snow avalanche: Two shallow layers model. *EGU General Assembly 2013*, 13947.

Narayanaswamy, M., Crespo, A.J.C., Gómez-Gesteira, M. and Dalrymple, R.A. (2010). SPHysics-FUNWAVE hybrid model for coastal wave propagation. *Journal of Hydraulic Research*, 48(S1), 85-93.

Newman, J.N. (2018). Marine hydrodynamics. *MIT Press*, Cambridge, MA.

Nick, F.M., Vieli, A., Andersen, M.L., Joughin, I., Payne, A., Edwards, T.L., Pattyn, F. and Van de Wal, R.S. (2013). Future sea-level rise from Greenland's main outlet glaciers in a warming climate. *Nature*, 497(7448), 235-238.

Noble, D.R. and Torczynski, J.R. (1998). A lattice-Boltzmann method for partially saturated computational cells. *International Journal of Modern Physics C*, 9(08), 1189-1201.

Noda, E. (1970). Water waves generated by landslides. *Journal of the Waterways, Harbors and Coastal Engineering Division*, 96(4), 835-855.

Nye, J.F. (1957). The distribution of stress and velocity in glaciers and ice-sheets. *Proceedings of the Royal Society of London. Series A. Mathematical and Physical Sciences*, 239(1216), 113-133.

Nye, J.F. (1952). The mechanics of glacier flow. *Journal of Glaciology*, 2(12), 82-93.

O'Neel, S., Marshall, H.P., McNamara, D.E. and Pfeffer, W.T. (2007). Seismic detection and analysis of icequakes at Columbia Glacier, Alaska. *Journal of Geophysical Research: Earth Surface*, 112(F3).

OpenFOAM extensions (2016). Online under https://sourceforge.net/u/hjasak/foam-extend-4.0/ci/master/tree/ (accessed on 18 January 2021).

Patankar, S.V. and Spalding, D.B. (1972). A calculation procedure for heat, mass and momentum transfer in three-dimensional parabolic flows. *International Journal of Heat Mass Transfer*, 15, 1787–1806.

Paterson, W.S.B. (1994). The Physics of Glaciers, 3rd edition. *Pergamon*, Oxford.

Palacios, F., Colonno, M.R., Aranake, A.C., Campos, A., Copeland, S.R., Economon, T.D., Lonkar, A.K., Lukaczyk, T.W., Taylor, T.W.R. and Alonso, J.J. (2013). Stanford University unstructured (SU2): An open-source integrated computational environment for multi-physics simulation and design. In *51st AIAA Aerospace Sciences Meeting including the New Horizons Forum and Aerospace Exposition*, Grapevine, USA.

Panizzo, A., Bellotti, G. and De Girolamo, P. (2002). Application of Wavelet transform analysis to landslide generated waves. *Coastal Engineering*, 44(4), 321-338.

Panizzo, A., De Girolamo, P. and Petaccia, A. (2005). Forecasting impulse waves generated by subaerial landslides. *Journal of Geophysical Research: Oceans*, 110(C12025).

Peskin, C.S. (1972). Flow patterns around heart valves: A numerical method. *Journal of Computational Physics*, 10, 252-271.

Pralong, A. and Funk, M. (2005). Dynamic damage model of crevasse opening and application to glacier calving. *Journal of Geophysical Research: Solid Earth*, 110(B1).

Qiu, L.C., Tian, L., Liu, X.J. and Han, Y. (2019). A 3D multiple-relaxation-time LBM for modeling landslide-induced tsunami waves. *Engineering Analysis with Boundary Elements*, 102, 51-59.

Rabczuk, T., Gracie, R., Song, J.H. and Belytschko, T. (2010). Immersed particle method for fluid–structure interaction. *International Journal for Numerical Methods in Engineering*, 81(1), 48-71.

Rayleigh, L. (1876). On the resistance of fluids. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 2(13), 430-441.

Reeh, N. (1968). On the calving of ice from floating glaciers and ice shelves. *Journal of Glaciology*, 7(50), 215-232.

Reeh, N. (1985). Was the Greenland ice sheet thinner in the late Wisconsinan than now? *Nature*, 317(6040), 797-799.

Romano, A., Lara, J.L., Barajas, G., Di Paolo, B., Bellotti, G., Di Risio, M., Losada, I.J. and De Girolamo, P. (2020). Tsunamis generated by submerged landslides: Numerical analysis of the near-field wave characteristics. *Journal of Geophysical Research: Oceans*, e2020JC016157.

Ruffini, G., Heller, V. and Briganti, R. (2021). Numerical characterisation and efficient prediction of landslide-tsunami propagation over a wide range of idealised bathymetries (under review).

Ruffini, G., Heller, V. and Briganti, R. (2019). Numerical modelling of landslide-tsunami propagation in a wide range of idealised water body geometries. *Coastal Engineering*, 153, 103518.

Russell, J.S. (1837). Report on waves. *Meeting of the British Association for the Advancement of Science Liverpool*, 417–496.

Sælevik, G., Jensen, A. and Pedersen, G. (2009). Experimental investigation of impact generated tsunami; related to a potential rock slide, Western Norway. *Coastal Engineering*, 56(9), 897-906.

Sergeant, A., Mangeney, A., Stutzmann, E., Montagner, J.P., Walter, F., Moretti, L. and Castelnau, O. (2016). Complex force history of a calving-generated glacial earthquake derived from broadband seismic inversion. *Geophysical Research Letters*, 43(3), 1055-1065.

Sergeant, A., Yastrebov, V.A., Mangeney, A., Castelnau, O., Montagner, J.P. and Stutzmann, E. (2018). Numerical modeling of iceberg capsizing responsible for glacial earthquakes. *Journal of Geophysical Research: Earth Surface*, 123(11), 3013-3033.

Smagorinsky, J. (1963). General circulation experiments with the primitive equations: I. The basic experiment. *Monthly Weather Review*, 91(3), 99-164.

Steig, E.J., Ding, Q., White, J.W., Küttel, M., Rupper, S.B., Neumann, T.A., Neff, P.D., Gallant, A.J., Mayewski, P.A., Taylor, K.C. and Hoffmann, G. (2013). Recent climate and ice-sheet changes in West Antarctica compared with the past 2,000 years. *Nature Geoscience*, 6(5), 372-375.

Stokes, G.G. (1847). On the theory of oscillatory waves. *Mathematics and Physics Papers*, 1, 197-229.

Sui, Y., Chew, Y.T., Roy, P. and Low, H.T. (2007). A hybrid immersed-boundary and multi-block lattice Boltzmann method for simulating fluid and moving-boundaries interactions. *International Journal for Numerical Methods in Fluids*, 53(11), 1727-1754.

Sun, P.N., Colagrossi, A., Marrone, S., Antuono, M. and Zhang, A.M. (2018). Multi-resolution Delta-plus-SPH with tensile instability control: Towards high Reynolds number flows. *Computer Physics Communications*, 224, 63-80.

Synolakis, C.E. (1991). Green's law and the evolution of solitary waves. *Physics of Fluids A: Fluid Dynamics*, 3(3), 490-491.

Synolakis, C.E. and Skjelbreia, J.E. (1993). Evolution of maximum amplitude of solitary waves on plane beaches. *Journal of Waterway, Port, Coastal, and Ocean Engineering*, 119(3), 323-342.

Tan, H. and Chen, S. (2017). A hybrid DEM-SPH model for deformable landslide and its generated surge waves. *Advances in Water Resources*, 108, 256-276.

Tan, H., Ruffini, G., Heller, V. and Chen, S. (2018). A numerical landslide-tsunami hazard assessment technique applied on hypothetical scenarios at Es Vedrà, offshore Ibiza. *Journal of Marine Science and Engineering*, 6(4), 111, 1-22.

The Guardian (2018). Huge iceberg threatens tiny Greenland village. Online under https://www.theguardian.com/world/2018/jul/14/huge-iceberg-threatens-village-in-greenland (accessed on 18 January 2021).

Top Fives (2018). 5 Icebergs Flipping Over - incredible. Online under https://www.youtube.com/watch?v=hxy-0zpJwxs (accessed on 18 January 2021).

Tsai, V.C., Rice, J.R. and Fahnestock, M. (2008). Possible mechanisms for glacial earthquakes. *Journal of Geophysical Research: Earth Surface*, 113(F3).

Ursell, F. (1953). The long-wave paradox in the theory of gravity waves. In *Mathematical Proceedings of the Cambridge Philosophical Society*, 49(4), 685-694.

Vacondio, R., Rogers, B.D., Stansby, P.K., Mignosa, P. and Feldman, J. (2013). Variable resolution for SPH: A dynamic particle coalescing and splitting scheme. *Computer Methods in Applied Mechanics and Engineering*, 256, 132-148.

Van den Broeke, M., Box, J., Fettweis, X., Hanna, E., Noël, B., Tedesco, M., Van As, D., Van de Berg, W.J. and van Kampenhout, L. (2017). Greenland ice sheet surface mass loss: Recent developments in observation and modeling. *Current Climate Change Reports*, 3(4), 345-356.

Van der Veen, C.J. (1999). Crevasses on glaciers. *Polar Geography*, 23(3), 213-245.

Van der Veen, C.J. (1998). Fracture mechanics approach to penetration of surface crevasses on glaciers. *Cold Regions Science and Technology*, 27(1), 31-47.

Vaňková, I. and Holland, D.M. (2016). Calving signature in ocean waves at Helheim Glacier and Sermilik Fjord, East Greenland. *Journal of Physical Oceanography*, 46(10), 2925-2941.

Velicogna, I., Sutterley, T.C. and Van Den Broeke, M.R. (2014). Regional acceleration in ice mass loss from Greenland and Antarctica using GRACE time-variable gravity data. *Geophysical Research Letters*, 41(22), 8130-8137.

Violeau, D. and Rogers, B.D. (2016). Smoothed particle hydrodynamics (SPH) for free-surface flows: Past, present and future. *Journal of Hydraulic Research*, 54(1), 1-26.

Viroulet, S., Cébron, D., Kimmoun, O. and Kharif, C. (2013). Shallow water waves generated by subaerial solid landslides. *Geophysical Journal International*, 193(2), 747-762.

Wang, H. and Belytschko, T. (2009). Fluid–structure interaction by the discontinuous-Galerkin method for large deformations. *International Journal for Numerical Methods in Engineering*, 77(1), 30-49.

Weiss, J. (2004). Subcritical crack propagation as a mechanism of crevasse formation and iceberg calving. *Journal of Glaciology*, 50(168), 109-115.

Weiss, R., Krastel, S., Anasetti, A. and Wünnemann, K. (2013). Constraining the characteristics of tsunami waves from deformable submarine slides. *Geophysical Journal International*, 194(1), 316-321.

Weller, H.G., Tabor, G., Jasak, H. and Fureby, C. (1998). A tensorial approach to computational continuum mechanics using object-oriented techniques. *Computers in Physics*, 12(6), 620-631.

Wiegel, R.L. (1964). Oceanographical engineering. *Prentice-Hall*, London.

Wiegel, R.L., Noda, E.K., Kuba, E.M., Gee, D.M. and Tornberg, G.F. (1970). Water waves generated by landslides in reservoirs. *Journal of the Waterways, Harbors and Coastal Engineering Division*, 96(2), 307-333.

Wilkes, D.R., Peters, H., Croaker, P., Marburg, S., Duncan, A.J. and Kessissoglou, N. (2017). Non-negative intensity for coupled fluid–structure interaction problems using the fast multipole method. *The Journal of the Acoustical Society of America*, 141(6), 4278-4288.

Wouters, B., Gardner, A.S. and Moholdt, G. (2019). Global glacier mass loss during the GRACE satellite mission (2002-2016). *Frontiers in Earth Science*, 7, 96.

Wu, C.H.K. and Nguyen, V.T. (2017). Aerodynamic simulations of offshore floating wind turbine in platform-induced pitching motion. *Wind Energy*, 20(5), 835-858.

Wu, K., Yang, D. and Wright, N. (2016). A coupled SPH-DEM model for fluid-structure interaction problems with free-surface flow and structural failure. *Computers & Structures*, 177, 141-161.

Xu, J. (2007). Benchmarks on tera-scalable models for DNS of turbulent channel flow. *Parallel Computing*, 33(12), 780-794.

Xu, Y.Q., Tian, F.B. and Deng, Y.L. (2013). An efficient red blood cell model in the frame of IB-LBM and its application. *International Journal of Biomathematics*, 6(01), 1250061.

Yang, S. (2016). Hazards of impulse waves generated by ice calving. Interim report, BEng Individual Investigative Project, The University of Nottingham, Nottingham.

Yang, Q., Jones, V. and McCue, L. (2012). Free-surface flow interactions with deformable structures using an SPH–FEM model. *Ocean Engineering*, 55, 136-147.

Yavari-Ramshe, S. and Ataie-Ashtiani, B. (2016). Numerical modeling of subaerial and submarine landslide-generated tsunami waves—recent advances and future challenges. *Landslides*, 13(6), 1325-1368.

Zhang, Q. and Hisada, T. (2001). Analysis of fluid–structure interaction problems with structural buckling and large domain changes by ALE finite element method. *Computer Methods in Applied Mechanics and Engineering*, 190(48), 6341-6357.

Zwally, H.J., Li, J., Brenner, A.C., Beckley, M., Cornejo, H.G., Dimarzio, J., Giovinetto, M.B., Neumann, T.A., Robbins, J., Saba, J.L. and Yi, D. (2011). Greenland ice sheet mass balance: Distribution of increased mass loss with climate warming; 2003–07 versus 1992–2002. *Journal of Glaciology*, 57(201), 88-102.

Zweifel, A. (2004). Impulswellen: Effekte der Rutschdichte und der Wassertiefe, Doctoral dissertation 15596, ETH Zurich, Zurich (in German).

Zweifel, A., Hager, W.H. and Minor, H.E. (2006). Plane impulse waves in reservoirs. *Journal of Waterway, Port, Coastal, and Ocean Engineering*, 132(5), 358-368.

Appendix A

Implementation of the new flow solver

The newly implemented flow solver *interDyMIbFoam* is extended based on the already available multiphase solver *interIbFoam* in Foam-extend 4.0. The newly introduced functionalities are calling the motion solver and supporting the handling of moving immersed boundaries. Therefore, the following header files need to be added: *dynamicFvMesh.H* and *createDynamicFvMesh.H*. The list of the header files before applying the main function is

```
# include "fvCFD.H"
# include "dynamicFvMesh.H"
# include "interfaceProperties.H"
# include "twoPhaseMixture.H"
# include "turbulenceModel.H"
# include "pimpleControl.H"
# include "immersedBoundaryFvPatch.H"
# include "immersedBoundaryAdjustPhi.H"
```

In the main function of *interDyMIbFoam*, *mesh.update()* is required to call the motion solver to update the motion state of the rigid body (e.g. the iceberg acceleration, velocity and the displacement). Note that after determining the new position of the immersed boundary, *createIbMashs.H* needs to be called to update the immersed boundary position. The source code of the main function is presented hereafter. Note that further code such as fields initiation (*createFields.H*), solving velocity (*UEqn.H*) and pressure equations (*pEqn.H*) also need to be modified accordingly for moving immersed boundary simulations.

```
int main(int argc, char *argv[ ])
{
    #include "setRootCase.H"
    #include "createTime.H"
    #include "createDynamicFvMesh.H"
    #pimpleControl pimple(mesh);
```

```
    # include "readGravitationalAcceleration.H"
    # include "initContinuityErrs.H"
    # include "createFields.H"
    # include "createControls.H"
    # include "CourantNo.H"
    # include "setInitialDeltaT.H"
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //
    Info<< "\nStarting time loop\n" << endl;
    while (runTime.run())
    {
        // Make the fluxes absolute
        fvc::makeAbsolute(phi, U);
        # include "setDeltaT.H"
        runTime++;
        Info<< "Time = " << runTime.timeName() << nl << endl;
        bool meshChanged = mesh.update();
        reduce(meshChanged, orOp<bool>());
        # include "createIbMasks.H"
        # include "readControls.H"
        # include "immersedBoundaryCourantNo.H"
        # include "volContinuity.H"
        volScalarField gh("gh", g & mesh.C());
        surfaceScalarField ghf("ghf", g & mesh.Cf());
        if (correctPhi && meshChanged)
        {
            # include "correctPhi.H"
        }
        // Make the fluxes relative to the mesh motion
        fvc::makeRelative(phi, U);
        if (checkMeshCourantNo)
        {
            # include "meshCourantNo.H"
        }
        // Pressure-velocity corrector
```

```
        while (pimple.loop())
        {
            twoPhaseProperties.correct();
            # include "UEqn.H"
            // — PISO loop
            while (pimple.correct())
            {
                # include "pEqn.H"
            }
            # include "limitU.H"
            p = pd + rho*gh;
            p.correctBoundaryConditions();
            # include "alphaEqn.H"
            turbulence->correct();
        }
        runTime.write();
        Info<< "ExecutionTime = " << runTime.elapsedCpuTime() << " s"
            << " ClockTime = " << runTime.elapsedClockTime() << " s"
            << nl << endl;
    }
    Info<< "End\n" << endl;
    return 0;
}
```

Appendix B

Trajectory inference using the motion sensor

The method of inferencing a trajectory using the motion sensor is applicable for all 5 iceberg calving mechanisms in Heller et al. (2019c; 2020). The raw data from the motion sensor were three local acceleration components recorded with the accelerometer and three global rotation angle components obtained from the gyroscope. Prior to post-processing, the noise from the raw data was removed by a low-pass filter with cut-off frequency at 10 Hz. In order to inference the trajectory of the block, the transformation of acceleration components from local to global coordinate system is required. There are three standard rotation matrices used in this coordinate transform, which rotate vectors by an angle $\boldsymbol{\theta}(\theta_x, \theta_y, \theta_z)$ relative to the x-, y- and z-axes, respectively.

$$R_x(\theta_x) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta_x & -\sin\theta_x \\ 0 & \sin\theta_x & \cos\theta_x \end{bmatrix}$$

$$R_y(\theta_y) = \begin{bmatrix} \cos\theta_y & 0 & \sin\theta_y \\ 0 & 1 & 0 \\ -\sin\theta_y & 0 & \cos\theta_y \end{bmatrix} \tag{B.1}$$

$$R_z(\theta_z) = \begin{bmatrix} \cos\theta_z & -\sin\theta_z & 0 \\ \sin\theta_z & \cos\theta_z & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

The rotation of the block can be expressed as matrix multiplication, and the global acceleration $\mathbf{a}_a$ can thus be obtained by

$$\mathbf{a}_a = [R_z(\theta_z)R_y(\theta_y)R_x(\theta_x)]^{-1}\mathbf{a}_l + \mathbf{g}, \tag{B.2}$$

where $\mathbf{a}_l$ denotes the global acceleration vector and $\mathbf{g} = (0,\ 0,\ -9.81)\ \mathrm{ms}^{-2}$. Therefore, the block velocity and displacement can be obtained by integrating $\mathbf{a}_a$ once and twice over time, respectively.

For overturning cases, where the block performed a pitch motion (Section 4.2.3), only θ_y was used for the trajectory inference. This is to avoid the accumulation of small errors from both $\boldsymbol{\theta}$ and $\mathbf{a}_l$ with time. Therefore, in mechanism D and E (gravity- and buoyancy-dominated overturning), the velocity components V_x and V_z were calculated with

$$V_x = R\frac{d\theta_y}{dt}\cos\theta_y, \tag{B.3}$$

$$V_z = -R\frac{d\theta_y}{dt}\sin\theta_y, \tag{B.4}$$

where R denotes the distance between the motion sensor and the centre of rotation (Section 4.2.3). In Eqs. (B.3) and (B.4) $\frac{d\theta_y}{dt}$ was obtained by a central difference scheme

$$(\frac{d\theta_y}{dt})_{t=t_\mathrm{n}} = \frac{1}{2}(\frac{{\theta_y}^{\mathrm{n}} - {\theta_y}^{\mathrm{n-1}}}{t^{\mathrm{n}} - t^{\mathrm{n-1}}} + \frac{{\theta_y}^{\mathrm{n+1}} - {\theta_y}^{\mathrm{n}}}{t^{\mathrm{n+1}} - t^{\mathrm{n}}}). \tag{B.5}$$

In Eq. (B.5) the superscript n−1, n and n+1 indicate the previous, current and next moments in time, respectively.

Appendix C

Implementation of the solution of the theoretical floating heaving sphere case

The method presented here of obtaining the analytical solution proposed by Hulme (1982) for the floating heaving sphere was implemented in Matlab. The key point of obtaining the analytical solution is to calculate the unknown complex constant series p_{i} in Eq. (5.5). According to Hulme (1982), p_{i} satisfies the following infinite linear system of equations

$$p_{\mathrm{i}} + (\frac{\sigma^2}{g} a_r) \sum_{\mathrm{j}=1}^{\infty} p_{\mathrm{i}} M_{\mathrm{ij}} i = d_{\mathrm{i}} \ (\mathrm{i} = 1, 2, 3, ...), \tag{C.1}$$

where $g = 9.81$ kg/m^2 denotes the gravitational acceleration, i the imaginary unit, $\sigma = 2\pi/\omega$ the angular frequency and a_r the radius of the sphere. In Eq. (C.1)

$$M_{\mathrm{ij}} = \frac{4\mathrm{i}+1}{2\mathrm{i}+1}[I(2\mathrm{i}, 2\mathrm{j}-1; 0) - 2I(2\mathrm{i}, 1; 0)I(0, 2\mathrm{j}-1; 0)], \tag{C.2}$$

$$d_{\mathrm{i}} = \frac{4\mathrm{i}+1}{2\mathrm{i}+1}[J(2\mathrm{i}, ka_r) - 2J(0, ka_r)I(2\mathrm{i}, 1; 0)], \tag{C.3}$$

where $k = \sigma^2/g$ denotes the wave number, $I(\mathrm{m,n;0}) = \int_0^1 P_{\mathrm{m}}(x)P_{\mathrm{n}}(x)dx$ the notation where $P_{\mathrm{m}}(x)$ and $P_{\mathrm{n}}(x)$ are m- and n-order Legendre polynomials, respectively. In Eq. (C.3)

$$\begin{aligned} J(\mathrm{m}, ka_r) \ &= \ -I(\mathrm{m}, 0; 0) - ka_r \textstyle\sum_{\mathrm{n}=1}^{\infty} \frac{-ka_r{}^{\mathrm{n}}}{(\mathrm{n}-1)!} \frac{\partial}{\partial \mathrm{j}} [I(\mathrm{m}, \mathrm{j}; 0)]_{\mathrm{j}=\mathrm{n}} \\ &\quad + ka_r \textstyle\sum_{\mathrm{n}=1}^{\infty} \frac{-ka_r{}^{\mathrm{n}}}{\mathrm{n}!} \{\mathrm{n}[\Gamma(\mathrm{n}+1) + \pi i - \ln(ka_r)] - 1\} I(\mathrm{m}, \mathrm{n}; 0), \end{aligned} \tag{C.4}$$

where $\Gamma(\mathrm{n}+1) = -\lambda + \sum_{\mathrm{i}=1}^{\mathrm{n}} \frac{1}{\mathrm{i}}$ and $\lambda = 0.577$ denotes the Euler–Mascheroni constant. Further, according to Hulme (1982),

$$I(2\mathrm{i}, 2\mathrm{j}; 0) = \begin{cases} 0, & \mathrm{i} \neq \mathrm{j} \\ \frac{1}{4\mathrm{i}+1}, & \mathrm{i} = \mathrm{j} \end{cases} \tag{C.5}$$

$$I(2\mathrm{i}, 2\mathrm{j}-1; 0) = (\frac{-1}{4})^{\mathrm{i}+\mathrm{j}-1} \frac{(2\mathrm{i})!(2\mathrm{j}-1)!}{(2\mathrm{i}-2\mathrm{j}-3)(2\mathrm{i}+2\mathrm{j}-2)(\mathrm{i}!\mathrm{j}!)^2}. \tag{C.6}$$

Therefore, when n is an even number

$$\frac{\partial}{\partial \mathrm{j}}[I(\mathrm{m}, \mathrm{j}; 0)]_{\mathrm{j}=\mathrm{n}} = \begin{cases} 0, & \mathrm{n} \neq \mathrm{m} \\ \frac{-4}{(4\mathrm{n}+1)^2}, & \mathrm{n} = \mathrm{m}. \end{cases} \tag{C.7}$$

and when n is an odd number

$$\begin{aligned} \tfrac{\partial}{\partial \mathrm{j}}[I(\mathrm{m}, \mathrm{j}; 0)]_{\mathrm{j}=\mathrm{n}} \quad = \quad & \tfrac{-1^{\frac{\mathrm{m}+\mathrm{n}+1}{2}}}{4^{\frac{\mathrm{m}+\mathrm{n}-1}{2}}} \tfrac{\mathrm{m}!}{[(\frac{\mathrm{m}}{2})!]^2} \tfrac{\mathrm{n}!}{(\mathrm{m}-\mathrm{n})(\mathrm{m}+\mathrm{n}+1)} \tfrac{1}{[(\frac{\mathrm{n}-1}{2})!]^2} [\tfrac{\pi}{2} - \ln 2 \\ & + \tfrac{\Gamma(\mathrm{n}+1)(\mathrm{m}-\mathrm{n})(\mathrm{m}+\mathrm{n}+1)+2\mathrm{n}+1}{(\mathrm{m}-\mathrm{n})(\mathrm{m}+\mathrm{n}+1)} - \Gamma(\tfrac{\mathrm{n}+1}{2})], \end{aligned} \tag{C.8}$$

Thus far, constants M_{ij} and d_{i} in Eq. (C.1) can be obtained and Eq. (C.1) was truncated as an N × N equation system (here N = 20) to solve the constant series p_{i}. The complex constant C is then determined by

$$C = \frac{1}{2}[J(0, ka_r) - ka_r \sum_{\mathrm{i}=1}^{\mathrm{N}} p_{\mathrm{i}} I(0, 2\mathrm{i}-1; 0)]^{-1}. \tag{C.9}$$

Finally, the velocity potentials in Eq. (5.5) can be fully obtained as

$$\phi_0 = \int_0^{\mathrm{N}_\tau} \frac{\tau}{\tau - k} J_0(\tau r) d\tau, \tag{C.10}$$

$$\phi_{\mathrm{i}} = \frac{k}{2\mathrm{i}} \frac{1}{r^{2\mathrm{i}}} P_{2\mathrm{i}-1}(0) + \frac{1}{r^{2\mathrm{i}+1}} P_{2\mathrm{i}}(0). \tag{C.11}$$

In Eq. (C.10) $\mathrm{N}_\tau = 500$ was selected and r is the radial distance.

Appendix D

Turbulence model in the Foam-extend IBM toolkit

In this Reynolds-Averaged Navier-Stokes (RANS) equations based k-ϵ turbulence model, the turbulent kinetic energy k and the dissipation of the turbulent kinetic energy ϵ are introduced in order to solve μ_t in Eq. (3.4). The governing equations for k and ϵ are

$$\rho k + \frac{\partial}{\partial x_{\mathrm{i}}}(\rho u_{\mathrm{i}} k) = \frac{\partial}{\partial x_{\mathrm{i}}}(\frac{\mu_t}{\sigma_k}\frac{\partial k}{\partial x_{\mathrm{i}}}) + \mu_t(\frac{\partial u_{\mathrm{i}}}{\partial x_{\mathrm{j}}} + \frac{\partial u_{\mathrm{j}}}{\partial x_{\mathrm{i}}})\frac{\partial u_{\mathrm{i}}}{\partial x_{\mathrm{j}}} - \rho\epsilon, \tag{D.1}$$

$$\rho\epsilon + \frac{\partial}{\partial x_{\mathrm{i}}}(\rho u_{\mathrm{i}} \epsilon) = \frac{\partial}{\partial x_{\mathrm{i}}}(\frac{\mu_t}{\sigma_\epsilon}\frac{\partial \epsilon}{\partial x_{\mathrm{i}}}) + C_{1\epsilon}\frac{\epsilon}{k}\mu_t(\frac{\partial u_{\mathrm{i}}}{\partial x_{\mathrm{j}}} + \frac{\partial u_{\mathrm{j}}}{\partial x_{\mathrm{i}}})\frac{\partial u_{\mathrm{i}}}{\partial x_{\mathrm{j}}} - C_{2\epsilon}\rho\frac{\epsilon^2}{k}. \tag{D.2}$$

μ_t is then calculated with

$$\mu_t = \rho C_\mu \frac{k^2}{\epsilon}. \tag{D.3}$$

There are five unknown variables to be solved in Eqs. (3.3), (3.4) and (D.1) to (D.3), namely the velocity u_i (i = x, y, z), density ρ, k, ϵ and μ_t, and the five constants $C_{1\epsilon} = 1.44$, $C_{2\epsilon} = 1.92$, $C_\mu = 0.09$, $\sigma_k = 1.0$ and $\sigma_\epsilon = 1.3$. The solving process for u_i and ρ has been given in Section 3.3.3. The steps within the turbulence correction function (Fig. 3.3) are solving Eqs. (D.2), (D.1) and (D.3) accordingly as shown in Fig. D.1.

The k-ϵ turbulence model with the Immersed Boundary Method (IBM) support is already available in Foam-extend 4.0, relying on a new wall function *immersedBoundaryVelocityWallFunction* implemented by Jasak et al. (2014) based on the regular k-ϵ turbulence model (without IBM support). In order to apply it in simulations of multiphase flows and moving immersed boundaries, some modifications need to be made. This is because the boundary condition of velocity in *immersedBoundaryVelocityWallFunction* involves a function correcting the velocity of the immersed boundary, making it differ from the no-slip boundary condition of the block. Further, the pressure force had some unphysical numerical oscillations due to the correction of the immersed boundary velocity. The generated iceberg-tsunamis were larger than in the laboratory under these

shortcomings. Therefore, the calculation of immersed boundary velocity in *immersedBoundaryVelocityWallFunction* was modified to adopt the same velocity obtained by the no-slip boundary condition (Eq. (3.50)) in this work.

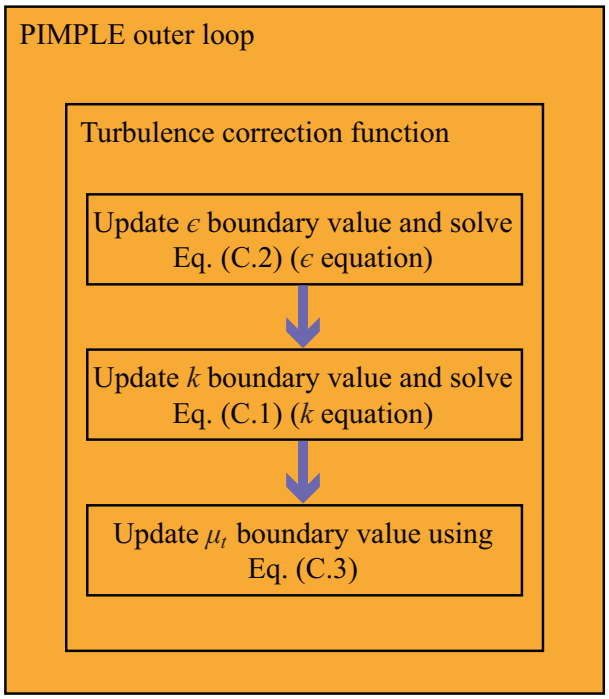


Figure D.1 Steps in the turbulence correction function