THE UNIVERSITY *of* EDINBURGH

High-Fidelity Computational Modelling of Fluid–Structure Interaction for Moored Floating Bodies

Tristan de Lataillade

A thesis presented for the degree of

Engineering Doctorate (EngD)

2019

Industrial Doctoral Centre for Offshore Renewable Energy

(IDCORE)

The University of Edinburgh

The University of Exeter

The University of Strathclyde

Abstract

The development and implementation process of a complete numerical framework for high-fidelity Fluid–Structure Interaction (FSI) simulations of moored floating bodies using Computational Fluid Dynamics (CFD) with the Finite Element Method (FEM) is presented here. For this purpose, the following three main aspects are coupled together: Two-Phase Flow (TPF), Multibody Dynamics (MBD), and mooring dynamics.

The fluid–structure problem is two-way and fully partitioned, allowing for high modularity of the coupling and computational efficiency. The Arbitrary Lagrangian–Eulerian (ALE) formulation is used for describing the motion of the mesh-conforming fluid–solid interface, and mesh deformation is achieved with linear elastostatics. Mooring dynamics is performed using gradient deficient Absolute Nodal Coordinate Formulation (ANCF) elements with a two-way mooring–structure coupling and a one-way fluid–mooring coupling. Hydrodynamic loads are applied accurately along mooring cables using the solution of the fluid velocity provided by the TPF solver. For this purpose, fluid mesh elements containing cable nodes that do not conform to the fluid mesh are located with a computationally efficient particle-localisation algorithm.

As it is common for partitioned FSI simulations of solids moving within a relatively dense fluid to experience unconditional instability from the added mass effect in CFD, a non-iterative stabilisation scheme is developed here. This is achieved with an accurate and dynamic estimation of the added mass for arbitrarily shaped structures that is then applied as a penalty term to the equations of motion of the solid. It is shown that this stabilisation scheme ensures stability of FSI simulations that are otherwise prone to strong added mass effect without affecting the expected response of structures significantly, even when using fully partitioned fluid–structure coupling schemes.

Thorough verification and validation for all aspects of the FSI framework ultimately show that the produced numerical results are in good agreement with experimental data and other inherently stable numerical models, even when complex nonlinear events occur such as vortices forming around sharp corners or extreme wave loads and overtopping on moving structures. It is also shown that the mooring dynamics model can successfully reproduce nonlinearities from high frequency fairlead motions and hydrodynamic loads. The large-scale 3D simulation of a floating semi-submersible structure moored with three catenary lines ties all the models and tools developed here together and shows the capability of the high-fidelity FSI framework to model complex systems robustly and accurately.

Declaration

I confirm that this thesis presented for the degree of Engineering Doctorate, has:

(i) been composed entirely by myself

(ii) been solely the result of my own work

(iii) not been submitted for any other degree or professional qualification

Tristan de Guilhem de Lataillade

Acknowledgements

First of all, I would like to thank my industrial supervisor, Dr Aggelos Dimakopoulos, for his guidance throughout my doctoral work. Aggelos has been very supportive during my time at HR Wallingford, and I benefited from his broad knowledge and experience in fluid dynamics. He always made himself available to provide insightful suggestions and scientific advice that have helped me greatly for the present research.

I am especially grateful to my academic supervisor, Prof Lars Johanning, who has been a great mentor to me, as well as a source of inspiration and guidance. His passion for his work is contagious: if mooring dynamics has become such an interesting topic to me as well, it is, without a doubt, thanks to Lars. My gratitude also goes to my other academic supervisors, Prof David Ingram and Dr Tahsin Tezdogan, who both gave me precious advice about CFD practices. Prof David Ingram is the director of IDCORE, and I take this opportunity to thank him and the IDCORE team again for creating and managing such a fantastic programme that has taught me so much.

I would also like to warmly thank Dr Christopher Kees — the father of the Proteus software and with whom I have worked closely throughout these years of research — for the transmission of invaluable knowledge and technical skills that will serve me for life. I greatly appreciate the recent opportunity that Chris gave me to work in the U.S. at ERDC, allowing me to carry on with research and development related to the work presented in this thesis (and beyond). I also thoroughly enjoyed the many occasions he created for working with other researchers internationally, and the Proteus developers workshops that have always been enlightening.

I would like to thank my friends from IDCORE, HR Wallingford, and the Proteus team who have made these years a rich and unforgettable experience. Among them, a special mention goes to Haydel Collins whose help allowed the heaviest numerical simulation of this thesis to run to completion on a (sometimes capricious) HPC.

The unconditional moral support of my family was very important to me as well during these years, and I feel lucky to be part of it. Finally, I would like to thank my wife, Isabella, for being such an amazing person, for believing in me more than myself in tougher times, for all her love, and for happily embarking on our (seemingly) never-ending international adventures.

Table of Contents

Abstract iii

Declaration v

Acknowledgements vii

Figures and Tables xiii

Acronyms xvii

Nomenclature xxi

I Introduction 1

I.1 Research Background and Needs . . . 1

I.2 Aims and Objectives . . . 2

I.3 Original Contributions . . . 4

I.4 Thesis Structure . . . 4

II General Background and Literature Review 7

II.1 Historical Context . . . 9

II.2 Numerical Modelling of Fluid Dynamics . . . 10

II.2.1 Potential Flow . . . 11

II.2.2 Computational Fluid Dynamics (CFD) . . . 12

II.3 Fluid–Structure Coupling . . . 14

II.3.1 Monolithic Approach . . . 14

II.3.2 Partitioned Approach . . . 15

II.4 Added Mass and Coupling Instabilities . . . 18

II.5 Moving Solids and Fluid Mesh . . . 19

II.5.1 Arbitrary Lagrangian–Eulerian (ALE) Mesh . . . 20

II.5.2 Mesh Adaptivity . . . 21

II.6 Moorings . . . 22

II.6.1 Design Considerations for Mooring Systems . . . 22

II.6.2 Numerical Modelling of Mooring Systems . . . 23

II.7 State of Research . . . 25

III Uncoupled Models: Governing Equations and Numerical Implementation 27

III.1 Two-Phase Flow . . . 29

III.1.1 Governing Equations of Fluid Flow . . . 29

III.1.2 Free Surface Representation and Tracking . . . 32

III.1.3 Finite Element Method (FEM) . . . 34

III.1.4 Typical Boundary Conditions . . . 38

III.2 Wave Theory . . . 39

III.2.1 Governing Equations . . . 39

III.2.2 Wave Generation and Absorption . . . 41
III.3 Rigid Body Dynamics . . . 42
III.4 Mooring Modelling . . . 44
III.4.1 Mooring Statics and Quasi-Statics . . . 44
III.4.2 Mooring Dynamics . . . 50
IV Coupling Strategy **55**
IV.1 General Workflow of Coupling . . . 57
IV.2 Fluid–Structure Coupling . . . 58
IV.2.1 Interface Coupling . . . 58
IV.2.2 Coupling Scheme . . . 60
IV.2.3 Added Mass Effect . . . 64
IV.3 Moving Domains . . . 69
IV.3.1 Arbitrary Lagrangian–Eulerian (ALE) Formulation . . . 69
IV.3.2 Mesh Deformation Approaches . . . 71
IV.3.3 Mesh Quality and Smoothing . . . 77
IV.4 Moorings Coupling . . . 79
IV.4.1 Mooring–Structure Coupling . . . 79
IV.4.2 Fluid–Mooring Coupling . . . 80
IV.4.3 Particle-Localisation Algorithm . . . 82
V Numerical Simulations **85**
V.1 Setup of a Typical Numerical Simulation . . . 87
V.1.1 Spatial Definition and Discretisation of the Numerical Domain . . . 87
V.1.2 Typical Initial Conditions for Two-Phase Flow Problem . . . 88
V.2 Two-Phase Flow . . . 90
V.2.1 Sloshing Wave . . . 90
V.2.2 Parallel Performance . . . 92
V.3 Stabilisation for Partitioned Fluid–Structure Coupling . . . 94
V.3.1 Numerical Assessment of the Added Mass Effect . . . 94
V.3.2 Added Mass Estimation for Arbitrarily Shaped Structures . . . 95
V.3.3 Non-Iterative Stabilisation Scheme . . . 97
V.4 Fluid–Structure Interaction (FSI) for Floating Bodies . . . 101
V.4.1 Free Decay (Heave) . . . 101
V.4.2 Free Decay (Roll) . . . 104
V.4.3 Response Under Regular Wave Loads . . . 106
V.4.4 Response Under Extreme (Focused) Wave Loads . . . 110
V.5 Moorings . . . 116
V.5.1 Statics Model Validation . . . 116
V.5.2 Static Convergence of Dynamic Model . . . 119
V.5.3 Mooring Dynamics Validation . . . 121
V.6 Moored Semi-Submersible Platform for Floating Wind Turbine . . . 126
V.6.1 Setup . . . 126
V.6.2 Uncoupled Results . . . 129
V.6.3 Fully Coupled Results . . . 131
VI Discussion **135**

VI.1 On Simulation Results and Numerical Challenges 135
VI.2 On Software Development 138
VI.3 On Limitations of the FSI Framework 140
VII Conclusion **143**
Published Work **145**
References **147**

Figures and Tables

List of Figures

I.1 Snapshot of proof-of-concept high-fidelity FSI simulation of a moored floating cylinder 2

II.1 Illustration of Reynolds-Averaged Navier–Stokes (RANS), Large Eddy Simulation (LES), Direct Numerical Simulation (DNS) principles on a velocity time series, assuming steady state flow 13
II.2 Illustration of Lagrangian ($\Omega_{\tilde{\mathbf{x}}}$), Eulerian ($\Omega_{\mathbf{x}}$), and ALE ($\Omega_{\boldsymbol{\xi}}$) domains in 1D 20

III.1 Illustration of arbitrary domain for two-phase flow and solid phase 27
III.2 Values taken by Volume of Fluid (VOF) and level set around implicit boundary 33
III.3 Lagrange elements for FEM 36
III.4 Piecewise-linear finite element basis function Φ_i on a triangular mesh around node $\mathcal{N}_i$ 37
III.5 Illustration of relaxation zone and blending function in typical numerical tank 42
III.6 State of body: position and rotation 43
III.7 Catenary shape for different values of a 45
III.8 Possible configurations for catenary line 47

IV.1 Diagram of numerical models workflow and coupling strategy 59
IV.2 Schematic representation of the Conventional Serial Staggered (CSS) scheme 61
IV.3 Schematic representation of the Improved Serial Staggered (ISS) scheme 63
IV.4 Illustration of the added mass effect encountered on partitioned coupling schemes when $m_a = 2m$ 65
IV.5 Mapping between spatial, material, and reference domains 70
IV.6 Snapshots (close-up) of mesh deformation around moving mesh-conforming boundaries 73
IV.7 snapshots of mesh transformation using target area monitoring function 76
IV.8 Quality metrics of a triangular mesh element when node $\mathbf{x}_{\mathcal{N}} = (1, 0)$ is moved from its ideal position 77
IV.9 Schematic representation of constraints and contact entities between mooring, structures, and seabed 79
IV.10 Triangular load applied on mooring cable elements 80
IV.11 Mesh search for $\mathbf{r}_s$ by boundary intersection with initial guess $\mathbf{r}_p^{(0)}$. 83

V.1 Comparison of constantly and gradually refined meshes with equivalent number of elements . . . 88
V.2 Schematic representation of domain for sloshing body of water in a fixed tank 90
V.3 Numerical results and analytical solution for the pressure on the left boundary at $x_3 = 0.5h_{\text{mwl}}$ with $\Delta t = 1.25^{-4}$second 91

V.4 Temporal convergence of the solution for the pressure from a sloshing wave on left boundary with constant refinement $h_{e0} = \frac{a}{10}$ . . . 91
V.5 Parallel efficiency of the TPF model on three mesh refinements . . . 93
V.6 Schematic representation of numerical domain for assessment of the added mass effect . . . 94
V.7 Vertical force acting on rectangular body of dimensions $1\text{m} \times 0.5\text{m}$ subject to added mass effect with $\rho_s = \rho_f$ and $\Delta t = 0.001\text{s}$ . . . 95
V.8 Envelope of vertical force acting on rectangular body subject to added mass effect with $\rho_s = \rho_f$ 96
V.9 Spatial convergence of added mass estimation model . . . 97
V.10 Pressure field for calculating added mass with different unit accelerations . . . 98
V.11 Error induced by added mass stabilisation scheme for different time discretisation levels (Δt), added mass overestimation factors (C_{am}), and mass (m) to actual added mass (m_a) ratio . . . 100
V.12 Schematic representation of numerical domain for free oscillation in heave of a floating cylinder . 101
V.13 Sensitivity analysis on heaving cylinder case: RMS errors from temporal and spatial discretisations and added mass stabilisation . . . 102
V.14 Time-series and sensitivity analysis of the free oscillation of a heaving cylinder . . . 103
V.15 Schematic representation of numerical domain for free roll oscillation of a floating caisson . . . 104
V.16 Free oscillation in roll of floating caisson . . . 106
V.17 Schematic representation of numerical domain for the simulation of wave-induced oscillation of rolling caisson . . . 107
V.18 Snapshots of wave-induced roll motion of floating caisson for $\frac{\omega}{\omega_n} = 0.77$ and $H = 0.06\text{m}$ . . . 108
V.19 Response Amplitude Operator (RAO) of rolling caisson under regular wave loads . . . 110
V.20 Schematic representation of numerical domain for the simulation of 2 Degrees of Freedom (DOFs) floating caisson under extreme wave loads . . . 111
V.21 Free surface elevation of focused wave over time at $x_1 = x_f$ and over space at time $t = t_f$ . . . 112
V.22 Response of floating body prone to added mass effect without stabilisation scheme . . . 113
V.23 Snapshots of floating body hit by focused wave ($t_f = 20\text{s}$) . . . 114
V.24 Response in heave and roll of floating body to focused wave loads . . . 115
V.25 Variation of the calculated added mass over time for floating body hit by focused wave (in global coordinates) . . . 116
V.26 Layout of mooring system and WEC for statics model validation . . . 117
V.27 Mooring static results for different surge offsets of a Wave Energy Converter (WEC) with three multi-segmented catenary lines . . . 118
V.28 Original and final position of cable for spatial convergence test of catenary position at equilibrium 119
V.29 Spatial sensibility analysis of cable position at equilibrium . . . 120
V.30 Schematic representation of the numerical setup for mooring dynamic validation case . . . 121
V.31 Mooring statics analysis: experimental, static, and dynamic models comparison . . . 122
V.32 Spatial sensitivity on cable dynamics with $\Delta t = 1 \times 10^{-4}\text{s}$ on case 16 . . . 123
V.33 Indicator diagrams of horizontal tension at fairlead . . . 124
V.34 Energy loss as a function of pretension . . . 125
V.35 Snapshot of rendered numerical domain for coupled simulation of OC4-DeepCwind platform . . . 126
V.36 Geometry of OC4-DeepCwind semi-submersible . . . 126

V.37 Schematic representation of numerical domain (top-down view) . . . 127
V.38 Tensions at fairleads for uncoupled OC4-DeepCwind semi-submersible simulation . . . 129
V.39 Spectrum of tensions at fairleads for uncoupled semi-submersible validation case (over 40 periods) 130
V.40 Response of OC4-DeepCwind semi-submersible platform to wave loads in fully coupled simulation 131
V.41 Tensions at fairleads of OC4-DeepCwind semi-submersible platform in fully coupled simulation . 132
V.42 Velocity along upstream mooring line (i.e. line 1) of the OC4-DeepCwind platform under regular wave loads with $H = 206.08\text{mm}$ and period $T = 1.71\text{s}$, for one complete oscillation . . . 133

List of Tables

III.1 Values of variables to recover Navier–Stokes Equations (NSEs) from the Convection-Diffusion-Reaction (CDR) equation eq. (III.6) . . . 32
III.2 Set of typical boundary conditions used in TPF simulations presented in this thesis . . . 38
III.3 Values of tensions at anchor and fairlead for static analysis of catenary cables . . . 48

IV.1 Unit acceleration for boundary conditions at the fluid–structure boundary of added mass model . 68

V.1 Guide for numerical simulations of verification and validation cases showing enabled models . . 85
V.2 Physical constants used in numerical simulations . . . 89
V.3 Characteristics of different systems used for parallel simulations . . . 93
V.4 Period of oscillation (in seconds) for rolling caisson (peak-to-peak values) . . . 106
V.5 Wave characteristics and resulting roll response for computing Response Amplitude Operator (RAO) of rolling caisson . . . 109
V.6 Characteristics of mooring lines for [65] test case . . . 117
V.7 Characteristics of lines for mooring dynamics validation case . . . 121
V.8 Parameters and results for mooring line damping test cases . . . 122
V.9 OC4-DeepCwind semi-submersible platform characteristics. . . . 127
V.10 Mooring system characteristics for the OC4-DeepCwind semi-submersible platform . . . 128

Acronyms

ALE	Arbitrary Lagrangian–Eulerian.
ALS	Accidental Limit State.
ANCF	Absolute Nodal Coordinate Formulation.
BGS	Block Gauss–Seidel.
BVP	Boundary Value Problem.
CDR	Convection-Diffusion-Reaction.
CFD	Computational Fluid Dynamics.
CFL	Courant–Friedrichs–Lewy.
CPS	Conventional Parallel Staggered.
CSS	Conventional Serial Staggered.
DEM	Discrete Element Method.
DNS	Direct Numerical Simulation.
DNV GL	Det Norske Veritas Germanischer Lloyd.
DOF	Degree of Freedom.
ERDC	Engineer Research and Development Center.
FDM	Finite Difference Method.
FEM	Finite Element Method.
FLS	Fatigue Limit State.
FSI	Fluid–Structure Interaction.
FVM	Finite Volume Method.
GSS	Generalised Serial Staggered.
HPC	High Performance Computing.

IBM	Immersed Boundary Method.
IMR	Ideal Weight Inverse Mean Ratio.
IPS	Improved Parallel Staggered.
ISS	Improved Serial Staggered.
IVP	Initial Value Problem.
JONSWAP	Joint North Sea Wave Project.
LES	Large Eddy Simulation.
LHS	Left Hand Side.
LS	Level Set.
MBD	Multibody Dynamics.
MPF	Multiphase Flow.
NSEs	Navier–Stokes Equations.
OOM	Out of Memory.
OREC	Offshore Renewable Energy Converter.
OWC	Oscillating Water Column.
OWT	Offshore Wind Turbine.
PCC	Pearson Correlation Coefficient.
PDE	Partial Differential Equation.
PIC	Particle-In-Cell.
RANS	Reynolds-Averaged Navier–Stokes.
RAO	Response Amplitude Operator.
RBF	Radial Basis Function.
RHS	Right Hand Side.
RMS	Root Mean Square.
RTT	Reynolds Transport Theorem.
SLS	Serviceability Limit State.
SPF	Single-Phase Flow.
SPH	Smoothed-Particle Hydrodynamics.

TEC	Tidal Energy Converter.
TLP	Tension-Leg Platform.
TPF	Two-Phase Flow.
ULS	Ultimate Limit State.
USACE	U.S. Army Corps of Engineers.
VOF	Volume of Fluid.
WEC	Wave Energy Converter.
WT	Wind Turbine.

Nomenclature

Operators and Notation Conventions

∇a	Gradient of a, $\nabla a = \frac{\partial a}{\partial \mathbf{x}}$
$\nabla \cdot a$	Divergence of a
Δa	Laplace operator, $\Delta a = \nabla \cdot \nabla a$
$\frac{\partial a}{\partial b}$	Partial derivative of a in regards to b
$\frac{\mathrm{d}a}{\mathrm{d}b}$	Total derivative of a in regards to b
$\dot{a}$	Time derivative of a, $\dot{a} = \frac{\partial a}{\partial t}$
$\ddot{a}$	Second time derivative of a, $\ddot{a} = \frac{\partial^2 a}{\partial t^2}$
$\int_c a \mathrm{d}b$	Integral of a in regards to b over c
$\sum_i^j a$	Sum of a from i to j
$\|a\|$	Norm of a
$\|a\|_F$	Frobenius norm of a
$a\Big\|_b$	Evaluation of a at b (holding b fixed)
$a \cdot b$	Inner product between a and b
$a \times b$	Cross product between a and b
$\mathbf{A}$	Matrix notation (bold capital)
$\bar{\bar{\mathbf{a}}}$	Tensor notation
$\mathbf{a}$	Vector notation (bold)
$\vec{a}$	Alternative vector notation
$a \otimes b$	Outer product between a and b

Domain, Mesh, and Spatial Variables

Ω	Domain
Ω_h	Discretised domain
$\partial\Omega$	Boundary of domain
$\Gamma_{\mathrm{f}\cap\mathrm{s}}$	Fluid-structure boundary
Γ	Partial boundary of domain, $\Gamma \in \partial\Omega$
$\Gamma_\sim$	Free surface
$\Omega_\mathbf{x}$	Spatial domain
$\Omega_{\widehat{\mathbf{x}}}$	Material domain
$\Omega_{\boldsymbol{\xi}}$	Reference domain (ALE)
Ω_f	Fluid domain ($\Omega_f \in \Omega$, $\Omega_f = \Omega_a \cup \Omega_w$)
Ω_w	Water domain ($\Omega_w \in \Omega$)
Ω_a	Air domain ($\Omega_a \in \Omega$)
$\mathbf{x}$	Spatial coordinates, $\mathbf{x} = \begin{pmatrix} x_1 & x_2 & x_3 \end{pmatrix}^\mathsf{T}$
$\widehat{\mathbf{x}}$	Material coordinates ($\widehat{\mathbf{x}} \in \Omega_{\widehat{\mathbf{x}}}$)
$\boldsymbol{\xi}$	Reference coordinates ($\boldsymbol{\xi} \in \Omega_{\boldsymbol{\xi}}$)
$\Gamma^\mathcal{E}$	Mesh element boundary
$\mathcal{E}$	Mesh element
$\{\Gamma^\mathcal{E}_\mathcal{E}\}_i$	Set of boundaries of element $\mathcal{E}_i$
$\{\mathcal{N}_\mathcal{E}\}_i$	Set of nodes of element $\mathcal{E}_i$
$\mathcal{M}$	Mesh
$n_\mathcal{E}$	Number of elements

$n_{\mathcal{N}}$	Number of nodes
$\mathcal{N}$	Mesh node
$\{\mathcal{E}_{\mathcal{N}}\}_i$	Set of elements around $\mathcal{N}_i$
$\{\mathcal{N}^\star\}_i$	Set of nodes around $\mathcal{N}_i$
V	Volume
Φ	Mapping from $\Omega_{\boldsymbol{\xi}}$ to $\Omega_{\mathbf{x}}$
$\widehat{\Phi}$	Mapping from $\Omega_{\boldsymbol{\xi}}$ to $\Omega_{\widehat{\mathbf{x}}}$
Ψ	Mapping from $\Omega_{\widehat{\mathbf{x}}}$ to $\Omega_{\mathbf{x}}$

Fluid Dynamics

$\bar{\bar{\boldsymbol{\sigma}}}$	Cauchy stress tensor
CFL	Courant–Friedrichs–Lewy (CFL) value
η	Free surface elevation
$\mathbf{f}$	Flux
$\bar{\bar{\mathbf{I}}}$	Identity tensor
ϕ_{sdf}	Signed distance function
$\bar{\bar{\epsilon}}$	Strain rate tensor
$\bar{\bar{\boldsymbol{\tau}}}$	Viscous shear stress tensor
$\mathbf{u}_{\boldsymbol{\xi}}$	Velocity of mesh
$\widehat{\theta}$	VOF
$\mathbf{u}$	Velocity of fluid $\mathbf{u} = \begin{pmatrix} u_1 & u_2 & u_3 \end{pmatrix}^{\mathsf{T}}$
$\mathbf{u}_f$	Velocity of fluid $\mathbf{u}_f = \mathbf{u}$
ϕ	Arbitrary fluid property (intensive)
c	Arbitrary fluid property (extensive)
$\mathbf{f}_\phi$	Flux of ϕ
R_ϕ	Source of ϕ
$\dot{\mathbf{u}}$	Acceleration of fluid
p	Pressure
ρ	Density
ρ_f	Fluid density
ρ_a	Air density
ρ_f	Fluid density
ρ_w	Water density
μ	Dynamic viscosity
ν	Kinematic viscosity
D	Diffusion coefficient
h_{mwl}	Mean water level
Φ	Velocity potential
a	Wave amplitude
ω	Wave angular frequency
η	Wave elevation
H	Wave height
k	Wavenumber
ϕ	Wave phase

Body Dynamics

$\mathbf{s}$	State vector, $\mathbf{s} = \begin{pmatrix} \mathbf{r} & \Theta \end{pmatrix}^{\mathsf{T}}$
$\mathbf{r}$	Position of barycentre $\mathbf{r} = \begin{pmatrix} r_1 & r_2 & r_3 \end{pmatrix}^{\mathsf{T}}$
$\dot{\mathbf{r}}$	Velocity of barycentre
$\ddot{\mathbf{r}}$	Acceleration of barycentre
X	Displacement in surge
Y	Displacement in sway
Z	Displacement in heave
Θ	Rotational vector $\Theta = \begin{pmatrix} \theta & \phi & \Psi \end{pmatrix}^{\mathsf{T}}$
θ	Pitch
ϕ	Roll

Ψ	Yaw
$\boldsymbol{\omega}$	Angular velocity
$\dot{\boldsymbol{\omega}}$	Angular acceleration
m	Mass
$\mathbf{f}$	Generalised force vector $\mathbf{f} = \begin{pmatrix} \vec{f} & \vec{m} \end{pmatrix}^{\mathsf{T}}$
$\vec{f}$	3×1 force vector
$\vec{m}$	3×1 moment vector
$\bar{\bar{\mathbf{I}}}_t$	Inertia tensor
$\mathbf{A}$	Added mass matrix (estimate)
$\widehat{\mathbf{A}}$	Added mass matrix (true value)
$\mathbf{f}_f$	Hydrodynamic force
$\widehat{\mathbf{f}}_f$	Hydrodynamic force without added mass
$\mathbf{M}$	Mass matrix
C_d	Damping coefficient

Moorings

C_m	Added mass coefficient
$\dot{\mathbf{u}}_r$	Fluid acceleration relative to cable
C_d	Drag coefficient
a	Catenary shape variable
A_0	Unstretched cross-sectional area
d	Horizontal anchor-fairlead distance
d_0	Unstretched diameter of line
e	Stretch of line
EA_0	Axial stiffness
$\mathbf{f}_m$	Inertia force
$\mathbf{f}_b$	Buoyancy force
$\mathbf{f}_c$	Collision force
$\mathbf{f}_d$	Drag force
$\mathbf{f}_e$	External force
h	Vertical anchor-fairlead distance
L	Line length
L_s	Lifted line length
ρ_c	Density of cable
s	Distance along cable
w_0	Submerged weight
x_0	Horizontal span of line
$n_{\mathcal{E}}$	Number of elements
$n_{\mathcal{N}}$	Number of nodes
$\mathbf{T}$	Tension
T_A	Axial tension
T_H	Horizontal tension
T_V	Vertical tension
$\mathbf{u}_r$	Fluid velocity relative to cable

Functions and Function Spaces

Φ	Basis for trial function
Ψ	Basis for test function
$\mathscr{H}$	Hilbert function space
J	Jacobian
$\mathscr{L}$	Lebesgue function space
$\mathbb{R}$	Real function space
$\mathscr{R}$	Residual
$\mathscr{H}$	Sobolev function space
φ	Solution / trial function
φ_h	Discrete solution / trial funtion

w	Test function	**General Variables**	
$\mathscr{W}$	Space of admissible test functions	Δt	Time step
$\mathscr{W}_h$	Space of admissible discrete test functions	ϵ	Error
$\mathscr{V}$	Space of admissible trial functions	$\mathbf{g}$	Gravitational acceleration
$\mathscr{V}_h$	Space of admissible discrete trial functions	$\bar{\bar{\mathbf{I}}}$	Identity tensor
θ	Heaviside function	ϵ_{tol}	Tolerance
θ_ϵ	Smooth heaviside function	t	Time

CHAPTER

Introduction

I.1 Research Background and Needs

High-fidelity Fluid–Structure Interaction (FSI) models using Computational Fluid Dynamics (CFD) are becoming increasingly viable commercially, largely due to rapid developments in computational science that have led to a sharp and steady increase in available computational power at a greatly reduced cost during the past decades. Within this context, the research presented here describes the development, implementation, and testing of a complete numerical framework for high-fidelity simulations of moored floating structure. This implies an accurate representation of all relevant FSI engineering aspects: Multiphase Flow (MPF) simulation for air and water phases, Multibody Dynamics (MBD) for floating bodies and their mooring systems, and a stable and robust coupling approach.

When assessing the response of floating structures to environmental loads, physical testing of prototype scale models is the ideal way to gather data and assess the feasibility of the considered application, as it produces the actual real-world response. However, doing so often requires enormous investments from the earliest stages of device development due to the iterations in design that are necessary to ensure that the tested structure fits the requirements. Scaled physical models can be a good alternative but remain costly as potentially complex construction of the model and tank testing are still required, and it can prove impossible to reproduce the targeted environmental conditions in a flume due to scaling limitations. This is where numerical modelling becomes an interesting alternative, as it allows the simulation of fluid flow and its effect on arbitrary structures of any design, in any environmental conditions, at any scale, and at a much lower cost. Drawbacks of numerical models should however not be ignored: they are usually based on simplifying assumptions and provide an approximate solution with an accuracy depending on the spatial and temporal refinement level of the simulation, as well as other numerical inputs and errors. When compared to simpler, more linear numerical models, high-fidelity models tend to describe highly nonlinear processes and aim at producing numerical approximations that are as close as possible to the real-world equivalent of the numerical simulations.

Numerous mainstream numerical models exist for specific applications, such as fluid dynamics, solid mechanics, or mooring dynamics, but rarely are they coupled together to combine several of these applications robustly and accurately. Furthermore, existing software combining fluid, body, and mooring dynamics might still fail to fit the desired requirements for specific aspects of the coupled problem, and many questions can arise from their potential

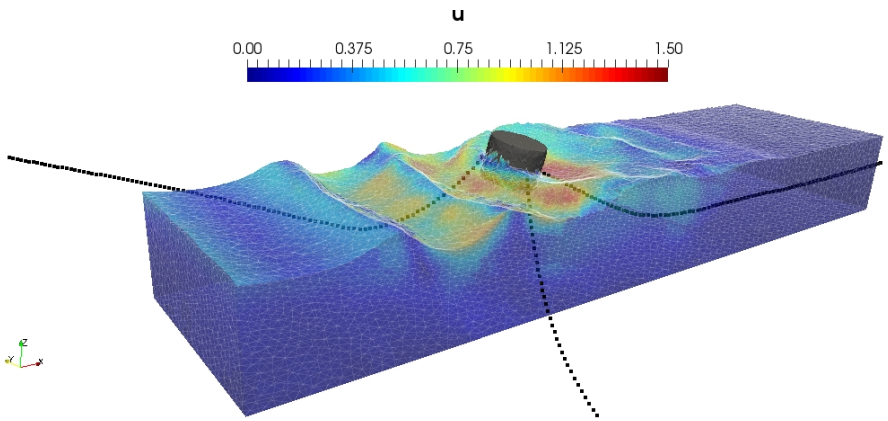


Figure I.1: Snapshot of proof-of-concept high-fidelity FSI simulation of a moored floating cylinder (as originally published in [87])

lack of modularity. Is the CFD software using Finite Difference Method (FDM), Finite Volume Method (FVM), or Finite Element Method (FEM)? Can it deal with moving boundaries using the Arbitrary Lagrangian–Eulerian (ALE) approach? Is the MBD solver using the desired time integration scheme? Does it include collision detection? Is the mooring dynamics solver using a quasi-statics, lumped mass, or FEM approach? Does it take elasticity or torsion into account? This non-exhaustive list of potential questions can be extended according to the requirements and complexity of the model. Therefore, creating an FSI framework and choosing the relevant approaches must be done carefully, and the framework must ensure modularity to some extent, in case a certain aspect of the coupling must be replaced or improved upon in order to better fit the specific needs of the numerical application considered. This can become particularly constraining when using commercial software as it usually only grants limited access to the source code (if at all), does not allow any modification of code, and implies a monetary cost that can sometimes be prohibitive. It then becomes difficult, if not impossible, to couple software that is highly specialised in one of the aspects of the problem with software that is highly specialised in another aspect. For this reason, and in the context of this thesis, all software that has been developed and used is entirely open-source, granting free access to anybody for usage and modification of the code, as well as ensuring reproducibility of the simulations presented.

I.2 Aims and Objectives

The research presented here focuses entirely on the development of high-fidelity numerical simulations of moored floating offshore structures under various environmental conditions. The following main aims are therefore addressed:

1. Ensure that the models using incompressible Two-Phase Flow (TPF) simulations (fluid flow, free surface tracking) are robust and reliable under the right numerical parameters;
2. Select and implement techniques for wave generation and absorption in numerical wave tanks in order to generate the intended waves and eliminate any spurious or reflected waves;
3. Ensure that nonlinearities from the fluid flow, floating structures, and mooring dynamics are well captured;
4. Implement mesh motion techniques for allowing the fluid–structure interface to be displaced within the fluid mesh;

5. Establish and implement an efficient and robust strategy for the coupling of MPF and MBD;
6. Eliminate instabilities of the selected coupling schemes such as the added mass effect;
7. Assess the robustness and reliability of the coupling for the simulation of FSI under typical and extreme environmental conditions;
8. Increase computational efficiency where possible and necessary;
9. Release all work as open-source software for the purposes of research reproducibility and further development.

The main selected numerical techniques for the work presented in this thesis are:

- FEM for CFD (e.g. fluid flow, free surface tracking), including auxiliary models featuring Partial Differential Equations (PDEs) (e.g. fluid mesh motion, added mass estimation);
- Discrete Element Method (DEM) capable MBD solver for structures, to allow for collision detection;
- Statics and quasi-statics model solving the catenary equations for estimating moorings tensions at equilibrium;
- FEM method for mooring dynamics using gradient deficient Absolute Nodal Coordinate Formulation (ANCF) beam theory.

This list of objectives requires careful verification and validation of all aspects of the FSI framework, starting with each of the stand-alone models and gradually coupling them together. With all objectives listed above fulfilled, the end product is a robust and efficient high-fidelity simulation tool for moored floating structures. A snapshot of an early proof-of-concept simulation produced with the framework developed in this thesis is shown in fig. I.1.

As mentioned earlier, an important aspect of the coupled framework to be implemented is its open-source nature as it grants direct access to the source code which, in our case, allows for modifications and additions to the code developed specifically for the needs of the research presented here. Any additions to the code are shared with the scientific community which, in turn, can provide their own modifications that can be beneficial for the purpose of this research project and beyond. For these reasons, the main software that has been selected, developed, and used for designing and implementing the numerical solutions in this thesis are:

Proteus: an open-source computational simulation toolkit developed by the Engineer Research and Development Center (ERDC) of the U.S. Army Corps of Engineers (USACE) and HR Wallingford. This toolkit can be technically used to solve any PDE using FEM, and is specifically used for PDEs related to MPF and FSI here, such as the Navier–Stokes Equations (NSEs), free surface tracking, ALE mesh, and added mass estimation. The source code is available online at:

`https://github.com/erdc/proteus`.

Chrono: an open-source multi-physics simulation engine developed as a community project led by the University of Wisconsin–Madison and the University of Parma-Italy. It is used here for its MBD capabilities as well as fully coupled FEM cable dynamics with gradient deficient ANCF formulation and collision detection through DEM. The source code is available online at:

`https://github.com/projectchrono/chrono`.

Most additions and modifications done by the author for the work presented in this thesis can be accessed and reviewed on the online repository of Proteus (see link stated above), which also includes modifications of the Chrono source code that were necessary for fluid–structure coupling. Note that all models, concepts, and algorithms de-

veloped for this research are described in this thesis in a language-agnostic manner to remain applicable to any programming language or other software libraries and frameworks.

I.3 Original Contributions

The FSI framework has been entirely developed for the work presented in this thesis, making it by definition, a novel contribution. All coupling aspects between fluid, structure, and moorings were selected, adapted, and implemented to answer the needs mentioned above, and the needs of industrial partners: HR Wallingford and the U.S. Army ERDC. When coupled together, these models and tools make for a unique and novel coupling strategy.

Other novelties include a non-iterative added mass effect stabilisation scheme for fully partitioned and explicitly coupled FSI problems that uses a dynamically estimated value of the added mass. This scheme is proven to be able to provide numerical results at an accuracy that is comparable to inherently stable (i.e. monolithic) schemes for various relevant engineering cases presenting in this thesis (e.g. floating body experiencing regular or extreme waves loads).

In terms of mooring dynamics, gradient deficient ANCF elements have been applied for the first time in the context of modelling mooring dynamics of floating structures with hydrodynamics from CFD, to the author's knowledge. An additional novel contribution is the use of different particle localisation algorithms to retrieve the fluid velocity solution from the fluid mesh in order to accurately apply drag and inertia loads along mooring lines that are non-conforming to the fluid mesh.

All the models developed for this work and the aforementioned contributions are used altogether in a large 3D FSI simulation of a moored floating platform for offshore wind turbines under nonlinear wave loads, proving their efficiency and accuracy for realistic engineering cases.

I.4 Thesis Structure

Following this introduction, the thesis contains 6 subsequent chapters, which are briefly described here.

General background and literature review in chapter II describe the main approaches that can be used in the field of FSI for moored floating structures, as well as the various challenges, solutions, and uncertainties ensuing from it. This chapter is closed with a brief review of existing coupled models for moored floating bodies and the needs for a novel tool as developed for this thesis.

The equations governing the physics and numerics behind the uncoupled models of the FSI framework developed here are laid out in chapter III, along with the numerical approaches implemented to solve it. These include the equations of incompressible fluid dynamics, the free surface tracking method, wave theory, the equations of motion for rigid body dynamics, and mooring statics (catenary equation) and dynamics (beam theory). Additionally, the general procedure to solve a generic transport equation with FEM is shown, along with the main boundary conditions that are used in the TPF simulations produced for this research.

While the chapter described above focuses on the numerical background and methods used for the uncoupled models, chapter IV gives an overview of the coupling strategy and the resulting tools that have been developed for the FSI framework. This includes the implementation of appropriate fluid–solid coupling schemes, mesh deformation

models for moving solid structures within the fluid mesh, a non-iterative stabilisation scheme for the added mass effect, and the mooring–solid and mooring–fluid coupling.

In chapter V, various numerical FSI simulations are introduced and their results analysed. This chapter focuses on the verification and validation of the various aspects of the FSI framework with the following simulations: TPF with sloshing wave in enclosed tank, capabilities and limitations of the fluid-structure stabilisation scheme, FSI for unmoored floating bodies under free oscillation as well as regular and extreme wave-induced oscillation, verification and validation of the mooring models (statics, quasi-statics, and dynamics), and the simulation of a of a moored floating semi-submersible structure for Offshore Wind Turbine (OWT) under nonlinear wave loads.

In chapter VI, discussions of the various findings, limitations, and further work of the research are summarised, along with general remarks about the development of this high-fidelity framework. Finally, a global conclusion closes the thesis in chapter VII.

CHAPTER II

General Background and Literature Review

The purpose of this chapter is to establish an overview of the known background and main challenges, and to present the main different approaches possible for Fluid–Structure Interaction (FSI) simulations of moored offshore floating structures. The different numerical tools and techniques that have been selected, implemented and used for the research presented in this thesis are described in more details in the subsequent chapters.

The chapter starts with section II.1 by introducing historical context of real-world applications of moored floating offshore structures and the current state of the research relevant to the work presented herein. Subsequently, the various challenges of floating structures and their mooring systems are discussed, with an emphasis on smaller-scale devices such as Offshore Renewable Energy Converters (ORECs) where dynamic effects can be more prevalent. The aim is to provide a deeper understanding of these real-world challenges for the selection and development of appropriate tools to be used for high-fidelity simulations. In section II.2, the main approaches for simulating fluid dynamics are described and their main advantages and drawbacks are identified. The various existing coupling schemes for FSI are then explained in details in section II.3. The instability known as the added mass effect caused by partitioned coupling schemes is described in section II.4, along with the existing solutions to tackle this problem. Section II.5 deals with solutions for including moving solids in the fluid domain. Section II.6 gives an overview of general mooring design considerations and of the different techniques to model them numerically. Finally, section II.7 presents some of the recent developments in coupled models for the numerical simulation of moored floating bodies, and highlights some of their difference with the FSI framework developed for this research.

II.1 Historical Context . . . 9
II.2 Numerical Modelling of Fluid Dynamics . . . 10
II.2.1 Potential Flow . . . 11
II.2.2 Computational Fluid Dynamics (CFD) . . . 12
II.3 Fluid–Structure Coupling . . . 14
II.3.1 Monolithic Approach . . . 14
II.3.2 Partitioned Approach . . . 15
II.3.2.1 Implicit Schemes . . . 15
II.3.2.2 Semi-Implicit Schemes . . . 16
II.3.2.3 Explicit Schemes . . . 17
II.4 Added Mass and Coupling Instabilities . . . 18
II.5 Moving Solids and Fluid Mesh . . . 19
II.5.1 Arbitrary Lagrangian–Eulerian (ALE) Mesh . . . 20
II.5.2 Mesh Adaptivity . . . 21
II.6 Moorings . . . 22
II.6.1 Design Considerations for Mooring Systems . . . 22
II.6.2 Numerical Modelling of Mooring Systems . . . 23
II.7 State of Research . . . 25

II.1 Historical Context

Research on numerical models for floating structures and their mooring systems originated mainly from studies related to large vessels and oil & gas platforms. The recent research effort in ORECs for alternative energy production has however led towards a growing interest in numerical simulations for smaller-scale structures. These new applications include floating Wave Energy Converters (WECs), Tidal Energy Converters (TECs), and Offshore Wind Turbines (OWTs), all being much smaller-scale than more traditional offshore structures and usually deployed in sites with harsh environmental conditions for energy conversion purposes. While the applications and sites can be very different, some of the experience gained from the research effort towards larger structures can be used for these more recent, smaller-scale structures. Undeniably, many of the traditional tools must be adapted or replaced to respond to the new challenges that are brought by the numerical modelling of structures such as ORECs. Note that smaller structures exist in the oil & gas industry (e.g. loading buoys) and that smaller scale vessels also face similar challenges to ORECs.

Before comparing the main differences between the traditional and more recent type of offshore floating structures in terms of physical applications and numerical simulations, historical context concerning the different types of ORECs can help to highlight their differences. Arguably the ancestors to Wind Turbines (WTs), windmills were already used in antiquity to convert wind power into rotational energy for grain grinding and water pumping. The first onshore WT producing electricity was however developed in 1887 in Scotland by James Blyth. Modern commercial designs were developed in the second half of the 20th century and, for the first time three decades ago, WTs dared going beyond the shoreline in Denmark [9] on monopiles. Since then, OWTs have started reaching deeper water depths as a result of the recent developments on moored floating platforms with the first commercial floating wind farm delivering electricity to the Scottish grid in 2017. The first watermills – arguably the ancestor of TECs – were also developed during antiquity for similar applications as windmills. Consequently, they bear a similar design to windmills, but they use hydropower instead of wind power. Modern TECs use tidal currents directly as a means to produce electricity (not to be confused with tidal dams taking advantage of the tidal range) and are designed similarly to modern OWTs with horizontal and vertical axis turbines. They are, however, still mostly in the demonstration phase with the first small farms currently being deployed. Concerning wave energy, interest and first attempts at converting it can be traced as far back as 1799, with a patent deposited by Girard in France stating “Mechanical means of taking advantage of the rising and falling motion of sea waves, as driving forces” (translated here from the French: “Moyens mécaniques de tirer parti de l’ascension et de l’abaissement des vagues de la mer, comme forces motrices”) [100], but it was only in the past few decades that interest for such devices and research related to it started to become widespread (with a thousand patented WECs already by 2002 [27]). As of today, the vast majority of WECs remain in the conceptual or early demonstration phase. Several prototype scale devices such as the attenuator Pelamis was deployed on several sites but was ultimately decommissioned after Pelamis Wave Power went bankrupt due to a lack of investors. It is clear that all of these ORECs are still in their infancy when it comes to offshore and particularly moored floating structure design, even though some of them are based on well-established historical designs.

More traditional moored structures – vessels and oil & gas platforms – are designed so that their response

modes are very different from the expected wave excitation frequencies of their deployment site. This allows for the unavoidable motion resonances to happened away from the most energetic part of the wave spectra. They are also ideally stationed in calm waters, although this is not always a possibility (e.g. several deployment sites in Norway or West of Africa). Traditionally, and due to the limited response of the structures to waves, it was considered sufficiently appropriate to use linear models for structure hydrodynamics and quasi-static models to estimate the tensions in mooring systems. This was usually justified by the fact that nonlinear effects of the fluid around the structure and dynamic effects along the line could be assumed to be negligible or to be taken into account with safety factors. Additional reasons for using those simple numerical models were their low computational cost, and ease of implementation relative to their nonlinear and dynamic counterparts. It was however shown recently that, when placed in deep water, nonlinear and dynamic effects affecting the response of the floating structure can be significant, with up to 80% of the total damping induced by the mooring lines themselves [73]. This has led to tightened requirements in terms of mooring system design, which are reflected in the iterations of guidelines for offshore standards from Det Norske Veritas Germanischer Lloyd (DNV GL). In particular, the DNVGL-OS-E301 guidelines about position mooring used to state that a quasi-static approach was appropriate for water depths up to 200 meters in 2004 [32], reducing it to a maximum 100 meters in 2010 [33], and now states that quasi-statics is only appropriate for mean or low-frequency displacements of a floating structure and that a dynamic approach is necessary for wave-induced motion, unless it is demonstrated that dynamic effects are negligible [38, 39]. This evolution can be explained by the increasing computational power available at a lower cost over the years, making complex simulations more practical and accessible, as well as the development of ORECs requiring dynamic analysis even in deeper water. For example, many WECs are designed for the resonance of the device to occur at the expected wave excitation frequencies in order to convert as much energy as possible (e.g. attenuators and point absorbers). In this case, dynamic effects are prominent and the mooring system must be carefully considered as it can significantly affect the response of the overall system. This applies to all structures placed in shallower waters, including oil & gas. The mooring system bears a cost not to be ignored, but it should ideally remain a relatively low percentage of the overall cost of the system. For the interested reader, [66] gives an overview of the importance of the mooring system for different types of WECs.

II.2 Numerical Modelling of Fluid Dynamics

The flow of real-world fluids is complex and can be unpredictable. As no analytical solution exist to describe fluid flow realistically, numerical approximations are used through numerous methods with vastly different approaches. Some numerical models are more accurate than others depending on the assumptions made by the chosen method and, when it comes to FSI applications, the numerical approach for the simulation of fluid flow must be chosen according to the application intended, the computational resources available, and the level of approximations of the approach itself. A brief review of the main techniques is presented in this section, highlighting the advantages and drawbacks of each method. For the purposes of this work, we assume that all fluids are incompressible. Note that compressibility effects may become significant when air entrainment or aeration processes are important in the context of FSI applications (e.g. compressibility effects during extreme loads [30] or in the context of Oscillating

Water Column (OWC) devices [36]). Again, such cases are not covered here.

II.2.1 Potential Flow

Potential flow theory uses Euler's equation [42] to describe fluids that are irrotational and inviscid. Irrotationality implies that there is no vorticity in the fluid, while inviscidity implies that there is no viscous damping. As this ideal fluid is not encountered in reality, John von Neumann coined the term "dry water" for such description of the flow [52]. Indeed, these assumptions make the numerical fluid act in a very different manner to a real fluid. For this reason, in FSI applications, it is common to apply empirical terms to the equations of motion of the structures in order to account for the shortcomings of the potential flow method and to recover a semblance of the expected response, such as viscous damping terms (e.g. in [85]). The main issue with adding a viscous damping coefficient is that its value must be calibrated from experimental data, defeating the purpose of reliably using a numerical approach without having to conduct expensive physical experiments specifically designed for the numerical problem to investigate. Therefore, using empirical viscous damping coefficients can provide satisfactory accuracy if physical data for similar devices and under similar environmental conditions exist, but it becomes unreliable when analysing a novel device design or new environmental conditions because damping coefficients would then need to be chosen in a relatively arbitrary manner. Furthermore, the risk of overscaling or underscaling the empirical coefficients in order to best fit the physical experiment data can introduce biases on top of the ones that are already present in the physical model (e.g. scaling effects). Another major limitation of the potential flow approach is the inability to model breaking waves and green water events such as overtopping [40].

For FSI applications and from a general engineering point of view, potential flow is therefore mostly appropriate for the simulation of structures experiencing small and slow displacements under wave loads, where the viscous forces due to flow separation occurring around sharp corners are small enough to assume that they are insignificant, and where overtopping events do not occur. Typical applications fitting these constraints are the simulations of large offshore structures such as oil & gas platforms placed in calm waters. This does not apply to smaller-scale structures such as ORECs that experience larger and faster motion under wave loads as they are typically placed in areas where environmental loads can be high, leading to significant viscous effects and overtopping. Using an inviscid and irrotational fluid can therefore lead to widely different behaviour between a numerically simulated device using potential flow and its real-world counterpart interacting with a real fluid, depending on the application.

The advantages of potential flow over the other more complex numerical methods are its computational efficiency and its minimal numerical dissipation. Computational efficiency is mainly due to the dimensionality of the problem being reduced by one when compared to methods such as Computational Fluid Dynamics (CFD), as only the boundaries of the numerical domain need to be spatially discretised. Because the interior of the fluid domain is not spatially discretised, numerical dissipation is also minimal when compared to methods such as CFD. Despite these non-negligible advantages, and as discussed above, potential flow often lacks the accuracy needed for high-fidelity simulations and is therefore preferable for developing early stage concepts rather than for full-fledged and more realistic numerical simulations modeling real-world devices. In [24], where a steady wave-making problem acting on a fixed floating structure is presented, it is stated that models including viscous effects are preferable over a potential flow model in order to obtain accurate free surface elevation and pressure field on the hull of the structure,

despite the additional computational cost. According to [101], a potential flow model is satisfactory for simulating a kitefoil as numerical results show that only 10% of the total drag forces are due to viscous effects. It is also concluded in [96] that potential flow theory is adequate enough when compared to CFD in the case of a Tension-Leg Platform (TLP) for floating wind, apart from the mean surge motion. It is clear from these few examples that the applicability of potential flow depends greatly on the application considered and the desired level of accuracy. Despite the advantages of potential flow over other methods (particularly its computational efficiency), it is also clear that potential flow can be considered unsuitable for high-fidelity modelling of fluid flow due to the assumptions made of inviscidity and irrotationality, as well as the use of empirical or semi-empirical coefficients for FSI applications.

■ II.2.2 Computational Fluid Dynamics (CFD)

CFD encompass a multitude of approaches that derive the Navier–Stokes Equations (NSEs), which can describe rotational and viscous fluids. The NSEs were discovered in five independent instances by Navier in 1822 (published in 1823 [95]), Cauchy in 1823, Poisson in 1829, Saint-Venant in 1837, and Stokes in 1845. The history behind the "five births" of the NSEs is described in more detail in [31]. Note that because rotationality and viscosity of the fluid is taken into account in CFD, it Feynman called it the flow of "wet water" in contrast with the flow of "dry water" for potential flow theory [52].

The most common CFD techniques use a mesh-based approach where the fluid domain is discretised spatially to solve the NSEs with Finite Difference Method (FDM), Finite Volume Method (FVM), or Finite Element Method (FEM). One of the first written records that pioneered mesh-based CFD can be traced back to the work of [29], with the use of FDM, and the introduction of the well-known Courant–Friedrichs–Lewy (CFL) stability condition that bears the name of the authors of that work. More detail on the historical context of CFD development up to the 21st century focusing primarily on FDM and FEM can be found in [123] and context concerning FEM from the 1990s to the early 2010s can be found in [1]. With a mesh-based approach, the NSEs are solved in each mesh cell until a global solution has been found, and this process is repeated until convergence is reached for a given time step. Therefore, the accuracy of this method depends greatly on the refinement level of the mesh used for the simulation and the tolerance chosen for convergence. This leads to one of the biggest drawbacks of CFD compared to simpler methods such as potential flow: the computational cost for solving a single time step is significantly higher due to larger global matrices. Finer meshes also imply smaller time steps and therefore more iterations are needed to complete a simulation when a stabilising time-stepping scheme is used such as the CFL condition (i.e. smaller elements mean that smaller steps must be taken).

The main advantages of CFD over simpler methods such as potential flow are that nonlinearities of the flow, its effect on structures, and the effect of structures on the flow are inherently taken into account. For these reasons, it can be seen as a more realistic method for simulating complex fluid flow than potential flow, but it is by no means a perfect approach as it still idealises certain aspects of the fluid. One of its main limitations is turbulent flows: scales of such phenomena can be complex to resolve accurately without leading to prohibitive computational cost. Turbulence must therefore be either assumed insignificant enough to ignore or resolved by a sufficiently high refinement of the mesh around turbulent fronts (or through approximations based on empirical data). Turbulence modelling is a major topic and research focus in the CFD community (and has been so for decades) and can be divided into three

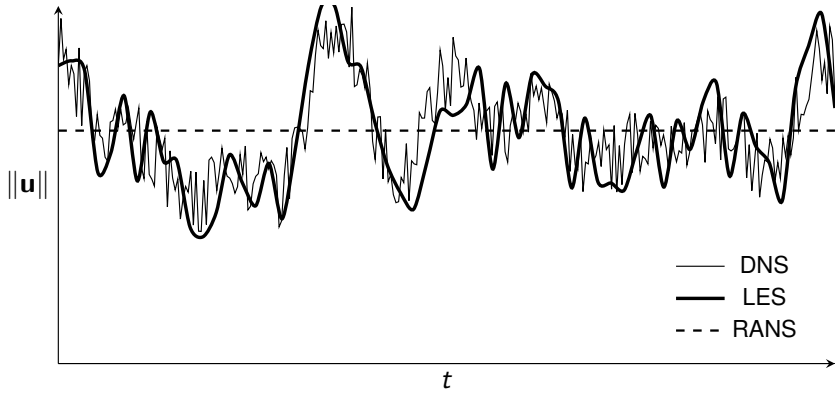


Figure II.1: Illustration of RANS, LES, DNS principles on a velocity time series, assuming steady state flow

main approaches: the first one is Direct Numerical Simulation (DNS), where the fluid flow is solved on a mesh fine enough to resolve turbulence of the smallest scales and a time step small enough to cope with the largest scales of flow motion. This leads to high – and often prohibitive – computational cost as DNS requires resolving eddies up to viscous (Kolmogorov) scale. The second method, Large Eddy Simulation (LES), directly resolves large energy-containing eddies and uses subgrid-scale models for smaller-scale eddies. While LES is significantly more computationally efficient than DNS, high spatial and temporal refinement covering large eddies that form during simulations is still needed with this approach. The third method, Reynolds-Averaged Navier–Stokes (RANS), applies time-averaging on the NSEs using empirical turbulence transport models for closure. RANS is the method with the highest computational efficiency but it is relatively sensitive to the nature of the turbulent transport models. An illustration of the three methods is shown in fig. II.1. It appears from this illustration that the DNS approach resolves all scales of turbulence, while using RANS essentially averages the effect of this turbulence on the fluid, as does LES but with more fluctuations due to its capacity to resolve larger eddies. The DNS method theoretically provides the most realistic solution, but its demand in computational power is prohibitive for most engineering applications. The model and empirical coefficients chosen for the RANS approximation can lead to underestimation or overestimation of turbulent effects on the flow but it still provides a far more reliable approach for simulating turbulent effects when compared to potential flow. The coefficients for the RANS method are usually chosen depending on the estimated global Reynolds number of the flow and on the geometry of the structures. The LES approach can be a convenient middle ground, but still includes a reduced aspect of the drawbacks from the two other methods: a relatively high computational cost, and dependence on empirical approximations on lower-scale eddies.

On top of the traditional mesh-based CFD methods described above, meshless approaches such as the Smoothed-Particle Hydrodynamics (SPH) method that was first introduced by [90, 57] have also been recently developed. SPH, as implied by its name, is a particle-based method and, in the same logic as potential flow (“dry water”) and mesh-based CFD (“wet water”), one could call SPH the flow of “solid water”, as the fluid is represented with solid particles. In opposition to discretisation in space for mesh-based methods, the SPH method can be considered as a discretisation in mass, with each SPH particle a finite mass of the discretised continuum. While mesh-based methods usually describe the fluid in the Eulerian frame, SPH uses the Lagrangian frame. The main advantage of the SPH approach is that complex events such as breaking waves and overtopping are more easily described than with mesh-based

methods, but the implementation of boundary conditions in such approach is less trivial than mesh-based methods. Parallelisation is also known to be more troublesome due to the particles technically able to travel from one side of the numerical domain to the other. Some techniques use hybrid Eulerian-Lagrangian formulations such as Particle-In-Cell (PIC) [15] in an attempt to combine the flexibility of the Lagrangian description of the fluid for complex events with the efficiency of the Eulerian formulation for spatial discretisation (such as in [25]).

II.3 Fluid–Structure Coupling

When simulating FSI with moving structures, the fluid–solid interface must be coupled to ensure kinematic continuity (same displacement and velocity) and dynamic continuity (equilibrium of stresses). Numerous fluid–structure coupling approaches exist, and the choice of a specific coupling scheme is highly dependent on the problem considered as well as the objectives of the numerical model (e.g. efficiency, accuracy, stability, modularity). Coupling approaches can be divided into two main categories: monolithic coupling and partitioned coupling, with the latter divisible in several subcategories (explicit, implicit, and semi-implicit), as described in more details below. In this section, unconditional stability (or instability) implies that the scheme is stable (or unstable) regardless of numerical parameters and methods employed (e.g. time discretisation technique), as opposed to conditional stability (or instability) that depends on the choice of certain parameters or methods.

II.3.1 Monolithic Approach

In the monolithic approach, the fluid and solid problems are intertwined and form a single, combined solver as the solution for both solid and fluid is found simultaneously and is interdependent. This method is also called direct, global, or fully coupled approach in the literature. Depending on the physics involved in the application considered, each problem to be solved leads to a different global system of equations and therefore to a different monolithic formulation. In the case of FSI with CFD and other nonlinear problems, monolithically coupled problems are often solved using the Newton-Raphson method. Because it uses a single, global FSI system of equations, this coupling procedure has the advantage of being highly accurate while offering unconditional stability due to the exact convergence of the solution between the fluid and the solid. One of its main disadvantages is that it can become impractical to develop and can hinder the use of independent and potentially highly specialised external libraries simulating specific physical processes. Monolithic schemes encourage an *ad hoc* approach in terms of development or extension of its capabilities, as the inclusion of additional physics often means redesigning and re-implementing the scheme (if at all possible) to accommodate for the new terms in the global system of equations. Compared to simpler coupling schemes, a complication for FSI applications using a monolithic scheme is that the velocity of the fluid–solid interface is an additional nonlinear term that appears in the fluid convection term of the global equation. For this reason, the nonlinear fully coupled monolithic system contains ill-conditioned non-definite positive matrices which is computationally expensive to solve [74] and, as the complexity of the global monolithic problem grows, it can also be difficult to develop efficient preconditioners. Additionally, and again due to the use of a single system of equations, the temporal discretisation for the fluid and the solid must be the same. This can prove inconvenient when either the fluid or the solid requires a much finer time discretisation to successfully run with the intended accuracy, as it will consequently force the other physical aspect to also adopt the same time step.

This lack of modularity makes it harder to implement or update state-of-the-art techniques specific to each of the coupled aspects.

II.3.2 Partitioned Approach

In a partitioned approach, the fluid and solid solvers are completely separated. This method is also called staggered, segregated, or time-lagged approach in the literature. Each solver here can be seen as a black-box because it does not exchange any information with other solvers while it is solving its own sub-problem. Each solver can still use the output of other solvers: in the case of FSI, the fluid solver uses the solid displacement as input, and the solid solver uses the fluid pressure and shear forces. The aim of a partitioned scheme is to find a suitable approximation of the solution that would be provided by a fully coupled monolithic scheme, ideally in a stable manner.

This type of approach allows more flexibility and modularity when compared to the monolithic approach, as it is possible to replace or improve a specific solver when needed without affecting the other solvers internally. This means that access and modification of the source code of each solver are not always necessary, as opposed to a monolithic scheme. Additionally, the temporal discretisation used on either solver can differ from the one used on the other solver due to the partitioned nature of the system of equations. The main disadvantage of the approach is the potential instability issues due to the partition of the models that causes a temporal "lag" between the solutions. If the coupling problem is nonlinear, and because of this lag, energy conservation is not ensured at the fluid–solid interface. A very common instability problem in CFD applications using partitioned schemes is the added mass effect, which is discussed in more detail in section II.4. Many different coupling schemes fall under the partitioned approach, and they can be divided into three main subcategories that are described below: implicit, explicit, and semi-implicit schemes.

II.3.2.1 Implicit Schemes

In implicit schemes, solutions for the fluid and the solid are calculated separately, but iterations are used until convergence between the models is reached (up to a user-defined tolerance). This approach is also called strong coupling or iterative coupling in the literature. For models that depend on each other, the solution found on the previous iteration of one model is used in the new iteration of the other model. These iterations can be seen as prediction or correction steps where the fluid and solid states must be stored and retrieved at each iteration. In theory, the more iterations are used, the closer the solution of an implicit scheme is to the solution that a monolithic scheme would provide.

The most common approaches to converge to a solution are: Block Gauss–Seidel (BGS), block Jacobi, and block Newton. When using BGS iterations, the solution for the solid at iteration k is used for the solution of the fluid at iteration $k+1$, which is in turn used for the solution of the solid at $k+1$. This can be compactly written using fixed-point formulation as follows:

$$\mathbf{s}^{(k+1)} = S \circ F\left(\mathbf{s}^{(k)}\right) \tag{II.1}$$

with $\mathbf{s}$ the state of the solid, F representing the fluid solver, S representing the solid solver, and $F\left(\mathbf{s}^{(k)}\right)$ returning

the fluid state at iteration $k+1$. The residual that must be brought below a given tolerance can be expressed as:

$$\mathscr{R}^{(k+1)} = S \circ F\left(\mathbf{s}^{(k)}\right) - \mathbf{s}^{(k)} \tag{II.2}$$

With block Jacobi iterations, the solution of the fluid and solid at iteration $k+1$ both use the solutions found at iteration k, which leads to the following fixed-point formulation:

$$\mathbf{s}^{(k+1)} = S \circ F\left(\mathbf{s}^{(k-1)}\right) \tag{II.3}$$

with $F\left(\mathbf{s}^{(k-1)}\right)$ the fluid state at iteration k, and the corresponding residual expressed as:

$$\mathscr{R}^{(k+1)} = S \circ F\left(\mathbf{s}^{(k-1)}\right) - \mathbf{s}^{(k)} \tag{II.4}$$

Note that the only notable advantage in using the block Jacobi over BGS is the possibility to do calculations for the fluid and solid solver in parallel if at all desirable.

Block Newton is the most challenging coupling procedure in terms of implementation but has the advantage of not depending on the order in which each subsystem (solid and fluid) is solved. Examples of Newton-like methods applied to partitioned problems can be found in [91].

While these iterations ensure convergence to a common solution between the solid and the fluid (given that there is no instability), the main disadvantage of the implicit approach is that it can become computationally inefficient if a large number of iterations is needed to bring the residual under a certain tolerance. Indeed, solving for the solid motion and especially the fluid dynamics several times per time step is often computationally prohibitive in large CFD applications. Additionally, the requirement of storing and retrieving the state of the models after a prediction step can be an issue for complex implementations. For example, Multibody Dynamics (MBD) codes that involve collision detection, joints and constraints, or even multi-physics processes, are not necessarily designed for allowing arbitrary backtracking in time. This is especially true when different solvers are used for different processes, such as body dynamics solved internally but collision detection by an external solver. This is mostly a software engineering problem, but it is not to be overlooked when designing a coupling method if predictive steps are intended.

II.3.2.2 Semi-Implicit Schemes

Semi-implicit schemes, also called explicit-implicit in the literature, consist in treating part of the interface boundary conditions as implicit (usually the Neumann part affecting the stresses applied on the solid) while keeping an explicit treatment for the other part (usually the Dirichlet part affecting the fluid velocity). With this method, only one main iteration for the fluid velocity is needed per global time step while sub-iterations are used for convergence between the fluid pressure and the solid displacement, followed by a correction step for the fluid velocity. As a result, the computational cost is reduced when compared to a fully implicit (or implicit–implicit) method, especially when the fluid step is expensive relative to the solid step (which is usually the case in CFD). These pressure–solid iterations still require the ability to fully predict and correct solid displacement solution, which can be an issue depending on the capabilities of the solid solver and the global complexity of the problem, as discussed above. Note also that this implicit treatment of the traction at the interface can only be applied if the fluid solver uses a projection scheme that separates the fluid velocity and pressure equations, such as the Chorin–Teman scheme [26, 121, 120] used

in [17, 49, 6, 94] for implementing semi-implicit schemes. A relatively recent overview of projection methods for incompressible flow can be found in [63].

II.3.2.3 Explicit Schemes

Explicit schemes also solve the fluid and solid steps separately, but only once per time step. This approach is also called weak or loose coupling in the literature. It is, by far, the partitioned coupling method that requires the least computational power, at a trade-off in terms of accuracy and stability. There are many types of explicit schemes, which can be divided into the following subcategories: synchronous or asynchronous, and serial or parallel. Synchronous methods find solutions for the fluid and solid at matching temporal values (i.e. solution at $t^{(n)}$ for both the fluid and solid), while asynchronous methods solve the different models with a shift in time (e.g. solution at $t^{(n)}$ for the solid, and $t^{(n+\frac{1}{2})}$ for the fluid). A parallel scheme uses the solutions of the coupled solvers at the same time $t^{(n)}$ to advance to the solution $t^{(n+1)}$, while a serial scheme first solves one of the models to $t^{(n+1)}$ and uses the solution from that model at $t^{(n+1)}$ for solving the other model from $t^{(n)}$ to $t^{(n+1)}$.

The most commonly used explicit scheme, due to its ease of implementation, is the Conventional Serial Staggered (CSS) scheme. It is equivalent to a single iteration of the BGS method per time step. Its parallel counterpart, the Conventional Parallel Staggered (CPS), is equivalent to a single iteration of the block Jacobi per time step. It is shown in [43] that serial schemes are more accurate and more stable than their parallel counterparts, and that asynchronous methods can be more accurate than synchronous methods. From [108], it is also shown that the CSS is only first-order time-accurate at best, even if the fluid and solid solvers are second-order accurate. Some explicit schemes use a prediction step, such as the Generalised Serial Staggered (GSS) introduced by [108, 107], where the predicted position of the solid is calculated as follows:

$$\mathbf{s}_P^{(n+1)} = \mathbf{s}^{(n)} + \alpha_0 \dot{\mathbf{s}}^{(n)} \Delta t + \alpha_1 \left(\dot{\mathbf{s}}^{(n)} - \dot{\mathbf{s}}^{(n-1)} \right) \Delta t \tag{II.5}$$

with α_0 and α_1 two given coefficients according to the order of time integration desired. This prediction is used to move the fluid–solid interface to match the predicted position $\mathbf{s}_P^{(n+1)}$. From here, the solid uses the fluid solution at $t^{(n+1)}$ to find $\mathbf{s}^{(n+1)}$ from $\mathbf{s}^{(n)}$. The Improved Serial Staggered (ISS) scheme, introduced in [89, 43], corresponds to an asynchronous GSS scheme with an offset of half a time step ($\frac{\Delta t}{2}$) between the solid and the fluid solution. The Improved Parallel Staggered (IPS) scheme is also introduced by [108] but yields less accurate results than the ISS scheme, which could be expected from a parallel staggered scheme when compared to its serial equivalent. It is also formally shown in [45] that the ISS scheme is second-order time-accurate if solid solver uses the midpoint rule and if the Arbitrary Lagrangian–Eulerian (ALE) mesh integrator is also second-order accurate. Note however that to obtain the advantage of a second-order accuracy, the ISS scheme requires the time integration for the solid solver to be restricted to the midpoint rule and that, therefore, the global time step must be known in advance in order to predict the movement of the solid mesh to half a time step. This is sometimes not possible when, for example, a constraint such as CFL is used on the time-stepping scheme of the fluid. The ISS scheme can also use a time step that is five times larger than a CSS scheme to reproduce equivalent results [43].

The main issue with fully explicit schemes is their stability when, for example, subject to strong added mass effect. We mention here that unconditionally stable explicit schemes have been successfully developed for compress-

ible fluids [44], and incompressible fluids [51, 20, 19]. More details about the added mass effect and stabilisation of coupling schemes are provided in the following section.

II.4 Added Mass and Coupling Instabilities

The discussion here focuses on the added mass effect that can lead to instabilities in CFD applications involving FSI, which is related to – but must not be confused with – the actual added mass of a structure. The concept of added mass was first introduced by Friedrich Bessel in 1828 [117], who observed that the period of oscillation of a pendulum increased when submerged in a dense fluid instead of a vacuum, and therefore concluded that the effective mass of the system must be larger due to the surrounding fluid. In fact, because a solid and a fluid cannot occupy the same space at the same time, and when a solid moves with a non-zero acceleration in a fluid, it must also move the surrounding fluid. This additional effort for moving the fluid leads to what can be seen as additional mass changing the inertia of the body, hence the term “added mass”.

In potential flow models, the added mass is taken into account by simply adding the estimated added mass matrix to the total mass matrix of the structure (or of the overall system). These added mass values can be estimated analytically if the structure has a simple geometry or can be found experimentally for more complex geometries. The added mass effect never occurs in potential flow, as the added mass is directly integrated in the mass matrix due to the inability of the fluid model to take the added mass into account. CFD simulations inherently take into account the effect of a moving solid on the fluid and vice-versa, which means that the added mass is also taken into account when retrieving forces from the fluid. The added mass should therefore not be added to the mass matrix of the solid in this case, or it will essentially be counted twice.

When using monolithic schemes where the solution of the fluid and solid are processed together, the effect of the added mass is calculated accurately, and the simulation is stable. However, when using partitioned schemes, an instability known as the added mass effect may occur. The added mass effect is due to the nature of a partitioned coupling that introduces a lag between the fluid and solid solutions, where the forces from the fluid found at the previous time step (or iteration) are applied to the solid, thus omitting the additional added mass forces resulting from changes in acceleration while the solid moves to its new position. This omission leads to inaccuracies and, in some cases, to unconditionally unstable simulations. The importance of the added mass effect depends on the characteristics of the solid, such as its relative density to the surrounding fluid and/or its aspect ratio. The following conditions can worsen the added mass effect [23, 48]: for a given solid geometry, reducing the solid density relative to the surrounding fluid density; for a given solid density, increasing the length of the domain. Additionally, if a simulation is already unstable due to the added mass effect, refining the time discretisation or using higher-order time integration aggravates the instability [54], implying that lower accuracy must be traded for greater stability.

On top of potentially making explicit coupling schemes unconditionally unstable, the added mass effect can also affect the convergence rate of implicit schemes, which can lead to significantly lower computational efficiency as more iterations are needed to reach convergence. In the most extreme cases, the added mass effect can also render implicit schemes non-convergent. For semi-implicit schemes where pressure segregation is used, simulations subject to strong added mass effect can also yield non-convergent results. To recover better convergence rates,

and given an appropriate first guess of the added mass, relaxation factors can be used as in [116, 35, 134, 133]. The added mass of arbitrarily shaped bodies can be estimated accurately with a technique using Partial Differential Equations (PDEs) that was introduced by [116].

For explicit schemes, Robin–Neumann boundary conditions can be used on the fluid–solid interface, and stabilisation can be achieved by extrapolations of solid velocities and fluid stress as in [50] and with prediction-correction steps as in [7, 8]. Another method from [20] proposes a coupling through Robin and Dirichlet–Nitsche boundary conditions at the interface and uses a dimensionless penalty term on the pressure from the fluid solver to stabilise the scheme. It is also possible, although less convenient, to use another penalty term that has the dimension of a length to include viscous contributions as described in the previous work of the same authors [21]. This method suffers from low temporal discretisation accuracy ($O\left(\Delta t^{\frac{1}{2}}\right)$), but prediction-correction iterations (called defect-correction by the authors presenting this method) can be used to improve the time-accuracy of the solution to $O\left(\Delta t\right)$. A single iteration can suffice to yield satisfactory results [20], but one can argue that using any prediction-correction iteration leads to a scheme that is not truly explicit anymore. It is also possible to use Robin–Robin coupling to stabilise an FSI simulation, as in [51, 19]. While Nitsche's method and Robin–Robin type couplings can be added mass free, they both need the imposition of additional constraints to the rate of the discretisation parameters to achieve the desired stability [19].

In any case, it is clear that there are many different ways to tackle the added mass effect and that this problem is the subject of ongoing and active research. It also appears that higher stability of a scheme must be traded for lesser accuracy or higher computational cost.

II.5 Moving Solids and Fluid Mesh

In mesh-based CFD methods, the motion of structures in the fluid domain requires either deforming the mesh when the structural boundaries are conforming to the fluid mesh or to track the motion of the structure through the fluid mesh when the boundaries are non-conforming.

In a conforming boundary approach, mesh nodes are shared between the fluid mesh and solid mesh at the fluid–solid interface. This approach allows straightforward and accurate integration of the fluid pressure induced by the solid at its boundaries. The main disadvantage is that, because the mesh must conform to the fluid–solid interface, it must deform to accommodate the motion of the solid. This is particularly troublesome when solid motion is extensive (such as displacements over relatively long distances or full revolutions of the solid structure), and can lead to mesh quality deterioration or even to tangled meshes in the most extreme cases. Another limitation is the fact that it is not able to handle collisions between two structures as the mesh cannot be fully compressed at the point or area of contact.

Non-conforming boundary approaches use two independent meshes: one for the fluid and one for the solid, unless the latter is represented solely using implicit techniques such as a signed distance function. This approach for tracking the fluid–solid interface is known as the Immersed Boundary Method (IBM). The solid in this case can freely travel through the fluid mesh as its boundaries are implicitly tracked. Interpolation from the solid boundary to the mesh nodes or quadrature points is required to apply relevant boundary conditions on the fluid (e.g. no-slip or

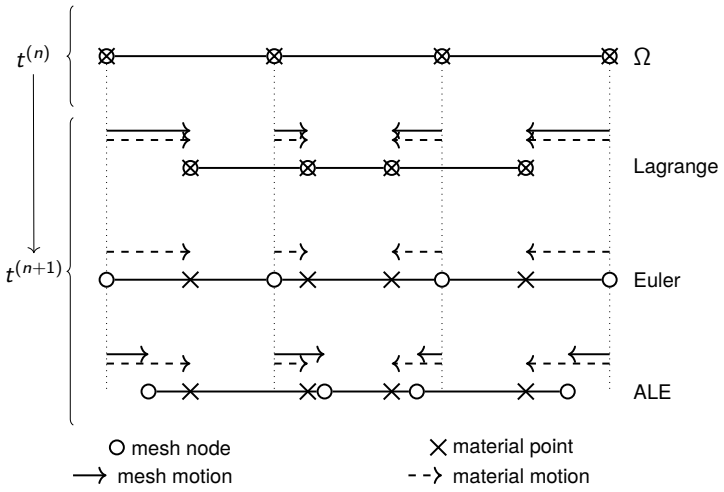


Figure II.2: Illustration of Lagrangian ($\Omega_{\widehat{\mathbf{x}}}$), Eulerian ($\Omega_{\mathbf{x}}$), and ALE ($\Omega_{\boldsymbol{\xi}}$) domains in 1D

free-slip conditions), and to calculate the fluid forces acting on the solid. A notable advantage of this approach is that it allows movement of the solid without deteriorating the fluid mesh, and can also handle large displacements without problems as well as colliding bodies. One of the disadvantages is that integrating variables over the implicit boundary is usually not as accurate as a conforming mesh technique, requiring a finer mesh around the interface for an equivalent accuracy. In order to keep up with refinement requirements around the solid boundary as it moves through the fluid mesh, it is often preferable to use mesh adaptivity or mesh deformation techniques specifically designed for implicit boundaries. A good overview of various IBM techniques can be found in [92].

The remainder of this section describes the principles of ALE mesh, along with a brief description of some of the most common mesh motion techniques. These techniques can be applied to both conforming and non-conforming solids, the former requiring mesh deformation to move the fluid–solid interface and the latter benefiting from mesh deformation to keep a refined mesh around its implicit boundaries.

■ II.5.1 Arbitrary Lagrangian–Eulerian (ALE) Mesh

The ALE mesh method, introduced by [69], is a hybrid approach between the Eulerian (where the mesh is fixed) and Lagrangian (where the mesh follows the fluid) frames of reference. It allows for an arbitrary motion of the mesh following the movement of the fluid–structure interface. To understand the ALE method, three different frames of reference that are used to describe the physics of a continuous medium are introduced alongside an illustration in fig. II.2.

In the Lagrangian formulation, the frame of reference called the material frame of reference coincides with the particles of the medium. Its corresponding domain, the material domain $\Omega_{\widehat{\mathbf{x}}}$, deforms with the medium by following the relative movement of the particles to each other (with particle material coordinates $\widehat{\mathbf{x}} \in \Omega_{\widehat{\mathbf{x}}}$). This formulation is popular in solid mechanics where deformations are relatively small but can become cumbersome in fluid dynamics where unsteady flows could lead to highly deformed and potentially tangled meshes that would require extra care and periodical remeshing. For this reason, Lagrangian methods for fluids typically employ particle discretisation schemes, such as SPH or PIC schemes. When using mesh-based approaches for modelling Two-Phase Flow

(TPF), the free surface can be represented as a moving boundary conforming to the moving mesh.

In the Eulerian formulation, the frame of reference is the spatial frame of reference and is fixed in relation to the terrestrial reference frame. Its corresponding domain, the spatial domain $\Omega_{\mathbf{x}}$ with $\mathbf{x} \in \Omega_{\mathbf{x}}$, does not deform over time. This formulation is traditionally preferred for CFD with no moving explicit boundaries. The moving interfaces inside the domain can be tracked implicitly using transport equations, such as the free surface with Volume of Fluid (VOF) and Level Set (LS).

In the ALE formulation, an arbitrary frame of reference that is usually neither Lagrangian nor Eulerian in regards to the medium, combines elements of both approaches aforementioned. Its corresponding domain, the reference domain $\Omega_{\boldsymbol{\xi}}$ with $\boldsymbol{\xi} \in \Omega_{\boldsymbol{\xi}}$, can be used to map a spatial domain to a material domain and vice-versa. The mesh nodes can move in space independently of the material, resulting in a mesh motion different from material motion from a Eulerian frame. Note that the mesh motion model is usually separated from the model that provides the quantities of interest (such as fluid velocity). However, because both models share the same mesh, the governing equations of the model of interest must take into account the displacement of the mesh, and this is achieved by introducing additional terms such as mesh velocity to the NSEs.

In the context of FSI, the ALE approach is suitable for handling deformations of the fluid domain due to the motion of solids with mesh-conforming boundaries. This could not be described with a purely Eulerian approach as the fluid mesh must be deformed to accommodate for the moving structure. In a purely Lagrangian approach, following the material displacement of the fluid can become computationally challenging, if not unfeasible due to mesh entanglement. Note however that when using the ALE approach on a conforming solid–fluid mesh, the nodes lying on the solid boundaries can be seen purely as Lagrangian from the solid perspective as they follow the solid material displacements. When the mesh-conforming solid boundaries move, it is also necessary to displace the interior nodes of the fluid mesh to prevent element distortion or entanglement, and to preserve an acceptable global quality of the mesh. There are many techniques for deforming the mesh using physical analogies, interpolation, or mesh control methods. Physical analogies include linear springs (e.g. [10]), and linear elasticity (e.g. [122]). Interpolation methods include Radial Basis Function (RBF) (e.g. [14]). Mesh control methods use monitoring functions that assign criteria to the mesh elements (such as its volume or shape) and move the mesh using the difference between the target and actual characteristics of the mesh (e.g. [62]). A comparison of several of these mesh motion methods in terms of computational efficiency and some of their respective advantages and drawbacks is given by [113].

II.5.2 Mesh Adaptivity

Although this is not the focus of the work presented in this thesis, it is worth mentioning mesh adaptivity as an alternative to mesh deformation for including moving structures in a fluid mesh. Mesh adaptivity relies on remeshing of the fluid domain in a targeted manner by adding or removing the mesh nodes and reconnecting the surrounding nodes to form new elements. If the solution is known, such as an analytical solution, adaptivity can be applied directly to the areas requiring refinement for most accurate results. In most cases, the solution is not known in advance, so the adaptivity is usually done with *a posteriori* error estimate. To reduce the global energy norm, there are several types of approaches such as the residual method [3] and recovery method [136]. Instead of the general

FEM error, the metric used can also be a quantity of interest such as the free surface location or vorticity of the flow. This is called goal-oriented error estimation [102]. Note that using *a posteriori* error estimate to refine the mesh is also possible for ALE mesh technique if using a monitoring function.

A known issue with mesh adaptivity is load balance in parallel, as a processor can see its local domain be greatly refined relative to the domain of other processors. This can significantly slow down a simulation as all processors must finish their individual calculations before being able to move to the next time step in FEM applications. A solution to this problem is dynamic load-balancing [34].

II.6 Moorings

II.6.1 Design Considerations for Mooring Systems

Before describing mooring systems, the different types of moored floating structures are introduced here as they can heavily influence the design of mooring systems. Floating structures, especially in the context of ORECs, can be classified into two main categories: motion-dependent and motion-independent [80]. Motion-independent structures do not require motion to function and their response to wave loads is usually purposely restricted as much as possible for best operating conditions. Typical examples of motion-independent structures that operate with minimal motion include oil & gas platforms, floating OWCs, overtopping WECs, and floating OWTs. On the contrary, the frequency of resonance of motion-dependent structures is tuned to respond to a specific range of wave frequencies, usually for the purposes of energy extraction. Typical examples of such structures include mostly WECs such as point absorbers and attenuators. In the case of motion-dependent devices, it is particularly important to consider the mooring system as an integral part of the structure as it can heavily influence the frequency of resonance of the overall system. In the case of motion-independent devices, the natural frequency of the mooring system is traditionally chosen to be far away from the natural frequency of the structure and the expected main wave frequencies of the deployment site. This means that the dynamic effects in the mooring lines are expected to be less significant. These dynamic effects must however not be ignored on smaller scale devices (e.g. floating OWC, OWT) placed in highly energetic sites with harsh environmental conditions where significant hydrodynamic loads are expected on the device.

In all cases, the main characteristics to consider when designing a mooring system are station keeping of the structure, the footprint of the system, limit states of the mooring cables, redundancy of the mooring system, installation criteria and limitations, maximum excursion of the device, as well as overall the cost. Depending on the application, station keeping objectives for the floating structures is to be as stable as possible (e.g. motion-independent device), or to allow response for one or more Degrees of Freedom (DOFs) of the structure while restricting the movement of other DOFs as much as possible (e.g. heaving point absorber). Depending on the space limitations of the deployment site, the mooring footprint of the structure must be considered, along with the space needed between each structure if several of them are deployed on the same site, as well as other restrictions such as navigation lanes. Note that a larger footprint usually incurs a higher cost for a catenary mooring system (with the notable exception of TLP), meaning that reducing it as much as possible is often desired as it provides an economic advantage. The limit states of mooring systems can be defined as: Ultimate Limit State (ULS) which is the maximum load that the system can withstand before breaking, Accidental Limit State (ALS) for the ability of lines

to resist accidents such as breakage of one of the lines of the system (which can lead to significantly higher loads on other lines), Fatigue Limit State (FLS) for resisting cyclic loading over an extended period of time, and Serviceability Limit State (SLS) corresponding to loads under typical conditions. Finally, in case of failure of one or several lines, the mooring system should be equipped to withstand the additional loads on the other lines until maintenance can be undertaken for replacement or repair. Mooring systems with a single line offer no redundancy in case of line breakage and are therefore usually avoided. The non-exhaustive list of design considerations described above will affect the configuration of the mooring system and the material of the lines themselves. An overview of different mooring materials such as chains, wires, and ropes can be found in [77, 129, 127]. In terms of configurations, the mooring system can be set as a catenary, be fully stretched (e.g. for TLP), or use more complex layouts with subsurface buoys and clump weights such as “lazy-S”, “steep-S”, “lazy-wave”, “steep-wave”, or “pliant-wave”. For a more detailed overview of the various challenges concerning the mooring systems of floating WECs and their design specifications, the reader can refer to the work of [80, 77].

Another aspect to consider when designing a mooring system is the nonlinear effects induced by the motion of the structure and the environmental loads acting on the cable. The main contributors to the damping of a mooring line are drag forces, friction with the seabed, and internal friction within the cable. The drag forces become a particularly important factor in regards to the overall damping when the line is submerged in a relatively high-density fluid (such as water). Damping of a mooring line can be assessed experimentally through decaying oscillation [72] or driven sinusoidal motion [128, 79]. The former method involves moving a moored structure to a known distance from its equilibrium position and releasing it: the resulting decaying oscillation is then used to estimate the damping from the log decrement of the amplitude after each cycle. The latter method involves forcefully moving a structure in a sinusoidal motion and calculating the energy dissipation from the recorded displacement-tension curve at the fairlead. These experiments can also be used as a benchmark for validating the capability of numerical models to simulate nonlinear damping effects appropriately.

■ II.6.2 Numerical Modelling of Mooring Systems

The two main ways of numerically modelling cables attached to structures are: the quasi-static approach, and the dynamic approach. The quasi-static approach consists in applying forces on the structures from tensions at the fairleads by considering that the cable is at static equilibrium at each time step. Tensions can be calculated directly from an analytical solution, such as the catenary equations. All dynamic effects are however completely ignored as, by definition, the quasi-static approach assumes static equilibrium of the line. The main advantage of this method is that it is usually significantly more efficient computationally than a fully dynamic method. It is therefore useful for preliminary analysis and parametric studies of mooring lines and their characteristics (e.g. axial stiffness, pretension, etc), or for simulations with very little dynamic effects involved (e.g. relatively slow motion of a device) but it does not yield accurate results when nonlinear dynamic effects are expected in a simulation.

When it comes to mooring dynamics, two main techniques are used: the lumped-mass approach using MBD (e.g. see [64]) and the FEM approach (e.g. see [106, 105]). In the lumped-mass approach, the cable is divided into massless subsegments connected by nodes that inherit half of the weight of the adjacent subsegments. The cable essentially corresponds to several rigid bodies (or masses) that are linked to each other through springs and

revolute joints. The equations of motion from MBD are then directly applied to the cable nodes. Hydrodynamic loads are usually retrieved at the midpoint of the segments and evenly distributed to the two surrounding nodes, although some models also calculate the loads at the nodes themselves such as in [64]. Even though bending of the cable can be calculated and restrained, cable segments between nodes are usually assumed to be straight as they are simply massless springs. In the FEM approach, cable sections are discretised with elements that can be bent and distorted, giving additional accuracy when compared to a lumped-mass approach. An early attempt at using FEM for cable dynamics was published in 1960 [126], but this technique was applied reliably to cable dynamics only relatively recently. In [83], it is concluded that the FEM and lumped-mass methods can be considered equivalent in terms of results as long as accelerations are relatively small, but not for vibratory studies or severe manoeuvres of the fairlead where larger accelerations are in play and where the FEM method provides better results. The order of nonlinearity of the elements defines to which degree of accuracy the shape of the beam is described: higher-order schemes lead to higher computational cost, but the increased accuracy of such schemes can also be used to reduce the number of elements needed for reliable predictions [105].

It was traditionally common to simulate mooring dynamics in a fully uncoupled manner where the mooring dynamics and floating structure dynamics were computed completely separately. In this case, the tensions in the mooring system are calculated according to the full time-series of the position of the structure that is used directly to impose a prescribed motion on the fairleads. This approach is therefore mostly suitable for calculating mooring tensions numerically if experimental data of the motion of the full system (i.e. the structure and its mooring system) is already available. An example of uncoupled mooring analysis can be found in [64] where a lumped-mass model is successfully validated against experimental data using the known displacement over time of the full system. The uncoupled and coupled methods are also compared without using prescribed motion in [103], where it is concluded that the cable tensions and resulting forces applied on a vessel are in good agreement with experimental results when using the coupled method but are usually underestimated when using the uncoupled method.

Coupled methods for mooring–structure can be separated in two main categories: weak coupling and strong coupling. Note that weak coupling here is equivalent to partitioned coupling while strong coupling is equivalent to monolithic coupling, as described in section II.3. In a weak mooring–structure coupling scheme, the mooring and structural dynamics are computed separately, but the separated models communicate their results to each other at each time step. The body dynamics solver provides the position of the fairlead to the mooring solver, and the mooring solver provides the tension at the fairlead to the body dynamics solver. An example of weak mooring–structure coupling can be found in [106]. In a strong coupling scheme, the mooring and body dynamics are solved simultaneously as part of the same system of equations. Such coupling can be achieved by making the structure an integral component of the cable. In a dynamic model, the fairleads of the structure are directly modelled as nodes of the cable mesh, or constraints such as joints can be applied between one end of the cable and the fairleads. Strong coupling is more costly than weak coupling, especially if several lines are connected to the same structure as the global system of equations can become complex. Weak coupling allows for computing each line separately on different meshes instead of using a more complex global mesh for the overall system. If computational efficiency is a concern and depending on the accuracy desired, weak coupling can still yield very satisfactory results for a

fraction of the cost of the strong coupling [112].

The fluid–mooring coupling is usually one-way, where the effects of the fluid on the cable are modelled, but the effects of the cable on the fluid are considered negligible. For most applications, cables cannot realistically be two-way coupled with the fluid in CFD simulations without inducing a prohibitive computational cost as their axial dimension is usually significantly lower than the characteristic fluid mesh element size. For this reason, Morison's equation is usually employed for applying hydrodynamic loads on cables such as drag and added mass. As mentioned previously, the denser the fluid is, the more influential the drag is for the damping of the line. This can become a major source of uncertainty when mooring dynamics are applied numerically, especially when semi-empirical parameters are employed, which is the case when using Morison's equation with drag and added mass coefficients. These coefficients are influenced by the shape of the line (e.g. cylinder, wire, chain) and characteristics of the flow (e.g. Reynolds number, angle of attack relative to the line), and can be difficult to estimate as a result. For example, it has been shown in [18] that, while drift-only oscillations using well-established empirical values for drag coefficients show good agreement numerically, using the same drag coefficients on combined wave and drift motions can lead to drastic numerical overestimation in damping of the mooring line (around 70% in this specific case) when compared to experimental data. In [132], where hydrodynamic properties of studless chain links are investigated for steady flows using CFD, it is shown that drag coefficients decrease when the Reynolds number increases, and that the touch-down area of the cable is the most sensitive to these variations. More uncertainty can arise when using a scaled model to reproduce the behaviour of a prototype scale device, as the numerous parameters affecting the mooring system must be scaled accordingly. This scaling of mooring systems is no easy task and is the subject of ongoing research, with recent works from [11] deriving sets of dimensionless parameters for this purpose. In the work just cited, it appears clearly that the drag and added mass coefficients can be significantly different between prototype and model scale, as a mismatch in scaling of the diameter, length, or volume coefficients can significantly affect the response of a mooring line to hydrodynamic loads. These experimental uncertainties also apply to numerical models that use empirical coefficients to for hydrodynamic loads in one-way fluid–mooring coupling schemes.

II.7 State of Research

After introducing the main components for creating a framework for high-fidelity simulations of moored floating structures, some examples of coupled models using FSI and mooring dynamics are reviewed. One of the earliest attempts at coupling mooring and floating structure dynamics was performed by [103] who concluded that the traditional separated (i.e. uncoupled) approach may be severely inaccurate in comparison to the coupled approach. These coupled models for floating, moored structures, similar to the one considered herein, have been a relatively recent development. Traditionally, stand-alone potential flow or CFD models have been used to first calculate forces on a floating structure (either fixed or in motion), and then the mooring loads would be estimated through numerical springs, statics look-up tables, or dynamic models using the top end motion of the fairlead for approximating tensions of mooring lines. These well-known and simpler uncoupled approaches were considered as an industry standard until relatively recently and can be found both as commercial and open-source software. The uncoupled approach

can also be used for frequency domain analysis, as shown in [53]. More recently, higher fidelity models have been developed using lumped mass and finite element approaches for simulating mooring dynamics, coupled with linear and nonlinear models for the hydrodynamics of floating structures. In [103], potential flow was used for fluid dynamics and FEM for mooring dynamics with strong coupling with the floating structure. In [98], the numerical study of a WEC was undertaken using CFD with Ansys CFX coupled to mooring dynamics with OrcaFlex, but the details of the coupling strategy are not detailed. In [65], the lumped mass model OrcaFlex was used for mooring dynamics of a buoy along with the linear radiation-diffraction code Hydrostar to calculate its hydrodynamic coefficients. In [64], the response of a semi-submersible platform for offshore wind turbines was simulated using the lumped mass MoorDyn model with weak coupling to the potential flow solver Fast. In [106], the finite volume CFD solver OpenFOAM and the high-order finite element model MooDy were used for simulating a moored buoy, using weak coupling between the moorings and the floating structure with a lagging quadratic interpolation, and assuming the fluid to be at rest when calculating drag and inertia forces. As pointed out in this latter case and [131], the number of existing studies for the simulation of floating WECs showing thorough verification and validation need to be extended.

As can be seen from the few mentioned examples above, most of the existing coupled models use simplified, linearised methods for one of the coupled aspects (e.g. potential flow for fluid dynamics), a weak coupling between the floating structure and its mooring system, and/or assume the fluid to be at rest throughout the simulation, even when waves are present. In this thesis, the open-source FSI framework for high-fidelity simulations of moored floating structures is tailor-made, allowing for the development of an advanced coupling strategy that responds to the needs of the industrial partners (HR Wallingford and the U.S. Army Engineer Research and Development Center (ERDC)). All aspects – fluids, structures, and moorings – are therefore carefully reviewed, selected, implemented, coupled, verified, and validated to ensure an appropriate level of accuracy, reliability, and modularity. Following this literature review, the subsequent chapters detail this development and implementation process in depth.

CHAPTER III

Uncoupled Models: Governing Equations and Numerical Implementation

This chapter describes the governing equations and implementation methods behind each model separately (uncoupled). The concept of domains and boundaries that partition the Fluid–Structure Interaction (FSI) problem geometrically is first introduced here. In the context of Two-Phase Flow (TPF), the numerical fluid domain Ω_f can be decomposed into two separate subdomains: the water domain Ω_w and the air domain Ω_a, with $\Omega_f = \Omega_w \cup \Omega_a$. The boundary between the two-phases, $\Gamma_\sim = \Omega_w \cap \Omega_a$, is called the free surface. When floating bodies are involved, the solid domain Ω_s is introduced, leading to the global numerical domain $\Omega = \Omega_w \cup \Omega_a \cup \Omega_s$ with boundaries $\partial\Omega$, and the fluid–solid interface $\Gamma_{f\cap s}$. Figure III.1 illustrates the different domains and boundaries just mentioned, giving context to several of the governing equations derived in this chapter.

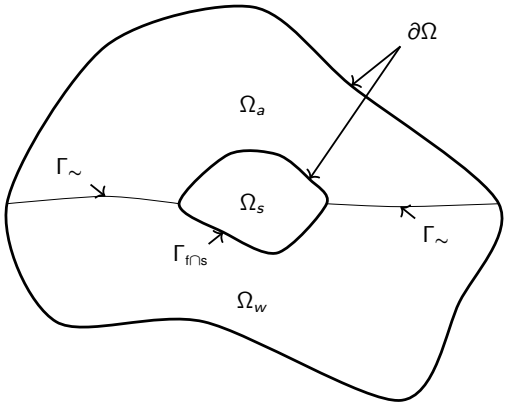


Figure III.1: Illustration of arbitrary domain for two-phase flow and solid phase

The theory behind TPF is introduced in section III.1 with the integral and differential forms of fluid flow equations, ultimately leading to the derivation of the Navier–Stokes Equations (NSEs) for incompressible fluids. Free surface tracking is described in section III.1.2 and, in section III.2, the different types of waves and the associated theory behind it, as well as the method for absorbing waves numerically, are described. Section III.3 presents the equations of motion for Multibody Dynamics (MBD) in the context of rigid bodies. Finally, moorings statics and dynamics are described in section III.4.2. The reader is reminded that all the theory here is related to the uncoupled models. For the coupling strategy and additional models necessary for the coupled simulation of FSI, refer to chapter IV.

III.1 Two-Phase Flow . . . 29
III.1.1 Governing Equations of Fluid Flow . . . 29
III.1.1.1 Transport Equation . . . 29
III.1.1.2 Conservation of Mass . . . 30
III.1.1.3 Conservation of Momentum . . . 31
III.1.1.4 Navier–Stokes Equations for Incompressible Flows . . . 31
III.1.2 Free Surface Representation and Tracking . . . 32
III.1.3 Finite Element Method (FEM) . . . 34
III.1.3.1 Weak Formulation . . . 35
III.1.3.2 Spatial Discretisation . . . 36
III.1.3.3 Temporal Discretisation . . . 37
III.1.4 Typical Boundary Conditions . . . 38
III.2 Wave Theory . . . 39
III.2.1 Governing Equations . . . 39
III.2.1.1 Regular Waves . . . 39
III.2.1.2 Irregular and Extreme Waves . . . 40
III.2.2 Wave Generation and Absorption . . . 41
III.3 Rigid Body Dynamics . . . 42
III.4 Mooring Modelling . . . 44
III.4.1 Mooring Statics and Quasi-Statics . . . 44
III.4.1.1 Fully Stretched Line . . . 46
III.4.1.2 Fully Lifted Line . . . 48
III.4.1.3 Partly Lifted Line . . . 49
III.4.1.4 Line with No Horizontal Span . . . 50
III.4.2 Mooring Dynamics . . . 50
III.4.2.1 Linear Elasticity for Isotropic Materials . . . 50
III.4.2.2 Beam Theory . . . 52
III.4.2.3 Absolute Nodal Coordinate Formulation (ANCF) . . . 53

III.1 Two-Phase Flow

The theory behind the TPF formulation used for this work is described in this section. Firstly, the governing equations of fluid flow are derived in section III.1.1. The representation of the free surface separating the two phases and its tracking method are then presented in section III.1.2. Spatial and temporal discretisation using the Finite Element Method (FEM) for solving those models is explained in section III.1.3, before listing typical boundary conditions used for TPF simulations in section III.1.4. Note that fluid dynamics and all models using the fluid domain are defined and simulated numerically using the Proteus toolkit.

III.1.1 Governing Equations of Fluid Flow

Considering a time varying control volume or domain $\Omega(t)$ that can be of any arbitrary size, any property (or quantity) of the fluid varies in space $\mathbf{x}$ and time t within $\Omega(t)$. This property can be defined by its extensive value (the actual quantity), $c(\mathbf{x}, t)$, or its intensive value (the amount of c per unit mass), $\phi(\mathbf{x}, t)$. They relate to each other with:

$$\phi(\mathbf{x}, t) = \frac{\mathrm{d}c(\mathbf{x}, t)}{\mathrm{d}m} \tag{III.1}$$

with m the mass. In integral form:

$$\int_{\Omega(t)} c(\mathbf{x}, t)\,\mathrm{d}\mathbf{x} = \int_{\Omega(t)} \rho\phi(\mathbf{x}, t)\,\mathrm{d}\mathbf{x} \tag{III.2}$$

with ρ the density. As $\Omega(t)$ can be of any size, considering an infinitesimally small control volume leads to the differential form of the relation between intensive and extensive property of the fluid:

$$c(\mathbf{x}, t) = \rho\phi(\mathbf{x}, t) \tag{III.3}$$

For convenience and clarity, the space and time dependency notations are dropped from here with $c = c(\mathbf{x}, t)$, $\phi = \phi(\mathbf{x}, t)$, and $\Omega = \Omega(t)$.

III.1.1.1 Transport Equation

This subsection describes the governing equations for the transport of a fluid property. The total rate of change of c in the control volume Ω is equal to the sum of the instantaneous change of c and the net flux (inflow minus outflow) across the boundary $\partial\Omega$ of Ω. This principle can be expressed in integral form with the Reynolds Transport Theorem (RTT):

$$\frac{\mathrm{d}}{\mathrm{d}t}\int_{\Omega} c\mathrm{d}\mathbf{x} = \int_{\Omega} \frac{\partial c}{\partial t}\mathrm{d}\mathbf{x} + \int_{\partial\Omega} c(\mathbf{u}\cdot\mathbf{n})\mathrm{d}\mathbf{x} \tag{III.4}$$

with $\mathbf{u}$ the fluid velocity, and $\mathbf{n}$ the unit normal vector pointing outwards from Ω on $\partial\Omega$. For reference, the RTT can also be written in terms of ϕ, simply as:

$$\frac{\mathrm{d}}{\mathrm{d}t}\int_{\Omega} \rho\phi\mathrm{d}\mathbf{x} = \int_{\Omega} \frac{\partial\rho\phi}{\partial t}\mathrm{d}\mathbf{x} + \int_{\partial\Omega} \rho\phi(\mathbf{u}\cdot\mathbf{n})\mathrm{d}\mathbf{x} \tag{III.5}$$

The transport of a property of the fluid can also be expressed in differential form. This is then called the Convection-Diffusion-Reaction (CDR) equation, which is also known as the generic transport equation. As its name indicates, the CDR equation introduces three distinct processes in order to describe the concentration of a fluid

property ϕ within a medium over time and space. Convection (also known as advection) is the movement induced by bulk motion of ϕ within a medium. In other terms, it is the transport of ϕ resulting from the movement of the surrounding medium itself. Diffusion is the natural tendency of reduction of gradients of ϕ within Ω. Given enough time for equilibrium to be reached, diffusion processes will result in a uniform spread of ϕ within Ω (uniform concentration). Note that as it is driven by the second law of thermodynamics, diffusion is an irreversible process. Finally, reaction is the creation or destruction of ϕ due to a range of phenomena that typically act as sources or sinks. The CDR equation combines these concepts together in a global transport equation expressed as:

$$\underbrace{\frac{\partial \rho\phi}{\partial t}}_{\text{Accumulation}} + \overbrace{\underbrace{\nabla \cdot (\mathbf{u}\rho\phi)}_{\text{Convection}} - \underbrace{\nabla \cdot (D\rho\nabla\phi)}_{\text{Diffusion}}}^{\text{Transport of } \phi} = \underbrace{R_\phi}_{\text{Reaction}} \tag{III.6}$$

with D the diffusion coefficient (also known as diffusivity), and R_ϕ the reaction term. The gain or loss of ϕ within Ω over time is called the accumulation. In eq. (III.6), it corresponds to the transient term $\frac{\partial \rho\phi}{\partial t}$, where $\frac{\partial \rho\phi}{\partial t} = 0$ leads to a steady-state problem and, by opposition, $\frac{\partial \rho\phi}{\partial t} \neq 0$ means that the problem is of unsteady nature (i.e. changing over time). As mentioned above, the reaction term R_ϕ describes the production (source with $R_\phi > 0$) or destruction (sink with $R_\phi < 0$) of ϕ within Ω. The convective term, $\nabla \cdot (\mathbf{u}\rho\phi)$, is of hyperbolic nature, where the direction of incoming information and the finite velocity at which this information is transported are defined by the convection velocity $\mathbf{u}$ directly in the vicinity of the point $\mathbf{x}$ considered. The diffusive term, $\nabla \cdot (D\rho\nabla\phi)$, is of elliptic nature, meaning that any point in the domain feels the influence of all other points instantaneously with information travelling at infinite velocity and in all directions. It is also common to introduce a flux term corresponding to the amount of ϕ that passes through Ω, or as the instantaneous rate of transport of ϕ at point $\mathbf{x}$ and time t. The flux $\mathbf{f}_\phi = \mathbf{f}(\phi)$ of ϕ encompassing both the convective and diffusive terms can be expressed as:

$$\mathbf{f}_\phi = \mathbf{u}\rho\phi - D\rho\nabla\phi \tag{III.7}$$

The CDR equation can therefore be expressed more succinctly in terms of $\mathbf{f}_\phi$ as:

$$\frac{\partial \rho\phi}{\partial t} + \nabla \cdot \mathbf{f}_\phi = R_\phi \tag{III.8}$$

III.1.1.2 Conservation of Mass

Within an enclosed control volume Ω, the fundamental law of conservation of mass applies to the fluid, which states that the total mass m must remain constant through time. In other terms, this means that no quantity can be added or removed. By definition, mass conservation within Ω in integral form is:

$$\frac{\mathrm{d}m}{\mathrm{d}t} = \frac{\mathrm{d}}{\mathrm{d}t}\left(\int_\Omega \rho d\mathbf{x}\right) = 0 \tag{III.9}$$

Using the RTT of eq. (III.4), and setting the extensive property of interest as the mass itself $c = m$, which leads to an intensive property $\phi = \frac{\mathrm{d}m}{\mathrm{d}m} = 1$, the integral form of the equation of mass conservation can also be expressed from eq. (III.5) as:

$$\int_\Omega \dot{\rho} + \int_{\partial\Omega} \rho(\mathbf{u} \cdot \mathbf{n}) ds = 0 \tag{III.10}$$

with $\dot{\rho} = \frac{\partial \rho}{\partial t}$. The equivalent differential form can be found by either considering an infinitesimally small Ω from eq. (III.10), or directly using the CDR equation (eq. (III.6)) with $c = \rho$, leading to $\phi = 1$, and without source or sink

($R_\phi = 0$). The differential form of the equation of mass conservation, which is known as the continuity equation, is therefore expressed as follows:

$$\dot{\rho} + \nabla \cdot (\rho \mathbf{u}) = 0 \tag{III.11}$$

III.1.1.3 Conservation of Momentum

As stated by Newton's second law, the rate of change of momentum of the fluid is proportional to the sum of forces $\mathbf{f}$ applied to it, which can be represented symbolically by:

$$\frac{\mathrm{d}}{\mathrm{d}t} \int_\Omega \rho \phi \mathrm{d}\mathbf{x} = \sum \mathbf{f} \tag{III.12}$$

where the external forces $\sum \mathbf{f}$ for a fluid are composed of the body forces (e.g. gravitational forces) and surface forces (e.g. pressure and shear forces). In integral form, the equation for conservation of momentum for such fluid is expressed as:

$$\frac{\mathrm{d}}{\mathrm{d}t} \int_\Omega \rho \phi \mathrm{d}\mathbf{x} = \int_\Omega \rho \mathbf{g} \mathrm{d}\mathbf{x} + \int_{\partial\Omega} \bar{\bar{\sigma}} \cdot \mathbf{n} \mathrm{d}\mathbf{x} \tag{III.13}$$

with $\mathbf{g}$ the gravitational acceleration, and $\bar{\bar{\sigma}}$ the Cauchy–Shwartz stress tensor. As can be seen in eq. (III.13), the body force component corresponds to the gravitational force exerted on the fluid in Ω, and the surface forces are represented with a stress vector field of $\bar{\bar{\sigma}} \cdot \mathbf{n}$ acting along $\partial\Omega$ (with $\mathbf{n}$ the unit normal vector pointing outwards of the boundary). For a Newtonian fluid, $\bar{\bar{\sigma}}$ encompasses the pressure and viscous forces with:

$$\bar{\bar{\sigma}} = -p\bar{\bar{\mathbf{I}}} + 2\mu\bar{\bar{\epsilon}} - \frac{2}{3}\mu (\nabla \cdot \mathbf{u}) \bar{\bar{\mathbf{I}}} \tag{III.14}$$

where μ is the dynamic viscosity of the fluid, $\bar{\bar{\mathbf{I}}}$ is the identity tensor and $\bar{\bar{\epsilon}}$ is the strain rate tensor defined by:

$$\bar{\bar{\epsilon}} = \frac{1}{2} (\nabla \mathbf{u} + \nabla \mathbf{u}^\intercal) \tag{III.15}$$

When using the divergence theorem on the surface integral of the of eq. (III.13), and choosing an infinitesimally small domain Ω, the conservation of momentum can be expressed in differential form as:

$$\frac{\partial \rho \mathbf{u}}{\partial t} + \nabla \cdot (\rho \mathbf{u} \otimes \mathbf{u}) - \nabla \cdot \bar{\bar{\sigma}} = \rho \mathbf{g} \tag{III.16}$$

with $\otimes$ the outer product.

III.1.1.4 Navier–Stokes Equations for Incompressible Flows

After the introduction of the fundamental principles described above, the system of governing equations for fluid flow can be described with the NSEs. As the fluid considered here is assumed incompressible, several simplifications can be made on the laws of conservation. Firstly, the continuity equation eq. (III.11) can be simplified as follows when ρ is constant over time:

$$\nabla \cdot \mathbf{u} = 0 \tag{III.17}$$

Secondly, when using this simplified continuity equation above and applying it to eq. (III.14), the Newtonian fluid stress tensor reduces to:

$$\bar{\bar{\sigma}} = -p\bar{\bar{\mathbf{I}}} + 2\mu\bar{\bar{\epsilon}} \tag{III.18}$$

The system of NSEs is obtained by regrouping the mass (eq. (III.11)) and momentum (eq. (III.16)) conservation laws, which are as follows:

$$\begin{cases} \dfrac{\partial \rho \mathbf{u}}{\partial t} + \nabla \cdot (\rho \mathbf{u} \otimes \mathbf{u}) - \nabla \cdot \bar{\bar{\boldsymbol{\sigma}}} = \rho \mathbf{g} \\ \dot{\rho} + \nabla \cdot (\rho \mathbf{u}) = 0 \end{cases} \tag{III.19}$$

For incompressible Single-Phase Flow (SPF) where the density is constant, using eq. (III.17) allows for reducing eq. (III.19) to the following:

$$\begin{cases} \rho \dot{\mathbf{u}} + \rho \mathbf{u} \cdot \nabla \mathbf{u} + \nabla p - \nabla \cdot \bar{\bar{\boldsymbol{\tau}}} = \rho \mathbf{g} \\ \nabla \cdot \mathbf{u} = 0 \end{cases} \tag{III.20}$$

with $\bar{\bar{\boldsymbol{\tau}}} = 2\mu\bar{\bar{\epsilon}}$ the viscous shear stress tensor. When viscosity is constant, further simplification of eq. (III.21) is possible with $\nabla \cdot \bar{\bar{\boldsymbol{\tau}}} = \mu \Delta \mathbf{u}$, leading to:

$$\begin{cases} \dot{\mathbf{u}} + \mathbf{u} \cdot \nabla \mathbf{u} + \dfrac{1}{\rho} \nabla p - \nu \Delta \mathbf{u} = \mathbf{g} \\ \nabla \cdot \mathbf{u} = 0 \end{cases} \tag{III.21}$$

where $\nu = \frac{\mu}{\rho}$ is the kinematic viscosity. This does not apply to Multiphase Flow (MPF) due to the existence of density gradients at the free surface. The different components necessary for building the system of NSEs (momentum and continuity) can also be obtained directly from the CDR equation eq. (III.6), as shown in table III.1.

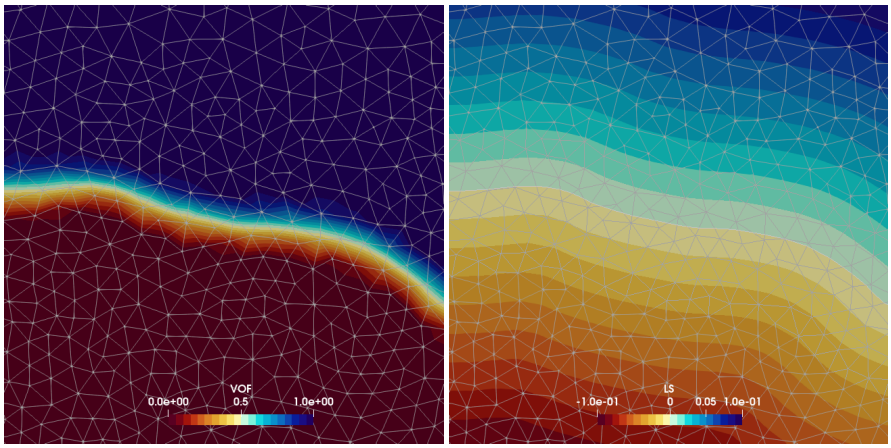
III.1.2 Free Surface Representation and Tracking

In TPF simulations, the free surface $\Gamma_\sim = \Omega_w \cap \Omega_a$ separating the two fluid phases is transported and tracked in an explicit or implicit fashion. The explicit method implies the separation of the two phases with the inclusion of an explicit boundary that deforms according to kinematics and dynamics of free surface motion. While generally more accurate due to the mesh conformity of the free surface boundary, explicit methods become inconvenient when the free surface gets deformed significantly as it can lead to mesh entanglement (e.g. in the case of breaking waves). These issues are non-existent when using an implicit method, which was therefore selected for tracking the free surface in the applications presented in this thesis.

The implicit representation of the free surface is done with a property that takes different values according to the phase in which it belongs. An example of such property to represent the free surface $\Gamma_\sim$ implicitly is the Volume of Fluid (VOF) [70] denoted $\widehat{\theta}$ here. When describing two phases such as air and water, the VOF takes a different value for each phase, for example: $\widehat{\theta} = 1$ for the air phase, $\widehat{\theta} = 0$ for the water phase, and $\widehat{\theta} = \frac{1}{2}$ for the free surface. The VOF representation is therefore discontinuous, which can lead to numerical difficulties. Another way

Table III.1: Values of variables to recover NSEs from the CDR equation eq. (III.6)

Equation	ϕ	D	R_ϕ
Continuity	1	0	0
x-momentum	u_1	μ	$-\frac{\partial p}{\partial x_1} + \rho g_1$
y-momentum	u_2	μ	$-\frac{\partial p}{\partial x_2} + \rho g_2$
z-momentum	u_3	μ	$-\frac{\partial p}{\partial x_3} + \rho g_3$
momentum (vector)	$\mathbf{u}$	μ	$-\nabla p + \rho \mathbf{g}$



(a) VOF (b) Level Set (signed distance)

Figure III.2: Values taken by VOF and level set around implicit boundary

to represent implicitly the free surface and the different phases in a continuous manner is the Level Set (LS) method [104, 118] that uses a signed distance function ϕ_{sdf}. The free surface is represented by the isosurface given by $\phi_{\text{sdf}} = 0$, while the value of the signed distance function at an arbitrary point $\mathbf{x}$ gives the actual distance from $\Gamma_\sim$ to $\mathbf{x}$. The sign of ϕ_{sdf} is used to represent in which phase $\mathbf{x}$ is placed with, in our case, $\phi_{\text{sdf}} < 0$ for the water phase and $\phi_{\text{sdf}} > 0$ for the air phase. In summary, the free surface and different phases can be represented implicitly with the VOF or LS with the following set of rules:

$$\begin{cases} 0 \leq \widehat{\theta} < \frac{1}{2}, & \phi_{\text{sdf}} < 0 \implies \mathbf{x} \in \Omega_w \\ \widehat{\theta} = \frac{1}{2}, & \phi_{\text{sdf}} = 0 \implies \mathbf{x} \in \Gamma_\sim \\ \frac{1}{2} < \widehat{\theta} \leq 1, & \phi_{\text{sdf}} > 0 \implies \mathbf{x} \in \Omega_a \end{cases} \tag{III.22}$$

A snapshot of the VOF and LS values around an arbitrary free surface is shown in section III.1.2.

As mentioned above, the property representing the free surface implicitly must be transported and tracked over time. The tracking technique described below was first introduced and implemented in [82], and is the one used for all the TPF applications presented here. This method tackles the problem of mass conservation that can be encountered when dealing with time-evolving LS methods. As the VOF representation described above is discontinuous, a smoothed (or regularised) heaviside function θ_ϵ is defined to allow a smooth transition between the two phases:

$$\widehat{\theta} = \theta_\epsilon(\phi_{\text{sdf}}) = \begin{cases} 0 & \phi_{\text{sdf}} < \text{-}\epsilon_s \\ \frac{1}{2}\left(1 + \frac{\phi_{\text{sdf}}}{\epsilon_s} + \frac{1}{\pi}\sin\left(\frac{\phi\pi}{\epsilon_s}\right)\right) & |\phi_{\text{sdf}}| \leq \epsilon_s \\ 1 & \phi_{\text{sdf}} > \epsilon_s \end{cases} \tag{III.23}$$

with ϵ_s a user defined parameter. For the work presented here, $\epsilon_s = 1.5h_e$ was found to be sufficiently smooth, h_e being the characteristic mesh element size. When $0 < \widehat{\theta} < 1$, the cell is assumed to contain a mixture of the two fluids. With this smooth representation of the free surface, the VOF is first displaced over time using a hyperbolic transport equation:

$$\dot{\widehat{\theta}} + \nabla \cdot \left(\mathbf{u}\widehat{\theta}\right) = 0 \tag{III.24}$$

Meanwhile, the LS is transported with the following non-conservative convection equation:

$$\frac{\partial \phi_{\text{sdf}}}{\partial t} + \mathbf{u} \cdot \nabla \phi_{\text{sdf}} = 0 \tag{III.25}$$

Once the free surface has evolved, ϕ_{sdf} usually does not remain a true signed distance function: $\|\nabla \phi_{\text{sdf}}\| \neq 1$ near the free surface. The following eikonal equation must therefore be satisfied to recover the properties of a signed distance function on ϕ_{sdf}:

$$\begin{cases} \|\nabla \phi_{\text{sdf}}\| = 1 & \forall \mathbf{x} \in \Omega \setminus \Gamma_{\sim} \\ \phi_{\text{sdf}} = 0 & \forall \mathbf{x} \in \Gamma_{\sim} \end{cases} \tag{III.26}$$

To recover such characteristics, the following equation can be solved [118]:

$$\frac{\partial \phi_{\text{sdf}}}{\partial t} + \text{sign}(\phi_{\text{sdf}}) \left(\|\nabla \phi_{\text{sdf}}\| - 1 \right) = 0 \qquad \forall \mathbf{x} \in \Omega \setminus \Gamma_{\sim} \tag{III.27}$$

Although eq. (III.27) can be solved several times if needed to reach a steady-state, only one iteration is usually needed as ϕ_{sdf} is typically only slightly deteriorated around the free surface. Because a non-conservative convection equation was used to transport ϕ_{sdf}, mass was not conserved. A correction must therefore be calculated and applied to ϕ_{sdf} using the position of the free surface as defined by the mass conservative solution of the VOF from eq. (III.24):

$$\begin{cases} \theta_{\epsilon} \left(\phi_{\text{sdf}} + \phi'_{\text{sdf}} \right) - \widehat{\theta} = \kappa_u \Delta \phi'_{\text{sdf}} & \forall \mathbf{x} \in \Omega \\ \nabla \phi'_{\text{sdf}} \cdot \mathbf{n} = 0 & \forall \mathbf{x} \in \partial \Omega \end{cases} \tag{III.28}$$

with κ_u a parameter penalising the deviation of ϕ'_{sdf} from a global constant. Finally, both the VOF and level set are updated to their final value:

$$\widehat{\theta} = \theta_{\epsilon}(\phi_{\text{sdf}} + \phi'_{\text{sdf}}) \tag{III.29}$$

$$\phi_{\text{sdf}} \coloneqq \phi_{\text{sdf}} + \phi'_{\text{sdf}} \tag{III.30}$$

Following the steps described above that consist in transport of VOF and LS from eq. (III.24) and eq. (III.25), mass correction of eq. (III.28), and redistancing of eqs. (III.29) and (III.30), mass conservation is achieved when deforming the free surface.

■ III.1.3 Finite Element Method (FEM)

In the context of this research, the FEM method is used for solving the Navier-Stokes equations and free surface tracking model, as well as many of the models described in subsequent chapters such as mesh motion with linear elastostatics and added mass estimation. To illustrate the method, and because all the governing equations described above can be derived from the CDR equation, eq. (III.6) is used here assuming a constant density as is the case in incompressible flows, leading to:

$$\dot{\phi} + \nabla \cdot (\mathbf{u}\phi) - \nabla \cdot (D \nabla \phi) = R_{\rho} \tag{III.31}$$

with $R_{\rho} = \frac{R}{\rho}$ for ease of notation.

III.1.3.1 Weak Formulation

FEM is a residual based method that gives a controlled approximation φ to the true solution ϕ. The residual of the solution φ from eq. (III.31) can be expressed as:

$$\mathscr{R}(\varphi) = \dot{\varphi} + \nabla \cdot (\mathbf{u}\varphi - D\nabla\varphi) - R_\rho \tag{III.32}$$

It is clear that with $\mathscr{R}(\varphi) = 0$, we have $\varphi = \phi$. As it is usually not possible to retrieve the exact solution, the aim of FEM is to find a solution for which $\mathscr{R}(\varphi) < \epsilon_{\text{tol}}$ with ϵ_{tol} the user defined tolerance that essentially controls the accuracy of the approximate solution. Because second derivatives appear in eq. (III.32), it is expected that the first and second order derivatives of φ are continuous in Ω. In other terms, it is expected that $\varphi \in \mathscr{C}^2$, with $\mathscr{C}^k$ the set of real-valued functions that are continuous in Ω up to their derivatives of order k:

$$\mathscr{C}^k(\Omega) = \left\{ \varphi \colon \Omega \to \mathbb{R} \colon \int_\Omega w^k < \infty \right\} \tag{III.33}$$

Due to this restrictive requirement on φ, eq. (III.31) is known as the strong form of the Partial Differential Equation (PDE). In FEM, the PDE is converted to integral form and multiplied by a chosen test function w (also known as weight function) to be expressed in what is known as the weak or variational form of the PDE. The residual in the weak form is expressed as follows, and is known as the method of weighted residuals:

$$\int_\Omega w\mathscr{R}(\varphi)\,\mathrm{d}\mathbf{x} = 0 \tag{III.34}$$

$$\int_\Omega (w\dot{\varphi} - \nabla w \cdot (\mathbf{u}\varphi - D\nabla\varphi) - wR_\rho)\,\mathrm{d}\mathbf{x} + \int_{\partial\Omega} w(\mathbf{u}\varphi - D\nabla\varphi)\cdot\mathbf{n}\mathrm{d}\mathbf{x} = 0 \tag{III.35}$$

where Green's formula (eq. (III.36)) is used on the flux term of eq. (III.35) to remove second derivatives with:

$$\int_\Omega w\nabla\cdot(\mathbf{u}\varphi - D\nabla\varphi)\,\mathrm{d}\mathbf{x} = -\int_\Omega \nabla w\cdot(\mathbf{u}\varphi - D\nabla\varphi)\,\mathrm{d}\mathbf{x} + \int_{\partial\Omega} w(\mathbf{u}\varphi - D\nabla\varphi)\cdot\mathbf{n}\mathrm{d}\mathbf{x} \tag{III.36}$$

The weak form allows what is called relaxation/relaxing of the PDE, by leading to less constraining (or "weaker") requirements on the solution φ. The test function w must fit some criteria set by its space of admissible test functions $\mathscr{W}$, with $w \in \mathscr{W}$:

$$\mathscr{W}(\Omega) = \left\{ w \in \mathscr{H}^1(\Omega) : w = 0 \quad \forall \mathbf{x} \in \Gamma_D \right\} \tag{III.37}$$

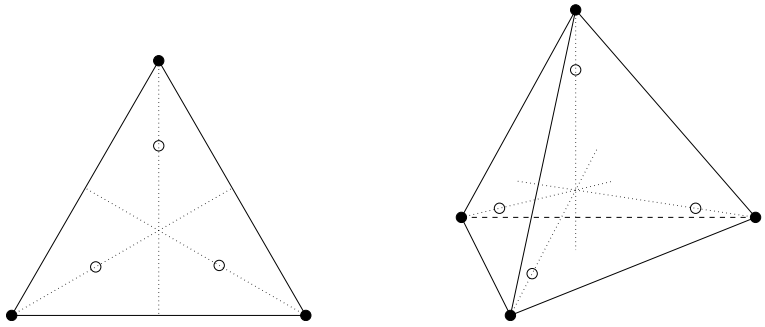
with Γ_D the partial (or total) boundary where Dirichlet conditions are defined over $\partial\Omega$ meaning that w must vanish on Γ_D (see section III.1.4), and where $\mathscr{H}^1(\Omega)$ is the Sobolev space:

$$\mathscr{H}^1(\Omega) = \left\{ w \in \mathscr{L}^2(\Omega),\ \nabla w \in \mathscr{L}^2(\Omega) \right\} \tag{III.38}$$

which requires that w and its gradient ∇w are in the Lebesgue space $\mathscr{L}^2(\Omega)$:

$$\mathscr{L}^2(\Omega) = \left\{ w \colon \Omega \to \mathbb{R} \colon \int_\Omega w^2 < \infty \right\} \tag{III.39}$$

that in turn requires that w and ∇w have finite integrals over Ω. In other terms, the test function and its first derivative are required to be bounded in Ω. The approximation of the solution φ, called the trial function in this context, must also have a suitable function space $\mathscr{V}$, with $\varphi \in \mathscr{V}$ and $\mathscr{V} = \mathscr{H}^1$. To summarise, the reason that eq. (III.35) is called the weak form (while eq. (III.6) would be the equivalent strong form) is that second order partial derivatives are not present after integration, and that $\mathscr{R}(\varphi)$ (or rather $w\mathscr{R}(\varphi)$) no longer needs to be continuous anymore but only integrable over Ω.



(a) Triangle (b) Tetrahedron

Figure III.3: Lagrange elements for FEM
●: node, ○: quadrature point

III.1.3.2 Spatial Discretisation

The problem is then discretised spatially in order to be solved numerically under the FEM framework. This process is known as the triangulation or meshing of the domain Ω, which produces a finite element space composed of elements such as triangles or tetrahedra as shown in fig. III.3. The triangulated domain is denoted Ω_h and is composed of $n_{\mathcal{E}}$ elements $\mathcal{E}_i \in \Omega_h$, and $n_{\mathcal{N}}$ nodes $\mathcal{N}_i \in \Omega_h$. The elements covering the whole domain must be full-dimensional cells ($\mathring{\mathcal{E}} \neq \varnothing$), non-overlapping ($\mathring{\mathcal{E}}_i \cap \mathring{\mathcal{E}}_j = \emptyset \quad \forall \mathcal{E}_i, \mathcal{E}_j \in \Omega_h \wedge \mathcal{E}_i \neq \mathcal{E}_j$), and conform, meaning that if the intersection between two different elements is not zero, then it must be a face, edge, or vertex.

The trial function, or weak solution, φ of the continuous variational problem must also be discretised in order to be solved numerically by approximating φ with a discrete solution $\varphi_h \in \mathscr{V}_h \subset \mathscr{V}$. For a Boundary Value Problem (BVP) problem solved within an FEM framework, φ_h should be equal to the exact solution φ_i at what is known as the Degrees of Freedom (DOFs) of Ω_h. For linear Lagrange (P1) elements, such as those presented in fig. III.3, the DOFs coincide with the set of nodes $\{\mathcal{N}_1, \cdots, \mathcal{N}n_{\mathcal{N}}\}$ of Ω_h. For any point in the finite element space $\mathscr{V}_h$ that does not coincide with a DOF of Ω_h, basis functions Φ spanning $\mathscr{V}_h$ with $\{\Phi_1, \Phi_2, \cdots, \Phi_{n_{\mathcal{N}}}\}$ are used to interpolate the solution. The discrete solution φ_h is expressed in terms of basis function Φ as:

$$\varphi_h(\mathbf{x}) = \sum_{j=1}^{n_{\text{DOF}}} \varphi_j \Phi_j(\mathbf{x}) \tag{III.40}$$

where the values $\varphi_1, \cdots, \varphi_{n_{\text{DOF}}}$ are the values of φ at the DOFs. The nodal or Lagrange basis function Φ, with $n_{\text{DOF}} = n_{\mathcal{N}}$, must have compact support with $\Phi_i = 1$ at $\mathcal{N}_i$, and $\Phi_i = 0$ at $\mathcal{N}_j$ with $\mathcal{N}_j \neq \mathcal{N}_i$, meaning that Φ_i is compactly supported within the patch of elements surrounding the node $\mathcal{N}_i$. It must also be continuous with smoothness $\mathscr{C}^0$ within an element. In fact, Φ_i is only discontinuous exactly at $\mathcal{N}_i$, and continuous within each element $\mathcal{E} \in \{\mathcal{E} : \mathcal{N}_i \in \mathcal{E}\}$. See fig. III.4 for an illustration of such a basis function.

The test function w is also discretised into a finite element space $\mathscr{W}_h$ with $w_h \in \mathscr{W}_h \subset \mathscr{W}$. The basis functions Ψ spanning $\mathscr{W}_h$ are actually used as test functions themselves ($w_h = \Psi$). When $\Psi = \Phi$, as is the case for all simulations presented in this thesis, the method is called the Galerkin method, as opposed to the Petrov–Galerkin method when $\Psi \neq \Phi$.

Note that for the CDR equation, the gradient of the solution must be also computed. When using linear Lagrange elements, $\nabla\varphi_h$ is a discontinuous vector field due to the piecewise continuous nature of the scalar field φ_h.

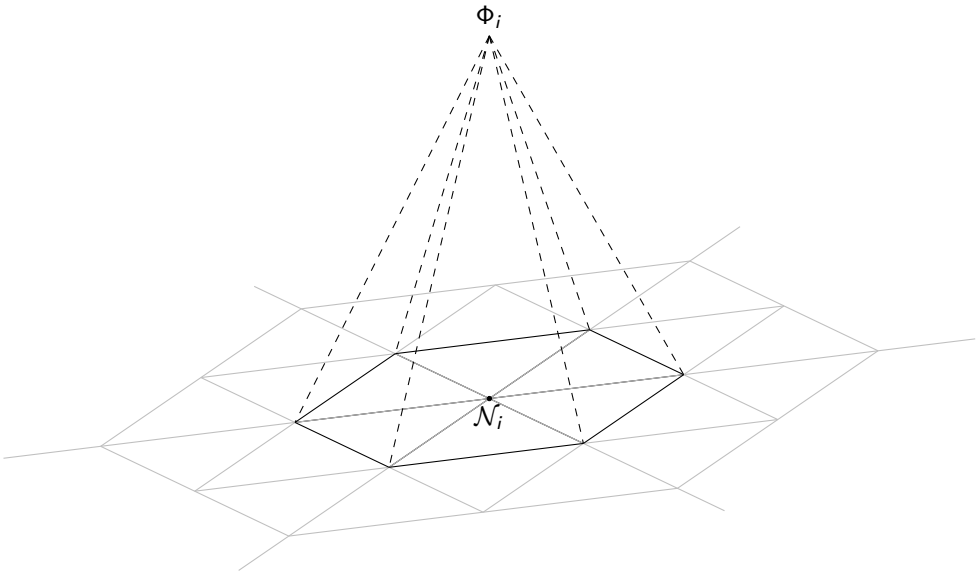


Figure III.4: Piecewise-linear finite element basis function Φ_i on a triangular mesh around node $\mathcal{N}_i$

Gradients can however be recovered at the DOFs of the mesh if necessary: for Lagrange elements, gradients at a node $\mathcal{N}_i$ can be obtained by averaging the gradients inside elements around $\mathcal{N}_i$, either with a simple average or with a weighted average of the gradients (e.g. according to the volume of elements). Once those gradients have been recovered, it is possible to smooth the value of the gradient within elements by using:

$$\nabla\varphi_h(\mathbf{x}) = \sum \nabla\varphi_j \Phi_j(\mathbf{x}) \tag{III.41}$$

The spatial discretisation used for all simulations presented in this thesis is as follows: triangles in 2D and tetrahedra in 3D, using P1 Lagrange elements with affine linear nodal basis and simplex Gaussian quadrature.

III.1.3.3 Temporal Discretisation

When solving for a system that is not in steady-state, the PDE must also be discretised in time. Usually, time t is discretised with 1D Finite Difference Method (FDM):

$$t \in \left[0 = t^{(0)}, t^{(1)}, \cdots, t^{(N-1)}, t^{(N)} = T_{\text{sim}}\right] \tag{III.42}$$

with $t^{(n)}$ the value of time at time step n, and $N+1$ the total number of time steps. Note that throughout this thesis, the value of a variable at time $t^{(n)}$ is written with a superscript $^{(n)}$ (e.g. $\varphi^{(n)} = \varphi\left(t^{(n)}\right)$). Different techniques exist for time-stepping, which can be divided in two main categories: explicit (e.g. Forward Euler as in eq. (III.43)) and implicit (e.g. backward Euler as in eq. (III.44)):

$$\frac{\varphi^{(n+1)} - \varphi^{(n)}}{\Delta t} = \mathscr{R}\left(\varphi^{(n)}\right) \tag{III.43}$$

$$\frac{\varphi^{(n+1)} - \varphi^{(n)}}{\Delta t} = \mathscr{R}\left(\varphi^{(n+1)}\right) \tag{III.44}$$

Higher-order methods also exist such as Runge–Kutta. For all simulations presented here however, the time discretisation method used for the fluid is the first-order accurate implicit backward Euler (eq. (III.44)) due to its simplicity

and relatively low computational cost. The time step $\Delta t = t^{(n+1)} - t^{(n)}$ can be fixed to a specific value, or controlled dynamically by using the following Courant–Friedrichs–Lewy (CFL) condition:

$$\text{CFL} = u_k \frac{\Delta t}{\Delta x} \leq \text{CFL}_{\text{max}} \tag{III.45}$$

with u_k the speed in the direction k and Δx a distance interval (usually the characteristic length of an element). For stability, it is ensured that $\text{CFL}_{\text{max}} < 1$, meaning that the fluid cannot travel a distance that is longer than one element within a time step. This condition is used in most of the FSI simulations presented here, as it allows for a dynamically accurate time discretisation.

■ III.1.4 Typical Boundary Conditions

There are two main types of boundary conditions: Dirichlet conditions and Neumann conditions. Dirichlet conditions impose specified values directly to the solution at the boundary. In the context of the CDR equation (eq. (III.6)), it specifies the value of ϕ at the boundary. Neumann boundary conditions impose values to the derivative (or gradient) of the solution, corresponding to $\nabla\phi$ in the CDR equation. When both Neumann and Dirichlet boundary conditions are used on the same boundary, they are named mixed, Newton, Robin, or Cauchy boundary conditions, amongst others and depending on the conditions imposed. In the context of the TPF solver used for the simulations presented here, boundary conditions can be imposed as Dirichlet conditions or on different components of the CDR: the advective term ($\nabla \cdot (\rho \mathbf{u} \phi)$) or the diffusive term ($\nabla \cdot (D \nabla \phi)$). The most typical boundary conditions used for the TPF presented in this thesis are described in more details below.

Boundary conditions on solid, material boundaries are traditionally imposed as no-slip or free-slip conditions. No-slip condition assumes that the fluid velocity is the same as the velocity of the boundary. For a non-moving boundary, this simply corresponds to Dirichlet conditions of $\mathbf{u} = 0$. Free-slip (sometimes simply called slip) condition only requires the normal component of the velocity of the fluid to be the same as the velocity of the boundary, and assumes that there is no friction between the solid boundary and the fluid. For a non-moving boundary, this is equivalent to $\mathbf{u} \cdot \mathbf{n} = 0$. Usually, free-slip conditions are used when viscous effects are considered negligible or undesirable, or if the mesh size is too large at the boundary to capture the boundary layer effects. In most cases, with the right mesh refinement, no-slip boundary conditions are preferred as they more accurately represent physical behaviour of most engineering applications.

In order to let the fluid escape the domain, atmospheric boundary conditions are commonly applied. In this case, the pressure p and VOF $\widehat{\theta}$ are imposed as Dirichlet conditions. In typical engineering applications where

Table III.2: Set of typical boundary conditions used in TPF simulations presented in this thesis

	dirichlet					advective					diffusive		
	u_1	u_2	u_3	p	$\widehat{\theta}$	u_1	u_2	u_3	p	$\widehat{\theta}$	u_1	u_2	u_3
Free-slip	—	—	—	—	—	0	0	0	—	0	0	0	0
No-slip	0	0	0	—	—	—	—	—	—	0	—	—	—
Atmosphere	0	0	—	0	1	—	—	—	—	—	—	—	0
Wave	f	f	f	f	f	—	—	—	—	—	—	—	—
Non-material	—	—	—	—	—	—	—	—	—	0	0	0	0

f: custom function (defined by user); —: no condition applied
Note: the atmosphere boundary is assumed to be perpendicular to the constant $\mathbf{g} = (0, 0, -9.81)$

the atmosphere boundary is perpendicular to the gravitational acceleration $\mathbf{g}$, it is common to set $p = 0$ (only the relative pressure in Ω_f matters) and the value corresponding to air for the VOF ($\widehat{\theta} = 1$ in our case). The latter ensures that any fluid escaping the domain is replaced by air.

Non-material boundary conditions are imposed on boundaries within Ω_f that are not supposed to be physical boundaries. For example, when relaxation zones are introduced (see section III.2.2), the boundary separating the sponge layer and the rest of the domain is non-material. When the simulation is parallelised, where each processor owns a subdomain of Ω_f, the boundary between each subdomain is also non-material. In this case, the diffusive component of the fluid velocity and advective term of the VOF are set to zero.

The different types of boundary conditions described above are compiled in table III.2, showing the values taken for the Dirichlet, advective, and diffusive components of all fluid variables.

III.2 Wave Theory

In this section, the governing equations of wave dynamics are derived in section III.2.1. These equations concern linear and nonlinear regular waves, as well as spectral waves (random and focused). The method to generate and absorb waves with relaxation zones is then described in section III.2.2.

III.2.1 Governing Equations

III.2.1.1 Regular Waves

Regular waves, also known as monochromatic waves, can be divided in two main categories: linear and nonlinear waves. A wave is defined as linear if its steepness is small: $\frac{H}{\lambda} \ll 1$, with H the wave height, and λ the wavelength; and if the water is deep: $\frac{\lambda}{h_{\text{mwl}}} \ll 1$, with h_{mwl} the water depth. If these conditions apply, Airy wave theory (which is derived from potential flow theory) is used to describe the wave kinematics. The wave elevation profile of a linear wave is represented as a sinusoidal signal:

$$\eta = a\cos(kx_1 - \omega t + \phi) \tag{III.46}$$

with η the wave elevation at a point x_1 along the direction of propagation of the wave at a time t, $a = \frac{H}{2}$ the wave amplitude, $k = \frac{2\pi}{\lambda}$ the wavenumber, ω the angular frequency of the wave, and ϕ the wave phase. The fluid velocity potential Φ, related to the fluid velocity with $\mathbf{u} = \nabla\Phi$, is expressed as:

$$\Phi = \frac{\omega}{k} a \frac{\cosh(kx_3)}{\sinh(kh_{\text{mwl}})} \sin(kx_1 - \omega t + \phi) \tag{III.47}$$

with x_3 the vertical coordinates of the point considered, assuming that $x_3 = 0$ at the seabed and $x_3 = h_{\text{mwl}}$ at the mean surface level. The horizontal velocity u_1, and vertical velocity u_3 can therefore be calculated as follows:

$$u_1 = \omega a \frac{\cosh(kx_3)}{\sinh(kh_{\text{mwl}})} \cos(kx_1 - \omega t + \phi) \tag{III.48}$$

$$u_3 = \omega a \frac{\sinh(kx_3)}{\sinh(kh_{\text{mwl}})} \sin(kx_1 - \omega t + \phi) \tag{III.49}$$

As k is unknown, the dispersion equation for linear waves can be used as a relation between wavelength, water depth, and gravitational acceleration:

$$\lambda = \frac{gT^2}{2\pi} \tanh\left(2\pi \frac{h_{\text{mwl}}}{\lambda}\right) \tag{III.50}$$

with T the wave period, and g the gravitational acceleration.

As previously mentioned, the equations described above for wave profile and kinematics are only valid for regular linear waves. When the wave steepness becomes large, the wave profile cannot be represented by a single sinusoidal signal as nonlinear waves feature a higher and sharper crest with a shallower and flatter trough which is determined by its wave period, wave height, and the water depth. The order to nonlinearity up to the wave breaking criteria can be estimated according to the wave classification of [88]. A common technique to describe a nonlinear waves is the Fenton Fourier Transform theory [47, 46], where the solution is found with an iterative process and is expressed as a Fourier series. With this technique, the wave elevation and velocity are expressed as:

$$\eta = \sum_{i=1}^{n_F} A_i \cos\left(i(kx_1 - \omega t) + \phi\right) \tag{III.51}$$

$$u_1 = \frac{\omega}{k} - \bar{U} + \sqrt{\frac{g}{k}} \sum_{i=1}^{n_F} iB_i \frac{\cosh\left(ikx_3\right)}{\cosh\left(ikh_{\text{mwl}}\right)} \cos\left(i(kx_1 - \omega t) + \phi\right) \tag{III.52}$$

$$u_3 = \sqrt{\frac{g}{k}} \sum_{i=1}^{n_F} iB_i \frac{\sinh\left(ikx_3\right)}{\cosh\left(ikh_{\text{mwl}}\right)} \sin\left(i(kx_1 - \omega t) + \phi\right) \tag{III.53}$$

with A_i and B_i the ith Fourier series coefficients for wave amplitude and velocity, respectively, n_F the number of Fourier series coefficients, and $\bar{U}$ is the mean horizontal fluid velocity at any level.

III.2.1.2 Irregular and Extreme Waves

Irregular waves, also called random waves, are used to describe waves that can be encountered in real sea states. Because irregular waves are a stochastic process, they are described through statistical properties (significant wave height, peak period) and spectral properties (spectral energy distribution). The wave profiles are then reconstructed by combining linear components of the spectral distribution according to a chosen number of frequencies. A commonly used description of random sea states can be obtained with the Joint North Sea Wave Project (JONSWAP) spectrum for deep and intermediate waves that is expressed as follows [68]:

$$S_p(f_i) = \beta_J H_s^2 T_p^{-4} f_i^{-5} e^{-\frac{5}{4}(T_p f_i)^{-4}} \cdot \gamma^{e^{\frac{-(f_i T_p - 1)^2}{2\sigma^2}}} \tag{III.54}$$

where T_p is the peak period, T_i is the component period, f_i is the component frequency, H_s is the significant wave height, γ is the peak enhancement factor (influencing the sharpness of the spectral peak), with usually $1 < \gamma < 7$ (mean 3.3) and with:

$$\sigma = \begin{cases} 0.07 & T_i \geq T_p \\ 0.09 & T_i \leq T_p \end{cases} \tag{III.55}$$

and [58]:

$$\beta_J = \frac{0.0624(1.094 - 0.01915 \ln \gamma)}{0.230 + 0.0336\gamma - 0.185(1.9 + \gamma)^{-1}} \tag{III.56}$$

Note that other spectral descriptions can be used for random waves such as the Pierson–Moskovitz spectrum, from which the JONSWAP spectrum is derived. Once the spectral distribution has been generated with the chosen

parameters, the free surface elevation and water velocity are reconstructed using:

$$\eta = \sum_{i=1}^{n_f} a_i \cos\left(k_i x_1 - \omega_i t + \phi_i\right) \tag{III.57}$$

$$u_1 = \sum_{i=1}^{n_f} \omega_i a_i \frac{\cosh\left(k_i x_3\right)}{\sinh\left(k_i h_{\text{mwl}}\right)} \cos\left(k_i x_1 - \omega_i t + \phi_i\right) \tag{III.58}$$

$$u_3 = \sum_{i=1}^{n_f} \omega_i a_i \frac{\sinh\left(k_i x_3\right)}{\sinh\left(k_i h_{\text{mwl}}\right)} \sin\left(k_i x_1 - \omega_i t + \phi_i\right) \tag{III.59}$$

with n_f the number of frequency components, k_i, ω_i, ϕ_i the wavenumber, wave angular frequency, and wave phase of the ith frequency component respectively. Using this reconstruction method, the wave amplitude of each component is calculated as follows:

$$a_i = \sqrt{2 S_i \Delta f_i} \tag{III.60}$$

with S_i the spectral energy and Δf_i the frequency interval of the ith linear component. On top of describing random sea states, spectral distributions can be used to define extreme waves such as focused waves (sometimes also called freak waves). Focused waves are highly nonlinear waves where all frequency components converge at a specific point in time and space. In order for a spectral wave to focus at time t_f and point x_f, it is clear that the following expression is needed for the wave elevation (the same logic applies to wave velocity components):

$$\eta = \sum_{i=1}^{n_f} a_i \cos\left(k_i(x_1 - x_f) - \omega_i(t - t_f)\right) \tag{III.61}$$

Therefore, the phase shift ϕ_i needed for each frequency component i to focus at the intended time and space in eqs. (III.57) to (III.59) is calculated as follows:

$$\phi_i = -k_i x_f + \omega_i t_f \tag{III.62}$$

To reach the chosen target amplitude a_f of a focused wave, the amplitude of each frequency components is then set as:

$$a_i = a_f \frac{\sqrt{2 S_i \Delta f_i}}{\sum_{j=1}^{n_f} \sqrt{2 S_j \Delta f_j}} \tag{III.63}$$

III.2.2 Wave Generation and Absorption

Similarly to lab experiments, numerical simulations are prone to undesired wave reflection due to a smaller domain compared to real world applications. Several techniques exist for mitigating this phenomenon, such as the relaxation zone method, internal source wave makers, solid boundary motion (numerical paddles) and radiation boundary conditions. An overview of these methods is available in [130]. The relaxation zone technique is chosen for the work presented here due to its capacity to reliably absorb waves and ease of implementation through the addition of a single source term to the NSEs. Its implementation is therefore briefly described in this section.

Relaxation zones — also known as sponge layers — do not require movement of any physical boundaries, nor the use of radiation boundary conditions, and generally use Dirichlet boundary conditions for imposing the wave kinematics (velocity, pressure, and free surface position) directly at the generating boundary. Wave absorption capabilities are then introduced within the internal fluid domain, rather than at the boundaries themselves. In [76], a

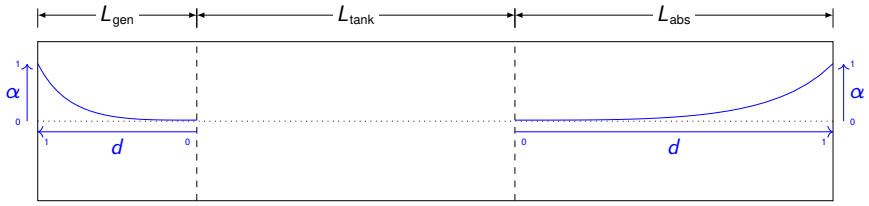


Figure III.5: Illustration of relaxation zone and blending function in typical numerical tank

target value for the fluid velocity and the position of the free surface is gradually imposed over a dedicated zone of the numerical fluid domain. In the implementation used here, only the fluid velocity is imposed gradually in the relaxation zone, keeping the Dirichlet imposition of the free surface position at the generating boundary but not within the fluid domain. Because the free surface elevation is not imposed within Ω_f, mass conservation is inherently ensured. As can be seen in [37], solely imposing the fluid velocity yields results that are equivalent to the ones presented in [76] who also imposes the free surface position. Relaxation zones can be separated into two main categories: absorption zones that damp incident or transmitted waves, and generation zones that absorb reflected waves. For this purpose, absorption zones simply impose a target velocity of zero while generation zones impose a target velocity that matches the generated wave conditions. This target velocity is imposed through a source term in the NSEs as follows:

$$\rho\left(\dot{\mathbf{u}} + \mathbf{u}\cdot\nabla\mathbf{u} - \mathbf{g}\right) - \nabla\cdot\bar{\bar{\boldsymbol{\sigma}}} = \alpha_b\alpha_0\left(\mathbf{u} - \mathbf{u}_t\right) \tag{III.64}$$

with α_b a blending function, α_0 a constant, and $\mathbf{u}_t$ the target velocity. The blending function used here is the same as in [76]:

$$\alpha_b = \frac{e^{d^{3.5}} - 1}{e - 1} \tag{III.65}$$

with $0 < d < 1$ the scaled distance from the boundary to the end of the relaxation zone. Using this scaling, we have $d = 1$ at the boundary (i.e. the source term is fully imposed), and $d = 0$ at the end of the relaxation zone (i.e. the source term vanishes at the end of the zone). An illustration of the relaxation zones and the blending functions within them is shown in fig. III.5. For eq. (III.64), it has also been found that using $\alpha_0 = \frac{30}{T}$ was sufficient for imposing relaxation [99], where T is the period of the generated or incoming wave. In [37], this constant is expressed in terms of angular frequency with an equivalent value of $\alpha_0 = 5\omega \approx \frac{30}{T}$. In the case of random wave conditions, the value of the period or angular frequency is chosen according to the peak period expected from the sea state. The length of relaxation zones is usually scaled with the wavelength λ of the waves to absorb. In the simulations presented in chapter V, the length of absorption zones are $L_{abs} = 2\lambda$, and the length of generation zones are $L_{gen} = 1\lambda$, similarly to [76] and [37].

III.3 Rigid Body Dynamics

Because their boundaries do not deform in space, it is convenient to represent rigid bodies as particles. In the FSI framework built here, the boundaries of rigid bodies are only used to integrate the external forces and moments, and for potential collision detection, but they do not play a role when it comes to kinematics or equations of motion.

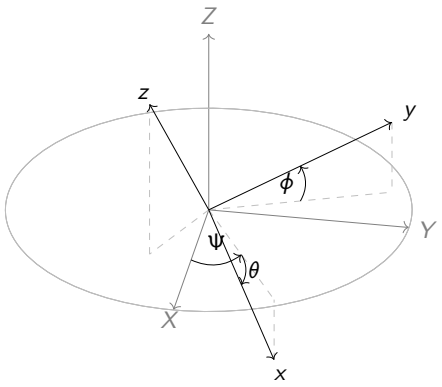


Figure III.6: State of body: position and rotation

The position vector of a body $\mathbf{r} = \mathbf{r}(t)$ can be written in 3D as:

$$\mathbf{r} = \begin{pmatrix} r_1 & r_2 & r_3 \end{pmatrix}^\intercal \tag{III.66}$$

with r_1 the position along the X-axis, r_2 the position along the Y-axis, and r_3 the position along the Z-axis. In the context of rigid body dynamics, $\mathbf{r}$ usually coincides with the position of the barycentre (i.e. centre of mass) of the body. The translational velocity vector is denoted $\dot{\mathbf{r}}$ and the translational acceleration vector is denoted $\ddot{\mathbf{r}}$. The rotation of a body $\Theta = \Theta(t)$ is represented as follows in 3D:

$$\Theta = \begin{pmatrix} \phi & \theta & \Psi \end{pmatrix}^\intercal \tag{III.67}$$

with ϕ the roll angle (around the X-axis), θ the pitch angle (around the Y-axis), and Ψ the yaw angle (around the Z-axis). The rotational velocity vector is denoted $\boldsymbol{\omega}$ and the rotational acceleration vector is denoted $\dot{\boldsymbol{\omega}}$.

The state vector $\mathbf{s}$, also called generalised coordinates, encompasses both the translational and rotational information of the body ($\mathbf{r}$ and Θ) as follows:

$$\mathbf{s} = \begin{pmatrix} \mathbf{r} & \Theta \end{pmatrix}^\intercal \tag{III.68}$$

In MBD solvers, the state of rigid bodies are described by their position $\mathbf{s}$ and velocity $\dot{\mathbf{s}}$, while $\ddot{\mathbf{s}}$ is found with the equations of motion. Following Newton's second law, the motion of a rigid body is described as:

$$\mathbf{M}\ddot{\mathbf{s}} = \mathbf{f} \tag{III.69}$$

with $\mathbf{M}$ the mass matrix of the body and $\mathbf{f}$ the sum of forces and moments acting on the body. The mass matrix of a rigid body is composed of its mass m and moment of inertia $\bar{\bar{\mathbf{I}}}_t$ (3×3 tensor) to form a 6×6 matrix that is expressed as follows:

$$\mathbf{M} = \begin{pmatrix} \bar{\bar{\mathbf{I}}} m & 0 \\ 0 & \bar{\bar{\mathbf{I}}}_t \end{pmatrix} \tag{III.70}$$

with $\bar{\bar{\mathbf{I}}}$ a 3×3 identity matrix. Additional terms can be added to this mass matrix as needed, such as added mass terms that can fill off-diagonals values. The generalised force vector $\mathbf{f}$ here is a 6×1 vector that combines the 3×1 vectors of forces $\vec{f}$ and moments $\vec{m}$ acting on the body as follows:

$$\mathbf{f} = \begin{pmatrix} \vec{f} & \vec{m} \end{pmatrix}^\intercal \tag{III.71}$$

The total forces $\vec{f}$ and moments $\vec{m}$ are used to calculate the new translational and rotational accelerations as follows:

$$m\ddot{\mathbf{r}} = \vec{f} \tag{III.72}$$

$$\bar{\bar{\mathbf{I}}}_t \dot{\boldsymbol{\omega}} = \vec{m} \tag{III.73}$$

For this work, rigid body dynamics are simulated with the Chrono library where, for the purpose of this thesis, the solver was modified in order to allow for the full inclusion of 6×6 mass matrices (including off-diagonal terms) instead of defining bodies only with their mass m and moment of inertia $\bar{\bar{\mathbf{I}}}_t$.

III.4 Mooring Modelling

In this section, the different methods for numerical modelling of mooring cables that have been implemented and used for this thesis are described. The logic and algorithms for calculating static tensions in catenary mooring lines are first described in section III.4.1. These algorithms are also the basis for mooring quasi-statics. The general theory of linear elasticity and beam theory is then described in section III.4.2.1 and applied to mooring dynamics. Note that the static / quasi-static model was designed from scratch as a standalone solver for the purposes of the work presented here, and that the Chrono library was used as a base for building a mooring dynamics framework.

III.4.1 Mooring Statics and Quasi-Statics

Consider a cable fixed in space by its two extremities (anchor and fairlead in the context of moorings), subject only to a uniform gravitational force, and at equilibrium. The shape of such cable hanging under its own weight is known as a catenary and its characteristics can be obtained analytically. The parametric equations of a catenary are defined as follows:

$$x_1 = a \operatorname{arcsinh} \frac{s}{a} \tag{III.74}$$

$$x_2 = a \cosh \frac{x_1}{a} \tag{III.75}$$

with s the distance along the catenary, and a a parameter that defines the shape of the catenary. To illustrate these equations, examples of catenary shapes for different parameters a are shown in fig. III.7. Note that these equations are derived in 2D as the catenary is expressed in the plane including the two extremities of the cable and parallel to the gravitational acceleration vector (a simple transformation would be needed to retrieve coordinates from a 3D perspective). When performing static analysis of cables, the unknown parameter of interest is usually the cable tension:

$$\mathbf{T} = \begin{pmatrix} T_H & T_V \end{pmatrix}^\intercal \tag{III.76}$$

with T_H and T_V the horizontal and vertical components of the tension, respectively. The parametric equations of a catenary can also be written as follows [4]:

$$x_1 = a \left(\operatorname{arcsinh} \left(\frac{s}{a} + t_a \right) - \operatorname{arcsinh} t_a \right) \tag{III.77}$$

$$x_2 = a \left(\cosh \left(\frac{x_1}{a} + \operatorname{arcsinh} t_a \right) - \cosh \left(\operatorname{arcsinh} t_a \right) \right) \tag{III.78}$$

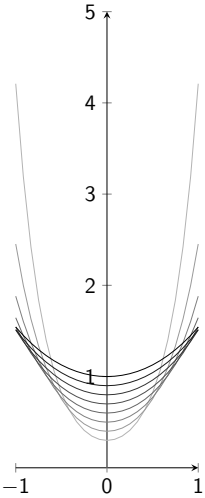


Figure III.7: Catenary shape for different values of a
Note: $0.3 \leq a \leq 1$ in increments of 0.1. A smaller a is translated into a lighter curve.

with $t_a = \frac{T_{V,a}}{T_H}$ representing the slope of the catenary at the anchor. The horizontal tension along the lifted part of the cable is constant and is found with $T_H = aw_0$ where w_0 is the linear submerged weight of the cable. The vertical tension varies along the cable according its weight at the point considered, meaning that $T_V = T_{V,a} + w_0 s$, with $T_{V,a}$ the vertical tension at the anchor and s the distance along the lifted cable. Alternatively, for any point of the lifted cable, vertical tension can be calculated as:

$$T_V = T_H \tan(\theta) \tag{III.79}$$

with $\theta = \theta(x_1)$ the angle formed by the catenary at the point considered:

$$\theta(x_1) = \arctan\left(\frac{\partial x_2}{\partial x_1}\right) = \arctan\left(\sinh\left(\frac{x_1}{a}\right)\right) \tag{III.80}$$

All cases here assume a flat seabed perpendicular to the gravitational acceleration vector, and only the weight of the cable is taken into account for solving the catenary equations. We define here the horizontal distance d and vertical distance h between the anchor $\mathbf{r}_a = \begin{pmatrix} r_{1,a} & r_{2,a} \end{pmatrix}^\intercal$ and the fairlead $\mathbf{r}_f = \begin{pmatrix} r_{1,f} & r_{2,f} \end{pmatrix}^\intercal$ as follows:

$$d = r_{1,f} - r_{1,a} \quad ; \quad h = r_{2,f} - r_{2,a} \tag{III.81}$$

If stretching is taken into account, the total length of the cable is expressed as $L + e$ with L the unstretched cable length and e the stretch of the cable:

$$e = T_A \frac{L}{EA_0} \tag{III.82}$$

with T_A the axial tension, and EA_0 the axial stiffness of the cable. When the line is stretched, the submerged weight of the cable must be averaged as follows:

$$\bar{w_0} = w_0 \frac{L_s}{L_s + e} \tag{III.83}$$

with L_s the unstretched lifted cable length. When dealing with moorings that can partly lay on the seabed, four different configurations are possible: fully stretched line (straight cable from anchor to fairlead), fully lifted line, partly lifted line (part of the line on the seabed), and lifted line with no horizontal span, as shown in fig. III.8. For

Algorithm 1 Calculation of static tension in catenary mooring line at the fairlead ($\mathbf{T}_f$) and anchor ($\mathbf{T}_a$)

function GETCABLETENSION(d, h, L, w_0, EA_0)
  $e = \frac{h}{\left(\frac{EA_0}{w_0 h}+1\right)}$ ▷ First guess for e assuming $L_s = h$
  if $L + e \geq h + d$ **then** ▷ Cable with zero horizontal span
    $\mathbf{T}_a = \begin{pmatrix} 0 & 0 \end{pmatrix}^\mathsf{T}$
    $\mathbf{T}_f = \begin{pmatrix} 0 & w_0 (h - e) \end{pmatrix}^\mathsf{T}$
  else
    $f(a) \leftarrow a\left(\cosh \frac{d}{a} - 1\right) - h$ ▷ First guess for a assuming fully lifted line and $t_a = 0$
    $a \leftarrow$ BISECTION($f(a)$)
    $e \leftarrow \frac{L}{EA_0}\sqrt{(aw_0)^2 + (w_0 L)^2}$
    if $L + e > a \sinh \frac{d}{a}$ **then** ▷ Partly lifted cable
      $a, L_s \leftarrow$PARTLYLIFTEDELASTIC(d, h, L, w_0, EA_0)
      $\mathbf{T}_a = \begin{pmatrix} 0 & 0 \end{pmatrix}^\mathsf{T}$
      $\mathbf{T}_f = \begin{pmatrix} a\bar{w}_0 & w_0 L_s \end{pmatrix}^\mathsf{T}$
    else ▷ Fully lifted cable
      $a \leftarrow$ FULLYLIFTEDELASTIC(d, h, L, w_0, EA_0)
      if $\exists a$ **then** ▷ Converged to a fully lifted solution
        $\mathbf{T}_a = \begin{pmatrix} a\bar{w}_0 & a\bar{w}_0 \tan\theta_f - w_0 L \end{pmatrix}^\mathsf{T}$
        $\mathbf{T}_f = \begin{pmatrix} a\bar{w}_0 & a\bar{w}_0 \tan\theta_f \end{pmatrix}^\mathsf{T}$
      else ▷ Fully stretched line
        $e = \sqrt{h^2 + d^2} - L$
        $\mathbf{T}_a = \left(\frac{eEEA_0}{L}\cos\theta_f \quad \frac{eEEA_0}{L}\sin\theta_f - w_0 L\right)^\mathsf{T}$
        $\mathbf{T}_f = \left(\frac{eEEA_0}{L}\cos\theta_f \quad \frac{eEEA_0}{L}\sin\theta_f\right)^\mathsf{T}$
  return $\mathbf{T}_a$, $\mathbf{T}_f$

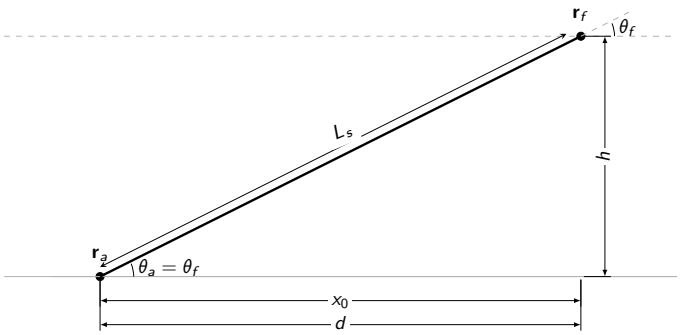
these different cases, the tensions at the anchor and fairlead are calculated as shown in table III.3, derived from the equations described above. Algorithm 1 describes the process for getting the tension of a mooring cable at the fairlead and the anchor given the length of the cable L, the submerged weight of the cable w_0, and the horizontal distance d and the vertical distance h between the anchor and the fairlead. The algorithm first checks if the line is long enough to fall straight on the seabed and, if not, it proceeds to check if the line is partly or fully lifted. From there, subsequent algorithms according to partly lifted (algorithm 3) or fully lifted (algorithm 2) configuration are called. The algorithms and numerical implementation dealing with the different possible configurations of the cables for static analysis are described in further details in the next four subsections. Note that the ability to do static analysis of a multi-segmented cable (cable composed of segments that can have different material properties) was also implemented for the work presented in this thesis. This is a relatively trivial addition when the cable is assumed rigid, as the only difference in this case is the need to average the submerged weight of the lifted cable according to the submerged weight of each lifted segment when calculating the tension. When elasticity plays a role, the elongation of each segment is assumed to be evenly distributed along the relevant segment. For the sake of clarity, the algorithms described in this section do not include multi-segmented cable notation logic.

III.4.1.1 Fully Stretched Line

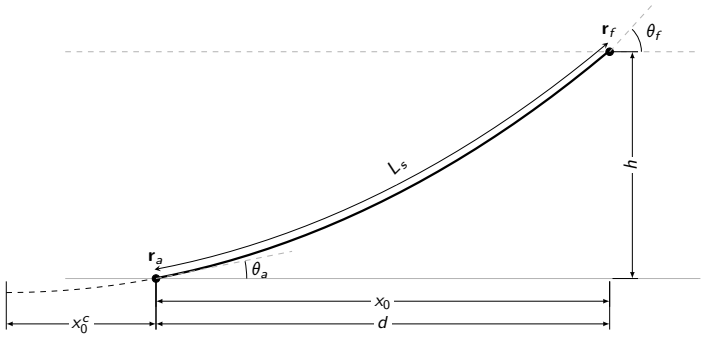
The mooring line is considered fully stretched (see fig. III.8a) if the following applies:

$$L \leq \sqrt{h^2 + d^2} \;\wedge\; L + e \approx \sqrt{h^2 + d^2} \tag{III.84}$$

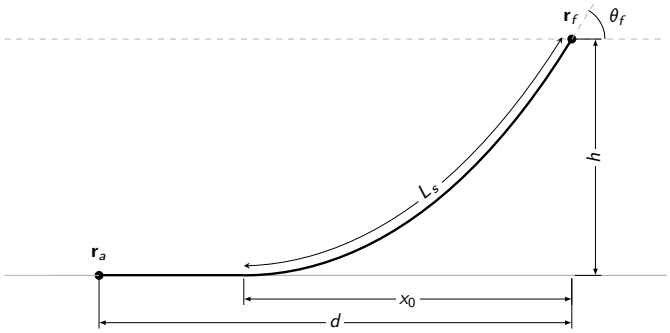
When elasticity is taken into account, it is straightforward to find the tension in the cable as stretching of the line is directly calculated as $e = \sqrt{h^2 + d^2} - L$, leading to a total tension T_A at the fairlead that is calculated by applying



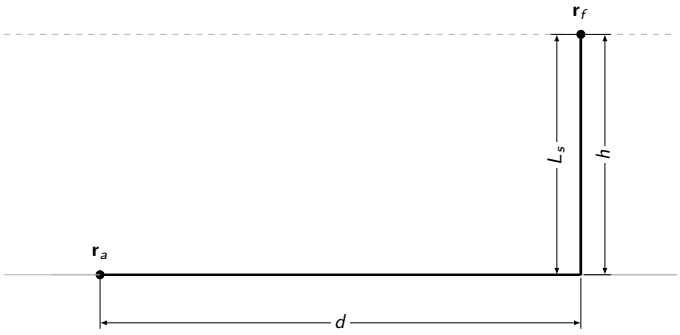
(a) Fully stretched mooring line



(b) Fully lifted mooring line



(c) Partly lifted mooring line



(d) Mooring line with no horizontal span

Figure III.8: Possible configurations for catenary line

Algorithm 2 Getting catenary solution for fully lifted cable

function FULLYLIFTEDELASTIC(d, h, L, w_0, EA_0, ϵ_{tol})
  $t_a = t_a^{\max} = \frac{h}{d}$; $t_a^{\min} = 0$; $e = 0$; $L_s = 0$; $\epsilon = \epsilon_{\text{tol}} + 1$
  while $\|\epsilon\| > \epsilon_{\text{tol}}$ **do**
    $t_a \leftarrow 0.5\left(t_a^{\min} + t_a^{\max}\right)$
    $a \leftarrow$ BISECTION$\left(a\left(\cosh\left(\frac{d}{a} + \operatorname{arcsinh} t_a\right) - \cosh\left(\operatorname{arcsinh} t_a\right) - 1\right) - h\right)$
    $L_{se} \leftarrow \sqrt{\left(2a\sinh\left(\frac{d}{2a}\right)\right)^2 + h^2}$
    $e \leftarrow \frac{L\sqrt{(a\bar{w}_0)^2 + (a\bar{w}_0\tan(\theta_f))^2}}{EA_0}$
    $\epsilon \leftarrow L + e - L_{se}$
    if $\epsilon > 0$ **then** $t_a^{\max} \leftarrow t_a$
    else if $\epsilon < 0$ **then** $t_a^{\min} \leftarrow t_a$
  return a

function FULLYLIFTEDRIGID(d, h, L)
  $f(a) \leftarrow 2a\sinh\left(\frac{d}{2a}\right) - \sqrt{L^2 - h^2} - a$
  $a \leftarrow$ BISECTION$(f(a))$
  return a

Hooke's law directly:

$$T_A = \frac{eEA_0}{L} \tag{III.85}$$

The horizontal and vertical components, T_H and T_V respectively, are then found by using simple trigonometry with the angle formed between the flat seabed and the straight mooring line $\theta_f = \arctan\left(\frac{h}{d}\right)$. Note that the horizontal tension remains constant along the table but that, for the vertical tension, the contribution due to the weight of the cable must still be taken into account according to the position s along the cable, leading to:

$$T_H(s) = T_A\cos(\theta_f) \quad ; \quad T_V(s) = T_A\sin(\theta_f) - (L - s)\,w_0 \tag{III.86}$$

III.4.1.2 Fully Lifted Line

The line is considered fully lifted (see fig. III.8b) if the following condition applies:

$$L_s = L \;\wedge\; L_s + e \geq \sqrt{h^2 + d^2} \tag{III.87}$$

Because the position of the anchor and fairlead are known as well as the lifted line length ($L_s = L$), the following expressions apply, using the definitions from eqs. (III.75) and (III.81):

$$h = 2a\sinh\frac{r_{1,a} + r_{1,f}}{2a}\sinh\frac{r_{1,a} - r_{1,f}}{2a} \tag{III.88}$$

$$L_s = 2a\cosh\frac{r_{1,a} + r_{1,f}}{2a}\sinh\frac{r_{1,a} - r_{1,f}}{2a} \tag{III.89}$$

Table III.3: Values of tensions at anchor and fairlead for static analysis of catenary cables

	fully stretched	fully lifted	partly lifted	no span
$T_{H,f}$	$T_{A,f}\cos\theta_f$	$a\bar{w}_0$	$a\bar{w}_0$	0
$T_{V,f}$	$T_{A,f}\sin\theta_f$	$T_{H,f}\tan\theta_f$	$w_0 L_s$	$w_0 L_s$
$T_{A,f}$	$\frac{eEA_0}{L}$	$\sqrt{T_{H,f}^2 + T_{V,f}^2}$	$\sqrt{T_{H,f}^2 + T_{V,f}^2}$	$T_{V,f}$
$T_{H,a}$	$T_{H,f}$	$T_{H,f}$	0	0
$T_{V,a}$	$T_{V,f} - w_0 L$	$T_{V,f} - w_0 L$	0	0
$T_{A,a}$	$\sqrt{T_{H,a}^2 + T_{V,a}^2}$	$\sqrt{T_{H,a}^2 + T_{V,a}^2}$	0	0

Algorithm 3 Getting catenary solution for partly lifted cable

function PARTLYLIFTEDELASTIC(d, h, L, w_0, EA_0, ϵ_{tol})
  $x_0 = x_0^{\max} = d$; $x_0^{\min} = 0$; $e = 0$; $L_s = L$; $\epsilon = \epsilon_{\text{tol}} + 1$
  while $\|\epsilon\| > \epsilon_{\text{tol}}$ **do**
    $x_0 \leftarrow 0.5\left(x_0^{\max} + x_0^{\min}\right)$
    $f(a) \leftarrow a\left(\cosh\left(\frac{x_0}{a}\right) - 2\right) - h$
    $a \leftarrow$ BISECTION$(f(a))$
    $L_{se} \leftarrow h\sqrt{1 + \frac{2a}{h}}$
    $e \leftarrow \sqrt{\frac{TL}{EA_0}}$
    $e \leftarrow \frac{L_s\sqrt{(a\bar{w}_0)^2 + (\bar{w}_0 L_s)^2}}{EA_0}$
    $\epsilon \leftarrow L + e + x_0 - d - L_{se}$
    if $\epsilon > 0$ **then** $x_0^{\max} \leftarrow x_0$
    else if $\epsilon < 0$ **then** $x_0^{\min} \leftarrow x_0$
  return a, $L_{se} - e$

which yields the following transcendental equation in a.

$$\sqrt{L_s^2 - h^2} = 2a \sinh \frac{d}{2a} \tag{III.90}$$

Finding the root of eq. (III.90) can be done with the Newton–Raphson or bisection methods. The algorithm describing the numerical process for retrieving the solution for a is shown in algorithm 2 for both the rigid and elastic fully lifted cable cases. From the new solution a, the horizontal tension which is again the same at any point along the cable is directly calculated as $T_H = aw_0$ (or $T_H = a\bar{w}_0$ if elasticity is taken into account), but the tangent vector (or angle) along the cable must be found to deduct the vertical and total tension with simple trigonometry. Using eq. (III.75) directly does not work in this case, because the trough of the catenary equation given by a is at $x_1 = 0$, while it is known in this fully lifted line case that $\frac{dx_2}{dx_1} \neq 0$ at the anchor. Indeed, the solution for a from eq. (III.90) corresponds only to finding a segment of the catenary solution that has the length L_s and height h, but does not ensure that the extremity of this segment corresponding to the anchor is at $x_1 = 0$. Using eq. (III.88) and eq. (III.89), we can write:

$$\frac{h}{L_s} = \tanh \frac{r_{1,a} + r_{1,f}}{2a} \tag{III.91}$$

and from here, we can calculate the coordinate x_1^c of the anchor in the catenary solution space:

$$x_1^c = \frac{1}{2}\left(2a \operatorname{arctanh}\left(\frac{h}{L_s}\right) - d\right) \tag{III.92}$$

which allows calculation of the vertical tension along the cable as follows:

$$T_V(s) = T_H \tan\left(\theta\left(a \operatorname{arcsinh}\left(\frac{s}{a}\right) + x_1^c\right)\right) \tag{III.93}$$

with $\theta = \theta(x_1)$ from eq. (III.80).

III.4.1.3 Partly Lifted Line

The line is considered partly lifted (see fig. III.8c) if the following condition applies:

$$L_s < L + e < h + d \tag{III.94}$$

In this case, we have $x_0 < d$ for the horizontal span of the line, and we need to find a catenary solution that satisfies:

$$d = L - L_s + x_0 \tag{III.95}$$

Iterations are therefore needed to find the right value for x_0. From an initial guess for the horizontal span $x_0^{(i)}$, a solution $a^{(i)}$ is calculated with the transcendental equation:

$$a^{(i)}\left(\cosh\left(\frac{x_0^{(i)}}{a^{(i)}}\right)-2\right)=h \tag{III.96}$$

From here, the corresponding lifted line length is calculated as:

$$L_s^{(i)}=h\sqrt{1+\frac{2a^{(i)}}{h}} \tag{III.97}$$

Then, if $L-L_s^{(i)}+x_0^{(i)}-d>0$, the new guess $x_0^{(i+1)}$ is scaled down, and if $L-L_s^{(i)}+x_0^{(i)}-d<0$, the new guess $x_0^{(i+1)}$ is scaled up. This is repeated until the right x_0 and L_s are found to fulfill eq. (III.95) up to a user-defined tolerance. The iterative process is shown in more details in algorithm 3. Once a solution is found, the tensions in the cable is calculated as follows:

$$T_H(s)=\begin{cases}0 & \forall s<d-x_0\\ aw_0 & \forall s\geq d-x_0\end{cases}\quad;\quad T_V(s)=\begin{cases}0 & \forall s<d-x_0\\ (s-(d-x_0))\,w_0 & \forall s\geq d-x_0\end{cases} \tag{III.98}$$

III.4.1.4 Line with No Horizontal Span

The line does not have any horizontal span (i.e. $x_0=0$ as shown in fig. III.8d) if the following condition is fulfilled:

$$L+e>h+d \tag{III.99}$$

In this case, the lifted part of the mooring line is considered to be perpendicular to the seabed (assuming that it is flat) with $L_s=h$, and with the rest of the line resting on the seabed. The fairlead tension is equal to the weight of the lifted cable, with:

$$T_H(s)=0\quad\forall s\leq L_s\quad;\quad T_V(s)=\begin{cases}0 & \forall s<d\\ (s-d)\,w_0 & \forall s\geq d\end{cases} \tag{III.100}$$

III.4.2 Mooring Dynamics

This section describes the main governing equations for mooring dynamics using beam theory. The general principles of linear elasticity are first introduced in section III.4.2.1 as they are the base of beam theory. Those principles are then simplified for general beam theory in section III.4.2.2, and the specific implementation used for mooring dynamics in the work presented here is described in more detail in section III.4.2.3. Note that throughout this section, the elastic materials are assumed to be isotropic.

III.4.2.1 Linear Elasticity for Isotropic Materials

The governing equations of linear elasticity can be divided into three main categories: the strain-displacement relations or kinematic equations, the constitutive laws, and the equilibrium equations. The kinematic equations describe the deformation $\mathbf{h}=\begin{pmatrix}h_1 & h_2 & h_3\end{pmatrix}^\intercal$ of a beam element with a displacement gradient that corresponds to the normal strain in eq. (III.101) and shear strain in eq. (III.102):

$$\epsilon_{11}=\frac{\partial h_1}{\partial \widehat{x_1}}\quad;\quad\epsilon_{22}=\frac{\partial h_2}{\partial \widehat{x_2}}\quad;\quad\epsilon_{33}=\frac{\partial h_3}{\partial \widehat{x_3}} \tag{III.101}$$

$$\epsilon_{23} = \frac{1}{2}\left(\frac{\partial h_2}{\partial \widehat{x}_3} + \frac{\partial h_3}{\partial \widehat{x}_2}\right) \quad ; \quad \epsilon_{13} = \frac{1}{2}\left(\frac{\partial h_1}{\partial \widehat{x}_3} + \frac{\partial h_3}{\partial \widehat{x}_1}\right) \quad ; \quad \epsilon_{12} = \frac{1}{2}\left(\frac{\partial h_1}{\partial \widehat{x}_2} + \frac{\partial h_2}{\partial \widehat{x}_1}\right) \tag{III.102}$$

When the material is assumed to be isotropic, the stress along each axis is related to the strain with the following generalised Hooke's law for normal strains eq. (III.103) and shear strains eq. (III.104):

$$\epsilon_{11} = \frac{1}{E}\left(\sigma_{11} - \nu\left(\sigma_{22} + \sigma_{33}\right)\right) \quad ; \quad \epsilon_{22} = \frac{1}{E}\left(\sigma_{22} - \nu\left(\sigma_{11} + \sigma_{33}\right)\right) \quad ; \quad \epsilon_{33} = \frac{1}{E}\left(\sigma_{33} - \nu\left(\sigma_{11} + \sigma_{22}\right)\right) \tag{III.103}$$

$$\epsilon_{23} = \epsilon_{32} = \frac{\sigma_{23}}{G} \quad ; \quad \epsilon_{13} = \sigma_{31} = \frac{\sigma_{13}}{G} \quad ; \quad \epsilon_{12} = \sigma_{23} = \frac{\sigma_{12}}{G} \tag{III.104}$$

with ν the Poisson's ratio of the beam, and G the shear modulus:

$$G = \frac{E}{2\left(1+\nu\right)} \tag{III.105}$$

The stresses and strains can then be expressed as 6×1 vectors as follows:

$$\boldsymbol{\epsilon} = \begin{pmatrix} \epsilon_{11} & \epsilon_{22} & \epsilon_{33} & \epsilon_{23} & \epsilon_{13} & \epsilon_{12} \end{pmatrix}^{\mathsf{T}} \tag{III.106}$$

$$\boldsymbol{\sigma} = \begin{pmatrix} \sigma_{11} & \sigma_{22} & \sigma_{33} & \sigma_{23} & \sigma_{13} & \sigma_{12} \end{pmatrix}^{\mathsf{T}} \tag{III.107}$$

which can be used to define the constitutive equation:

$$\boldsymbol{\sigma} = \mathbf{C}\boldsymbol{\epsilon} \tag{III.108}$$

with $\mathbf{C}$ the 6×6 material stiffness matrix:

$$\mathbf{C} = \frac{E}{\left(1+\nu\right)\left(1-2\nu\right)} \begin{pmatrix} 1-\nu & \nu & \nu & 0 & 0 & 0 \\ \nu & 1-\nu & \nu & 0 & 0 & 0 \\ \nu & \nu & 1-\nu & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1-2\nu}{2} & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1-2\nu}{2} & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{1-2\nu}{2} \end{pmatrix} \tag{III.109}$$

Finally, the equilibrium equation for linear elasticity that describe the relation of the stresses with the forces applied to the material can be expressed as follows:

$$\nabla \cdot \bar{\bar{\boldsymbol{\sigma}}} + \mathbf{f} = \rho \ddot{\mathbf{h}} \tag{III.110}$$

with $\mathbf{f}$ the body force per unit volume and $\bar{\bar{\boldsymbol{\sigma}}}$ the stress tensor expressed as:

$$\bar{\bar{\boldsymbol{\sigma}}} = \begin{pmatrix} \sigma_{11} & \sigma_{12} & \sigma_{13} \\ \sigma_{22} & \sigma_{22} & \sigma_{23} \\ \sigma_{33} & \sigma_{32} & \sigma_{33} \end{pmatrix} \tag{III.111}$$

which is symmetric in the case of isotropic material as $\sigma_{12} = \sigma_{21}$, $\sigma_{13} = \sigma_{31}$, and $\sigma_{23} = \sigma_{32}$.

III.4.2.2 Beam Theory

Beams are slender structures with one of their dimensions that is significantly larger than the others. Conventionally, this longer dimension is set as its longitudinal axis. Because of this geometry, the cross-section of a beam does not deform significantly when experiencing transverse or axial loads. Therefore, the cross-section of a beam is assumed to be infinitely rigid (i.e. no deformation can occur) in its own plane, to remain plane after deformation, and to remain normal to the deformed axis of the beam (longitudinal axis). Defining a local system of coordinates $\boldsymbol{\xi} = (\xi, \eta, \zeta)$ with ξ along the longitudinal axis of the beam, the state vector $\mathbf{e}$ of the beam can be defined as:

$$\mathbf{e} = \begin{pmatrix} \mathbf{q}_0 & \mathbf{q}_1 \end{pmatrix}^\mathsf{T} \quad ; \quad \mathbf{q} = \begin{pmatrix} \mathbf{r}_s & \theta \end{pmatrix}^\mathsf{T} \tag{III.112}$$

with $\mathbf{q}_0$ the state vector (position and rotation) of the node at one end of the beam, $\mathbf{q}_1$ the state vector of the node at the other end of the beam, $\mathbf{r}_s$ the absolute position of the node, and θ the rotation of the node. To recover the position $\mathbf{r}_s$ of an arbitrary point $\boldsymbol{\xi}$ within an element $\mathbf{e}$, a predefined matrix of shape functions $\mathbf{N} = \mathbf{N}(\boldsymbol{\xi})$ is used as follows:

$$\mathbf{r}_s = \mathbf{N}\mathbf{e} \tag{III.113}$$

From this shape function, the mass matrix $\mathbf{M}$ of an element can be defined as:

$$\mathbf{M} = \int_V \rho \mathbf{N}^\mathsf{T} \mathbf{N} \mathrm{d}V \tag{III.114}$$

The assumptions stated above for isotropic beams allow the kinematic equations along each axis to depend only on one variable: the coordinates along the longitudinal axis ξ of the beam. The kinematic equations from eqs. (III.101) and (III.102) reduce to the single expression:

$$\epsilon_\xi = \epsilon_{11} = \frac{\partial h_\xi}{\partial \xi} + \zeta \kappa_\eta - \eta \kappa_\zeta \tag{III.115}$$

with ϵ_ξ the strain of the beam in the axial direction, κ_η and κ_ζ the curvature of the beam in the transverse directions:

$$\kappa_\eta = \frac{\partial^2 h_\zeta}{\partial \xi^2} = \frac{\partial \theta_\zeta}{\partial \xi} \quad ; \quad \kappa_\zeta = \frac{\partial^2 h_\eta}{\partial \xi^2} = \frac{\partial \theta_\eta}{\partial \xi} \tag{III.116}$$

Because the axial stresses are much greater than transverse stresses, it is assumed that $\sigma_{22} \approx 0$ and $\sigma_{33} \approx 0$, Hooke's law from eqs. (III.103) and (III.104) for a beam simply reduces to:

$$\sigma_\xi = \sigma_{11} = E \epsilon_\xi \tag{III.117}$$

Note that due to Poisson's effect when $\nu \neq 0$, ϵ_{22} and ϵ_{33} do not technically vanish with beam theory. However, in the case of elongated beams, it is assumed that Poisson's effect is very small, and therefore that $\nu \approx 0$. The axial force f_ξ and bending moments m_η and m_ζ are expressed as:

$$f_\xi = \int_A \sigma_\xi \mathrm{d}A \quad ; \quad m_\eta = \int_A \zeta \sigma_\xi \mathrm{d}A \quad ; \quad m_\zeta = -\int_A \eta \sigma_\xi \mathrm{d}A \tag{III.118}$$

Material characteristics of beams are often defined through their axial and bending stiffness. Axial stiffness for isotropic beams is expressed as follows:

$$\int_A E \mathrm{d}A = E \int_A \mathrm{d}A = E A_0 \tag{III.119}$$

with A_0 the undeformed cross-sectional area. Bending stiffness of the beam is expressed as EI, which is also known as the flexural rigidity, with I the moment of inertia for flexion of the cable ($I = I_\eta = I_\zeta$ for isotropic materials), and appears in the moment-curvature relationship:

$$m_\eta = -EI\kappa_\eta \quad ; \quad m_\zeta = -EI\kappa_\zeta \tag{III.120}$$

Note that bending effects can be ignored by setting $I := 0$, and compression effects can be removed by making the axial load term vanish when $\epsilon_\xi < 0$. The equations of motion of a beam element is then expressed as:

$$\mathbf{M}\ddot{\mathbf{e}} + \mathbf{f}_k = \mathbf{f}_e \tag{III.121}$$

with $\mathbf{f}_k$ the vector of elastic forces, $\mathbf{f}_e$ the vector of external forces.

III.4.2.3 Absolute Nodal Coordinate Formulation (ANCF)

The type of beams used to define mooring cables for the work presented here is based on the gradient deficient Absolute Nodal Coordinate Formulation (ANCF) formulation as described by [55] and that was made available in Chrono. This type of beam is not fully parameterized but only defined by a position vector and one or two direction gradient vectors per node. If the element is made of an isotropic material and shear is negligible, as is the case for the applications considered here, the nodes can be described with a single direction gradient. The vector of nodal coordinates of a gradient deficient ANCF element are written as:

$$\mathbf{e}^{\text{ANCF}} = \begin{pmatrix} \mathbf{q}_0^{\text{ANCF}} & \mathbf{q}_1^{\text{ANCF}} \end{pmatrix}^\intercal \quad ; \quad \mathbf{q}^{\text{ANCF}} = \begin{pmatrix} \mathbf{r}_s & \nabla_\xi \mathbf{r}_s \end{pmatrix}^\intercal \tag{III.122}$$

with $\nabla_\xi = \frac{\partial}{\partial \xi}$ the gradient along the abscissa of the beam. Following the formulation of [12], the matrix of shape functions can be written in 2D as:

$$\mathbf{N}(\bar{\xi}) = \begin{pmatrix} s_1\bar{\bar{\mathbf{I}}} & s_2\bar{\bar{\mathbf{I}}} & s_3\bar{\bar{\mathbf{I}}} & s_4\bar{\bar{\mathbf{I}}} \end{pmatrix} \tag{III.123}$$

with $\bar{\bar{\mathbf{I}}}$ the identity matrix and the shape functions:

$$\begin{aligned} s_1 = 1 - 3\bar{\xi}^2 + +2\bar{\xi}^3 \quad ; \quad s_2 = l\left(\bar{\xi} - 2\bar{\xi}^2 + \bar{\xi}^3\right) \\ s_3 = 3\bar{\xi}^2 - 2\bar{\xi}^3 \quad ; \quad s_4 = l\left(\bar{\xi}^3 - \bar{\xi}^2\right) \end{aligned} \tag{III.124}$$

with $\bar{\xi} = \frac{\xi}{l}$ the non-dimensional abscissa and l the length of the beam. The cross-section of the beams used here is assumed to be circular, with axial strain ϵ_ξ and curvature κ defined as follows:

$$\epsilon_\xi = \frac{1}{2}\left((\nabla_\xi \mathbf{r}_s)^\intercal \nabla_\xi \mathbf{r}_s - 1\right) \quad ; \quad \kappa = \frac{\left\|\nabla_\xi \mathbf{r}_s \times \nabla_\xi^2 \mathbf{r}_s\right\|}{\left\|\nabla_\xi \mathbf{r}_s\right\|^3} \tag{III.125}$$

The virtual work of elastic forces is defined as [55]:

$$\delta W_k = \int_0^l \left(\underbrace{EA_0\epsilon_\xi}_{\text{axial force}} \delta\epsilon_\xi + \underbrace{EI\kappa}_{\text{bending moment}} \delta\kappa \right) \mathrm{d}\xi \tag{III.126}$$

with δW_k the virtual work, $\delta\epsilon_\xi$ the virtual strain, and $\delta\kappa$ the virtual curvature.

CHAPTER IV

Coupling Strategy

While previous chapters are focusing on individual aspects of the uncoupled fluid and solid solvers, this chapter focuses on the coupling aspects of the Fluid–Structure Interaction (FSI) framework developed for this work.

Firstly, in section IV.1, an overview of the general workflow and communication between the different models is presented. In section IV.2, the fluid–structure coupling is shown before proceeding to describe the chosen partitioned schemes in more detail. Then, stability issues arising from the added mass effect due to partitioned coupling are discussed in section IV.2.3. This is followed by the description of a non-iterative stabilisation scheme for explicitly coupled FSI problems and of a Partial Differential Equation (PDE) model for estimating the added mass of an arbitrarily shaped body dynamically which have both been implemented to tackle this instability problem. The next section, section IV.3, focuses on the Arbitrary Lagrangian–Eulerian (ALE) formulation necessary for mesh conforming FSI with details about the mesh motion models implemented here. This includes mesh motion using the equations of linear elastostatics for mesh conforming boundaries, and mesh motion according to element volume monitoring function for implicit boundaries. Finally, the fluid–mooring coupling is described, including the retrieval of fluid effects on the mooring cables that are non-conforming to the fluid mesh in section IV.4.

IV.1 General Workflow of Coupling . . . 57
IV.2 Fluid–Structure Coupling . . . 58
IV.2.1 Interface Coupling . . . 58
IV.2.2 Coupling Scheme . . . 60
IV.2.2.1 Conventional Serial Staggered (CSS) Scheme . . . 62
IV.2.2.2 Improved Serial Staggered (ISS) Scheme . . . 62
IV.2.3 Added Mass Effect . . . 64
IV.2.3.1 Stabilisation for Explicit Coupling Schemes . . . 64
IV.2.3.2 Estimation of the Added Mass . . . 67
IV.3 Moving Domains . . . 69
IV.3.1 Arbitrary Lagrangian–Eulerian (ALE) Formulation . . . 69
IV.3.2 Mesh Deformation Approaches . . . 71
IV.3.2.1 Linear Elastostatics . . . 72
IV.3.2.2 Monitor Function . . . 73
IV.3.3 Mesh Quality and Smoothing . . . 77
IV.4 Moorings Coupling . . . 79
IV.4.1 Mooring–Structure Coupling . . . 79
IV.4.2 Fluid–Mooring Coupling . . . 80
IV.4.3 Particle-Localisation Algorithm . . . 82

IV.1 General Workflow of Coupling

An overview of the sequencing and communication between the different models is described here and is best read along with the diagram illustrating the general workflow in fig. IV.1. This section is meant to be used as a guide for the coupling where additional details of specific implementation aspects can be found in the relevant subsections. In order to have a complete overview of the framework, consider a problem that includes: Two-Phase Flow (TPF), mesh motion, floating bodies subject to the added mass effect, and mooring dynamics. After setting the initial conditions of the problem along with any other necessary preprocessing matter, the models are solved in the following order and looped over at each time step until the simulation is complete:

1. ALE mesh: this is the first model to be solved that sets the new deformed mesh configuration and computes the resulting mesh velocity $\mathbf{u}_{\boldsymbol{\xi}}^{(n+1)}$ needed by any subsequent model that uses the fluid mesh. Note that, while $\mathbf{u}_{\boldsymbol{\xi}}^{(n+1)}$ is technically produced for $t^{(n+1)}$, the mesh is actually moved to the position of the solid at time $t^{(n)}$ as required by the Conventional Serial Staggered (CSS) scheme (see section IV.2.2.1 for further details). If the ALE mesh model is solved using the equations of elastostatics, the displacement of the mesh nodes $\mathbf{h}_{\Gamma_{\mathrm{f}\cap\mathrm{s}}}^{(n+1)}$ at the fluid–solid interface $\Gamma_{\mathrm{f}\cap\mathrm{s}}$ must be used as boundary conditions to solve the elastostatics PDE and compute the resulting displacement $\mathbf{h}^{(n+1)}$ of all the mesh nodes in Ω_f. For this purpose, communication with the Multibody Dynamics (MBD) solver is necessary to retrieve the state $\mathbf{s}^{(n)}$ of the displaced solid from which $\mathbf{h}_{\Gamma_{\mathrm{f}\cap\mathrm{s}}}^{(n+1)}$ can be calculated from spatial transformation. If the mesh motion model uses the monitor function approach for implicit boundaries, the signed distance ϕ_{sdf} from the free surface model can be communicated when it is used within the monitor function. See section IV.3 for more details about these two mesh deformation models as well as the effect that ALE formulation has on the governing equations of the fluid models.

2. Navier–Stokes: using the new mesh configuration (if mesh motion is involved), this model provides the solution for the fluid velocity $\mathbf{u}^{(n+1)}$, pressure $\mathrm{p}^{(n+1)}$, and density $\rho^{(n+1)}$, the latter being useful to the added mass model when using TPF. In order to solve the Navier–Stokes Equations (NSEs) while taking into account the mesh deformation, the mesh velocity $\mathbf{u}_{\boldsymbol{\xi}}^{(n+1)}$ must be retrieved from the ALE mesh model and the NSEs must be modified accordingly, as described in section IV.3.1. See section III.1.1.4 for more details about the (uncoupled) NSEs model itself.

3. Free Surface: once the solution for the fluid has been found, the free surface is updated using the solution $\mathbf{u}^{(n+1)}$ from the Navier–Stokes model. Note that the global free surface tracking model used here is divided in four submodels that are each solving their own PDE: transport of Volume of Fluid (VOF), transport of Level Set (LS), redistancing, and mass correction. See section III.1.2 for more details about these different models and how they are used for mass conservative free surface motion.

4. Added Mass: this model is used to estimate the added mass of a moving solid dynamically by solving six PDEs that populate the 6×6 added mass matrix $\mathbf{A}$. For this purpose, and because this model solves for a pressure differential using the configuration of the fluid model, communication is needed with the Navier–Stokes model in order to retrieve the fluid density $\rho^{(n+1)}$ as it varies spatially when using TPF with implicit free surface tracking. The details of the implementation of the added mass model is shown in section IV.2.3.2.

5. MBD solver: finally, the solution for the bodies and moorings is given by the MBD solver. Communication is

needed here with several models. Firstly, the fluid forces $\mathbf{f}_f^{(n+1)}$ acting on the body boundary $\Gamma_{\mathrm{f}\cap\mathrm{s}}$ are calculated from the stresses coming from the Navier–Stokes model. Secondly, if added mass stabilisation is necessary, the added mass matrix must be communicated from the added mass estimation model in order to be included in the equations of motion of the body as shown in section IV.2.3.1 to counter the added mass effect. Thirdly, the fluid velocity $\mathbf{u}^{(n+1)}$, still from the Navier–Stokes model, must be retrieved along the mooring cables in order to compute the drag and inertia forces as shown in section IV.4 for the application of the forces, and in section IV.4.3 for finding $\mathbf{u}^{(n+1)}$ in the fluid mesh at arbitrary coordinates (cable nodes). With all this information communicated, the MBD solver can proceed and find a solution for the fully coupled mooring–structure problem that provides $\mathbf{s}^{(n+1)}$ to be used again in the mesh motion model of step 1 for the next time step.

The steps listed above are repeated until the time loop is over at $t = T_{\mathrm{sim}}$, and the simulation proceeds to the postprocessing steps before terminating. Note that some of the models can be deactivated in some cases, such as the added mass model if there is no strong added mass effect, or the ALE mesh model if the Ω_f is not moving.

IV.2 Fluid–Structure Coupling

IV.2.1 Interface Coupling

When coupling a fluid and solid problem, new conditions must be applied to both problems: the continuity of velocity and the continuity of stresses. These principles are applied at the interface $\Gamma_{\mathrm{f}\cap\mathrm{s}}$ between the two problems as follows:

$$\mathbf{u}_f = \mathbf{u}_s \qquad \forall \mathbf{x} \in \Gamma_{\mathrm{f}\cap\mathrm{s}} \tag{IV.1}$$

$$\bar{\bar{\boldsymbol{\sigma}}}_f \cdot \mathbf{n}_{\Gamma_{\mathrm{f}\cap\mathrm{s}}} = \bar{\bar{\boldsymbol{\sigma}}}_s \cdot \mathbf{n}_{\Gamma_{\mathrm{f}\cap\mathrm{s}}} \qquad \forall \mathbf{x} \in \Gamma_{\mathrm{f}\cap\mathrm{s}} \tag{IV.2}$$

with $\mathbf{u}_f$ the velocity of the fluid, and $\mathbf{u}_s$ the velocity of the solid. The continuity of velocity ensures that the interface between the two problems is always well defined, and is usually enforced on the fluid phase as a Dirichlet condition, while the continuity of stresses is applied as a Neumann condition and allows for the integration of fluid forces on the solid. This is referred to as a Dirichlet–Neumann coupling of the fluid–solid interface. Concerning the integration of fluid forces in the context of floating bodies, the total 6×1 force vector $\mathbf{f}$ can be decomposed as follows:

$$\mathbf{f} = \mathbf{f}_f + \mathbf{f}_e \tag{IV.3}$$

with $\mathbf{f}_f = \begin{pmatrix} \vec{f}_f & \vec{m}_f \end{pmatrix}^{\mathsf{T}}$ the hydrodynamic or fluid forces, and $\mathbf{f}_e$ the remaining external forces such as gravity and collision forces. The hydrodynamic forces $\vec{f}_f$ and moments $\vec{m}_f$, each a 3×1 vector, are calculated by integrating the pressure force that depends on the fluid pressure p and viscous force that depends on the fluid velocity gradient $\nabla\mathbf{u}$ and viscosity ν at the fluid–solid interface $\Gamma_{\mathrm{f}\cap\mathrm{s}}$. These pressure and viscous contributions can be represented directly into one expression using the fluid stress tensor:

$$\vec{f}_f = \int_{\Gamma_{\mathrm{f}\cap\mathrm{s}}} \bar{\bar{\boldsymbol{\sigma}}} \cdot \mathbf{n} \mathrm{d}\Gamma \tag{IV.4}$$

$$\vec{m}_f = \int_{\Gamma_{\mathrm{f}\cap\mathrm{s}}} (\mathbf{x} - \mathbf{r}) \times (\bar{\bar{\boldsymbol{\sigma}}} \cdot \mathbf{n}) \mathrm{d}\Gamma \tag{IV.5}$$

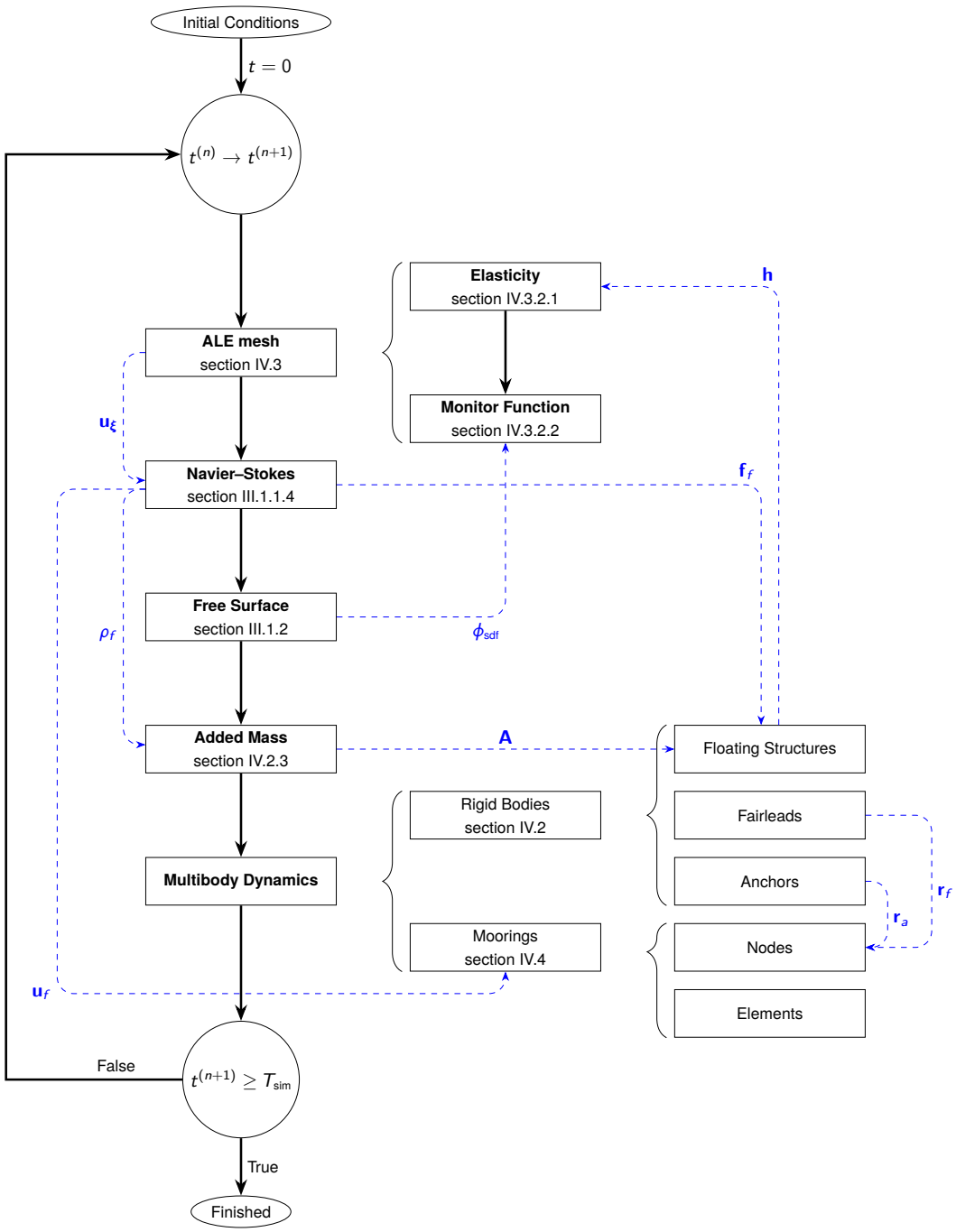


Figure IV.1: Diagram of numerical models workflow and coupling strategy

with $\mathbf{n}$ the normal vector to the boundary $\Gamma_{f\cap s}$, $\mathbf{x}$ a point on the boundary, $\mathbf{r}$ the position of the barycentre (or pivotal point) of the solid, and $\bar{\bar{\boldsymbol{\sigma}}}$ the Cauchy–Shwartz stress tensor as originally described in eq. (III.14).

■ IV.2.2 Coupling Scheme

Once the solid position has been updated by applying the fluid and other forces, the motion of the interface $\Gamma_{f\cap s}$ affects the fluid on the subsequent time step. This corresponds to a two-way coupling, where both the effect of the fluid on the solid, and of the solid on the fluid are taken into account, requiring careful interfacing between the models. With a multitude of coupling schemes to choose from, this section focuses on the selection of a scheme according to a list of requirements and constraints relevant to the purpose of this research. The main chosen criteria for an appropriate scheme are:

1. **Computational efficiency**, which is one of the main priorities as high-fidelity FSI simulations can be very demanding, especially when solving for the fluid due to the potentially large number of mesh cells;
2. **Stability**, which can be particularly affected when using partitioned coupling schemes on applications featuring strong added mass effect and that, in some cases, make the simulation unconditionally unstable;
3. **Accuracy**, which can also be affected by the scheme, especially when it is partitioned;
4. **Modularity**, which allows for improvement or replacement of the coupled solvers individually, as well as the expansion of the coupled problem to include other physical processes.

Additionally, because of the specific approaches and tools chosen for the uncoupled models used in this research, a few constraints apply. Firstly, the fluid solver is not projection-based, with the pressure and fluid velocity from the NSEs solved simultaneously and no segregation possible. Secondly, the MBD solver does not allow for retrieving a solution from a previous iteration when solving for complex systems (such as multiple bodies with mooring cables), and because external solvers are sometimes used for subproblems (such as collision detection), making it difficult to reset the solution to a previous state in a reliable manner. Note that these limitations can technically be overcome by further development on the different numerical tools, but that this additional work is beyond the scope of this thesis. Efficient coupling of the models is nonetheless possible despite these constraints, as will be shown below. For an overview of the different existing coupling schemes in the literature, the reader can refer to section II.3 that provides further details about the classification, advantages, and drawbacks of each type of scheme.

Monolithic schemes, while highly stable and giving optimal accuracy, are particularly cumbersome to implement and to maintain, especially when up-to-date state-of-the-art status for various physical processes is required. This difficulty is due to the fact that the coupling requires the solvers to be intertwined (or essentially form a single global solver), making the global system of equations grow in complexity every time a new physical process is integrated. Furthermore, it restricts the solvers to use specific approaches that allow for building such a global system of equations. For this reason, monolithic schemes do not fulfil the requirement of modularity. On the contrary, partitioned schemes are highly modular as they fully separate the fluid and solid solvers. There are, however, different levels of modularity within the different types of partitioned schemes. Implicit schemes require prediction-correction of the solid and fluid solver which can be impractical to implement when one of the solvers does not allow the complete retrieval of the solution of a previous iteration, as discussed earlier. Semi-implicit schemes only need to apply prediction-correction on the solid problem, but they also require a projection-based fluid solver. The latter particularly

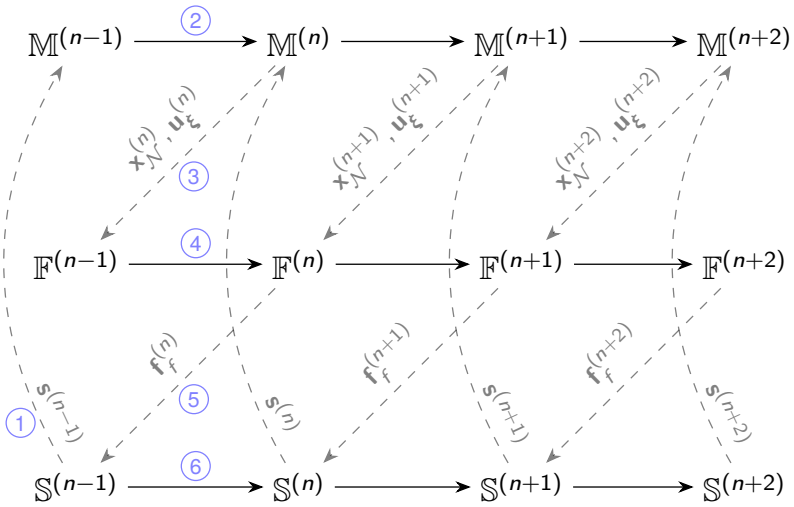


Figure IV.2: Schematic representation of the Conventional Serial Staggered (CSS) scheme
①: transfer structure position to mesh, ②: solve for mesh motion, ③: transfer mesh nodes position and velocity to fluid, ④: solve for fluid, ⑤: transfer fluid forces to structure, ⑥: solve for structure

affects modularity as it requires a specific numerical approach for solving the fluid problem. Fully explicit schemes have none of these restrictions and therefore offer the highest modularity possible. Regarding computational efficiency, which is perhaps the most important factor here, implicit schemes can be deemed prohibitive due to the expensive sub-iterations of the fluid and solid solvers they require until convergence of a solution between both. Semi-implicit schemes, while more efficient computationally than their fully implicit counterparts, still implicitly solve between the fluid pressure and structural equations of motion, implying that potentially non-negligible additional computational cost is needed for complex MBD systems. Fully explicit schemes are by far the most computationally efficient as they need the solvers to be called only once per time step, with no correction step required.

The glaring issue with fully explicit schemes is the fact that, in their simplest state, they are the most prone to instabilities such as the added mass effect. They are also generally considered less accurate than other schemes as no iteration is performed in order to have the fluid and solid solvers converge to a common solution. As will be shown in section IV.2.3, this stability issue can be tackled at a minimal cost in terms of accuracy and efficiency, and without affecting the modular aspect of the global coupling. Unlike the other issues mentioned for monolithic (lack of modularity), implicit (low computational efficiency), and semi-implicit (medium computational efficiency and need for projection-based fluid solver) schemes, the instability issue of explicit schemes can therefore be solved without affecting the other objectives listed above. For all these reasons, fully explicit coupling schemes appear to be the most appropriate method to fulfil all the requirements of the coupling strategy while circumventing all the potential constraints of the coupled solvers. As described in section II.3.2.3, several types of explicit schemes exist and they can be classified in the following categories: parallel, serial, synchronous, and asynchronous. Because parallel schemes are less accurate than their serial counterparts, the following two explicit schemes are implemented for this work: the synchronous CSS scheme and the asynchronous Improved Serial Staggered (ISS) scheme. The details of the implementation of these two schemes are presented in the next two subsections.

Algorithm 4 Conventional Serial Staggered (CSS) scheme

$n = 0$
$t = 0$
while $t < T_{\text{sim}}$ **do**
  Mesh: $\mathbf{x}_{\mathcal{N}}^{(n+1)}, \mathbf{u}_{\boldsymbol{\xi}}^{(n+1)} \leftarrow \mathbb{M}\left(\mathbf{s}^{(n)}\right)$
  Fluid: $\mathbf{f}^{(n+1)} \leftarrow \mathbb{F}\left(\mathbf{x}_{\mathcal{N}}^{(n+1)}, \mathbf{u}_{\boldsymbol{\xi}}^{(n+1)}\right)$
  Body: $\mathbf{s}^{(n+1)} \leftarrow \mathbb{S}\left(\mathbf{f}^{(n+1)}\right)$
  $n \leftarrow n + 1$
  $t \leftarrow t + \Delta t$

IV.2.2.1 Conventional Serial Staggered (CSS) Scheme

The CSS scheme is the most commonly used explicit coupling scheme due to its ease of implementation and high flexibility. Similarly to all explicit schemes, it can be seen as a black-box approach where each solver is called completely independently, with one solver only needing to manipulate the last output of the other solver. Even though the fluid and solid solutions are not produced simultaneously but in a lagged manner, the CSS is synchronous due to the fact that the solutions are produced for the same global temporal values.

As illustrated in fig. IV.2, the solution $\mathbb{M}^{(n+1)}$ for the mesh – node positions $\mathbf{x}_{\mathcal{N}}^{(n+1)}$ and node velocities $\mathbf{u}_{\boldsymbol{\xi}}^{(n+1)}$ – is first calculated from $t^{(n)}$ to $t^{(n+1)}$ using the last position of the solid $\mathbb{S}^{(n)}$ at $t^{(n)}$. In a second time, the solution $\mathbb{F}^{(n+1)}$ for the fluid is then computed from $t^{(n)}$ to $t^{(n+1)}$ using the previously calculated mesh node positions and velocities at $t^{(n+1)}$. Finally, the solution $\mathbb{S}^{(n+1)}$ for the solid is found from $t^{(n)}$ to $t^{(n+1)}$ by using the newly calculated fluid forces $\mathbf{f}_f^{(n+1)}$ and integrating them on the solid boundaries. Note that, alternatively, the mesh motion model for partitioned ALE schemes could be considered as an integral part of the fluid solver. If no mesh deformation is involved in the simulation, then $\mathbf{x}_{\mathcal{N}}^{(n+1)} = \mathbf{x}_{\mathcal{N}}^{(n)} = \mathbf{x}_{\mathcal{N}}^{(0)}$. Algorithm 4 shows the same procedure in the form of a pseudo-algorithm. More details about mesh motion models and ALE formulation is presented in section IV.3.

IV.2.2.2 Improved Serial Staggered (ISS) Scheme

The ISS scheme [89, 43] couples the fluid and body dynamics parts in an asynchronous manner using:

$$t_s \in \left\{ t^{(0)}, \cdots, t^{(n)}, \cdots, T_{\text{sim}} \right\} \tag{IV.6}$$

$$t_f \in \left\{ t^{(\frac{1}{2})}, \cdots, t^{(n+\frac{1}{2})}, \cdots, T_{\text{sim}} - \frac{\Delta t}{2} \right\} \tag{IV.7}$$

with t_s and t_f the values of time for solid and fluid time steps, respectively. The mesh motion model uses the same time step values as the fluid model. The midpoint rule is used for time stepping, with:

$$t^{(n)} = 0.5 \left(t^{(n-\frac{1}{2})} + t^{(n+\frac{1}{2})} \right) \tag{IV.8}$$

This scheme is illustrated in fig. IV.3 along with the pseudo-algorithm in algorithm 5. This asynchronous time stepping requires a predictive step for the mesh motion using $\mathbf{s}^{(n)}$ to get $\mathbf{x}_{\mathcal{N}}^{(n+\frac{1}{2})}$ and $\mathbf{u}_{\boldsymbol{\xi}}^{(n+\frac{1}{2})}$. This can be done by predicting the motion of the solid from $t^{(n)}$ to $t^{(n+\frac{1}{2})}$ with the solid velocity $\dot{\mathbf{s}}^{(n)}$ in order to find the positions of the mesh nodes $\mathbf{x}_{\Gamma_{\text{fcs}}}^{(n+\frac{1}{2})}$ at the boundaries of the solid. This prediction can be done as follows:

$$\mathbf{s}_P^{(n+\frac{1}{2})} = \mathbf{s}^{(n)} + \alpha_0 \dot{\mathbf{s}}^{(n)} \frac{\Delta t}{2} + \alpha_1 \left(\dot{\mathbf{s}}^{(n)} - \dot{\mathbf{s}}^{(n-1)} \right) \frac{\Delta t}{2} \tag{IV.9}$$

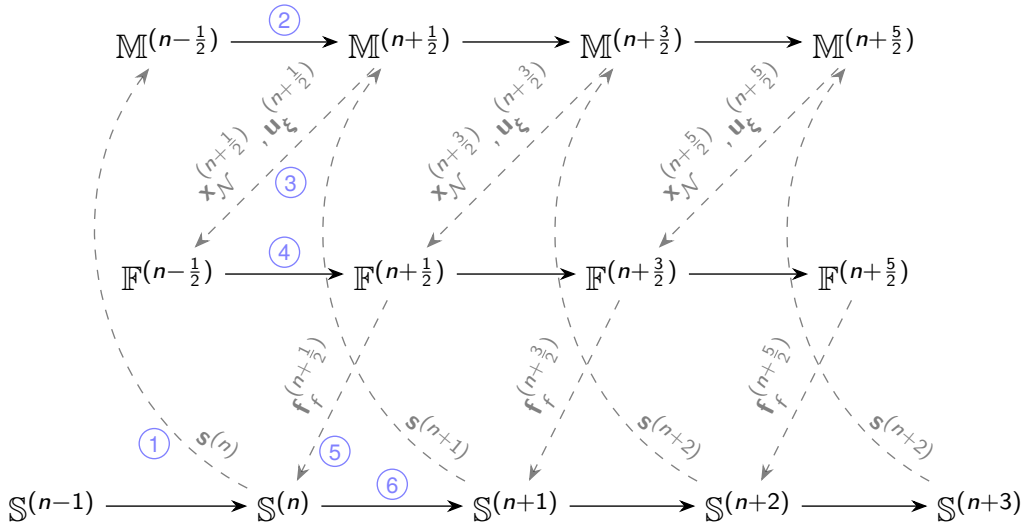


Figure IV.3: Schematic representation of the Improved Serial Staggered (ISS) scheme
①: transfer structure position to mesh, ②: solve for mesh motion, ③: transfer mesh nodes position and velocity to fluid, ④: solve for fluid, ⑤: transfer fluid forces to structure, ⑥: solve for structure

with α_0 and α_1 two given coefficients according to the order of time integration desired. These nodal displacements calculated on the boundary $\Gamma_{\mathrm{f}\cap\mathrm{s}}$ at $t^{(n+\frac{1}{2})}$ are then used as boundary conditions to compute the position $\mathbf{x}_{\mathcal{N}}^{(n+\frac{1}{2})}$ of the mesh nodes and their velocity $\mathbf{u}_{\boldsymbol{\xi}}^{(n+\frac{1}{2})}$ in Ω_f. While the mesh is updated to $t^{(n+\frac{1}{2})}$ using $\mathbf{s}_P^{(n+\frac{1}{2})}$ for the boundary conditions of the displacement at the solid boundaries, the structure itself remains at $t^{(n)}$. This means that the fluid forces $\mathbf{f}_f^{(n+\frac{1}{2})}$ found at $t^{(n+\frac{1}{2})}$ are applied on the solid at $t^{(n)}$ to find $\mathbf{s}^{(n+1)}$. On the subsequent step, the nodes at the solid boundaries can be seen as re-initialised to the actual position of the body at $t^{(n+1)}$ (instead of $t^{(n+\frac{1}{2})}$) by using the now known $\mathbf{s}^{(n+1)}$, and have their position at $t^{(n+\frac{3}{2})}$ estimated by using the known velocity $\dot{\mathbf{s}}^{(n+1)}$.

One issue with ISS is that the subsequent value of the fluid time step – $t^{(n+\frac{1}{2})}$ – must be known in advance in order to get the solid solution at the temporal midpoint of two fluid steps $t^{(n)}$. The commonly used Courant–Friedrichs–Lewy (CFL) condition can lead to varying time steps, making the midpoint rule difficult to apply. It has also been found in this work that, if added mass effects are significant, the ISS scheme is less stable than the CSS scheme. This increase in instability is due to the predictive step taken that actually emphasizes the oscillatory behaviour of forces acting on the body due to the added mass effect, making the numerical solution diverge faster. For these two main reasons, the CSS scheme has been selected for all simulations presented in chapter V.

Algorithm 5 Improved Serial Staggered (ISS) scheme

$n = 0$
$t = 0$
while $t < T_{\text{sim}}$ **do**
  Mesh: $\mathbf{x}_{\mathcal{N}}^{(n+\frac{1}{2})}, \mathbf{u}_{\boldsymbol{\xi}}^{(n+\frac{1}{2})} \leftarrow \mathbb{M}\left(\mathbf{s}^{(n)}, \dot{\mathbf{s}}^{(n)}\right)$
  Fluid: $\mathbf{f}^{(n+\frac{1}{2})} \leftarrow \mathbb{F}\left(\mathbf{x}_{\mathcal{N}}^{(n+\frac{1}{2})}, \mathbf{u}_{\boldsymbol{\xi}}^{(n+\frac{1}{2})}\right)$
  Body: $\mathbf{s}^{n+1} \leftarrow \mathbb{S}\left(\mathbf{f}^{(n+\frac{1}{2})}\right)$
  $n \leftarrow n + 1$
  $t \leftarrow t + \Delta t$

IV.2.3 Added Mass Effect

The added mass effect refers to the instability caused by partitioned FSI schemes when a solid body accelerates within a fluid. The moving solid must displace the surrounding fluid as it accelerates, and this effect is partially ignored in partitioned schemes when stepping for the solid, as is explained in additional detail below. For additional background on the added mass, the added mass effect and the different solutions that can be applied to it, the reader can refer to section II.4.

IV.2.3.1 Stabilisation for Explicit Coupling Schemes

When solving the structural equations of motion, the following is often used directly in Computational Fluid Dynamics (CFD) models for FSI:

$$\mathbf{M}\ddot{\mathbf{s}} = \mathbf{f}_f + \mathbf{f}_e \tag{IV.10}$$

with $\mathbf{M}$ the mass matrix, $\mathbf{f}_f$ the force from the fluid solver, and $\mathbf{f}_e$ the sum of external forces. In CFD, the added mass contributions at time $t^{(n)}$ are inherently taken into account in $\mathbf{f}_f^{(n)}$ and act as a force opposite to the body acceleration. If the added mass is known and its contributions can be isolated from $\mathbf{f}_f$, eq. (IV.10) can be written as:

$$\left(\mathbf{M} + \widehat{\mathbf{A}}\right)\ddot{\mathbf{s}} = \widehat{\mathbf{f}}_f + \mathbf{f}_e \tag{IV.11}$$

with $\widehat{\mathbf{f}}_f$ the fluid force exempt from added mass contributions, and $\widehat{\mathbf{A}}$ the added mass of the body. Equation (IV.11) can also be written as follows:

$$\mathbf{M}\ddot{\mathbf{s}} = \underbrace{\widehat{\mathbf{f}}_f - \widehat{\mathbf{A}}\ddot{\mathbf{s}}}_{\mathbf{f}_f} + \mathbf{f}_e \tag{IV.12}$$

where the negative sign of the added mass contribution can be explained by the fact that it corresponds to an effort that is always opposite to the acceleration of the body, as the body needs to push the surrounding fluid as it accelerates. Equation (IV.12) is stable as long as the fluid and solid problems are solved simultaneously, such as in monolithic coupling. With partitioned schemes, and particularly non-iterative coupling scheme such as CSS, the time discretised version of eq. (IV.12) can be written as eq. (IV.13) when solving for the solid at time $t^{(n+1)}$:

$$\mathbf{M}\ddot{\mathbf{s}}^{(n+1)} = \widehat{\mathbf{f}}_f^{(n)} - \widehat{\mathbf{A}}\ddot{\mathbf{s}}^{(n)} + \mathbf{f}_e^{(n)} \tag{IV.13}$$

with the superscripts $^{(n)}$ and $^{(n+1)}$ indicating the solutions at time $t^{(n)}$ and $t^{(n+1)}$. An important aspect is then omitted in this equation: the effect of the added mass contributions on the body due to its change of acceleration from $t^{(n)}$ to $t^{(n+1)}$ relative to the fluid. This omission in partitioned schemes can give rise to an unconditional instability when the added mass is greater than the mass of the body.

For illustrating a non-iterative partitioned scheme subject to the added mass effect in a simple manner, consider the acceleration of a single Degree of Freedom (DOF) body of mass m and added mass m_a at time step $t^{(n)}$ at position $r^{(n)}$ subject to a constant force f over time as well as an added mass force term that depends on the acceleration of the body at the previous time step ($-m_a\left(\ddot{r}^{(n-1)}\right)$) due to the lag of a partitioned scheme. This is essentially a 1D problem that can be written as:

$$m\ddot{r}^{(n+1)} = f^{(n)} - m_a\ddot{r}^{(n)} \tag{IV.14}$$

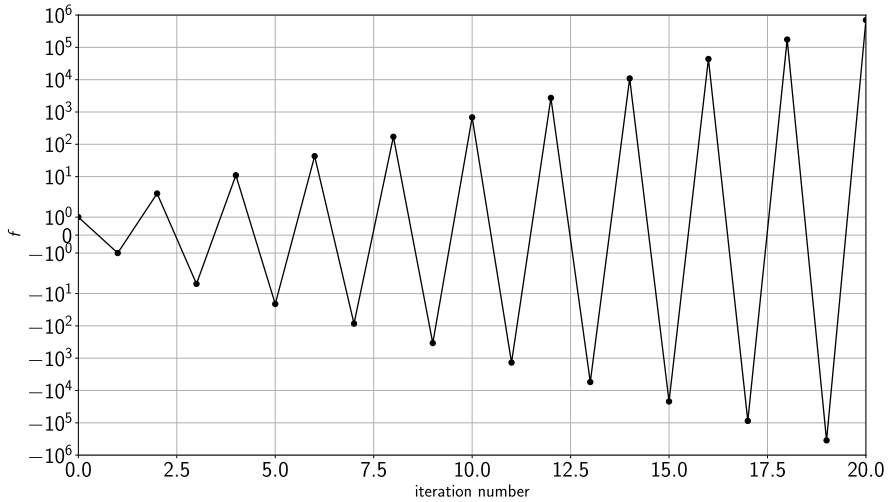


Figure IV.4: Illustration of the added mass effect encountered on partitioned coupling schemes when $m_a = 2m$

If the initial acceleration $r^{(0)} = 0$ and the added mass is known and set to $m_a = 2m$ the first few steps will yield:

$$
\begin{aligned}
mr^{(1)} &= f - m_a r^{(0)} = f - 2m0 = +1f \\
mr^{(2)} &= f - m_a r^{(1)} = f - 2mr^{(1)} = -1f \\
mr^{(3)} &= f - m_a r^{(2)} = f - 2mr^{(2)} = +3f \\
mr^{(4)} &= f - m_a r^{(3)} = f - 2mr^{(3)} = -5f \\
mr^{(5)} &= f - m_a r^{(4)} = f - 2mr^{(4)} = +10f \\
mr^{(6)} &= f - m_a r^{(5)} = f - 2mr^{(5)} = -19f
\end{aligned}
$$

It appears clearly here that the added mass contributions from the previous time step lead to an instability that grows in magnitude after each step of the non-iterative coupling. A graph representing this instability is plotted in fig. IV.4 for the first twenty steps. Going back to multi-dimensional matrix notation, a way to tackle this issue is to introduce a good estimation of the added mass matrix when eq. (IV.10) is discretised in time:

$$
(\mathbf{M} + \mathbf{A})\,\ddot{\mathbf{s}} = \widehat{\mathbf{f}}_f + \mathbf{f}_e \tag{IV.15}
$$

with $\mathbf{A}$ the estimated added mass matrix, not to be confused with $\widehat{\mathbf{A}}$ that corresponds to the actual added mass that was defined above. In CFD, the added mass contributions are inherently taken into account in $\mathbf{f}_f$, so $\widehat{\mathbf{f}}_f$ is not actually known explicitly. In this case, it becomes technically impossible to separate the added mass contribution from the forces of the Right Hand Side (RHS) of eq. (IV.10) to apply them as an inertial term on the Left Hand Side (LHS) as in eq. (IV.15). Note that the fact that the added mass is inherently taken into account here is in direct contrast with methods such as potential flow where $\widehat{\mathbf{f}}_f$ is provided but $\mathbf{f}_f$ is unknown, and where the added mass matrix of the solid is therefore estimated and directly added in the full mass matrix of the solid. Simply adding an estimation of the added mass on the LHS of eq. (IV.10) would modify the intended inertia — and therefore response — of the structure in a two-way fully coupled CFD model. Although it could stabilise the problem as the unstable added mass contributions from the RHS would be compensated, the inertia of the body would be significantly affected.

Due to the "lag" between the fluid and solid solutions when using partitioned coupling schemes in CFD , extra care must be given to counteract the added mass effect. As hinted above, the main problem here is that only the fluid forces from $t^{(n)}$ are taken into account, but not the additional effect of added mass resulting from the evolution of the solid from $t^{(n)}$ to $t^{(n+1)}$ within this same fluid. The aim here is to accurately estimate those added mass contributions from $t^{(n)}$ to $t^{(n+1)}$ in order to stabilise the coupling without affecting the expected response of the body. In other terms, the following equation is to be imposed:

$$\mathbf{M}\ddot{\mathbf{s}}^{(n+1)} = \mathbf{f}_f^{(n)} - \mathbf{A}\left(\ddot{\mathbf{s}}^{(n+1)} - \ddot{\mathbf{s}}^{(n)}\right) + \mathbf{f}_e^{(n)} \tag{IV.16}$$

where the term $\mathbf{A}\left(\ddot{\mathbf{s}}^{(n+1)} - \ddot{\mathbf{s}}^{(n)}\right)$ on the RHS of eq. (IV.16) can be seen as a penalty term, penalising large variations or oscillations in the body acceleration over time. This term is actually taking into account the added mass force due to the change in acceleration of the body within a time step. Oscillations due to the added mass effect are guaranteed to be damped out in eq. (IV.16) when using an appropriate $\mathbf{A}$ (i.e. $\mathbf{A} \geq \widehat{\mathbf{A}}$). As the only unknown term here is $\ddot{\mathbf{s}}^{(n+1)}$, eq. (IV.16) can be expressed for clarity as:

$$\left(\mathbf{M} + \mathbf{A}\right)\ddot{\mathbf{s}}^{(n+1)} = \mathbf{f}_f^{(n)} + \mathbf{A}\ddot{\mathbf{s}}^{(n)} + \mathbf{f}_e^{(n)} \tag{IV.17}$$

With this method, the estimated added mass matrix $\mathbf{A}$ must be added to the full mass matrix of the structure, and an additional force term – $\mathbf{A}\ddot{\mathbf{s}}^{(n)}$ – must be applied to the structure. In other words, the aim here is to cancel out the added mass contributions calculated by the fluid solver in $\mathbf{f}_f^{(n)}$ and to impose them fully as an inertial term affecting the full mass matrix $\left(\mathbf{M} + \mathbf{A}\right)$ of the body. Note that if $\mathbf{A}$ corresponds exactly to the actual added mass $\widehat{\mathbf{A}}$ that is taken into account in $\mathbf{f}_f^{(n)}$, eq. (IV.17) is then equivalent to eq. (IV.15), because $\mathbf{f}_f + \widehat{\mathbf{A}}\ddot{\mathbf{s}} = \widehat{\mathbf{f}}_f$. As mentioned in section II.4, a relaxation factor can be used alternatively to eq. (IV.17) as follows [134, 133]:

$$\mathbf{M}\ddot{\mathbf{s}} = \alpha_c\left(\mathbf{f}_f + \mathbf{f}_e\right) + \left(1 - \alpha_c\right)\mathbf{M}\ddot{\mathbf{s}} \tag{IV.18}$$

with the relaxation factor $\alpha_c = \frac{1}{1+\frac{\mathbf{M}}{\mathbf{A}}}$. As it appears clearly in eq. (IV.18), the contributions due to the added mass correction are all in the RHS when using this relaxation factor, meaning that the mass matrix of the body does not need to be modified. One argument in favour of using a relaxation factor is therefore that it does not require modification of the body dynamics solver, but only to impose an additional force term. While this is true, it must be clarified that, on the other hand, all the forces are affected by the introduction of α_c and must therefore be scaled accordingly. This can prove difficult when more complex mechanisms are introduced, such as MBD systems with joints, springs, collisions, or mooring cables, as modifications of the forces rising from these complexities must also be corrected with α_c. Scaling those forces can be a difficult task as one might actually have to go deeper in the solid solver logic in order to apply α_c appropriately. This can even lead to modifications of other models, such as when collision detection is handled by a specialised solver (as collision forces must also be scaled). Without relaxation factor, the only modification necessary is modifying the mass matrix of the body ($\mathbf{A} + \mathbf{M}$) and adding a single external force ($\mathbf{A}\mathbf{s}^{(n)}$), while the rest of the external contributions are left untouched. Because the structures presented in the simulations of this thesis involve MBD, collision detection, mooring cables, and because even more complexities could be added in the future, it was deemed more appropriate to use the added mass correction method directly as shown in eq. (IV.17). The MBD solver – Chrono– used for the purpose of this work had to be

modified in order to apply the added mass contributions appropriately as it did not originally use full 6×6 mass matrices to described rigid body dynamics, but only scalar mass values m along with 3×3 inertia tensors $\bar{\bar{\mathbf{I}}}_t$. Again, allowing 6×6 matrices to be used in the MBD solver is the only modification necessary to include the added mass contributions of a rigid body, with the application of a single external force ($\mathbf{As}^{(n)}$) and without scaling any of the external force terms, regardless of the complexity of the MBD problem.

IV.2.3.2 Estimation of the Added Mass

When using the stabilisation scheme of eq. (IV.17), an appropriate estimate of the added mass $\mathbf{A}$ is necessary (i.e. as close as possible to the actual added mass $\widehat{\mathbf{A}}$) in order to retain a certain level of accuracy in regards to the response of the body. The method implemented and presented here for estimating the added mass numerically here is based on a method that was first introduced by [116].

Consider two flow fields that have equal velocity $\mathbf{u}_a = \mathbf{u}_b$ at a given time. Although the fluid velocity is equal between the two fields, the fluid acceleration can be different. This difference in acceleration is used here to generate a pressure differential field caused by accelerating boundaries. From the NSEs (eq. (III.19)), and taking the difference between the two flow fields, all terms featuring the fluid velocity can be eliminated, resulting in:

$$\dot{\widetilde{\mathbf{u}}} = -\frac{1}{\rho}\nabla\widetilde{\mathrm{p}} \tag{IV.19}$$

with $\dot{\widetilde{\mathbf{u}}} = \dot{\mathbf{u}}_a - \dot{\mathbf{u}}_b$ and $\widetilde{\mathrm{p}} = \mathrm{p}_a - \mathrm{p}_b$. It is known that the fluid velocity field is also divergence-free through the continuity equation, leading to the following expression when taking the divergence of eq. (IV.19):

$$\nabla \cdot \left(-\frac{1}{\rho}\nabla\widetilde{\mathrm{p}} \right) = 0 \tag{IV.20}$$

One of the main differences with [116] here is that the density term is kept in eq. (IV.20), while [116] assumes that it is constant, leading to the Laplace equation $\Delta\widetilde{\mathrm{p}} = 0$. Dropping the density term here only works for Single-Phase Flow (SPF) with incompressible fluid where ρ is constant, or for Multiphase Flow (MPF) if free surface tracking is done explicitly in order to isolate the incompressible phase of interest when solving the equation. Keeping ρ in eq. (IV.20) allows for solving the problem appropriately when using implicitly tracked free surface.

To solve eq. (IV.20), boundary conditions corresponding to the body acceleration must be applied on its boundaries. The equation is solved for a unit acceleration $\boldsymbol{\alpha} = (\alpha_1, \alpha_2, \alpha_3)$. Using eq. (IV.19) with $\dot{\widetilde{\mathbf{u}}} = \boldsymbol{\alpha}$ on $\Gamma_{\mathrm{f}\cap\mathrm{s}}$, the following Neumann boundary conditions are applied to the fluid–structure boundary and other solid walls that are not part of the body considered, respectively:

$$\nabla\widetilde{\mathrm{p}} \cdot \mathbf{n} = -\rho\boldsymbol{\alpha} \cdot \mathbf{n} \qquad \forall \mathbf{x} \in \Gamma_{\mathrm{f}\cap\mathrm{s}} \tag{IV.21}$$

$$\nabla\widetilde{\mathrm{p}} \cdot \mathbf{n} = 0 \qquad \forall \mathbf{x} \in \Gamma_{\mathrm{wall}} \tag{IV.22}$$

This Boundary Value Problem (BVP) must be solved for each DOF of the solid body, using the appropriate unit acceleration of the DOF considered. In 2D, this corresponds to solving three BVPs, while six BVPs must be solved in 3D. The corresponding unit acceleration vectors to apply as boundary conditions on the solid boundaries for each of the DOFs considered are compiled in table IV.1. For a given $\boldsymbol{\alpha}_j$, integrating eq. (IV.23) for translational motion

and eq. (IV.24) for rotational motion over the structure surface allows for filling a column j of $\mathbf{A}$:

$$\mathbf{A}_{j,t} = \int_{\Gamma_{f\cap s}} \widetilde{p}_j \mathbf{n} d\Gamma \tag{IV.23}$$

$$\mathbf{A}_{j,r} = \int_{\Gamma_{f\cap s}} (\mathbf{x} - \mathbf{r}) \times \widetilde{p}_j \mathbf{n} d\Gamma \tag{IV.24}$$

with $\mathbf{r}$ the barycentre of the body, $\mathbf{n}$ the normal vector to boundary $\Gamma_{f\cap s}$, $\mathbf{x}$ a point on $\Gamma_{f\cap s}$, and $\mathbf{A}_{j,t}$ and $\mathbf{A}_{j,r}$ the translational and rotational components of the column j of the added mass matrix, respectively. The effect of the unit acceleration $\boldsymbol{\alpha}_j$ on the current DOF is given by $\mathbf{A}_{i,i}$, while the effect of $\boldsymbol{\alpha}_j$ on the other DOFs is given by $\mathbf{A}_{i,j}\ \forall i \neq j$. Again, for a body with all 6 DOFs in 3D, the equation must be solved 6 times in order to fill the 6×6 added mass matrix entirely, changing the unit acceleration component $\boldsymbol{\alpha}_j$ for the fluid–structure boundary conditions accordingly.

It is important to note that because eq. (IV.21) is solved in the global frame of reference and when the equation of motion of the body is solved in its local frame of reference (which is common practice with solid solvers), the added mass matrix $\mathbf{A}_g$ in the global frame of reference can be transformed in order to obtain the added mass matrix $\mathbf{A}_l$ in the local frame of reference, as follows:

$$\mathbf{A}_l = \mathbf{R}_{6\times6}{}^{\mathsf{T}} \mathbf{A}_g \mathbf{R}_{6\times6} \tag{IV.25}$$

with $\mathbf{R}_{6\times6}$ the rotation matrix defined as:

$$\mathbf{R}_{6\times6} = \begin{pmatrix} \mathbf{R} & 0 \\ 0 & \mathbf{R} \end{pmatrix} \tag{IV.26}$$

with $\mathbf{R}$ the 3×3 rotation matrix of the body relative to the global frame of reference.

There are other ways to estimate the added mass such as, for example, calculating its value at infinite frequency when using a potential flow solver. The issue with this method is that the added mass matrix calculated at infinite frequency remains constant during the whole simulation. In the case of MPF and floating structures, the position, orientation, and submergence of the structure relative to the different fluid phases actually matter and affect the value of the added mass. Ideally, $\mathbf{A}$ should be re-estimated when the free surface moves relative to the body. The only case where $\mathbf{A}$ would remain constant over time for a moving structure would be in a single-phase domain with boundaries that can be considered infinitely far away from the body boundaries. The method described above can be used to estimate $\mathbf{A}$ dynamically by recalculating it at each time step, and require no external tool as the finite element space of the fluid domain mesh can be directly used to solve the PDE of eq. (IV.21). If the magnitude of the temporal variation of $\mathbf{A}$ over a simulation is not very significant, only periodical estimations (or even only an initial estimation) can be assumed sufficient.

Table IV.1: Unit acceleration for boundary conditions at the fluid–structure boundary of added mass model

	surge	sway	heave	roll	pitch	yaw
j	1	2	3	4	5	6
α_1	1	0	0	0	$+\|x_3 - r_3\|$	$-\|x_2 - r_2\|$
α_2	0	1	0	$-\|x_3 - r_3\|$	0	$+\|x_1 - r_1\|$
α_3	0	0	1	$+\|x_2 - r_2\|$	$-\|x_1 - r_1\|$	0

Note: x_1, x_2, and x_3 the coordinates at the boundary, and r_1, r_2, and r_3 the coordinates of the barycentre of the solid body (or pivotal point)

IV.3 Moving Domains

This section focuses on the numerical handling of the moving fluid–solid interface $\Gamma_{f\cap s}$. Firstly, the ALE formulation and principles for moving domains are explained along with their effect on the governing equations in section IV.3.1. Then, two mesh deformation techniques that have been implemented for this work to accomodate for the displacement of $\Gamma_{f\cap s}$ are presented in section IV.3.2: the elastostatics model in section IV.3.2.1, and monitor function model in section IV.3.2.2. Finally, metrics to estimate the quality of the mesh, and mesh smoothing algorithms are introduced in section IV.3.3. For an introduction and definitions of Lagrangian, Eulerian, and ALE frames of reference, as well as an overview of different mesh deformation methods available in the literature, the reader can refer to section II.5.

IV.3.1 Arbitrary Lagrangian–Eulerian (ALE) Formulation

In the ALE approach for moving domains, three different domains — reference, spatial, and material — are mapped to each other independently from the approach used for mesh deformation. It is demonstrated below how field variables and equations such as the NSEs are modified in order to take into account this mesh deformation. Throughout this section, the material domain of the Lagrangian frame of reference is denoted $\Omega_{\widehat{\mathbf{x}}}$, with material coordinates $\widehat{\mathbf{x}} \in \Omega_{\widehat{\mathbf{x}}}$, the spatial domain of the Eulerian frame of reference is denoted $\Omega_{\mathbf{x}}$, with spatial coordinates $\mathbf{x} \in \Omega_{\mathbf{x}}$, and the reference domain of the ALE frame of reference is denoted $\Omega_{\boldsymbol{\xi}}$, with reference coordinates $\boldsymbol{\xi} \in \Omega_{\boldsymbol{\xi}}$. For each spatial point $\mathbf{x} \in \Omega_{\mathbf{x}}$ at any time t, a corresponding material point $\widehat{\mathbf{x}} \in \Omega_{\widehat{\mathbf{x}}}$ and a corresponding reference point $\boldsymbol{\xi} \in \Omega_{\boldsymbol{\xi}}$ exist. As shown in fig. IV.5, the domains are related to each other with the following maps: Φ mapping the reference domain to the spatial domain (eq. (IV.49)), $\widehat{\Phi}$ mapping the reference domain to the material domain (eq. (IV.28)), and Ψ mapping the material domain to the spatial domain (eq. (IV.29)):

$$\begin{aligned} \Phi : \Omega_{\boldsymbol{\xi}} \times [0, T_{\text{sim}}] &\to \Omega_{\mathbf{x}} \times [0, T_{\text{sim}}] \\ (\boldsymbol{\xi}, t) &\mapsto \Phi(\boldsymbol{\xi}, t) = (\mathbf{x}, t) \end{aligned} \tag{IV.27}$$

$$\begin{aligned} \widehat{\Phi} : \Omega_{\boldsymbol{\xi}} \times [0, T_{\text{sim}}] &\to \Omega_{\widehat{\mathbf{x}}} \times [0, T_{\text{sim}}] \\ (\boldsymbol{\xi}, t) &\mapsto \widehat{\Phi}(\boldsymbol{\xi}, t) = (\widehat{\mathbf{x}}, t) \end{aligned} \tag{IV.28}$$

$$\begin{aligned} \Psi : \Omega_{\widehat{\mathbf{x}}} \times [0, T_{\text{sim}}] &\to \Omega_{\mathbf{x}} \times [0, T_{\text{sim}}] \\ (\widehat{\mathbf{x}}, t) &\mapsto \Psi(\widehat{\mathbf{x}}, t) = (\mathbf{x}, t) \end{aligned} \tag{IV.29}$$

Note that Ψ represents the particle motion and can be expressed as $\Psi = \Phi \circ \widehat{\Phi}^{-1}$, implying that the mappings are not independent. Furthermore, if $\widehat{\Phi} = \mathbb{1}$, the formulation becomes equivalent to a purely Lagrangian approach where $\Omega_{\boldsymbol{\xi}} = \Omega_{\widehat{\mathbf{x}}}$; and if $\Phi = \mathbb{1}$, the formulation becomes equivalent to a purely Eulerian approach where $\Omega_{\boldsymbol{\xi}} = \Omega_{\mathbf{x}}$.

The transformations from a domain to another can be written in terms of Jacobians J. Following [71, 86, 5], the Jacobians can be used to find the expression of any field in any frame of reference. The Jacobian of the transformation Φ from the reference domain to the spatial domain is given in eq. (IV.30) and, for convenience and later derivation, the transformation from the material domain to the reference domain $\widehat{\Phi}^{-1}$ is expressed in eq. (IV.31)

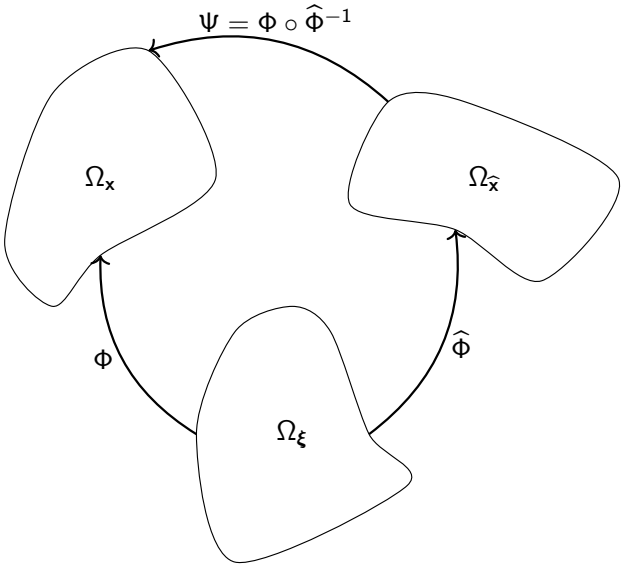


Figure IV.5: Mapping between spatial domain $\Omega_{\mathbf{x}}$, material domain $\Omega_{\widehat{\mathbf{x}}}$, and reference domains $\Omega_{\boldsymbol{\xi}}$

as follows:

$$J_{\Phi}\left(\boldsymbol{\xi}, t\right) = \frac{\partial \Phi}{\partial\left(\boldsymbol{\xi}, t\right)} = \begin{bmatrix} \frac{\partial \mathbf{x}}{\partial \boldsymbol{\xi}} & \frac{\partial \mathbf{x}}{\partial t}\Big|_{\boldsymbol{\xi}} \\ \frac{\partial t}{\partial \boldsymbol{\xi}} & \frac{\partial t}{\partial t}\Big|_{\boldsymbol{\xi}} \end{bmatrix} = \begin{bmatrix} \frac{\partial \mathbf{x}}{\partial \boldsymbol{\xi}} & \dot{\mathbf{x}}\Big|_{\boldsymbol{\xi}} \\ 0 & 1 \end{bmatrix} \tag{IV.30}$$

$$J_{\widehat{\Phi}^{-1}}\left(\widehat{\mathbf{x}}, t\right) = \frac{\partial \widehat{\Phi}^{-1}}{\partial\left(\widehat{\mathbf{x}}, t\right)} = \begin{bmatrix} \frac{\partial \boldsymbol{\xi}}{\partial \widehat{\mathbf{x}}} & \frac{\partial \boldsymbol{\xi}}{\partial t}\Big|_{\widehat{\mathbf{x}}} \\ \frac{\partial t}{\partial \widehat{\mathbf{x}}} & \frac{\partial t}{\partial t}\Big|_{\widehat{\mathbf{x}}} \end{bmatrix} = \begin{bmatrix} \frac{\partial \boldsymbol{\xi}}{\partial \widehat{\mathbf{x}}} & \dot{\boldsymbol{\xi}}\Big|_{\widehat{\mathbf{x}}} \\ 0 & 1 \end{bmatrix} \tag{IV.31}$$

with $\dot{\mathbf{x}}\Big|_{\boldsymbol{\xi}}$ the notation for evaluation of the time derivative of spatial coordinates $\mathbf{x}$ holding reference coordinates $\boldsymbol{\xi}$ fixed in space, which corresponds to the mesh velocity $\mathbf{u}_{\boldsymbol{\xi}} = \dot{\mathbf{x}}\Big|_{\boldsymbol{\xi}}$, and $\dot{\boldsymbol{\xi}}\Big|_{\widehat{\mathbf{x}}}$ the notation for evaluation of the time derivative of the reference coordinates $\boldsymbol{\xi}$ holding material coordinates $\widehat{\mathbf{x}}$ fixed in space. The transformation from the material domain to the spatial domain can then be found using the relation $\Psi = \Phi \circ \widehat{\Phi}^{-1}$ as follows:

$$\begin{aligned} J_{\Psi}\left(\widehat{\mathbf{x}}, t\right) = \frac{\partial \Psi\left(\widehat{\mathbf{x}}, t\right)}{\partial\left(\widehat{\mathbf{x}}, t\right)} = \frac{\partial\left(\Phi \circ \widehat{\Phi}^{-1}\right)}{\partial\left(\widehat{\mathbf{x}}, t\right)} &= \frac{\partial \Phi\left(\widehat{\Phi}^{-1}\left(\widehat{\mathbf{x}}, t\right)\right)}{\partial\left(\boldsymbol{\xi}, t\right)} \frac{\partial \widehat{\Phi}^{-1}\left(\widehat{\mathbf{x}}, t\right)}{\partial\left(\widehat{\mathbf{x}}, t\right)} \\ &= \frac{\partial \Phi\left(\boldsymbol{\xi}, t\right)}{\partial\left(\boldsymbol{\xi}, t\right)} \frac{\partial \widehat{\Phi}^{-1}\left(\widehat{\mathbf{x}}, t\right)}{\partial\left(\widehat{\mathbf{x}}, t\right)} \end{aligned} \tag{IV.32}$$

Or in matrix form using eq. (IV.30) and eq. (IV.31) into eq. (IV.32):

$$\begin{bmatrix} \frac{\partial \mathbf{x}}{\partial \widehat{\mathbf{x}}} & \dot{\mathbf{x}}\Big|_{\widehat{\mathbf{x}}} \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} \frac{\partial \mathbf{x}}{\partial \boldsymbol{\xi}} & \dot{\mathbf{x}}\Big|_{\boldsymbol{\xi}} \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \frac{\partial \boldsymbol{\xi}}{\partial \widehat{\mathbf{x}}} & \dot{\boldsymbol{\xi}}\Big|_{\widehat{\mathbf{x}}} \\ 0 & 1 \end{bmatrix} \tag{IV.33}$$

From which it appears clearly that the material velocity $\mathbf{u}_{\widehat{\mathbf{x}}} = \dot{\mathbf{x}}\Big|_{\widehat{\mathbf{x}}} \left(= \frac{\mathrm{d}\mathbf{x}}{\mathrm{d}t}\right)$ can be expressed as $\mathbf{u}_{\widehat{\mathbf{x}}} = \mathbf{u}_{\boldsymbol{\xi}} + \frac{\partial \mathbf{x}}{\partial \boldsymbol{\xi}} \cdot \dot{\boldsymbol{\xi}}\Big|_{\widehat{\mathbf{x}}}$, and where $\frac{\partial \mathbf{x}}{\partial \boldsymbol{\xi}} \cdot \dot{\boldsymbol{\xi}}\Big|_{\widehat{\mathbf{x}}}$ is known as the convective velocity, corresponding to the velocity at which the referential and material coordinates diverge. Considering a field variable ϕ with known material coordinates $\widehat{\mathbf{x}}$, it can be expressed

in the material domain $\Omega_{\widehat{\mathbf{x}}}$ as $\phi(\widehat{\mathbf{x}}, t)$ or, equivalently, in the reference domain $\Omega_{\boldsymbol{\xi}}$ as $\phi(\boldsymbol{\xi}, t) \circ \widehat{\Phi}^{-1}$. This leads to the following expression for the material derivative of ϕ:

$$\frac{\partial \phi(\widehat{\mathbf{x}}, t)}{\partial (\widehat{\mathbf{x}}, t)} = \frac{\partial \phi(\boldsymbol{\xi}, t)}{\partial (\boldsymbol{\xi}, t)} \frac{\partial \widehat{\Phi}^{-1}}{\partial (\widehat{\mathbf{x}}, t)} \tag{IV.34}$$

$$\begin{bmatrix} \frac{\partial \phi}{\partial \widehat{\mathbf{x}}} & \dot{\phi}\Big|_{\widehat{\mathbf{x}}} \end{bmatrix} = \begin{bmatrix} \frac{\partial \phi}{\partial \boldsymbol{\xi}} & \dot{\phi}\Big|_{\boldsymbol{\xi}} \end{bmatrix} \begin{bmatrix} \frac{\partial \boldsymbol{\xi}}{\partial \widehat{\mathbf{x}}} & \dot{\boldsymbol{\xi}}\Big|_{\widehat{\mathbf{x}}} \\ 0 & 1 \end{bmatrix} \tag{IV.35}$$

which yields the following expression for the time derivative of the material description of ϕ in the ALE frame of reference:

$$\dot{\phi}\Big|_{\widehat{\mathbf{x}}} = \dot{\phi}\Big|_{\boldsymbol{\xi}} + \dot{\boldsymbol{\xi}}\Big|_{\widehat{\mathbf{x}}} \frac{\partial \phi}{\partial \boldsymbol{\xi}} = \dot{\phi}\Big|_{\boldsymbol{\xi}} + (\mathbf{u}_{\widehat{\mathbf{x}}} - \mathbf{u}_{\boldsymbol{\xi}}) \cdot \frac{\partial \boldsymbol{\xi}}{\partial \mathbf{x}} \frac{\partial \phi}{\partial \boldsymbol{\xi}} \tag{IV.36}$$

$$= \dot{\phi}\Big|_{\boldsymbol{\xi}} + (\mathbf{u}_{\widehat{\mathbf{x}}} - \mathbf{u}_{\boldsymbol{\xi}}) \cdot \nabla_{\mathbf{x}} \phi \tag{IV.37}$$

where $(\mathbf{u}_{\widehat{\mathbf{x}}} - \mathbf{u}_{\boldsymbol{\xi}}) = \dot{\boldsymbol{\xi}}\Big|_{\widehat{\mathbf{x}}} \frac{\partial \mathbf{x}}{\partial \boldsymbol{\xi}}$ is the convective velocity discussed above. Therefore, recalling the Lagrangian and Eulerian formulations here, the different formulations possible for the time derivative of ϕ in each frame of reference is as follows:

$$\frac{\mathrm{d}\phi}{\mathrm{d}t} = \begin{cases} \dot{\phi}\Big|_{\widehat{\mathbf{x}}} & \text{Lagrangian} \\ \dot{\phi}\Big|_{\mathbf{x}} + \mathbf{u}_{\widehat{\mathbf{x}}} \cdot \nabla \phi & \text{Eulerian} \\ \dot{\phi}\Big|_{\boldsymbol{\xi}} + (\mathbf{u}_{\widehat{\mathbf{x}}} - \mathbf{u}_{\boldsymbol{\xi}}) \cdot \nabla \phi & \text{ALE} \end{cases} \tag{IV.38}$$

In the context of CFD and when the domain is deformed due to the motion of a fluid–solid interface $\Gamma_{\mathrm{f} \cap \mathrm{s}}$, the time derivative of the fluid velocity must be expressed using the ALE formulation. Combining eq. (III.19) and eq. (IV.38), the NSEs become:

$$\begin{cases} \rho \dot{\mathbf{u}}_f\Big|_{\boldsymbol{\xi}} + \rho (\mathbf{u}_f - \mathbf{u}_{\boldsymbol{\xi}}) \cdot \nabla \mathbf{u}_f = \nabla \cdot \bar{\bar{\sigma}} + \rho \mathbf{g} \\ \nabla \cdot \mathbf{u}_f = 0 \end{cases} \tag{IV.39}$$

Note that the field variable to retrieve here is the fluid velocity, and that this fluid velocity also happens to be the material velocity, leading to $\mathbf{u}_f = \mathbf{u}_{\widehat{\mathbf{x}}}$. Importantly, because Dirichlet boundary conditions are usually imposed according to the absolute spatial position of the point considered, Dirichlet imposed field variables should not take the mesh velocity into account if the boundary is moving. For example, applying $\mathbf{u}_{\boldsymbol{\xi}}$ to a Dirichlet imposed $\mathbf{u}_f$ introduces spurious velocities in the domain. Similarly, applying $\mathbf{u}_{\boldsymbol{\xi}}$ to a Dirichlet imposed VOF transports it along with the mesh on that boundary. However, for the motion of the fluid–solid interface $\Gamma_{\mathrm{f} \cap \mathrm{s}}$ where $\mathbf{u}_f$ must be equal to the velocity of the boundary, the Dirichlet value of $\mathbf{u}_f = 0$ (for no-slip) is used while taking $\mathbf{u}_{\boldsymbol{\xi}}$ into account.

■ IV.3.2 Mesh Deformation Approaches

In this section, the two main approaches for mesh deformation that were implemented and used for this work are introduced: the model of linear elastostatics in section IV.3.2.1 and the model of monitor function in section IV.3.2.2.

IV.3.2.1 Linear Elastostatics

When the fluid–solid interface $\Gamma_{f\cap s}$ is an explicit, mesh-conforming boundary, the mesh nodes along the interface $\mathcal{N} \in \Gamma_{f\cap s}$ must move at the same velocity as the interface itself. Furthermore, the remaining mesh nodes that are within the fluid domain $\mathcal{N} \in \Omega_f \setminus \Gamma_{f\cap s}$ must also be displaced in order to avoid significant deterioration or tangling of the mesh. Such mesh deformation can be achieved by essentially turning each element $\mathcal{E}$ into an elastic, deformable material. This physical analogy answers to the equation of linear elastostatics, which is derived from section III.4.2.1 for the following steady state problem:

$$\nabla \cdot \bar{\bar{\sigma}}_{\mathcal{E}} + \mathbf{f}_{\mathcal{E}} = 0 \tag{IV.40}$$

with $\mathbf{f}_{\mathcal{E}}$ the body force and $\bar{\bar{\sigma}}_{\mathcal{E}}$ the stress tensor of the element expressed as $\bar{\bar{\sigma}}_{\mathcal{E}} = \lambda \,\mathrm{tr}\,(\bar{\bar{\epsilon}}_{\mathcal{E}})\,\bar{\bar{\mathbf{I}}} + 2\mu\bar{\bar{\epsilon}}_{\mathcal{E}}$ where $\bar{\bar{\epsilon}}_{\mathcal{E}}$ is the strain tensor:

$$\bar{\bar{\epsilon}}_{\mathcal{E}} = \frac{1}{2}\left(\nabla \mathbf{h} + \nabla \mathbf{h}^{\mathsf{T}}\right) \tag{IV.41}$$

with $\mathbf{h}$ the displacement vector, and μ and λ the Lamé parameters:

$$\lambda = \frac{\nu E_{\mathcal{E}}}{(1+\nu)(1-2\nu)} \quad ; \quad \mu = \frac{E_{\mathcal{E}}}{2(1+\nu)} \tag{IV.42}$$

with $E_{\mathcal{E}}$ the Young's modulus, and ν the Poisson's ratio. The Young's modulus can be scaled in a non-uniform manner so that elements have preferential stiffness according to predefined criteria. In the implementation used in this work, $E_{\mathcal{E}}$ is scaled according to the element size as:

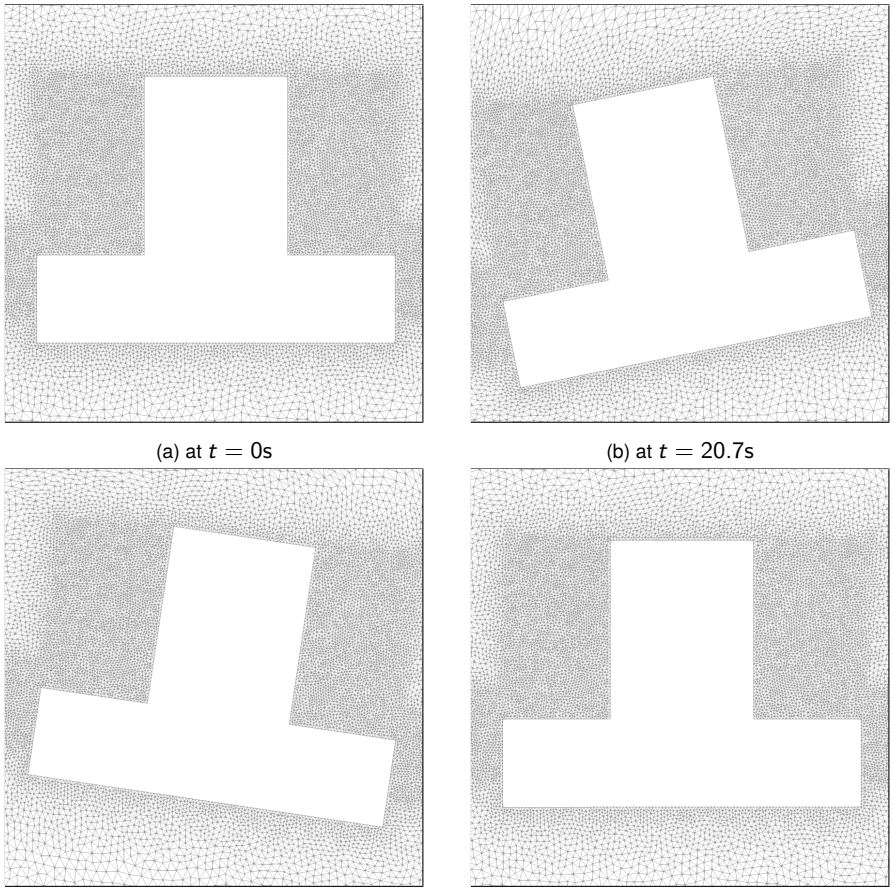
$$E_{\mathcal{E}} = \frac{E^u_{\mathcal{E}}}{\det J^{\mathcal{E}}} \tag{IV.43}$$

with $E^u_{\mathcal{E}}$ a user defined constant and $\det J^{\mathcal{E}}$ the determinant of the Jacobian of element $\mathcal{E}$, which is proportional to the volume of the element. Because only the element stiffness relative to other elements matters, an arbitrary constant $E^u_{\mathcal{E}} = 1$ is used here. The scaling of eq. (IV.43) stiffens the refined areas relative to the coarser areas of the mesh, thus preventing excessive deformation of refined areas of interest. According to [41], it is also possible to impose a non-zero and non-constant body force $\mathbf{f}_{\mathcal{E}}$ in order to impose spatially varying mesh element size, but this was not implemented here. Instead, the body force is set as $\mathbf{f}_{\mathcal{E}} = 0$, and the Poisson ratio as $\nu = 0.3$.

In order to solve eq. (IV.40), the known fluid–solid interface displacement is imposed as a Dirichlet boundary condition on $\Gamma_{f\cap s}$. In the case of rigid bodies, the displacement $\mathbf{h}$ of the nodes $\mathcal{N} \in \Gamma_{f\cap s}$ can be calculated with the following transformation:

$$\mathbf{h}^{(n+1)} = \underbrace{\mathbf{R}^{(n+1)}\mathbf{R}^{(n)\mathsf{T}} \cdot \left(\mathbf{x}^{(n)} - \mathbf{r}_p^{(n)}\right) - \left(\mathbf{x}^{(n)} - \mathbf{r}_p^{(n)}\right)}_{\text{rotation}} + \underbrace{\left(\mathbf{r}^{(n+1)} - \mathbf{r}^{(n)}\right)}_{\text{translation}} \qquad \forall \mathbf{x} \in \Gamma_{f\cap s} \tag{IV.44}$$

with $\mathbf{r}$ the barycentre of the rigid body, $\mathbf{r}_p$ the coordinates of the pivotal point (which is $\mathbf{r}_p = \mathbf{r}$ if the body is not constrained or colliding), and $\mathbf{R}$ the rotation matrix of the body. The translational displacement of the mesh nodes corresponds to the translation of the rigid body, while the rotational displacement depends current rotation of the body and on the distance of $\mathbf{x}^{(n)}$ from $\mathbf{r}_p^{(n)}$. On boundaries that remain fixed in space (e.g. a wave tank wall boundary Γ_{wall}), the mesh nodes can either be fixed with a Dirichlet condition of $\mathbf{h}^{(n+1)} = 0$, or allowed a sliding motion along the boundary with $\mathbf{h}^{(n+1)} \cdot \mathbf{n} = 0$.



(a) at $t = 0$s (b) at $t = 20.7$s

(c) at $t = 21.3$s (d) at $t = 30$s

Figure IV.6: Snapshots (close-up) of mesh deformation around moving mesh-conforming boundaries
Note: snapshots taken from simulation presented in section V.4.4

Snapshots of a mesh deformed according to the equations of linear elastostatics is shown in fig. IV.6, and corresponds to the simulation of a 2-DOFs floating body under extreme wave loads as shown in section V.4.4. It clearly appears from these snapshots that elements that are larger in volume deform more than finer elements, due to the stiffness scaling of eq. (IV.43).

IV.3.2.2 Monitor Function

In the case of implicit boundaries requiring local refinement, the volume of mesh elements can be controlled according to a target volume function, called a monitor function. This approach was introduced by [59, 60, 61], and is implemented here for long term goals of computational efficiency as it can be used for dynamic and preferential mesh refinement. The monitor function can be user-defined, automatically adapt to refine the target element volume around areas of interest, or be a combination of both. This approach can be particularly useful for refining around moving boundaries when using Immersed Boundary Method (IBM) or implicit free surface for MPF.

The monitor function $f = f(\mathbf{x}, t)$, imposes the variation of target element volume in space and time. The actual volume of an element $\mathcal{E}_i$ can be calculated as:

$$V(\mathcal{E}_i) = \int_{\mathcal{E}_i} 1 \mathrm{d}\mathbf{x} = \sum_{k=1}^{n_k} |\det J_k^{\mathcal{E}}| w_k \tag{IV.45}$$

with $J_k^{\mathcal{E}}$ the Jacobian at the quadrature point k, n_k the number of quadrature points, and w_k the weight of the quadrature point k. The volume can then be defined at a node $\mathcal{N}_i$ of the mesh by averaging the volume of all elements owning $\mathcal{N}_i$ as follows:

$$V\left(\mathcal{N}_i\right) = \frac{1}{n_{\mathcal{E}}} \sum_{j=1}^{n_{\mathcal{E}}} V\left(\mathcal{E}_j\right) \qquad \mathcal{E}_j \in \{\mathcal{E} : \mathcal{N}_i \in \mathcal{E}\} \tag{IV.46}$$

with $n_{\mathcal{E}}$ the number of elements around $\mathcal{N}_i$. From here, an element volume distribution function can be defined as a piecewise constant function $g_0^{\mathcal{E}}$ (constant per element) or as a smoother function $g_0^{\mathcal{N}}$ by interpolating the value recovered at the mesh nodes spanning $\{\mathcal{N} : \mathcal{N} \in \mathcal{E}_i\}$ of the element $\mathcal{E}_i$ as follows:

$$g_0^{\mathcal{E}}\left(\mathbf{x}\right) = V\left(\mathcal{E}_i\right) \quad \forall \mathbf{x} \in \mathcal{E}_i \tag{IV.47}$$

$$g_0^{\mathcal{N}}\left(\mathbf{x}\right) = \sum_{j=1}^{n_{\mathcal{N}\mathcal{E}}} V\left(\mathcal{N}_j^{\mathcal{E}}\right) \Phi_j(\xi) \quad \forall \mathbf{x} \in \mathcal{E}_i \tag{IV.48}$$

with $n_{\mathcal{N}\mathcal{E}}$ the number of nodes per element, $\mathcal{N}_j^{\mathcal{E}}$ the j-th node of $\mathcal{E}_i$, ξ the local coordinates of $\mathbf{x}$ in $\mathcal{E}_i$ and Φ_j the basis function of $\mathcal{N}_j^{\mathcal{E}}$. The aim of this moving mesh technique is to build a spatial mapping or transformation $\widetilde{\Phi}$ from coordinates $\mathbf{x}$ to new coordinates $\boldsymbol{\xi}_{\widetilde{\Phi}}$:

$$\begin{aligned} \widetilde{\Phi} : \Omega &\to \Omega_\xi \\ \mathbf{x} &\mapsto \widetilde{\Phi}\left(\mathbf{x}\right) = \boldsymbol{\xi}_{\widetilde{\Phi}} \end{aligned} \tag{IV.49}$$

where $\widetilde{\Phi}$ must satisfy the following relation between the volume function g_0 and monitor function f:

$$g_0(\mathbf{x}) |\det J_{\widetilde{\Phi}}| = f\left(\boldsymbol{\xi}_{\widetilde{\Phi}}\right) \tag{IV.50}$$

with $J_{\widetilde{\Phi}} = \frac{\partial \boldsymbol{\xi}_{\widetilde{\Phi}}}{\partial \mathbf{x}}$ the Jacobian matrix of the mapping $\widetilde{\Phi}$. The new coordinates $\boldsymbol{\xi}_{\widetilde{\Phi}}$ of a transformed point $\mathbf{x}$ are found with $\boldsymbol{\xi}_{\widetilde{\Phi}} = \widetilde{\Phi}(\mathbf{x})$. Following the method described in [59], the monitor function f and area function g_0 must first be scaled as a pre-step to fulfill the following condition:

$$\int_\Omega \frac{1}{f(\mathbf{x})} \mathrm{d}\mathbf{x} = \int_\Omega \frac{1}{g_0(\mathbf{x})} \mathrm{d}\mathbf{x} \tag{IV.51}$$

Note that if $g_0^{\mathcal{E}}$ is used for g_0, and because the value of $g_0^{\mathcal{E}}$ is equal to the volume of element $\mathcal{E}$ for any point in $\mathcal{E}$, the domain integral on the RHS of eq. (IV.51) is simply equal to the number of elements $n_{\mathcal{E}}$ in the mesh. The reciprocals of the monitor function and area function are denoted $\widetilde{f}$ and $\widetilde{g_0}$ respectively here, with the following scaling:

$$\widetilde{g_0} = \frac{1}{g_0} \quad ; \quad \widetilde{f} = \frac{1}{fC} \tag{IV.52}$$

with C the scaling coefficient equal to the following when using $g_0^{\mathcal{E}}$:

$$C = \frac{1}{n_{\mathcal{E}}} \int_\Omega \frac{1}{f\left(\mathbf{x}\right)} \mathrm{d}\mathbf{x} \tag{IV.53}$$

Once the scaled reciprocals are set, the following PDE must be solved to find the grid pseudo-velocity needed for displacing mesh nodes towards reaching the target element volumes:

$$\begin{cases} -\nabla \cdot \mathbf{v} = \widetilde{f} - \widetilde{g_0} & \forall \mathbf{x} \in \Omega \\ \mathbf{v} \cdot \mathbf{n} = 0 & \forall \mathbf{x} \in \partial\Omega \end{cases} \tag{IV.54}$$

with $\mathbf{v}$ the grid pseudo-velocity. The problem here is that Equation (IV.54) only provides conditions on the divergence of $\mathbf{v}$ within Ω, which is not sufficient to provide a unique solution. The grid pseudo-velocity $\mathbf{v}$ is therefore found by setting $\mathbf{v} \coloneqq \nabla w$ and solving the following pure Neumann problem to find a solution for the scalar field w:

$$\begin{cases} -\Delta w = \widetilde{f} - \widetilde{g_0} & \forall \mathbf{x} \in \Omega \\ \nabla w \cdot \mathbf{n} = 0 & \forall \mathbf{x} \in \partial\Omega \end{cases} \tag{IV.55}$$

The PDE above implicitly imposes the condition $\nabla \times \mathbf{v} = 0$, which in turn provides uniqueness of the solution for $\mathbf{v}$ up to an additive constant [62]. However, because of the constant null space of this pure Neumann problem, careful preconditioning is necessary by, for example, imposing a Dirichlet value of one DOF of the mesh in order to retrieve a unique global solution. Once the solution w has been found, the piecewise constant gradients of w are recovered at the nodes using a simple averaging of the gradients in the elements $\nabla w_{\mathcal{E}}$ around nodes:

$$\nabla w_{\mathcal{N}_i} = \frac{1}{n_{\mathcal{E}}} \sum_{j=1}^{n_{\mathcal{E}}} \nabla w_{\mathcal{E}_j} \qquad \mathcal{E}_j \in \{\mathcal{E} : \mathcal{N}_i \in \mathcal{E}\} \tag{IV.56}$$

or a weighted averaging:

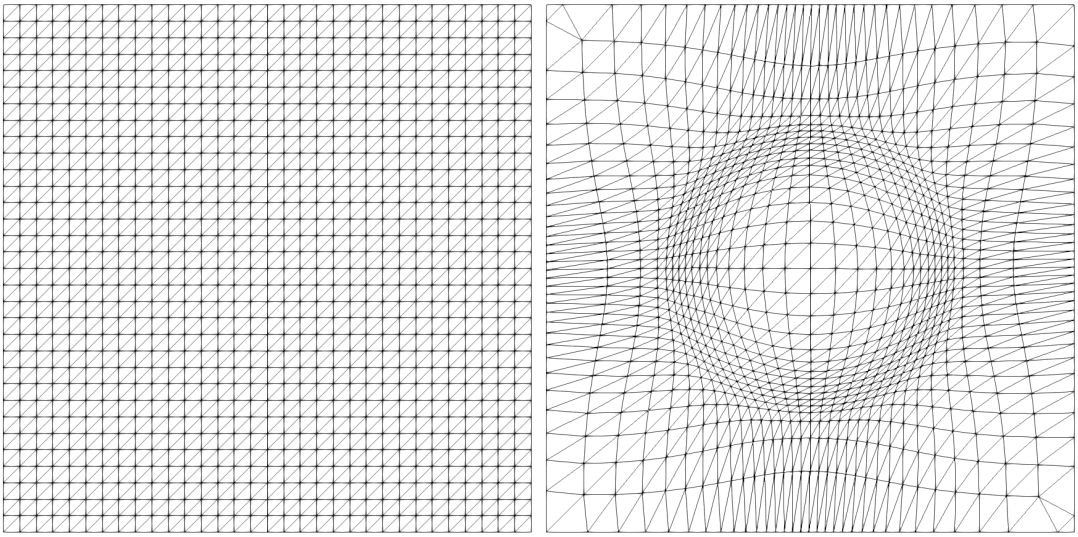
$$\nabla w_{\mathcal{N}_i} = \frac{1}{\sum_{j=1}^{n_{\mathcal{E}}} V_{\mathcal{E}_j}} \sum_{j}^{n_{\mathcal{E}}} V_{\mathcal{E}_j} \nabla w_{\mathcal{E}_j} \qquad \mathcal{E}_j \in \{\mathcal{E} : \mathcal{N}_i \in \mathcal{E}\} \tag{IV.57}$$

Other techniques can be used such as polynomial fitting, but for simplicity and efficiency, only the simple and weighted averaging techniques are used in this work, yielding satisfactory results when recovering variables.

Finally, an Initial Value Problem (IVP) is solved to displace the mesh nodes with pseudo-velocity $\mathbf{v} = \nabla w$ using a pseudo-time step Δt_s with $0 \leq t_s \leq 1$. For this purpose, we now look for a time-dependent mapping $\phi_s(\mathbf{x}, t_s) : \Omega \to \Omega(t_s)$ where the new coordinates of a point $\mathbf{x}$ are obtained at pseudo-time t_s with $\phi_s(\mathbf{x}, t_s)$ and setting the initial coordinates as $\phi_s(\mathbf{x}, 0) = \mathbf{x}$. The final coordinates at the end of the pseudo-time-stepping are then $\boldsymbol{\xi}_{\widetilde{\Phi}} = \Phi(\mathbf{x}) = \phi_s(\mathbf{x}, 1)$. The number of steps necessary to move the mesh nodes to their final position depends on the chosen value for Δt_s as $\phi_s^{i+1} = \phi_s^i + \Delta t_s \frac{\partial \phi_s}{\partial t_s}$ with $\frac{\partial \phi_s}{\partial t_s}$ the pseudo node velocity calculated as:

$$\frac{\partial \phi_s}{\partial t_s} = \frac{\nabla w(\phi_s)}{t_s \widetilde{f}(\phi_s) + (1 - t_s)\widetilde{g}(\phi_s)} \tag{IV.58}$$

where, for clarity, the dependency on $\mathbf{x}$ and t_s is dropped in the notation here with $\phi_s = \phi_s(\mathbf{x}, t_s)$. The initial step for calculating ϕ_s is done easily as the values of ∇w, $\widetilde{g_0}$, and $\widetilde{f}$ are already known at the nodes with $\phi_s(\mathbf{x}, 0) = \mathbf{x}$ (as ∇w was also recovered at the nodes) and can therefore be used directly in eq. (IV.58). For subsequent steps, if $\phi_s(\mathbf{x}, t_s) \neq \mathbf{x}$ for a given point, it becomes necessary to perform a search to find the element containing ϕ_s in order to retrieve the solution of ∇w and $\widetilde{g_0}$ on the background (i.e. original) mesh. This process can be very expensive as it must be done for all nodes that moved during the pseudo-time-stepping. The technique used for the localisation ϕ_s in this work is presented in another context in section IV.4.3 (applied to the localisation of mooring nodes within the fluid mesh) to which the reader can refer. Once the element $\mathcal{E}_{\phi_s}$ containing ϕ_s is found, the local coordinates ξ of ϕ_s within $\mathcal{E}_{\phi_s}$ are returned, and the values of ∇w and element volume at the displaced node within $\mathcal{E}_{\phi_s}$ are retrieved. Because the element volumes and ∇w are discontinuous scalar and vector fields respectively (on linear Lagrange element for the latter), a projection of the nodal recovered gradients and areas is used in order



(a) initial mesh ($t_s = 0$) (b) final mesh ($t_s = 1$)

Figure IV.7: snapshots of mesh transformation using target area monitoring function, with $n_{\mathcal{E}} = 2048$, $\epsilon = 0.025$, and $\delta t_s = 0.1$

to obtain a smooth solution for the displacement of the nodes. For the volumes, eq. (IV.48) is used, and the following is applied to ∇w in an element $\mathcal{E}_i$:

$$\nabla w = \sum_{i=1}^{n_{\mathcal{N}^{\mathcal{E}}}} \nabla w_{\mathcal{N}_i^{\mathcal{E}}} \Phi_i(\xi) \qquad \mathcal{N}_i^{\mathcal{E}} \in \{\mathcal{N} : \mathcal{N} \in \mathcal{E}_i\} \tag{IV.59}$$

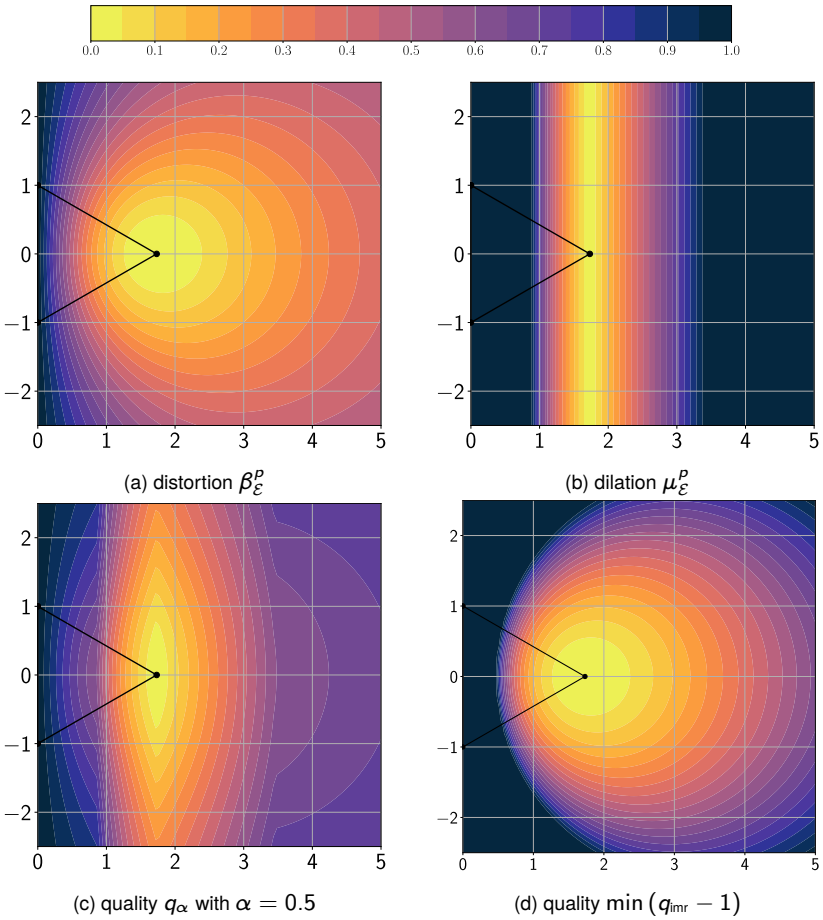
Note that using higher-order elements instead of linear Lagrange elements could lead to a continuous vector field for ∇w, but the volumes of the elements would remain a discontinuous vector field by definition. This process is repeated until $t_s = 1$, at which point the new coordinates of the mesh nodes are set with:

$$\mathbf{x}^{(n+1)} := \boldsymbol{\xi}_{\widetilde{\Phi}} = \Phi(\mathbf{x}^{(n)}) = \boldsymbol{\phi}_s(\mathbf{x}^{(n)}, 1) \tag{IV.60}$$

For illustrating the mesh motion model with monitor function implemented for this work, a small numerical simulation is presented here, based on the conceptual case from [61]. The numerical domain is a unit square $\Omega = [0, 1]^2$ spatially discretised with a triangular structured grid. A quadrilateral grid was used in [61] but, as both mesh are structured, the same number of nodes is used in both cases: $n_{\mathcal{N}} = 1089$. The number of elements for the triangular grid is however twice the number of elements of the quadrilateral grid, $n_{\mathcal{E}} = 2048$. The volume (or area) monitor function for the mesh is as follows:

$$f = \min\left(1, \max\left(\epsilon, \frac{\left\|0.25 - \sqrt{(x_1 - 0.5)^2 + (x_2 - 0.5)^2}\right\|}{0.25}\right)\right) \tag{IV.61}$$

with ϵ the area factor relative to the maximum element area. Because the triangular grid has twice the number of elements, $\epsilon = 0.025$ here in eq. (IV.61) (meaning that the minimum element area is 40 times smaller than the maximum area), instead of $\epsilon = 0.1$ for the quadrilateral grid case in [61]. As this is a steady problem with only pseudo-time-stepping, the simulation runs for only a single global time step, transforming the mesh from its initial position to its final position from $t_s^0 = 0$ to $t_s^{T_{\text{sim}}} = 1$ with $\Delta t_s = 0.1$. Figure IV.7 shows the initial configuration



(a) distortion $\beta_{\mathcal{E}}^{p}$

(b) dilation $\mu_{\mathcal{E}}^{p}$

(c) quality q_{α} with $\alpha = 0.5$

(d) quality $\min\left(q_{\text{imr}} - 1\right)$

Figure IV.8: Quality metrics of a triangular mesh element when node $\mathbf{x}_{\mathcal{N}} = (1, 0)$ is moved from its ideal position

of the grid at $t_s = 0$ and the final transformation at $t_s = 1$. The final configuration of the mesh nodes obtained here is virtually the same as the one obtained by [61], with a displacement to the same location at $t_s^{T_{\text{sim}}}$. This can be explained by the fact that only the relative element area matters for this simulation, and that by setting $\epsilon = 0.025$ here, the triangular mesh yields the same solution as the quadrilateral mesh (barring small discretisation errors). The pseudo-velocity solution is then used by the mesh nodes to move to their new position, with the nodes moving at the same rate regardless of the type of elements originally used because the same solution field is used.

■ IV.3.3 Mesh Quality and Smoothing

When meshes deform over time, their quality can deteriorate and must therefore be controlled. This can be a particularly important problem for relatively long simulations when nonlinear motions are introduced for moving the mesh, which is often the case for 6-DOFs mesh-conforming floating body. A way to overcome this problem is to separate the modes of motions in subparts of the mesh such as in [2]. This can be done by having part of the mesh deforming only with the rotational motion of the body (roll), another part deforming only according to one of the translational motion of the same body (e.g. heave), and another part deforming according to the other translational motion (e.g. sway). While this solution is easily applicable in 2D where there are only 3 DOFs, it becomes unviable in 3D where there are 6 DOFs. An alternative way to solve this problem is to introduce element quality metrics in

order to control the global quality of the mesh, and smooth it accordingly. As mentioned in section IV.3.2.1, the mesh deformation model using the equations of linear elastostatics stiffens elements according to their size, and while this theoretically deals with dilation of the element (change in volume), it does not deal with their distortion. Mesh quality metrics for a mesh element $\mathcal{E}$ are formulated in [16] as:

$$\beta_{\mathcal{E}} = \frac{\left(\frac{1}{n}\operatorname{tr}\left(J^{\mathcal{E}\mathsf{T}}J^{\mathcal{E}}\right)\right)^{\left(\frac{n_d}{2}\right)}}{\det J} \tag{IV.62}$$

$$\mu_{\mathcal{E}} = \frac{1}{2}\left(\frac{\det J_0^{\mathcal{E}}}{\det J^{\mathcal{E}}} + \frac{\det J^{\mathcal{E}}}{\det J_0^{\mathcal{E}}}\right) \tag{IV.63}$$

with $\beta_{\mathcal{E}}$ the distortion measure, $\mu_{\mathcal{E}}$ the dilation measure, $J^{\mathcal{E}}$ the Jacobian of the element, $J_0^{\mathcal{E}}$ the reference Jacobian (e.g. Jacobian of ideal shape or Jacobian at initial time step), and n_d the number of dimensions. Both quality metrics can instead be bounded from 0 (for a good quality element) to 1 (for a bad quality element) as:

$$\beta_{\mathcal{E}}^{p} = 1 - \beta_{\mathcal{E}}^{-1} \tag{IV.64}$$

$$\mu_{\mathcal{E}}^{p} = \max\left(\frac{\max\left(\det J^{\mathcal{E}}, \det J_0^{\mathcal{E}}\right)}{\min\left(\det J^{\mathcal{E}}, \det J_0^{\mathcal{E}}\right)}, 2\right) - 1. \tag{IV.65}$$

Note that in eq. (IV.65), $\mu_{\mathcal{E}}^{p}$ reaches a maximum value of 1 after a factor of relative dilation of 2, and that this could be adjusted as required. Those distortion and dilation values can then be used as a factor to stiffen the elements in an ALE framework using elastostatics. For this purpose, the metrics can be combined in a single weighted metric as follows:

$$q_{\alpha} = (1-\alpha)\beta_{\mathcal{E}}^{p} + \alpha\mu_{\mathcal{E}}^{p} \tag{IV.66}$$

with $0 \leq \alpha \leq 1$, q_{α} is also bounded from 0 (good quality) to 1 (worst quality). The Young's modulus of elements can then be expressed as:

$$E_{\mathcal{E}} = \frac{E_{\mathcal{E}}^{0}(q_{\alpha})^{a}}{\det J_0^{\mathcal{E}}} + \epsilon \tag{IV.67}$$

where a is a scaling factor for the quality metrics and ϵ a small constant acting as a tolerance (e.g. $\epsilon = 1e^{-6}$) to prevent potential values of $E^{\mathcal{E}} = 0$ in case of a non-deformed element ($q_{\alpha} = 0$). These metrics have been tested in this work to scale the stiffness of the mesh elements and, while it is preventing the quality of some elements to deteriorate over time when compared to the scaling of eq. (IV.43), it also prevents deteriorated elements from recovering a better quality. Other quality metrics can be used such as the Ideal Weight Inverse Mean Ratio (IMR) [93]:

$$q_{\text{IMR}} = \frac{\left\| J^{\mathcal{E}}(J_0^{\mathcal{E}})^{-1} \right\|_F^2}{2\left|\det\left(J^{\mathcal{E}}(J_0^{\mathcal{E}})^{-1}\right)\right|} \tag{IV.68}$$

with $\|\cdot\|_F$ the Frobenius norm. Using $\min\left(1, q_{\text{IMR}} - 1\right)$ for scaling the stiffness of mesh elements in this work yielded similar findings (i.e. it prevents elements from deteriorating but also prevents deteriorated elements from recovering). An illustration of the value of different metrics for a triangular element is shown in fig. IV.8 when one of its nodes is moved around its ideal position.

As the stiffness scaling described above for elastostatics mesh motion does not recover the quality of mesh elements, Laplacian smoothing can be used to improve the global quality of the mesh (or only a specific patch of

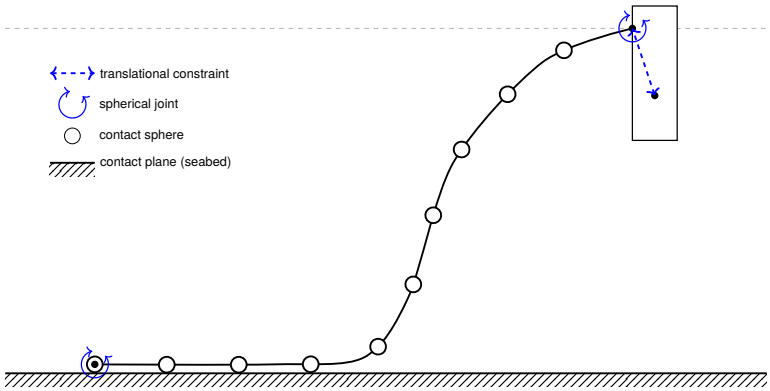


Figure IV.9: Schematic representation of constraints and contact entities between mooring, structures, and seabed

low-quality elements) by setting the new position of a node $\mathcal{N}_i$ as:

$$\mathbf{x}_{\mathcal{N}_i}^{\text{new}} := \frac{1}{n_{\mathcal{N}_j}} \sum_{j=1}^{n_{\mathcal{N}_j}} \mathbf{x}_{\mathcal{N}_j} \tag{IV.69}$$

with $\mathbf{x}_{\mathcal{N}_j}$ the coordinates of the j-th neighbouring node $\mathcal{N}_j$ of $\mathcal{N}_i$, and $n_{\mathcal{N}_j}$ the number of neighbouring nodes of $\mathcal{N}_i$. There are two ways to do Laplacian smoothing: simultaneously or sequentially. In the simultaneous version, all nodes $\mathbf{x}_{\mathcal{N}}^{new}$ are updated at once, keeping the original position of the nodes $\mathbf{x}_{\mathcal{N}}^{0}$ when solving eq. (IV.69). The sequential version is done by updating $\mathbf{x}_{\mathcal{N}_i} := \mathbf{x}_{\mathcal{N}_i}^{\text{new}}$ each time eq. (IV.69) is solved for a node, which means that the order in which the node positions are updated can affect the layout of the smoothed mesh. The simultaneous version usually provides better smoothing results than the sequential version [125], making it the preferred version in the implementation for this work.

IV.4 Moorings Coupling

In this section, the coupling strategy used for including mooring dynamics in the FSI framework is described. The mooring–structure coupling is first briefly described in section IV.4.1, followed by the fluid–mooring coupling with the application of hydrodynamic forces in section IV.4 and the retrieval of fluid variables from the fluid mesh for applying these forces accurately along the cable in section IV.4.3. Note that the coupling described here applies to the mooring dynamics model as described in the previous chapter, but not to the statics model.

IV.4.1 Mooring–Structure Coupling

The main principles described in this section for the mooring–structure coupling are schematically represented in fig. IV.9. Between the solid structure and its mooring system, a strong coupling is used where the structure and the cables are part of the same global system of equations, essentially making the fairlead a node of the Finite Element Method (FEM) system of equations for the mooring system. The position of the fairlead node is constrained to a given position relative to the position of the barycentre of the structure with a spherical mate constraint. This type of joint only constrains the position but allows any rotation of the line around the fairlead. Another spherical mate constraint is used for the anchor node which is attached to an arbitrary rigid body corresponding to the anchor that is fixed in space and time. Any movement of the anchor itself is prevented during the simulation. Other more

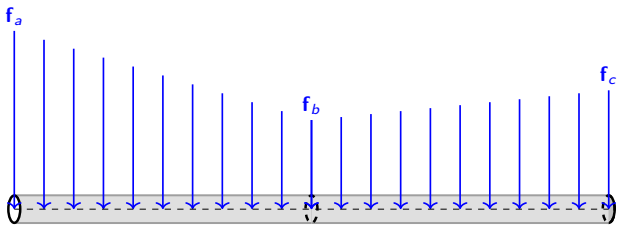


Figure IV.10: Triangular load applied on mooring cable elements

constraining joints can be also used according to requirements regarding the attachment of the fairlead or the anchor to the structure.

Collision detection is enabled on the mooring line in order to allow for contact between the cable and the seabed, as well as other obstacles. The seabed is simply represented by a solid plane (or box) with its own mechanical properties such as friction and stiffness. For the moorings, a cloud of collision-enabled spheres is created along the cable with a diameter matching the cable diameter or a chosen contact diameter. These spheres are placed at each node of the cable and can then enter in contact with other collision-enabled geometries such as the seabed and floating structures. This collision cloud cannot be used for cable-to-cable collision detection as only the nodes are able to collide with each other, while elements are free to go through each other undetected. Because no cable-to-cable collision is expected in the work presented here, using a cloud of spheres along the cable is deemed sufficient for collision detection purposes.

■ IV.4.2 Fluid–Mooring Coupling

The fluid–mooring coupling is a one-way coupling where the forces from the fluid are applied to the cable, but the effect of the cable on the fluid is ignored. The total external force $\mathbf{f}_e$ per unit length is applied along the cable as:

$$\mathbf{f}_e = \mathbf{f}_d + \mathbf{f}_m + \mathbf{f}_b + \mathbf{f}_c \tag{IV.70}$$

With $\mathbf{f}_d$ the drag force, $\mathbf{f}_m$ the inertia force, $\mathbf{f}_b$ the buoyancy force, and $\mathbf{f}_c$ the contact/collision force. The drag and inertia forces are both applied as normal and tangential components to the cable. In this section, the normal and tangential components of a vector are denoted with $_n$ and $_t$ subscripts respectively, and can be calculated as follows:

$$\mathbf{x}_t = \left(\mathbf{x} \cdot \hat{\mathbf{t}}\right) \hat{\mathbf{t}} \quad ; \quad \mathbf{x}_n = \mathbf{x} - \left(\mathbf{x} \cdot \hat{\mathbf{t}}\right) \hat{\mathbf{t}} \tag{IV.71}$$

with $\mathbf{x}$ any vector, and $\hat{\mathbf{t}}$ the direction vector:

$$\hat{\mathbf{t}} = \frac{\mathbf{x}}{\|\mathbf{x}\|} \tag{IV.72}$$

Assuming that the cable is a slender cylindrical structure, the drag force is calculated with Morison's equation as:

$$\mathbf{f}_d = \frac{1}{2}\rho_f \left(d_n C_{d,n} \|\mathbf{u}_{r,n}\| \mathbf{u}_{r,n} + d_t C_{d,t} \|\mathbf{u}_{r,t}\| \mathbf{u}_{r,t}\right) \tag{IV.73}$$

with $\mathbf{u}_r = \mathbf{u}_f - \dot{\mathbf{r}}$ the relative velocity between the fluid and the cable, ρ_f the density of the displaced fluid, $C_{d,n}$ and $C_{d,t}$ the normal and tangential drag coefficient, and d_n and d_t the normal and tangential drag diameters

respectively, with $d_t = \frac{d_n}{\pi}$ usually (as is the case in the current implementation). The inertia force is calculated with:

$$\mathbf{f}_m = \rho_f \frac{\pi d_{tn}^2}{4} \left(C_{m,n} \dot{\mathbf{u}}_{r,n} + C_{d,t} \dot{\mathbf{u}}_{r,t} + \dot{\mathbf{u}}_f \right) \tag{IV.74}$$

with $\dot{\mathbf{u}}_r = \dot{\mathbf{u}}_f - \ddot{\mathbf{r}}$ the relative acceleration between the fluid and the cable, d_{tn} the inertia diameter (usually the same for tangential and normal components), and $C_{m,n}$ and $C_{m,t}$ the normal and tangential added mass coefficients. Note that the inertia force contains the added mass forces from the relative accelerations as well as the Froude Krylov force from the absolute acceleration of the fluid. Numerically, both $\mathbf{f}_d$ and $\mathbf{f}_m$ are calculated at each node of the mooring cable and applied as triangular loads along the longitudinal axis of each element of the cable, as shown in fig. IV.10. Finally, the buoyancy force is calculated as follows:

$$\mathbf{f}_b = \frac{\rho_c - \rho_f}{\rho_c} \mathbf{g} A_0 \tag{IV.75}$$

with ρ_c the density of the cable, and A_0 the cross-sectional area of the unstretched cable. Note that $\mathbf{f}_b$ is in $\mathrm{N\ m^{-1}}$ in eq. (IV.75) but that it is actually applied numerically as a volumetric load on the cable. Finally, the contact force is handled by the MBD solver if there is contact between a cable node and a collision-enabled geometry such as the seabed.

As mentioned above, the fluid–mooring coupling scheme is one-way only: the effect of the fluid on the cable is taken into account through the equations stated above, while the effect of the cable on the fluid is ignored. The latter effect is assumed to be negligible as the non-longitudinal dimensions of mooring cables are significantly smaller than the other components that are included in the simulations such as floating structures, meaning that the effect of the moorings on the fluid is not expected to affect the response of structures in any significant way. Furthermore, a two-way coupling would require an extremely fine mesh around the cable for reliable results, leading to a greatly increased computational cost of CFD simulations. Using a conforming mesh on the moorings would also be a challenge due to the significant deformation of the cable over time. If including the effects on the fluid from the moorings is necessary, the IBM could be used as a more flexible alternative, but an appropriate refinement of the mesh around the cable would still be required.

The fluid velocity and acceleration used to calculate the drag and inertia forces described above are retrieved directly from the solution of the Navier–Stokes model at each node of the cable. As the mooring cable does not conform to the fluid mesh, a search algorithm must be used to find the fluid mesh elements containing each cable node. This operation is costly when the fluid mesh is large, with the cost varying according to the efficiency of the localisation technique used. The methods used to track the cable nodes within the fluid mesh implemented in this work is described in details in section IV.4.3. Note that in many cases, the mooring cables can extend beyond the numerical fluid domain Ω_f where no solution from the Navier–Stokes model is available. In this case, one of the following is performed to estimate the fluid velocity and acceleration: either the fluid is assumed to be at rest beyond Ω_f or, if waves are generated within Ω_f, the same function that is used for boundary conditions can be used to retrieve the wave kinematics solution at any node of the cable beyond Ω_f.

■ IV.4.3 Particle-Localisation Algorithm

Considering a particle (such as a mooring cable node) with known coordinates $\mathbf{r}_s$ within a mesh $\mathcal{M}$, this section aims to define an efficient way to find the mesh element $\mathcal{E}_{\text{min}}$ containing $\mathbf{r}_s$. From $\mathcal{E}_{\text{min}}$, the solution of any model using $\mathcal{M}$ can be retrieved at $\mathbf{r}_s$. Before proceeding to the description of the particle-localisation algorithms implemented for this work, a few concepts and definitions related to variables of the mesh are introduced. The mesh is hereby denoted $\mathcal{M}$ and is composed of elements $\mathcal{E}$, nodes $\mathcal{N}$, and element boundaries $\Gamma^{\mathcal{E}}$. The set of nodes owned by an element $\mathcal{E}_i$ is defined as:

$$\{\mathcal{N}_{\mathcal{E}}\}_i = \{\mathcal{N} : \mathcal{N} \in \mathcal{E}_i\} \tag{IV.76}$$

The set of boundaries owned by an element is defined as:

$$\left\{\Gamma^{\mathcal{E}}_{\mathcal{E}}\right\}_i = \left\{\Gamma^{\mathcal{E}} : \Gamma^{\mathcal{E}} \in \mathcal{E}_i\right\} \tag{IV.77}$$

The set of elements surrounding $\mathcal{N}_i$ (i.e. neighbouring elements of $\mathcal{N}_i$) is defined as:

$$\{\mathcal{E}_{\mathcal{N}}\}_i = \{\mathcal{E} : \mathcal{N}_i \in \mathcal{E}\} \tag{IV.78}$$

The set of nodes sharing the elements that own $\mathcal{N}_i$ (i.e. neighbouring nodes of $\mathcal{N}_i$) is defined as:

$$\{\mathcal{N}^{\star}\}_i = \{\mathcal{N} : \mathcal{N} \in \{\mathcal{E}_{\mathcal{N}}\}_i \wedge \mathcal{N} \neq \mathcal{N}_i\} \tag{IV.79}$$

The aim of the particle-localisation algorithms is to find the element $\mathcal{E}_{\text{min}}$ containing a given point $\mathbf{r}_s$. The most obvious way to find $\mathcal{E}_{\text{min}}$ is to use the brute force approach, i.e. iterate over all elements of $\mathcal{M}$ in order to find which element contains $\mathbf{r}_s$. To check if $\mathbf{r}_s$ is in an element, the following relation can be used for simplex elements (e.g. triangle and tetrahedron):

$$\mathbf{r}_s \in \mathcal{E} \iff \sum_{i=0}^{n_d} \xi_i < 1 \quad \wedge \quad \xi_i > 0 \quad \forall i \in \{0, \cdots, n_d\} \tag{IV.80}$$

with $\xi = (\xi_1, \cdots, \xi_{n_d})$ the local coordinates of $\mathbf{r}_s$ in $\mathcal{E}$. This is a costly operation when performed on all elements of the mesh until $\mathcal{E}_{\text{min}}$ is found. Instead, it is possible to first look for the closest node $\mathcal{N}_{\text{min}}$ to $\mathbf{r}_s$ and, once $\mathcal{N}_{\text{min}}$ is found, iterate over the elements spanning $\{\mathcal{E}_{\mathcal{N}}\}_{\text{min}}$ to check the condition eq. (IV.80). To look for $\mathcal{N}_{\text{min}}$ from an initial guess $\mathcal{N}^{(i)}_{\text{min}}$, the next closest node $\mathcal{N}^{(i+1)}_{\text{min}}$ is taken as the closest node to $\mathbf{r}_s$ within the neighbouring nodes of $\mathcal{N}^{(i)}_{\text{min}}$ (i.e. within $\{\mathcal{N}^{\star}\}$), and this operation is repeated until the actual closest node $\mathcal{N}_{\text{min}}$ is found. This principle is illustrated in algorithm 6. More iterations might be needed if no element directly around $\mathcal{N}_{\text{min}}$ contains $\mathbf{r}_s$, which can happen in the case of highly stretched elements. If no element containing $\mathbf{r}_s$ is found around $\mathcal{N}_{\text{min}}$, eq. (IV.80) is applied to each element surrounding the neighbouring nodes of $\mathcal{N}_{\text{min}}$ in $\{\mathcal{N}^{\star}\}_{\text{min}}$ and that are not in $\{\mathcal{E}_{\mathcal{N}}\}_{\text{min}}$.

A potentially more efficient technique to find $\mathcal{E}_{\text{min}}$ is to use the method first introduced by [67], which uses element boundary intersection to find $\mathbf{r}_s$. An initial guess for the element $\mathcal{E}^{(i)}_s$ and starting coordinates $\mathbf{r}^{(i)}_p$ within that element must first be chosen. This can be done by selecting an arbitrary element around $\mathcal{N}_{\text{min}}$, or by selecting the element around $\mathcal{N}_{\text{min}}$ with its centroid closest to $\mathbf{r}_s$ for a better initial guess. We then compute the direction of

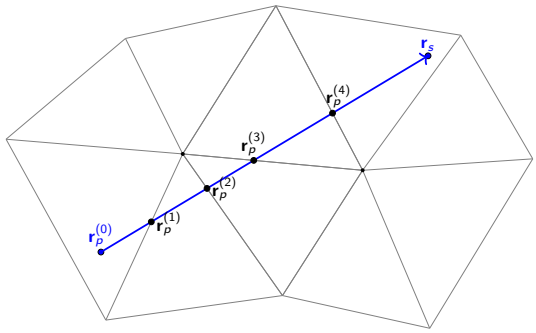


Figure IV.11: Mesh search for $\mathbf{r}_s$ by boundary intersection with initial guess $\mathbf{r}_p^{(0)}$.

the initial search distance d_s and search direction $\mathbf{t}_s$:

$$d_s^{(i)} = \left\| \mathbf{r}_s - \mathbf{r}_p^{(i)} \right\| \tag{IV.81}$$

$$\mathbf{t}_s = \frac{\mathbf{r}_s - \mathbf{r}_p}{d_s} \tag{IV.82}$$

Then, the search for $\mathcal{E}_{\text{min}}$ by element boundary intersection can be initiated. For each element boundary $\Gamma_i^{\mathcal{E}} \in \mathcal{E}_s^{(i)}$, the following condition is checked:

$$\left(\mathbf{r}_s - \mathbf{r}_p^{(i)} \right) \cdot \mathbf{n}_{\Gamma_i^{\mathcal{E}}} > 0 \tag{IV.83}$$

with $\mathbf{n}_{\Gamma_i^{\mathcal{E}}}$ the vector normal to $\Gamma_i^{\mathcal{E}}$ pointing outwards of the element. If the condition of eq. (IV.83) is fulfilled, it means that the line that contains $\mathbf{r}_s$ and $\mathbf{r}_p^{(i)}$ intersects the line (in 2D) or plane (in 3D) that contains $\Gamma_i^{\mathcal{E}}$. In this case, the distance of this intersection from $\mathbf{r}_p^{(i)}$ is calculated as:

$$\alpha_I = \frac{\left(\mathbf{x}_{\Gamma_i^{\mathcal{E}}} - \mathbf{r}_p^{(i)} \right) \cdot \mathbf{n}}{\mathbf{t}_s \cdot \mathbf{n}} \tag{IV.84}$$

with $\mathbf{x}_{\Gamma_i^{\mathcal{E}}}$ the centroid of the boundary $\Gamma_i^{\mathcal{E}}$. When several element boundaries of $\mathcal{E}_s^{(i)}$ fulfil eq. (IV.83), the boundary that yields the smallest positive α_I (denoted α_I^{min} and corresponding to the distance of the closest intersection) is selected as the intersecting boundary. The new search distance to $\mathbf{r}_s$ is then set as $d_s^{(i+1)} := d_s^{(i)} - \alpha_I^{\text{min}}$, and the new search coordinates become $\mathbf{r}_p^{(i+1)} := \mathbf{r}_p^{(i)} + \mathbf{t}_s \alpha_I^{\text{min}}$. As long as $d_s^{(i+1)} > 0$, the search for $\mathcal{E}_{\text{min}}$ is carried on from $\mathcal{E}_s^{(i+1)}$, which corresponds to the element sharing the selected intersecting boundary with $\mathcal{E}_s^{(i)}$. It is trivial to find $\mathcal{E}_s^{(i+1)}$ as only two elements can share the same boundary. If $d_s^{(i+1)} > 0$ but no element share the intersecting boundary, then $\mathbf{r}_s$ is considered outside of the mesh. If $d_s^{(i+1)} < 0$, then $\mathbf{r}_s$ must be within $\mathcal{E}_s^{(i)}$ and

Algorithm 6 Finding nearest mesh node $\mathcal{N}_{\text{min}}$ to given coordinates $\mathbf{r}_s$

```
function FINDNEARESTNODETOX(r_s, N_min)
    d = ||r_s − x_{N_min}||
    while False do
        N^0_min = N_min
        for N_j ∈ {N^⋆}_min do
            d = ||r_s − x_{N_j}||
            if d < d_min then
                d_min = d
                N_min = N_j
        if N_min = N^0_min then
            return N_min
```

Algorithm 7 Finding mesh element $\mathcal{E}_{\text{min}}$ containing coordinates $\mathbf{r}_s$

function FINDELEMENTCONTAININGX($\mathbf{r}_s$, $\mathcal{E}_{\text{min}}$)
  $\Gamma^{\mathcal{E}}_{\text{min}} = \text{None}$
  $\mathbf{r}_p = \mathbf{x}_{\mathcal{E}_{\text{min}}}$ ▷ barycentre of $\mathcal{E}_{\text{min}}$
  $d_s = \|\mathbf{r}_s - \mathbf{r}_p\|$
  $\mathbf{t}_s = \frac{\mathbf{r}_s - \mathbf{r}_p}{d}$
  while False **do**
    $\alpha_I^{\text{min}} = \infty$
    $\mathcal{E}^0_{\text{min}} := \mathcal{E}_{\text{min}}$
    for $\Gamma^{\mathcal{E}}_i \in \left\{\Gamma^{\mathcal{E}}_{\mathcal{E}}\right\}_{\text{min}}$ **do** ▷ list of boundaries of $\mathcal{E}_{\text{min}}$
      if $\mathbf{t}_s \cdot \mathbf{n}_{\Gamma^{\mathcal{E}}_i} > 0$ **then**
        $\alpha_I = \frac{(\mathbf{x}_{\Gamma^{\mathcal{E}}_i} - \mathbf{r}_s)\cdot \mathbf{n}_{\Gamma^{\mathcal{E}}_i}}{\mathbf{t}_s \cdot \mathbf{n}_{\Gamma^{\mathcal{E}}_i}}$ ▷ $\mathbf{x}_{\Gamma^{\mathcal{E}}_i}$ is the barycentre of $\Gamma^{\mathcal{E}}_i$
        if $0 < \alpha_I < \alpha_I^{\text{min}}$ **then**
          $\alpha_I^{\text{min}} := \alpha_I$
          $\Gamma^{\mathcal{E}}_{\text{min}} := \Gamma^{\mathcal{E}}_i$
    $d_s := d_s - \alpha_I^{\text{min}}$ ▷ Compute new distance
    if $d_s > 0$ **then**
      $\mathcal{E}_{\text{min}} := \mathcal{E} : \Gamma^{\mathcal{E}}_{\text{min}} \in \mathcal{E} \wedge \mathcal{E} \neq \mathcal{E}_{\text{min}}$ ▷ get next element sharing $\Gamma^{\mathcal{E}}_{\text{min}}$
      $\mathbf{r}_p := \mathbf{r}_p + \mathbf{t}_s \alpha_I^{\text{min}}$ ▷ Compute new coordinates to search from
    else ▷ found containing element
      return $\mathcal{E}_{\text{min}}$
    if $\mathcal{E}^0_{\text{min}} = \mathcal{E}_{\text{min}}$ **then**
      return None ▷ $\mathbf{r}_s$ is outside of the domain

this is verified with:

$$\mathbf{r}_s \in \mathcal{E}_s^{(i)} \iff \left(\mathbf{r}_s - \mathbf{x}_{\Gamma^{\mathcal{E}}_i}\right) \cdot \mathbf{n}_{\Gamma^{\mathcal{E}}_i} > 0 \quad \forall \Gamma^{\mathcal{E}}_i \in \mathcal{E}_s^{(i)} \tag{IV.85}$$

If eq. (IV.85) is fulfilled, then the element containing $\mathbf{r}_s$ is found: $\mathcal{E}_{\text{min}} = \mathcal{E}_s^{(i)}$. The whole algorithmic process implemented for this work to find the element $\mathcal{E}_{\text{min}}$ containing $\mathbf{r}_s$ is shown in algorithm 7, and a minimalist illustration is available in fig. IV.11 showing the iterating process for $\mathbf{r}_p^{(i)}$ until $\mathbf{r}_s$ is found.

In the case of mooring cables, once the search for all node coordinates $\mathbf{r}_s$ has been completed, the corresponding $\mathcal{E}_{\text{min}}$ and $\mathbf{r}_p$ are stored to be used as a starting point for search if the next global time step. Therefore, algorithm 6 is used only on the initial global step. Alternatively to using algorithm 6 on the initial step, an initial and arbitrary $\mathcal{E}^0$ can be chosen as a starting point on the initial time step, and algorithm 7 can be used directly from there. The interest in storing $\mathcal{E}_{\text{min}}$ and $\mathbf{r}_p$ is that the particles are expected to either still be within $\mathcal{E}_{\text{min}}$ on the next global step or to have moved to an element that is nearby, thus requiring minimal operations to recover the new $\mathcal{E}_{\text{min}}$. In summary, the technique used here for finding the fluid velocity at cable nodes coordinates within $\mathcal{M}$ is as follows: (1) at $t = 0$, find closest node to coordinates (see algorithm 6) and, from there, find closest element barycentre to $\mathbf{r}_s$ and select corresponding element as starting point; (2) find element $\mathcal{E}_{\text{min}}$ containing $\mathbf{r}_s$ through intersecting boundaries (see algorithm 7); (3) convert global coordinate $\mathbf{r}_s$ to local (element) coordinate ξ_s; (4) retrieve solution from model of interest (e.g. $\mathbf{u}_f$ from the NSEs model) in $\mathcal{E}_{\text{min}}$; (5) store $\mathcal{E}_{\text{min}}$ for next global time step.

In the case of parallel simulations, the algorithm must also include communication between processors. If the search initiated on a processor P^1 but the solution at $\mathbf{r}_s$ is found on processor P^3, this solution must be communicated back to the processor P^1 that initiated the search. Sending information back and forth between processors for finding $\mathbf{r}_s$ can be a relatively tedious process that is not described in details here for the sake of brevity but that was nonetheless implemented for the work presented here.

CHAPTER V

Numerical Simulations

Numerical modelling of relevant engineering applications is performed in this chapter in order to assess the viability of all aspects of the Fluid–Structure Interaction (FSI) framework for moored floating structure implemented for this work. A brief description of the setup for a typical numerical simulation is presented in section V.1 and subsequent sections focus on the thorough verification and validation of numerical models separately (uncoupled) and together (coupled). Simulations have been carefully selected to cover all aspects presented in previous chapters.

It is important to note that verification and validation are two distinct concepts, as Roache puts it succinctly [110]: verification is “solving the equations right”, while validation is “solving the right equations”. In the work presented here, assessing convergence of the solution of a model spatially or temporally is performing verification, and comparing numerical results to experimental data or to other numerical models is a validation case. Table V.1 can be used as a guide to the sections of this chapter, presenting the various simulations and indicating which aspects of the framework are included in each of these simulations for verification and validation.

Table V.1: Guide for numerical simulations of verification and validation cases showing enabled models

Aspect	Test case	Sec.	CFD	MBD	FSI	AME	WSI	MS	MD	PL
Two-Phase Flow Section V.2	Sloshing Wave	V.2.1	✓							
	Parallel Performance	V.2.2	✓							
Added Mass Section V.3	Added Mass Effect	V.3.1	✓	✓	✓					
	Added Mass Estimator	V.3.2	✓			✓				
	Stabilisation Error	V.3.3		✓						
Floating bodies Section V.4	Heaving cylinder	V.4.1	✓	✓	✓					
	Rolling caisson	V.4.2	✓	✓	✓					
	Wave-Induced Motion	V.4.3	✓	✓	✓		✓			
	Extreme Wave	V.4.4	✓	✓	✓	✓	✓			
Moorings Section V.5	Statics	V.5.1						✓		
	Static Sensitivity	V.5.2							✓	
	Dynamics damping	V.5.3		✓				✓	✓	
OC4-DeepCwind Section V.6	Uncoupled model	V.6.2		✓				✓	✓	
	Coupled model	V.6.3	✓	✓	✓	✓	✓	✓	✓	✓

CFD: Computational Fluid Dynamics, MBD: Multibody Dynamics, FSI: Fluid-Structure Interaction, AME: Added Mass Estimation, WSI: Wave-Structure Interaction, MS: Mooring Statics, MD: Mooring Dynamics, PL: Particle Localisation

V.1 Setup of a Typical Numerical Simulation . . . 87
V.1.1 Spatial Definition and Discretisation of the Numerical Domain . . . 87
V.1.2 Typical Initial Conditions for Two-Phase Flow Problem . . . 88
V.2 Two-Phase Flow . . . 90
V.2.1 Sloshing Wave . . . 90
V.2.2 Parallel Performance . . . 92
V.3 Stabilisation for Partitioned Fluid–Structure Coupling . . . 94
V.3.1 Numerical Assessment of the Added Mass Effect . . . 94
V.3.2 Added Mass Estimation for Arbitrarily Shaped Structures . . . 95
V.3.3 Non-Iterative Stabilisation Scheme . . . 97
V.4 Fluid–Structure Interaction (FSI) for Floating Bodies . . . 101
V.4.1 Free Decay (Heave) . . . 101
V.4.2 Free Decay (Roll) . . . 104
V.4.3 Response Under Regular Wave Loads . . . 106
V.4.4 Response Under Extreme (Focused) Wave Loads . . . 110
V.5 Moorings . . . 116
V.5.1 Statics Model Validation . . . 116
V.5.2 Static Convergence of Dynamic Model . . . 119
V.5.3 Mooring Dynamics Validation . . . 121
V.6 Moored Semi-Submersible Platform for Floating Wind Turbine . . . 126
V.6.1 Setup . . . 126
V.6.2 Uncoupled Results . . . 129
V.6.3 Fully Coupled Results . . . 131

V.1 Setup of a Typical Numerical Simulation

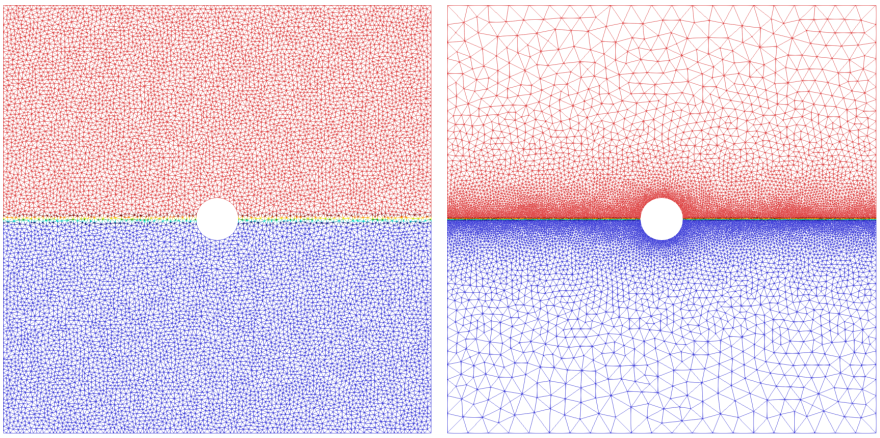
The main processes and steps for setting up a numerical simulation are briefly described in this section. In broad terms, it is necessary to perform the following before starting any simulation: (1) define the geometry of the fluid domain and the associated flags for boundaries and regions; (2) select models to be used in the simulation and the order in which they are called, as shown in fig. IV.1; (3) choose temporal discretisation scheme (backward Euler in our case); (4) choose spatial discretisation scheme (P1 Lagrange simplex elements in our case); (5) set initial conditions for each model; and (6) set boundary conditions for each model according to flagged geometrical entities of the domain. Once these preprocessing steps have been completed, the simulation can proceed to produce the requested data such as gauge data (e.g. integral gauge for Volume of Fluid (VOF), point gauge for pressure), the response of floating bodies, or mooring tensions, to name only a few of the possible data outputs that can be produced.

V.1.1 Spatial Definition and Discretisation of the Numerical Domain

The geometry of a numerical domain is represented with a set of vertices, segments, facets, volumes (in 3D), and regions. This geometrical information is required for the triangulation of the domain and for associating boundary conditions to different parts of the domain numerically. Vertices are a set of absolute coordinates $\mathbf{x}$ that represent the extremities of the geometries. Segments link vertices to each other, with each segment defined as the index of two vertices. Facets are defined with a set of segments that forms a closed loop. Finally, volumes are defined with a set of facets also forming a closed loop. Boundary flags are associated with each of these components (vertices, segments, and facets), which are then used to set the intended boundary conditions at the corresponding flags. The domain can also be divided into different regions, corresponding to facets in 2D or volumes in 3D. Regions are flagged so that elements within them can be manipulated if necessary. For example, the imposition of a source term in the Navier–Stokes Equations (NSEs) equations for relaxation zones is done by looping over all elements of the domain and imposing the source term only on the quadrature of elements that bear a region flag associated with an absorption or generation zone.

Once the geometry has been defined, the domain can be triangulated. All meshes used for the simulations presented in this chapter are always entirely composed of simplexes: triangles in 2D and tetrahedra in 3D. To generate uniformly refined meshes, the Triangle library [114] is used for 2D meshes and the Tetgen library [115] for 3D meshes. When spatial control over the refinement of the mesh is desired, the mesh is generated with the Gmsh library [56] that provides a set of convenient tools for gradual refinements. Because of the significant computational cost savings that it induces, most simulations presented here use a gradually refined mesh. Unless stated otherwise, the gradual coarsening of the mesh from a minimum characteristic element size h_{e0} is usually no more than 10% from an adjacent element. For illustration purposes, an example comparing a constantly refined mesh to a gradually refined mesh with h_{e0} around the free surface and the boundaries of a circular structure is shown in fig. V.1. In this particular case, both meshes use an equivalent number of elements $n_{\mathcal{E}} \approx 11 \times 10^3$ but the minimum characteristic element size is four times smaller around the areas of interest for the gradually refined mesh when compared to the constantly refined mesh. To have the same level of spatial discretisation around the

areas of interest with a constantly refined mesh, the number of elements would have to be more than fifteen times greater than the mesh shown in fig. V.1, leading to a total element count of $n_{\mathcal{E}} \approx 175 \times 10^3$. It is clear that when a specific mesh refinement is required for the description of the fluid around an area of interest and that such refinement becomes less primordial the further away from the area of interest, gradual mesh refinement allows for very significant reduction of the mesh size. Note also that finer elements lead to greater simulation time in unsteady simulations when the time step is controlled by the CFL condition.



(a) Mesh with constant refinement

(b) Mesh with gradual refinement around free surface and structure boundaries

Figure V.1: Comparison of constantly and gradually refined meshes with equivalent number of elements $N_{\mathcal{E}} \approx 11 \times 10^3$. For illustration purposes: air phase in red, water phase in blue, with a floating cylinder in the centre of the domain.

■ V.1.2 Typical Initial Conditions for Two-Phase Flow Problem

Before starting a simulation, initial conditions must be set within the fluid domain to define the different initial characteristics of the fluid phases. As all test cases presented here start with the fluid initially at rest, the usual initial conditions of Two-Phase Flow (TPF) simulations for a still water level will be described here. Note that it is relatively straightforward to extend the initial conditions to cases where the free surface is initially perturbed with, for example, a regular wave. If both fluid phases are at rest, then the initial velocity $\mathbf{u}_f^{(0)} = \mathbf{u}_f^{(0)}(\mathbf{x})$ is defined as:

$$\mathbf{u}_f^{(0)}(\mathbf{x}) = 0 \qquad \forall \mathbf{x} \in \Omega_f \tag{V.1}$$

As the fluid is at equilibrium and only subject to gravitational forces, the free surface separating the water phase Ω_w and air phase Ω_a is perpendicular to $\mathbf{g}$ and is placed at a height known as the mean water level h_{mwl} and defined by the user. Setting the gravitational acceleration vector parallel to the Z-axis (see table V.2 for its typical value), the initial free surface of the fluid at rest is perpendicular to the Z-axis, and the initial signed distance function from the free surface $\phi_{\text{sdf}}^{(0)} = \phi_{\text{sdf}}^{(0)}(\mathbf{x})$ can simply be defined as:

$$\phi_{\text{sdf}}^{(0)}(\mathbf{x}) = x_3 - h_{\text{mwl}} \qquad \forall \mathbf{x} \in \Omega_f \tag{V.2}$$

Equation (V.2) can be used directly as initial conditions for the Level Set (LS) model with $\phi_{\text{sdf}}^{(0)} < 0$ in Ω_w and $\phi_{\text{sdf}} > 0$ in Ω_a. The signed distance function is then used to calculate the initial conditions of the VOF model with

a smoothed Heaviside function as:

$$\widehat{\theta}^{(0)} = \theta_\epsilon\left(\phi^{(0)}\right) \qquad \forall \mathbf{x} \in \Omega_f \tag{V.3}$$

leading to a VOF value of $\widehat{\theta}_a = 1$ in the air phase and $\widehat{\theta}_w = 0$ in the water phase, and $0 < \widehat{\theta} < 1$ around the free surface up to a distance of $\pm 1.5 h_{e0}$ (see section III.1.2 for θ_ϵ). Still using $\phi_{\text{sdf}}^{(0)}$, the initial pressure field $\mathrm{p}^{(0)} = \mathrm{p}^{(0)}(\mathbf{x})$ in the fluid domain can be calculated as:

$$\mathrm{p}^{(0)}(\mathbf{x}) = \mathrm{p}_{\Gamma_{\text{atm}}}^{(0)} - g_3 \int_{\phi_{\text{sdf}}^0(\mathbf{x})}^{\phi_{\text{sdf}}^{\Gamma_{\text{atm}}}} (\rho_w + (\rho_w - \rho_a)\theta_\epsilon(\phi_{\text{sdf}}))\, \mathrm{d}\phi_{\text{sdf}} \qquad \forall \mathbf{x} \in \Omega_f \tag{V.4}$$

with $\phi_{\text{sdf}}^{\Gamma_{\text{atm}}}$ the signed distance from the mean water level to the top boundary of the tank (constant), ρ_w the water density, and ρ_a the air density. As long as Γ_{atm} is perpendicular to the gravitational acceleration vector, the pressure along Γ_{atm} is set as a constant, usually $\mathrm{p}_{\Gamma_{\text{atm}}}^{(0)} = 0$. Any constant value for $\mathrm{p}_{\Gamma_{\text{atm}}}^{(0)}$ could be used, as only the relative pressure in the fluid domain set by eq. (V.4) matters as an initial condition. Note that if the fluid is not at rest but rather perturbed initially by a phenomenon such as a wave, the TPF model can use the solution provided by wave theory for the wave elevation, velocity, and pressure.

When structures are within the fluid domain, the signed distance function defined in eq. (V.2) does not give the true distance between $\mathbf{x}$ and the free surface if the structure is placed between $\mathbf{x}$ and the free surface. This means that $\phi_{\text{sdf}}^{(0)}$ is not exact anymore, and the same can be said for any initial conditions derived from $\phi_{\text{sdf}}^{(0)}$ such as eq. (V.4). However, only a good approximation is needed for the initial conditions to allow for the models to recover the actual solution on the initial time step. In our implementation, setting $\phi_{\text{sdf}}^{(0)}$ to eq. (V.2) as an initial guess is sufficient for all TPF simulations presented here, regardless of the inclusion of solid structures within the fluid domain. On top of the above initial conditions, mesh conforming and moving structures require to set initial conditions for the elastic Arbitrary Lagrangian–Eulerian (ALE) mesh model, with the displacement $\mathbf{h}^{(0)} = \mathbf{h}^{(0)}(\mathbf{x})$ as:

$$\mathbf{h}^{(0)}(\mathbf{x}) = 0 \qquad \forall \mathbf{x} \in \Omega_f \tag{V.5}$$

which applies on all mesh nodes (no displacement).

Unless specified otherwise, all simulations presented here use the physical constants presented in table V.2, and time stepping is done with backward Euler for all models (TPF and Multibody Dynamics (MBD)). Also note that, throughout this chapter and unless stated otherwise, the dimensionless error ϵ of a quantity of interest f is expressed relative to a reference quantity f_0 (e.g. experimental value for validation or most refined numerical value for verification) as $\epsilon = \left|\frac{f - f_0}{f_0}\right|$.

Table V.2: Physical constants used in numerical simulations (unless specified otherwise)

Symbol	Value	Unit	Description
ρ_w	998.2	$\mathrm{kg\,m^{-3}}$	water density (at 20°C)
ν_w	1.004×10^{-6}	$\mathrm{m^2\,s^{-1}}$	water kinematic viscosity (at 20°C)
ρ	1.205	$\mathrm{kg\,m^{-3}}$	air density (at 20°C)
ν_a	1.5×10^{-5}	$\mathrm{m^2\,s^{-1}}$	air kinematic viscosity (at 20°C)
$\mathbf{g}$	$\begin{pmatrix}0 & 0 & -9.81\end{pmatrix}^\mathsf{T}$	$\mathrm{m\,s^{-2}}$	gravitational acceleration

V.2 Two-Phase Flow

V.2.1 Sloshing Wave

V.2.1.1 Setup

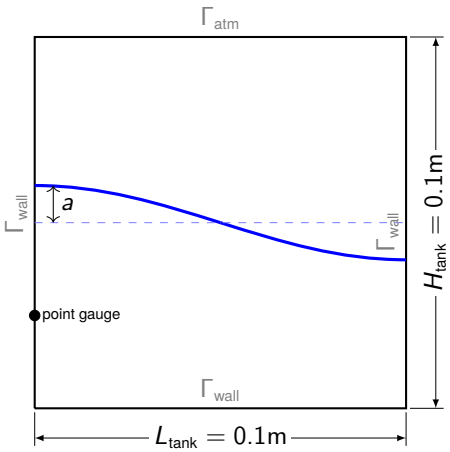
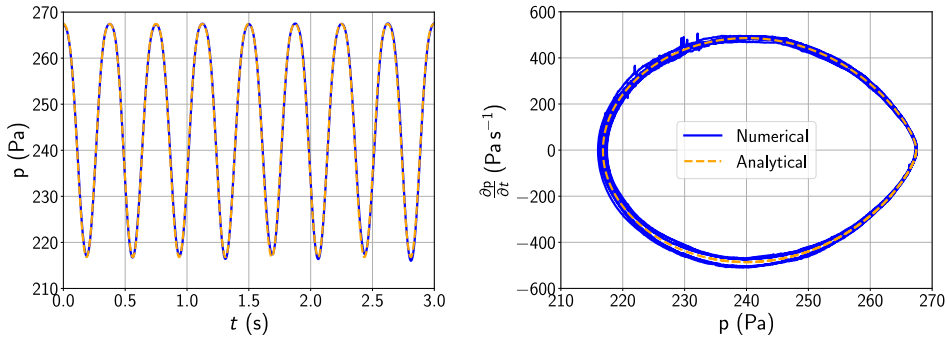


Figure V.2: Schematic representation of domain for sloshing body of water in a fixed tank
dashed blue line: mean water level; solid blue line: initial conditions of the free surface

The TPF model is validated here by simulating the sloshing motion of a body of water in an enclosed numerical tank. This case is traditionally used as a benchmark to validate Computational Fluid Dynamics (CFD) models involving surface flow and tracking of the free surface. The simulation presented here assumes an inviscid fluid and uses the same setup as found in other studies such as in [124, 109]. This means that no dissipation due to viscous effect should be observed in this simulation and any energy dissipation should solely be due to numerical errors, particularly numerical diffusion.

The fluid domain consists of a 2D tank forming a square with sides of length $L_{tank} = H_{tank} = 0.1\text{m}$, as shown schematically in fig. V.2. The tank is filled with water that forms a column of depth $h_{mwl} = 0.5L_{tank} = 0.05\text{m}$. The rest of the tank is filled with air and, as this is a conceptual case for inviscid fluids, the viscosity of both fluids is set to zero ($\nu_w = \nu_a = 0$). As initial conditions, the free surface forms a sinusoidal wave of amplitude $a = 0.005\text{m}$ and wavelength of $\lambda = 0.2\text{m}$ with its crest initially on the left boundary of the tank. The initial velocity and pressure fields of the water phase are derived with the third-order analytical solution for a sloshing wave of the first mode in a fixed tank, as originally derived by [119] who also provides the solutions for other modes. Free-slip boundary conditions are imposed on the bottom, left, and right boundaries Γ_{wall} of the tank, and atmospheric conditions are applied on the top boundary Γ_{atm} of the tank. A numerical point gauge is placed halfway within the water column on the left boundary to record the pressure over time.

The numerical domain is then discretised with an unstructured triangular mesh of constant characteristic element size $h_e = \frac{a}{10} = 5 \times 10^{-4}\text{m}$, and this setup is used for verification of the temporal convergence of the TPF model by running several simulations using different levels of temporal discretisation and for a total time $T_{sim} = 3\text{s}$.



(a) Pressure magnitude over time

(b) Time derivative of pressure as a function of pressure magnitude

Figure V.3: Numerical results and analytical solution for the pressure on the left boundary at $x_3 = 0.5h_{\text{mwl}}$ with $\Delta t = 1.25^{-4}$second

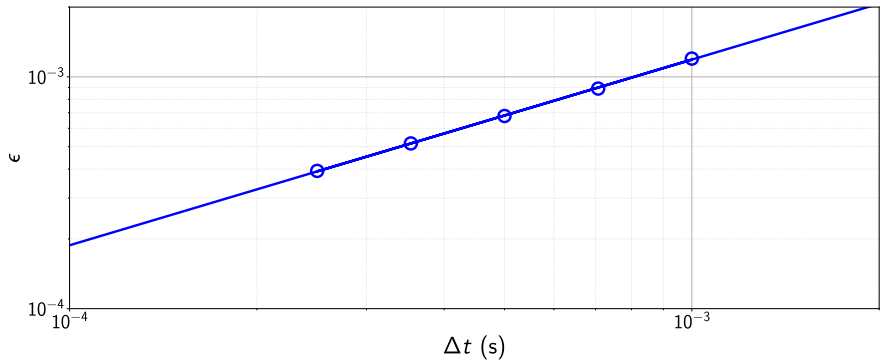


Figure V.4: Temporal convergence of the solution for the pressure from a sloshing wave on left boundary with constant refinement $h_{e0} = \frac{a}{10}$

V.2.1.2 Results

The time-series of the numerical pressure from the most refined numerical case ($\Delta t = 1.25 \times 10^{-4}$s) is shown in fig. V.3a along with the analytical solution. It appears clearly from the results of this highly refined case that the numerical signal of the pressure is in good agreement with the analytical signal over time . It is worth noting that there is a slight periodical divergence between the analytical and numerical solution when the free surface gets away from the initial condition (i.e. when the trough of the wave is on the left boundary), as can be seen in fig. V.3b. This minimal divergence oscillates between an overestimation of the magnitude of the pressure for odd-numbered periods and an underestimation for the even-numbered period. Despite this, the numerical solution for the pressure converges back to the value given as its initial condition after completing a period of oscillation. While the divergence from the analytical solution is minimal, it also appears clearly that the initial conditions are recovered accurately after each period of oscillation.

For each case with different Δt, the error in Pearson Correlation Coefficient (PCC) (denoted ϵ here) is used between the numerical and analytical solutions on the first five periods of oscillation. As shown in section V.2.1.2, there is clear convergence of the solution as the temporal discretisation is made finer, with a convergence order of 0.8. Overall, it is clear that there is good agreement between the numerical and analytical solution for this TPF problem, with numerical convergence to a unique solution as the simulation is refined.

V.2.2 Parallel Performance

In this thesis, all simulations that involve fluid dynamics are relatively large in terms of computational power needed and are therefore required run on several cores (i.e. several processors). This is known as distributed computing, and its scalability and efficiency must be assessed carefully in order to find bottlenecks and limits of the parallelisation of the code. The parallel performance of the TPF solver using different numbers of cores for a given mesh is therefore shown here. Note that the MBD solver used in this work can only run in serial and that its calculation times are usually minimal compared to the TPF solver. The parallel performance analysis is therefore carried out here with with no FSI involved and no MBD solver.

V.2.2.1 Setup

The sloshing wave case presented in section V.2.1 is used as a benchmark here for parallel performance of the TPF solver. The simulation is repeated with different numbers of cores n_c, allocating part of the fluid domain to each core equally with approximately the same number of mesh nodes per core. The simulation time is fixed to $T_{\text{sim}} = 1\text{s}$ with $\Delta t = 1 \times 10^{-3}\text{s}$, and the numerical domain is discretised uniformly with irregular triangular elements for three levels of refinement: a mesh $\mathcal{M}_1$ with $h_{e0} = 1 \times 10^{-3}\text{m}$ leading to $n_{\mathcal{E},1} = 32{,}404$ elements and $n_{\mathcal{N},1} = 16{,}201$ nodes, a mesh $\mathcal{M}_2$ with $h_{e0} = 5 \times 10^{-4}\text{m}$ leading to $n_{\mathcal{E},2} = 131{,}796$ elements and $n_{\mathcal{N},2} = 66{,}411$ nodes, and a mesh $\mathcal{M}_3$ with $h_{e0} = 2.5 \times 10^{-4}\text{m}$ leading to $n_{\mathcal{E},3} = 528{,}281$ and $n_{\mathcal{N},3} = 265{,}165$.

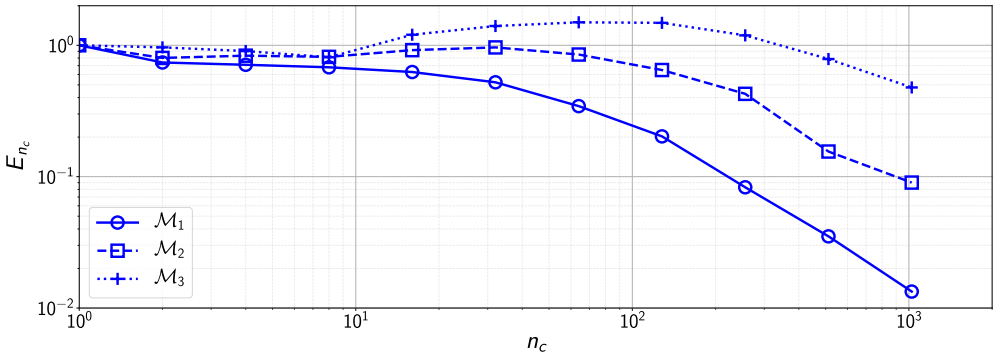
V.2.2.2 Results

The speed-up S_{n_c} and efficiency E_{n_c} for a parallel case with n_c cores are calculated as follows:

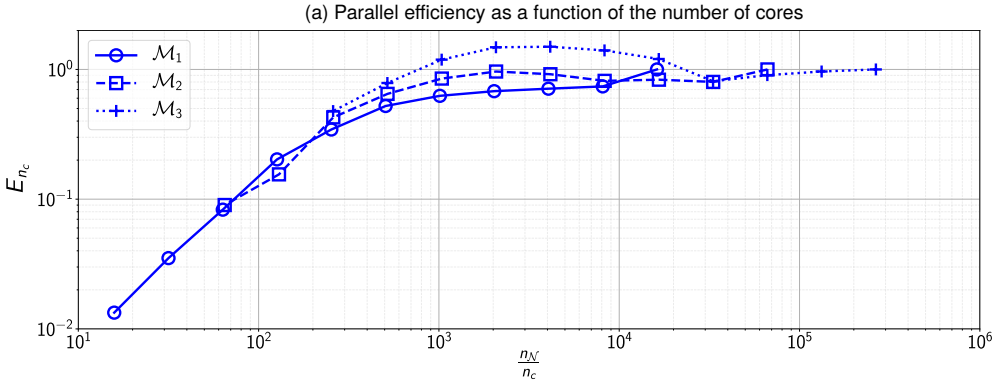
$$S_{n_c} = \frac{T_1}{T_{n_c}} \quad , \quad E_{n_c} = \frac{S_{n_c}}{n_c} \tag{V.6}$$

with T_1 the wall clock time needed to complete the simulation in series, and T_{n_c} the wall clock time needed to complete the simulation in parallel with n_c cores.

In serial, one time-step takes 9.44s (wall clock time) on average to be completed for $\mathcal{M}_1$, 75.02s for $\mathcal{M}_2$, and 684.73s for $\mathcal{M}_3$. The results for the parallel efficiency of the three different meshes are plotted in fig. V.5, as a function of the number of cores in fig. V.5a and as a function of the number of mesh nodes per core fig. V.5b. If parallelisation was ideal, all curves in fig. V.5 would be constant with $E_{n_c} = 1$. This is not possible in practice due to several factors affecting the overall parallel efficiency, such as time needed for communication between processors and parts of the code that are not parallelised and therefore running on all processors at once without taking advantage of distributing the problem. It appears clearly on fig. V.5a that, globally, efficiency decreases as the number of core increases and that, consequently, efficiency decreases when the number of mesh nodes per core decreases as shown in fig. V.5b. This can be mainly attributed to communication between cores becoming more prominent as more cores are used for the simulation. This drop of efficiency becomes steeper beyond a certain number of cores which depends on the original mesh size with, for the same number of cores, the parallelisation $\mathcal{M}_1$ being less efficient than $\mathcal{M}_2$, and the parallelisation of $\mathcal{M}_2$ being less efficient than $\mathcal{M}_3$. Looking at fig. V.5b, it becomes clear that a similar trend can be observed on the three meshes beyond a certain number of mesh nodes per core. For all meshes, parallelisation becomes significantly less efficient when there are less than a thousand



(a) Parallel efficiency as a function of the number of cores



(b) Parallel efficiency as a function of the number of mesh nodes per core

Figure V.5: Parallel efficiency of the TPF model on three mesh refinements

mesh nodes per core. Interestingly, for the case with the largest mesh ($\mathcal{M}_3$) and when $10 < n_c < 300$, super-linear speed-up is observed, i.e. $E_{n_c} > 1$. This phenomenon can happen depending on the configuration of the machine used and its parameters such as cache per processor, as well as the scale of the problem at hand.

Taking into account this parallel performance analysis, all simulations that use the TPF solver are aimed at having more than or around $\frac{n_\mathcal{N}}{n_c} = 1000$ nodes per core when parallel resources allow. As seen above, the overall performance risks suffering due to inter-processor communication when there are fewer nodes per core. Note that the framework has been built and tested successfully in parallel on various machines with different configurations and characteristics as listed in table V.3. This parallel analysis and all parallel simulations performed for the studies presented in this thesis used the Copper High Performance Computing (HPC). All simulations performed in series (e.g. pure MBD or mooring problem) used the laptop.

Table V.3: Characteristics of different systems used for parallel simulations

name	system type	processor type	nodes	cores/node	core speed	memory/node
		—	1	1	GHz	GB
S1	Laptop	Intel Core i7-7700HQ	1	8	2.8	16
S2	Copper HPC	AMD Interlagos Opteron	460	32	2.3	64
S3	Hydra HPC	2x Intel Xeon E5-2650 v2	6	16	2.6	64
S4	Hydra HPC	2x Intel Xeon E5-2690 v4	8	28	2.6	64
S5	Hydra HPC	2x Intel Xeon Gold 6154	8	36	3.0	64
S6	Archer HPC	2x Intel Xeon E5-2697 v2	4920	24	2.7	64

V.3 Stabilisation for Partitioned Fluid–Structure Coupling

The concept of added mass effect and the capabilities of the stabilisation scheme for explicit coupling are introduced here. This scheme is used in the coupled fluid–structure cases of the next section, and must therefore be verified to be reliably introduced in FSI simulations. This section is divided explores:

- the main parameters affecting the added mass effect section V.3.1,
- the convergence of the added mass estimator model in section V.3.2,
- the effect of the non-iterative added mass stabilisation scheme on the response of a body in section V.3.3.

V.3.1 Numerical Assessment of the Added Mass Effect

V.3.1.1 Setup

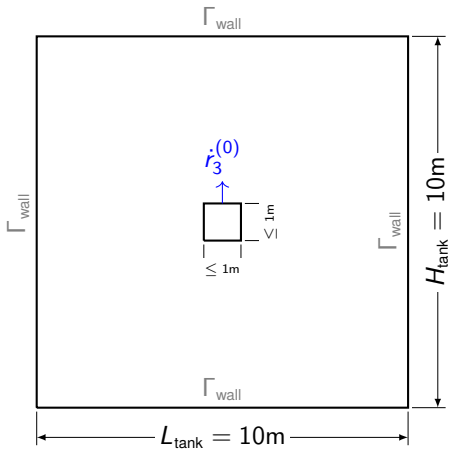


Figure V.6: Schematic representation of numerical domain for assessment of the added mass effect

In this set of simulations, the different properties that affect the added mass effect are investigated. As mentioned in section II.4, added mass instabilities can occur and grow depending on the geometry of the solid and the time discretisation. For this purpose, a rigid body of density $\rho_s = 1000\text{kg m}^{-3}$ is fully submerged into a single-phase fluid of density $\rho_w = 1000\text{kg m}^{-3}$. As $\rho_s = \rho_w$ and the coupling scheme used here is explicit (Conventional Serial Staggered (CSS) scheme), the simulation is theoretically at the edge of stability, as explained in section IV.2.3.1. The numerical setup of these 2D Single-Phase Flow (SPF) simulations consists of a rectangular solid of varying dimensions (with a maximum height and maximum width of $L_{max} = 1\text{m}$) and fully submerged in the centre of a tank of dimensions $L_{tank} = H_{tank} = 10L_{max}$. A schematic representation of the numerical domain is shown in fig. V.6.

These dimensions are chosen in order to have minimal effect on the value of the added mass of the solid from the walls of the tank which, ideally, should be considered infinitely far away from the solid boundaries. As the density of the fluid and the solid are the same, there is no movement and the simulation is at equilibrium when the fluid and solid are both initialised at rest. Therefore, a small disturbance is necessary to trigger the added mass effect, if it is present at all. For this reason, an initial velocity $\dot{r}_3^{(0)} = 0.001\text{m s}^{-1}$ is given to the rigid body. As we are interested in the effect of the height to width ratio of the structure, it is only allowed to heave freely and is fully constrained on

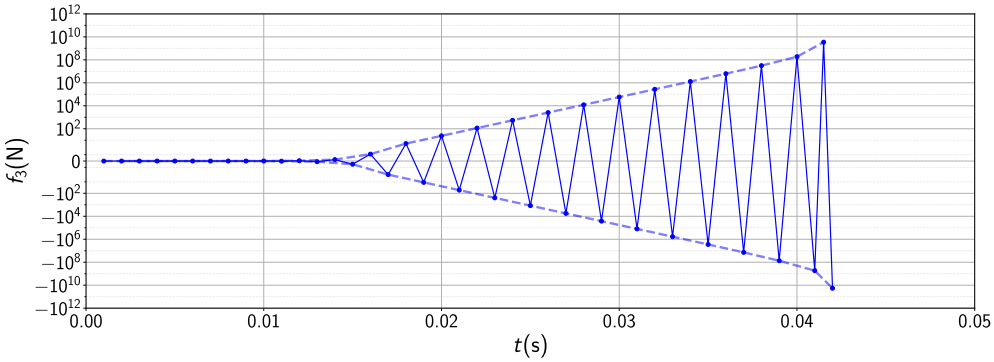


Figure V.7: Vertical force acting on rectangular body of dimensions $1\text{m} \times 0.5\text{m}$ subject to added mass effect with $\rho_s = \rho_f$ and $\Delta t = 0.001\text{s}$

all other Degrees of Freedom (DOFs). All simulations investigating the effect of the geometry of the rigid body on the added mass effect run with a fixed time step of $\Delta t = 0.001\text{s}$, while simulations assessing the effect of the time discretisation on the instability run with a fixed geometry of $1\text{m} \times 0.5\text{m}$.

V.3.1.2 Results

The time-series of the force experienced by a rigid body of dimensions $1\text{m} \times 0.5\text{m}$ with $\Delta t = 0.001\text{s}$ is shown in fig. V.7, where it clearly appears that the added mass instability is present and grows over time in an oscillatory fashion, until the simulation fails (after 43 iterations in this particular case). For clarity, and due to the highly oscillatory behaviour of the force signal, the envelope of the signal can be used to represent the divergence over time.

The envelopes of the signal of forces experienced by different geometry ratios over time are presented in fig. V.8a and it appears clearly that the larger the width relative to the height, the stronger the added mass effect affects the response of the body and the earlier the simulation fails. A similar observation can be made about temporal discretisation where reducing the time step leads to stronger oscillations of the hydrodynamic forces acting on the body due to the added mass effect, as shown in fig. V.8b.

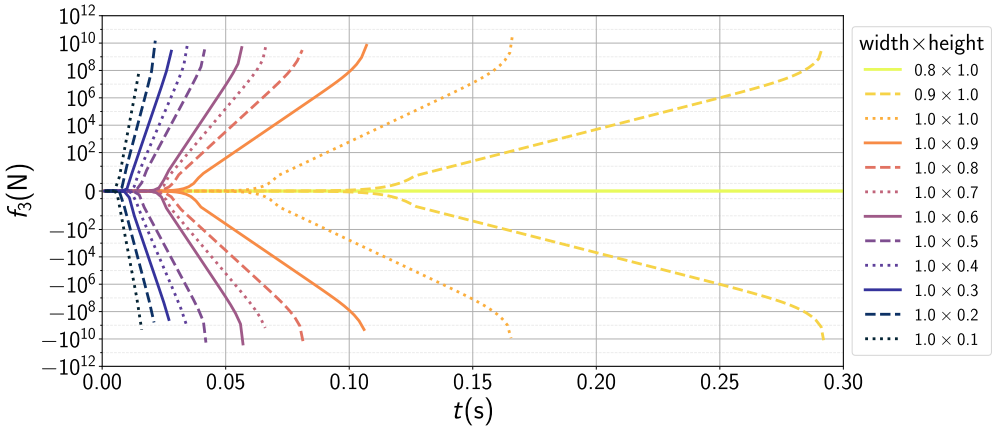
These numerical results confirm that the added mass effect, when present due to the geometry or relative density of the structure is unconditionally unstable for explicit schemes (i.e. temporal or spatial discretisation cannot solve the instability problem) and that a stabilisation scheme must be applied. The effect of the stabilisation scheme on the response of a body is investigated in the next section. Note that applying this stabilisation scheme on the submerged body for the simulations presented in this section would stabilise the response in all cases.

■ V.3.2 Added Mass Estimation for Arbitrarily Shaped Structures

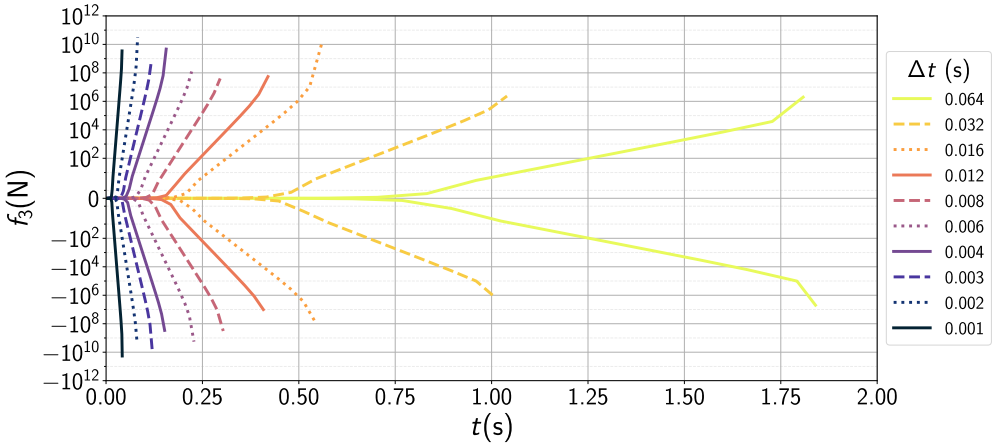
V.3.2.1 Setup

In this verification case, the added mass estimation model described in section IV.2.3.2 is used to calculate the added mass of a submerged structure in an SPF simulation. For this purpose, a square structure of length $D = 1\text{m}$ is placed in the center of a 2D numerical tank of dimensions $10\text{m} \times 10\text{m}$ and filled with a fluid of density $\rho_w = 1000\text{kg m}^{-3}$. The numerical setup is similar to the one shown in fig. V.6.

The Partial Differential Equation (PDE) is solved for each DOF (sway, heave, roll) by setting the related unit



(a) With $\Delta t = 0.001$s and different ratios of width $\times$ height



(b) With width $\times$ height $= 0.5$m $\times$ 1.0m and varying Δt

Figure V.8: Envelope of vertical force acting on rectangular body subject to added mass effect with $\rho_s = \rho_f$

accelerations as Dirichlet boundary conditions along the fluid–structure boundary $\Gamma_{\text{f}\cap\text{s}}$. Unit acceleration of zero is imposed on all boundaries of the tank Γ_{wall}. The numerical domain is discretised using a minimum element size h_{e0} that is gradually coarsened away from the structure with an increase of 10% in characteristic element size from adjacent elements. Note that this model solves a steady-state problem, so there is no temporal discretisation. The simulation is repeated for different values of h_{e0} ranging from $h_{e0} = D$ to $h_{e0} = \frac{D}{2^{11}}$.

V.3.2.2 Results

The result for the full added mass matrix of the body for the most refined case is as follows:

$$\mathbf{A} = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1213.641 & -2.027 \times 10^{-3} & 0.399 \times 10^{-3} & 0 & 0 \\ 0 & -2.026 \times 10^{-3} & 1213.643 & 0.625 \times 10^{-3} & 0 & 0 \\ 0 & 0.399 \times 10^{-3} & 0.626 \times 10^{-3} & 45.232 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \text{kg} \tag{V.7}$$

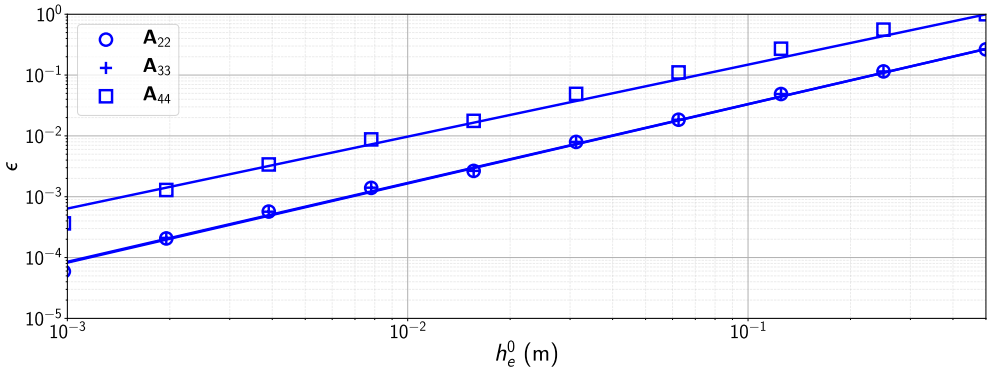


Figure V.9: Spatial convergence of added mass estimation model

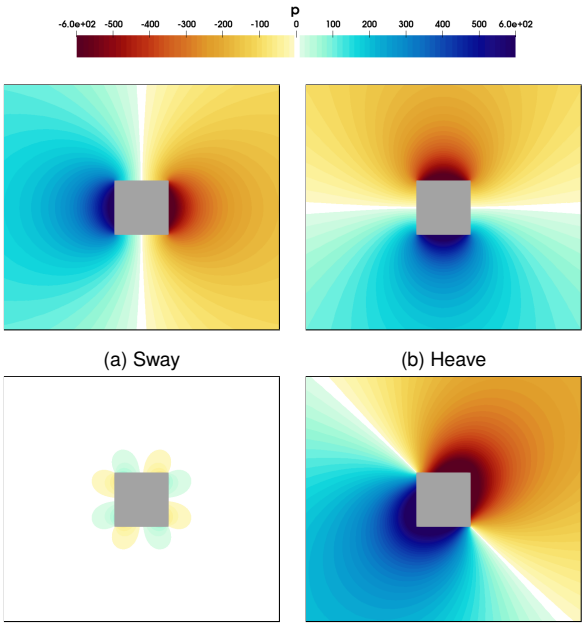
Note that this is presented as a 6×6 matrix for consistency with previous definitions and that, instead, a 3×3 matrix could technically be used as the case is in 2D with only 3 DOFs involved. Each column of this matrix represents the added mass due to the unit acceleration of one DOF on itself and on the other DOFs. In this case, the second column corresponds to sway, third column to heave, and fourth column to roll.

The resulting pressure fields from solving the PDE are shown in fig. V.10 for the most refined case using unit acceleration in sway, heave, and roll, as well as combined sway, heave, and roll. The spatial convergence of the model for these three DOFs is presented in fig. V.9, with $\mathbf{A}_{22}$ for sway, $\mathbf{A}_{33}$ for heave, and $\mathbf{A}_{44}$ for roll. Spatial order of convergence of 1.30 is observed for heave and sway, and 1.18 for roll. Note that the heave and sway values are virtually the same due to the geometry of the structure and the fact that it is placed in the centre of the numerical tank in a single-phase fluid.

It is clear from these results that the added mass estimator model converges to a unique solution as the mesh is refined and that the estimated value of the added mass is in good agreement with analytical values that could be obtained for a fully submerged square geometry. From [97], a square geometry has an added mass of $\mathbf{A}_{22} = \mathbf{A}_{33} = 4.754\rho_f \frac{1}{4} D^2$ for sway and heave (difference of 2.1% with values in eq. (V.7)), and an added mass of $\mathbf{A}_{44} = 0.725\rho_f D^4$ for roll (difference of 0.2% with value in eq. (V.7)).

■ V.3.3 Non-Iterative Stabilisation Scheme

This section has the purpose of showing the capabilities and limitations of the stabilisation scheme presented in section IV.2.3.1. Note that, in general, the stabilisation scheme is expected to use an accurate estimate of the added mass. The effect of the stabilisation on a non-iterative scheme is nevertheless evaluated here assuming that the actual added mass m_a of a body is overestimated by a factor of C_{am} (meaning that the estimated added mass is $C_{am} m_a$). The added mass estimator model is not used in this section but, when it is used (as is the case in several subsequent sections), it is expected to retrieve an estimate that is close to the actual added mass (i.e. $C_{am} \approx 1$), as shown in the previous section.



(a) Sway (b) Heave

(c) Roll (d) Sway, heave, and roll

Figure V.10: Pressure field for calculating added mass with different unit accelerations

V.3.3.1 Setup

In this conceptual case, the MBD solver is used with simple hydrodynamics forcing. For a 1 DOF floating body that is only subject to heave, the equation of motion becomes:

$$\left(m + m_a\right) \ddot{r}_3 = -mg_3 + \rho_f V_w g_3 \tag{V.8}$$

with m the mass of the body, m_a the added mass of the body, ρ_f the fluid density, V_w the volume of water displaced by the body, and g_3 the gravitational acceleration. Equation (V.8) can also be written as:

$$\left(m + m_a\right) \ddot{r}_3 = -\rho_f A_w g_3 \Delta r_3 \tag{V.9}$$

with Δx_3 the distance between the current position of the barycentre of the body and the position of the barycentre of the body at equilibrium, and A_w the horizontal bottom area of the body (assuming the body is a cuboid). Setting $A_w = 1$, the discretised equation of motion with the added mass correction scheme from section IV.2.3.1 becomes:

$$\underbrace{\left(m + C_{am} m_a\right)}_{\text{corrected mass matrix}} \ddot{r}_3^{(n+1)} = \underbrace{-\rho_f g_3 \Delta r_3^{(n)} - m_a \ddot{r}_3^{(n)}}_{\text{hydrodynamics}} + \underbrace{C_{am} m_a \ddot{r}_3^{(n)}}_{\text{correction term}} \tag{V.10}$$

with C_{am} the factor by which the actual added mass m_a is artificially overestimated. This is a 1D equivalent to eq. (IV.17), which can also be written explicitly as:

$$\underbrace{\left(\mathbf{M} + \mathbf{A}\right)}_{\text{corrected mass matrix}} \ddot{\mathbf{s}}^{(n+1)} = \underbrace{\widehat{\mathbf{f}}_f^{(n)} - \widehat{\mathbf{A}} \ddot{\mathbf{s}}^{(n)}}_{\text{hydrodynamics}} + \underbrace{\mathbf{A} \ddot{\mathbf{s}}^{(n)}}_{\text{correction term}} + \mathbf{f}_e^{(n)} \tag{V.11}$$

where $\widehat{\mathbf{A}}$ is the actual added mass (equivalent to m_a in eq. (V.10)), $\mathbf{A}$ is the estimated added mass (equivalent to $C_{am} m_a$ in eq. (V.10)), and $\mathbf{f}_e^{(n)}$ the external forces (equivalent to gravity forces that are included in the hydrodynamics of eq. (V.10)).

In all cases presented here, $C_{am} \geq 1$ so that simulations are always stable as the added mass is either exactly estimated ($C_{am} = 1$) or overestimated ($C_{am} > 1$). The equilibrium position of the body is $r_3 = 0$, from which it is displaced and released from an offset. Because hydrodynamics are applied here only with an external force acting on the body according to its position Δr_3, there is no damping of the response of the body, and it is therefore expected that the body oscillates indefinitely at a given amplitude equal to the original offset and at a given frequency. For convenience, the gravitational acceleration is scaled as follows in order to keep a period of oscillation of $T = 1$s regardless of the value of m_a relative to m:

$$g_3 = -\left(2\pi\right)^2 \frac{m + m_a}{\rho_f} \tag{V.12}$$

The fluid density is set as $\rho_f = 1000\text{kg}\,\text{m}^{-3}$, and the added mass as $m_a = 1000\text{kg}$. Each case runs for $T_{\text{sim}} = 50$s, corresponding to exactly fifty oscillations theoretically. The simulation is repeated with different masses for the body: $m = m_a$, $m = 0.1m_a$, and $m = 0.01m_a$. This range of masses cover both likely scenarios for the applications presented in this work with an added of the same order of magnitude as the mass of the body and unlikely scenarios with an added mass several orders of magnitude larger than the mass. For all three possible values of m, three temporal discretisation levels are considered: $\Delta t = T \times 10^{-4}$, $\Delta t = T \times 10^{-3}$, and $\Delta t = T \times 10^{-2}$. For each possible configuration of values of m and Δt, simulations are repeated with a factor of added mass overestimation ranging from $C_{am} = 1$ (case where the added mass is correctly estimated) to $C_{am} = 2^{19}$ (an unlikely scenario corresponding to a dramatic overestimation of the added mass).

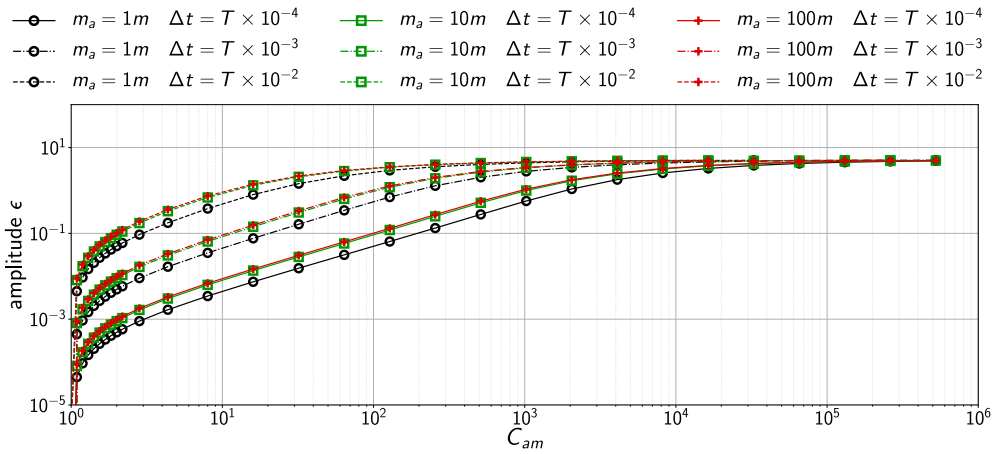
Therefore, these simulations investigate the effect of overestimating the added mass when using the non-iterative stabilisation scheme for different densities (or mass) of the solid relative to the fluid, different refinement levels of temporal discretisation, and different added mass overestimation levels.

V.3.3.2 Results

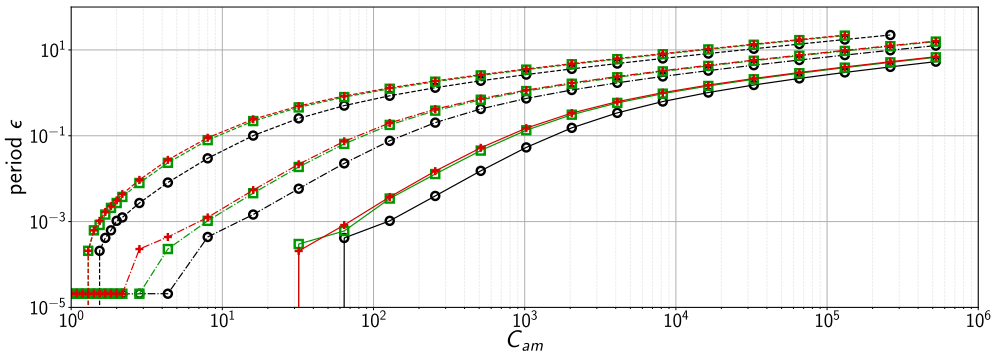
For a given m_a and time discretisation level, all results with $C_{am} > 1$ are compared to the results with $C_{am} = 1$, as the latter is the ideal case where the added mass is correctly estimated. The errors in amplitude and natural period of the response signal are shown in fig. V.11. These errors correspond to the log increment in amplitude per swing and average natural period of the oscillation over the total simulation time T_{sim}. The amplitude of the signal keeps increasing over time, while the natural period is modified but remains constant through time.

Note that each point on fig. V.11 corresponds to one simulation of $T_{\text{sim}} = 50$s for given values m, Δt, and C_{am}. It clearly appears that the stabilisation scheme leads to the following behaviour when the added mass is overestimated (and that this behaviour increases as C_{am} increases): gain of amplitude of the oscillation over time, and increase of natural period. It is also clear from these results that using finer the time discretisation leads to a lower error in the response of the body when compared to the base case of $C_{am} = 1$.

Considering the case where $m_a = m$ (at the edge of stability), with a time discretisation of $\Delta t = T \times 10^{-4}$, the gain in amplitude per swing is less than 1% for $C_{am} < 20$, and the error in period is below 1% for $C_{am} < 300$. With $\Delta t = T \times 10^{-3}$, the added mass can be overestimated by $C_{am} < 3$ before gaining more than 1% in amplitude per swing and by $C_{am} < 40$ before having an error in period of more than 1%. With $\Delta t = T \times 10^{-2}$, if the added mass is overestimated by $C_{am} = 2$, the amplitude gain is approximately 10% per swing. In this case,



(a) Gain in amplitude per swing compared to $C_{am} = 1$



(b) Gain in natural period compared to $C_{am} = 1$

Figure V.11: Error induced by added mass stabilisation scheme for different time discretisation levels (Δt), added mass overestimation factors (C_{am}), and mass (m) to actual added mass (m_a) ratio

$C_{am} < 1.2$ is necessary to obtain an error of less than 1% in amplitude and $C_{am} < 5$ for an error of less than 1% in period.

It is important to note that all the curves of amplitude gain from an added mass factor in the range $1 < C_{am} < 3$ are relatively steep, meaning that a small increase in C_{am} leads to a dramatically increased error. It also appears that the smaller the mass to added mass ratio, the larger the error is. However, this effect is relatively small compared to a change in time discretisation level of the same magnitude. The error in amplitude and period also appears to converge to a single value per case when $m \ll m_a$. Note that the error in gain of amplitude reaches a plateau of around $\epsilon \approx 5$ when C_{am} is very large, and the reason for this has not been clearly identified.

When a model for accurately estimating the added mass such as the one implemented for this work is used (see section IV.2.3.2), it is expected to retrieve a good estimation of the added mass (i.e. estimation with $C_{am} \approx 1$), meaning that the error in response should not be significant. Further investigation is shown in section V.4.1, where the TPF and MBD solvers are coupled for the free oscillation of a heaving body and where the added mass estimator is used to calculate an estimation of m_a with the stabilisation scheme and varying C_{am}. Based on the results shown here, an overestimated value of the added mass can be still used for stabilisation as an alternative to dynamic estimation using a PDE model, and it could be argued that even with $C_{am} \approx 2$, accuracy would not be significantly affected as long as the time discretisation is sufficiently fine.

V.4 Fluid–Structure Interaction (FSI) for Floating Bodies

In this section, the coupling between fluid and structure of the FSI framework is validated. The simulations presented here include the Navier–Stokes, free surface tracking, mesh motion, and rigid body dynamics models. When mentioned, the added mass estimator model and the added mass stabilisation scheme are also used. Additionally, wave absorption and generation zones are used in most cases. All these models are validated together with the following main cases:

- Free translational (heave) oscillation of floating body in section V.4.1,
- Free rotational (roll) oscillation of floating body in section V.4.2,
- Response of a floating body under regular wave loads for a range of wave frequencies in section V.4.3,
- Response of a floating body under extreme, focused wave loads in section V.4.4.

The last case also serves as validation for the added mass estimator and stabilisation scheme as the numerical simulation is subject to strong added mass effect.

V.4.1 Free Decay (Heave)

V.4.1.1 Setup

The simulation presented here is a validation case for the free oscillation of a floating cylinder that is only allowed to heave. The numerical setup is based on the experiment conducted by [75], where a floating cylinder of length $L = 1.83\text{m}$ (6ft) and diameter $D = 15.24\text{cm}$ (6in) is placed horizontally across the width of the tank at the mean water level. The clearance between the tank walls and the end-plates of the cylinder is 1.27cm (0.5in), which corresponds to 0.69% of the cylinder length, essentially making it possible to approximate the 3D experimental setup in a 2D numerical slice. When at rest, the cylinder of density $\rho_s = 0.5\rho_w$ is halfway submerged in a water column of height $h_{\text{mwl}} = 1.22\text{m}$ (4ft), and all its DOFs apart from heave are fully constrained. The numerical tank used here has a total length of $L_{\text{tank}} = 20\text{m}$ with absorption zones of length $L_{\text{abs}} = 9\text{m}$, enough to avoid any interference from reflected waves, and a height of $H_{\text{tank}} = 2h_{\text{mwl}} = 2.44\text{m}$. The cylinder is placed in the centre of the tank at a distance of 10m of either side boundary of the tank (see fig. V.12).

The initial conditions for this simulation are: fluid at rest and cylinder pushed down from its equilibrium position to an initial position of -2.54cm from the mean water level. The boundary conditions are free-slip on the tank bottom and side walls Γ_{wall}, atmospheric condition on the top boundary of the tank Γ_{atm}, and no-slip condition on

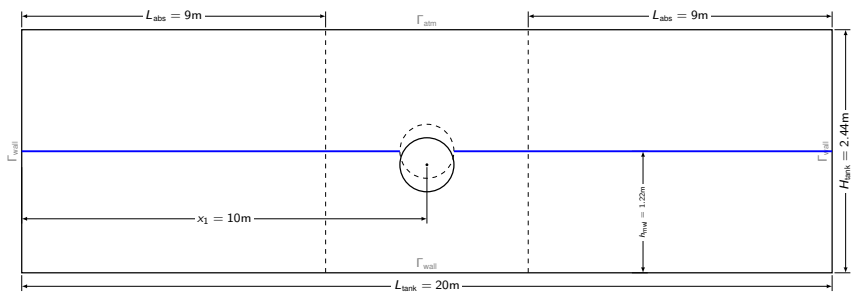
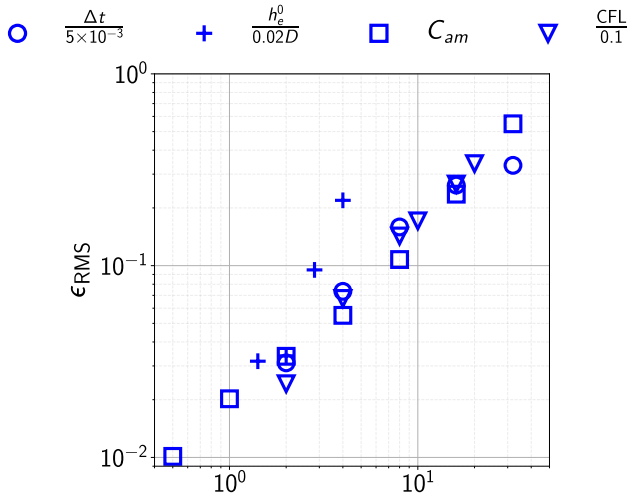


Figure V.12: Schematic representation of numerical domain for free oscillation in heave of a floating cylinder
Solid line: initial position, dashed line: equilibrium position



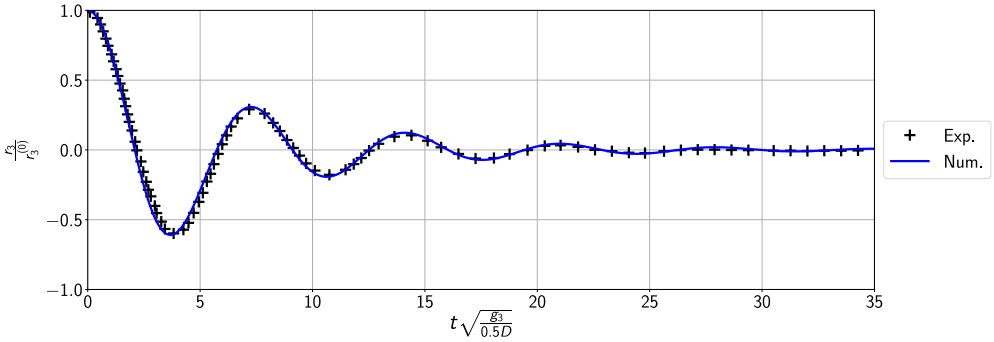
(a) Error in period

Figure V.13: Sensitivity analysis on heaving cylinder case: RMS errors from temporal and spatial discretisations and added mass stabilisation

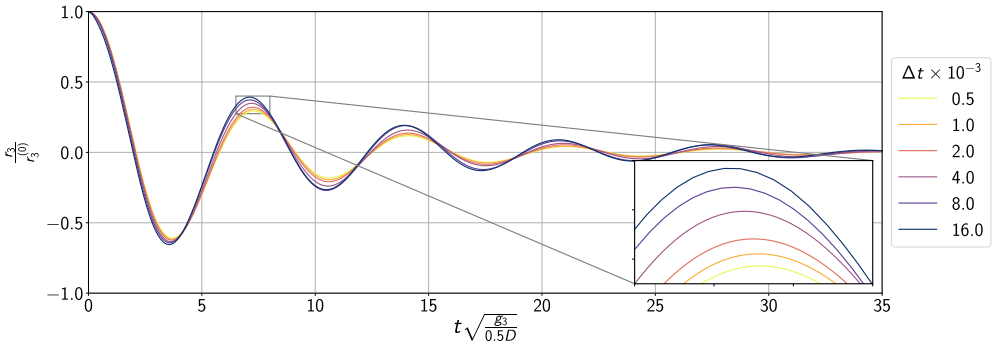
the floating cylinder boundaries $\Gamma_{\text{f}\cap\text{s}}$. The fluid domain is discretised using unstructured triangle elements with the following gradual refinement: minimum characteristic element size ($h_{e0} = 0.01D$ for the most refined case) kept constant up to a distance equal to the initial push of the cylinder around the mean water level and at the boundaries of the cylinder, and gradual coarsening of the mesh applied from the refined areas with a rate of 10% increase in characteristic element size. To assess convergence of the coupled TPF and MBD models, the simulation is repeated using different temporal (Δt and CFL) and spatial (h_{e0}) discretisation levels. Additionally, and even though this particular simulation is fully stable, the effect of the added mass stabilisation scheme is assessed by scaling the added mass returned by the added mass estimator model with a coefficient C_{am}. For all cases, the MBD solver uses a fixed time step of 1×10^{-5}s.

V.4.1.2 Results

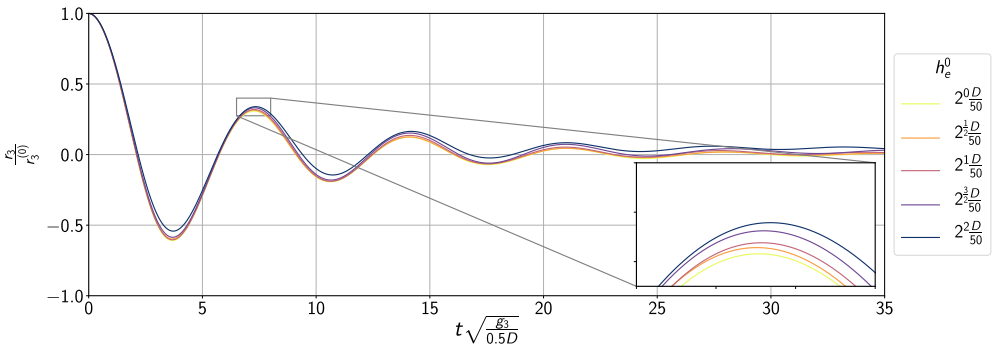
A time series of the numerical results for a highly refined case (CFL $= 0.1$, $h_{e0} = 0.02D$, and $C_{am} = 0$) along with the experimental results digitalised from [75] is presented in fig. V.14a, with the time normalised as $t\sqrt{\frac{|g_3|}{0.5D}}$, and the heave displacement normalised as $\frac{r_3}{r_3^{(0)}}$ (with $r_3^{(0)} = r_3(0)$). This particular case shown in had a mesh composed of $885,433$ elements and took 9.4h running in parallel on 96 cores to produce 4s of data. It is clear that for this highly refined case, the numerical curve is in good agreement with the experimental curve, giving a PCC of 0.995836 between the two. Sensitivity analysis is performed for the following: temporal convergence with Δt ranging from 0.5×10^{-3}s to 16×10^{-3}s as well as for the CFL condition ranging from 0.05 to 0.9 keeping $h_{e0} = 0.02D$ constant as shown in fig. V.14b; spatial convergence with h_{e0} ranging from $0.01D$ to $0.08D$ keeping $\Delta t = 1 \times 10^{-3}$ constant as shown in fig. V.14c; and for different added mass coefficients C_{am} keeping $\Delta t = 1\times10^{-3}$ and $h_{e0} = 0.02D$ constant as shown in fig. V.14d. For the latter sensitivity analysis, it is important to note that this test case does not suffer from instability due to the added mass effect. The sensitivity analysis on C_{am} is done here to mirror the conceptual case of section V.3.3 with an actual FSI case. It appears clearly from



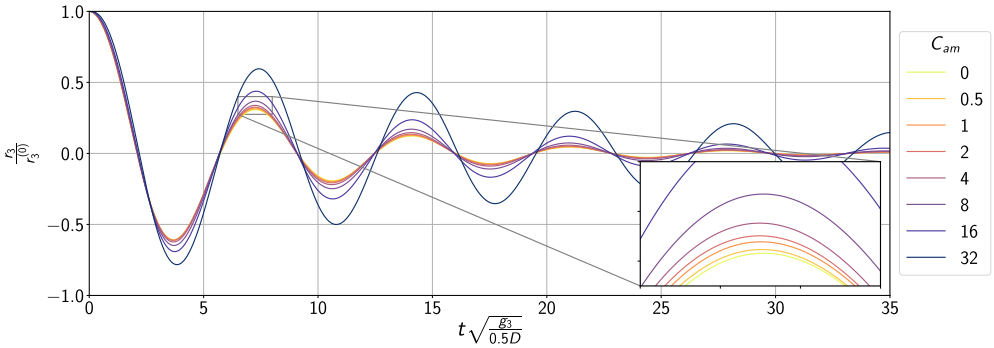
(a) Heave decay of floating cylinder for $\Delta t = 1e^{-3}$, $h_{e0} = 0.02D$, and $C_{am} = 0$. Experimental results digitalised from [75].



(b) Temporal sensitivity with $h_{e0} = 0.02D$



(c) Spatial sensitivity with $\Delta t = 1e^{-3}$s



(d) added mass correction sensitivity with $\Delta t = 1e^{-3}$ and $h_{e0} = 0.02D$

Figure V.14: Time-series and sensitivity analysis of the free oscillation of a heaving cylinder

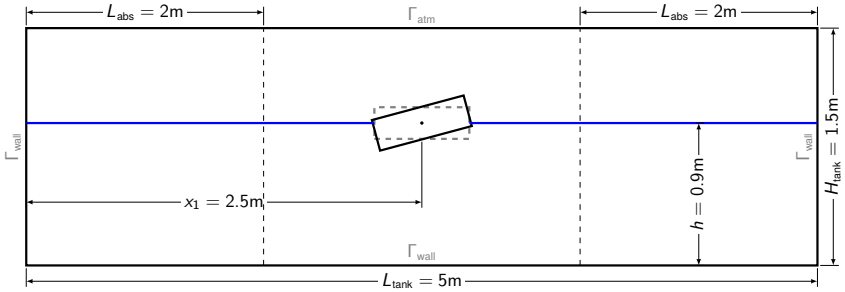


Figure V.15: Schematic representation of numerical domain for free roll oscillation of a floating caisson

these time-series that the solution converges as the spatial or temporal discretisation is refined, as well as when C_{am} is reduced.

A temporal and spatial sensitivity analysis is performed by checking the relative Root Mean Square (RMS) error of timeseries, as shown in fig. V.13. Errors are calculated by comparing the timeseries from each refinement level to results of the most refined case available, i.e. the reference cases that are $\Delta t = 5 \times 10^{-3}$s for temporal investigation with fixed time step, CFL $= 0.05$ for temporal investigation with varying time step, $h_{e0} = 0.02D$ for spatial investigation, and $C_{am} = 0$ for investigation on the effect of the added mass stabilisation scheme. For each case, the RMS error ϵ_{RMS} is defined as:

$$\epsilon_{\text{RMS}} = \sqrt{\frac{\sum_{i=1}^{n} \left(r_3^{(n)} - r_3^{(n),\text{ref}} \right)^2}{n_t}} \tag{V.13}$$

with $r_3^{(n)}$ the solution computed at $t^{(n)}$, $r_3^{(n),\text{ref}}$ the solution of the most refined case at time $t^{(n)}$, and n_t the number of time steps. It appears in fig. V.13 that the solution is converging temporally (CFL and Δt) with an order of approximately 1, while it is converging spatially (h_{e0}) with an order closer to 2. When varying C_{am}, an order of convergence of 1 is also observed. This sensitivity analysis shows that the fluid-structure coupling scheme converges to a single solution as the simulation is refined temporally and spatially, also validating the TPF, mesh motion, and MBD models used for these simulations.

■ V.4.2 Free Decay (Roll)

V.4.2.1 Setup

This simulation is based on the physical experiment conducted in [81], investigating the rolling motion of a floating rectangular caisson. The tank used for the physical experiment has a length of 35m, a height of 1.2m, a width of 0.9m, and is filled with a water column of height $h_{\text{mwl}} = 0.9$m. The floating caisson has a length of 0.3m, a height of 0.1m, a width of 0.9m, and is mounted on the tank walls through a pair of bars and hinges that are placed at the mean water level. Those hinges, acting together as a cylindrical joint allowing only rolling motion of the caisson, are aligned so that the axis of the joint goes through the centre of mass of the caisson. The roll moment of inertia has been calculated experimentally as $I_\phi = 0.236\text{kg m}^2$. Because the tank and the structure have virtually the same width, the experiment can be reproduced numerically using a 2D slice of the 3D experimental setup. Furthermore, absorption zones are used here in the numerical setup in order to absorb waves produced

by the oscillation of the structure, making it possible to dramatically reduce the length of the numerical tank when compared to the experimental setup. A tank of length of $L_{\text{tank}} = 5\text{m}$ with absorption zones of $L_{\text{abs}} = 2\text{m}$ on either end of the tank is therefore used numerically, as shown in the schematic representation of the numerical domain in fig. V.15.

For the free decay test, the initial conditions are: fluid at rest, and a roll angle of 15° for the caisson from the equilibrium position with no initial velocity. The boundary conditions are: no-slip boundary conditions on all boundaries of the caisson, free-slip on the bottom and side boundaries Γ_{wall} of the tank, and atmospheric boundary condition on the top boundary Γ_{atm} of the tank. The numerical simulation runs for at least $T_{\text{sim}} = 4\text{s}$ in order to record the oscillating signal for the same length of time as [81]. The fluid domain is spatially discretised with $h_{e0} = 0.005\text{m}$ around the mean water level and up to a distance of 0.45m of the floating body and uses a gradually coarsened mesh in the same manner as described in the previous case. The TPF solver uses a fixed time step $\Delta t = 5 \times 10^{-3}\text{s}$ while the MBD solver uses $\Delta t_s = 1 \times 10^{-5}\text{s}$.

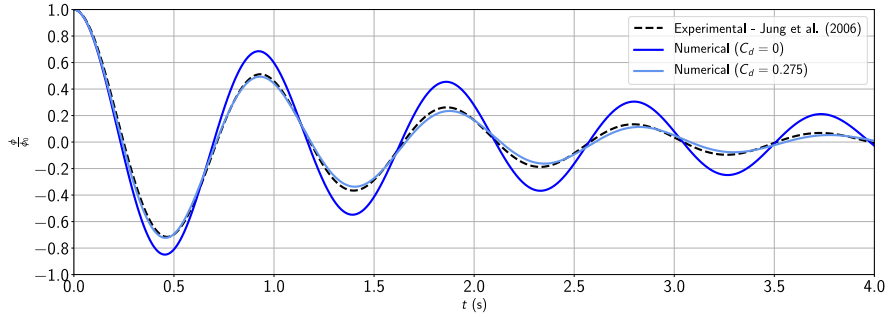
V.4.2.2 Results

The roll motion can be approximated analytically as:

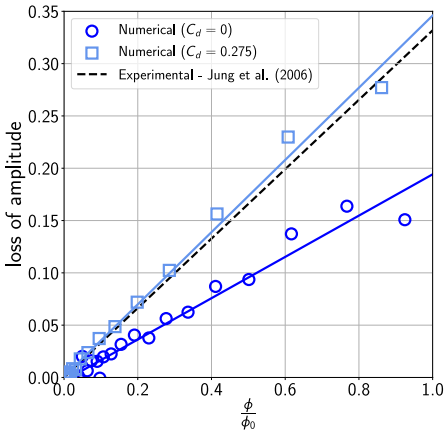
$$\ddot{\phi} + 2\eta_d \omega_n \dot{\phi} + \omega_n^2 \phi = 0 \tag{V.14}$$

with ϕ the roll, and with the following factors given by [81] for fitting the experimental curve: damping factor $\eta_d = 0.106$, and natural angular frequency $\omega_n = 6.78\text{rad}\,\text{s}^{-1}$. This equation is used here to plot and compare the experimental results to the numerical signal. Note that even though no issue with friction between the structure and the hinges is discussed in [81], it is argued in the literature that it is likely that friction actually took place and must be taken into account (see for example [22, 13]). Therefore, the numerical simulation is repeated for the following two scenarios: with a damping coefficient of $C_d = 0$ (no friction) and of $C_d = 0.275$ (friction coefficient from [22]). Note that C_d can be considered here as a fraction of the critical damping from eq. (V.14).

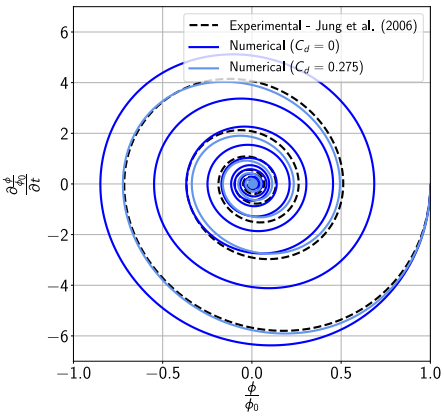
The time-series of the decaying oscillation for the two numerical scenarios and the experiment are shown in fig. V.16a. Extinction curves are represented with the loss of amplitude per swing (half-period) in fig. V.16b where each point represents the difference in amplitude between a peak (trough) and the following trough (peak), and with the continuous extinction of the roll as a function of the roll magnitude in fig. V.16c. There is a clear difference in terms of extinction of the signal between the cases with and without damping coefficients. The roll motion signal recorded from the numerical case with $C_d = 0.275$ follows the experimental signal relatively closely with a mean linear curve of coefficient 0.345 (4.02% difference with the theoretical coefficient), while the numerical case with $C_d = 0$ is clearly underestimating the loss of amplitude with a coefficient of 0.197 (59.48% difference with the theoretical curve). In terms of natural period, both signals are relatively close to the experimental natural period of 0.93s with the mean of the first four periods giving an overestimation of 0.35% for the frictionless case when compared to the experimental period, and 1.49% for the case with friction. The results for each peak-to-peak period are compiled in table V.4, where a slight shift can be clearly observed after the first oscillation in both the friction and frictionless cases. The period of oscillation to reach the first peak is virtually the same between the case with friction and the experiment, while the frictionless case is underestimating the period but, due to the period shifting after



(a) Timeseries of rolling motion



(b) Loss of amplitude per swing



(c) Continuous loss of amplitude

Figure V.16: Free oscillation in roll of floating caisson

Table V.4: Period of oscillation (in seconds) for rolling caisson (peak-to-peak values)

	0-1	1-2	2-3	3-4
$C_d = 0.000$	0.923	0.937	0.939	0.934
$C_d = 0.275$	0.930	0.948	0.947	0.951

this initial oscillation, the averaged period of oscillation of the frictionless case remains closer to the experimental period. Note that, as mentioned in [87], using different initial roll angles (5°, 10°, and 15°) gives virtually the same natural period of oscillation (relative error of less than 2% between the results at different initial angles), showing that different levels of mesh deformation (a greater initial roll angle leads to greater mesh deformation) does not affect the results in any significant manner.

■ V.4.3 Response Under Regular Wave Loads

V.4.3.1 Setup

The set of simulations presented here shows the response of a floating caisson to regular wave loads. The numerical setup is still based on the experimental setup of [81] used in the previous section, with the main difference being the length of the numerical tank that now varies according to the wavelength of the target wave to produce. For a given target wavelength λ, the total length of the tank is $L_{\text{tank}} = 5\lambda$, with a generation zone of length $L_{\text{gen}} = 1\lambda$ and an absorption zone of length $L_{\text{gen}} = 2\lambda$. The centre of mass of the floating body is placed 1λ away from the end of the generation zone and at the mean water level ($r_1 = 2\lambda$ and $r_3 = h_{\text{mwl}}$). A schematic representation of

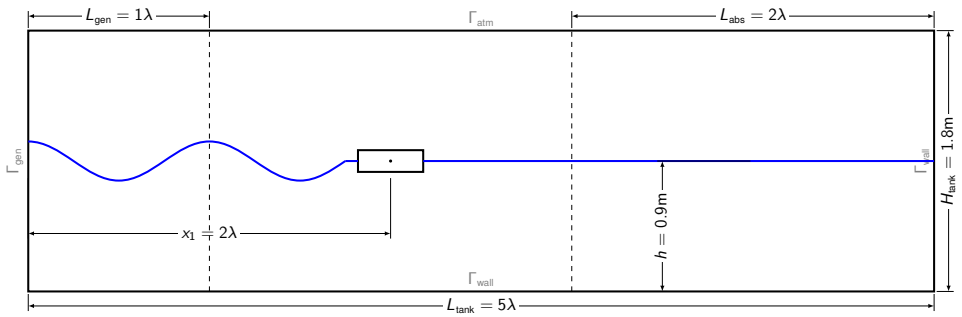


Figure V.17: Schematic representation of numerical domain for the simulation of wave-induced oscillation of rolling caisson

the numerical domain is shown in fig. V.17

Following [81], a total of 23 regular waves are generated in order to compute the Response Amplitude Operator (RAO) of the floating body with the different wave conditions shown in table V.5. Each test case runs numerically for 30 wave periods. Due to the nonlinearity of the waves (Stokes second-order), the Fenton approach with Fourier transform is used to compute their profile at the generating boundary and within the generation zone, as this ensures that the waves are close to their fully developed nonlinear profile as soon as they are numerically generated. For each case, the characteristic element size is set to $h_{e0} = 0.005$m up to a distance of 0.45m from the barycentre of the caisson and up to a distance equal to the wave amplitude ($\pm a$) around the free surface before gradual coarsening is applied. Time-stepping is controlled with CFL $= 0.1$ for the fluid and a fixed time step of $\Delta t_s = 1 \times 10^{-5}$s for the solid. The boundary conditions are the same as in the previous section with the exception of wave generation on the leftmost boundary of the numerical tank. The fluid and floating body are initially at rest, the latter with no initial rotation.

V.4.3.2 Results

The RAO, plotted in fig. V.19 and compiled in table V.5, shows the numerical roll response of the floating body against experimental data and linear theory as well as numerical results of a monolithic code from [25]. The last 10 periods of each case is used to calculate the RAO. Running in parallel on 160 cores for all cases, it took 4h to complete the case with the shortest wave period ($T = 0.6$s), and 48h for the case with the longest wave period ($T = 1.4$s). Concerning the case with $C_d = 0$, the numerical results are generally in better agreement with experimental results than [25], especially around the natural frequency of the caisson. The response under higher frequency waves, where $\frac{\omega}{\omega_n} > 1$, is in good agreement with the linear potential theory as well as the experimental results, with the exception of the response under $\frac{\omega}{\omega_n} = 1.15$, where the numerical results are close to the linear theory while the experiment shows a significantly lower response. This particular divergence in response can be explained by the steepness of the RAO curve for wave frequencies that are relatively close to the natural angular frequency ω_n of the floating body, meaning that even a small difference of wave frequency between the experimental and numerical setup can lead to a significant difference in response. At $\frac{\omega}{\omega_n} = 1.32$, three tests with waves of the same frequency but different heights are performed, and similar responses are obtained, confirming the experimental observations that the wave height has a negligible effect on the relative caisson response at high frequency. In

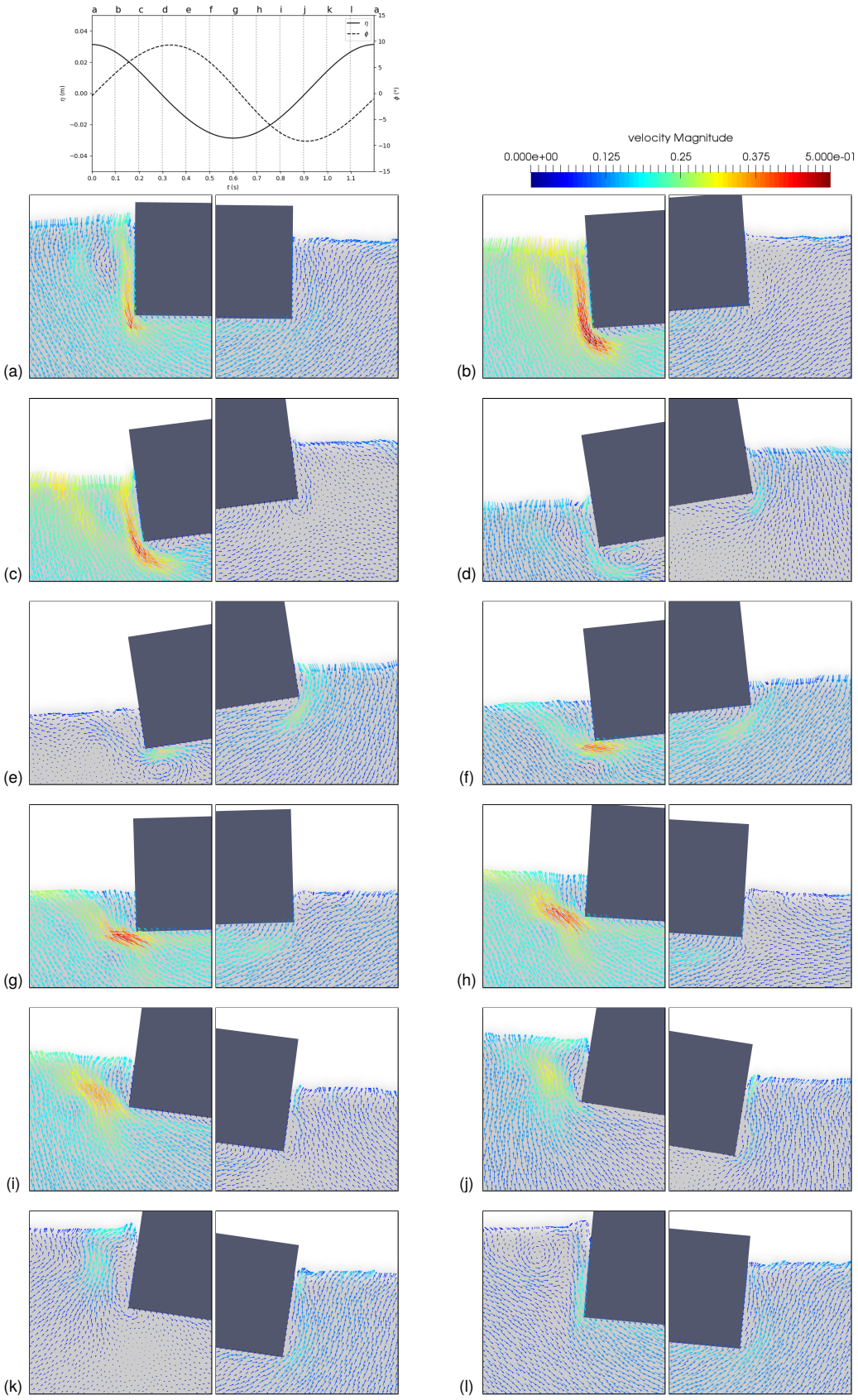


Figure V.18: Snapshots of wave-induced roll motion of floating caisson for $\frac{\omega}{\omega_n} = 0.77$ and $H = 0.06$m
Note: first published in [87]. Top left: temporal map of presented snapshots during one wave period

Table V.5: Wave characteristics and resulting roll response for computing RAO of rolling caisson

#	T	ω	λ	H	ka	$\phi\,(C_d = 0)$	$\phi\,(C_d = 0.275)$
—	s	$\mathrm{rad\,s^{-1}}$	m	m	—	rad	
1	0.50	12.57	0.39	0.010	0.0805	0.0021	0.0021
2	0.60	10.47	0.56	0.017	0.0950	0.0108	0.0106
3	0.70	8.98	0.77	0.015	0.0616	0.0234	0.0221
4	0.70	8.98	0.77	0.023	0.0944	0.0350	0.0326
5	0.70	8.98	0.77	0.029	0.1191	0.0439	0.0412
6	0.80	7.85	1.00	0.029	0.0912	0.0986	0.0826
7	0.85	7.39	1.13	0.033	0.0919	0.1678	0.1251
8	0.93	6.76	1.35	0.016	0.0372	0.1678	0.0956
9	0.93	6.76	1.35	0.027	0.0628	0.2526	0.1502
10	0.93	6.76	1.35	0.032	0.0745	0.2837	0.1736
11	0.93	6.76	1.35	0.040	0.0931	0.3364	0.2300
12	1.00	6.28	1.56	0.044	0.0887	0.3107	0.2184
13	1.10	5.71	1.88	0.057	0.0953	0.2346	0.2058
14	1.20	5.24	2.22	0.032	0.0453	0.0861	0.0814
15	1.20	5.24	2.22	0.060	0.0849	0.1653	0.1537
16	1.20	5.24	2.22	0.067	0.0948	0.1930	0.1770
17	1.30	4.83	2.57	0.060	0.0732	0.1322	0.1268
18	1.40	4.49	2.93	0.061	0.0653	0.1113	0.1080
19	1.50	4.19	3.29	0.062	0.0591	0.0996	0.0959
20	1.60	3.93	3.65	0.060	0.0516	0.0985	0.0906
21	1.80	3.49	4.36	0.061	0.0440	0.1097	0.0794
22	2.00	3.14	5.05	0.026	0.0162	0.0254	0.0162
23	2.00	3.14	5.05	0.059	0.0367	0.0845	0.0697

contrast, the response in roll under waves that have the same frequency as the natural frequency of the caisson ($\frac{\omega}{\omega_n} = 1$) varies greatly with wave height. At this frequency, the normalised numerical response increases as the wave height decreases, which is the expected behaviour according to the experimental investigation. The numerical response under lower frequency waves ($\frac{\omega}{\omega_n} < 1$) is generally closer to the experimental response rather than the linear potential theory. At these frequencies, nonlinearities of the flow around the structure have a significant effect on its response. This is showcased in fig. V.18, where the nonlinearity formed around the caisson appears under loads from the wave of period $T = 1.2$s and height $H = 0.06$m. From these snapshots, a vortex can be seen forming at the bottom left of the floating body just before the trough of the wave reaches the caisson. Due to viscous effects and the rotational direction of the vortex, the nonlinearity actually "feeds" the roll motion of the body (see snapshots (d) and (e) particularly), leading to the higher response observed numerically and experimentally when compared to linear theory. Using a damping coefficient of $C_d = 0.275$ leads to results that are generally in even better agreement with the experimental data, especially for relatively low frequencies. The response is however slightly underestimated around the peak frequency when compared to the experiment with $C_d = 0.275$, but this is less significant than the overestimation of the response with $C_d = 0$.

Overall, these results show that the FSI framework using a fully explicit coupling scheme can successfully predict the response of a 1-DOF floating structure across a complete range of wave frequencies, and can describe the effect of nonlinear events occurring in the flow that affect the response of the body at lower frequencies.

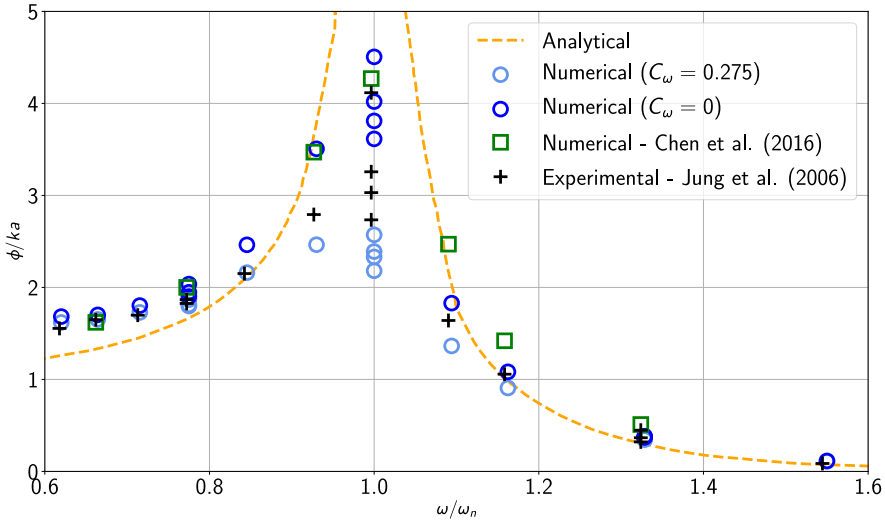


Figure V.19: Response Amplitude Operator (RAO) of rolling caisson under regular wave loads
Analytical (linear theory) curve and experimental points digitalised from [81]; monolithic results digitalised from [25]

■ V.4.4 Response Under Extreme (Focused) Wave Loads

V.4.4.1 Setup

The setup of this validation case is based on the experiment of [135] featuring a 2 DOFs (heave and roll) floating structure under extreme wave loads. The structure consists of a rectangular rigid body of length 0.500m and height 0.123m. In order to prevent any overtopping water from entering the hollow structure of the body and from being transmitted behind the structure, a rectangular superstructure of length 0.200m and height 0.250m is attached to the floating body. The overall width of the floating structure and its superstructure is 0.29m, and its mass is $m = 14.5\text{kg}$. The wave tank used for the experiment at the Research Institute for Applied Mechanics (RIAM) of Kyushu University has the following dimensions: length 18m, height 0.7m, and width 0.3m. The clearance between the wall and the floating structure is therefore 5mm on either side. Due to the design of this experiment, with a structure nearly as wide as the wave tank, a 2D slice of the 3D experiment can be used numerically once again. The water column inside the tank reaches a height of 0.4m when at rest, and a hinge (or cylindrical joint) is placed in the tank at $x_1 = 7\text{m}$ and $x_3 = 0.4\text{m}$. The structure is mounted on this hinge at a point placed 0.1m above its keel, or 20.4mm above its barycentre, giving a radius of gyration of 15.35mm around the hinge. The hinge itself is mounted on a heaving rod of mass 0.276kg, allowing the floating body to roll and heave but restraining it fully in sway. The extreme wave to be generated is focused in space at $x_f = 7\text{m}$ (corresponding to the x_1 position of the floating body) and in time at $t_f = 20\text{s}$ with peak frequency $f_p = 1\text{s}^{-1}$ and focus amplitude $a_f = 0.06\text{m}$. The domain is represented schematically in fig. V.20.

The numerical boundary conditions are free-slip on the tank bottom and right boundaries Γ_{wall}, atmospheric on the tank top boundary Γ_{atm}, unsteady TPF inlet on the tank left boundary Γ_{gen} (Dirichlet fluid velocity, pressure, and VOF), and no-slip boundary conditions on all boundaries of the floating structure $\Gamma_{\text{f}\cap\text{s}}$. The domain is discretised spatially with $h_{e0} = 0.005\text{m}$ around the structure and free surface level up to a distance of $\pm a_f$ before gradually coarsening the mesh (with 10% increase in characteristic element size between adjacent elements). Temporally,

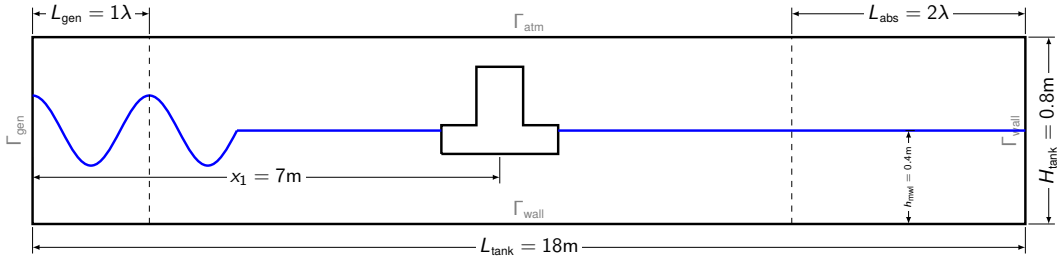


Figure V.20: Schematic representation of numerical domain for the simulation of 2 DOFs floating caisson under extreme wave loads

the conservative CFL value of 0.1 is used for the fluid in order to have an accurate description of the focused wave, and $\Delta t_s = 1 \times 10^{-5}$s is used for the structure.

When compared to the previously described FSI simulations, it is worth noting that this numerical simulation brings two new challenges that have to be tackled before producing satisfactory results: the generation and effect of the impact of an extreme wave (sometimes referred as “freak” wave) on a floating structure and, most importantly, the added mass effect that is triggered due to the characteristics of this structure (geometry and density).

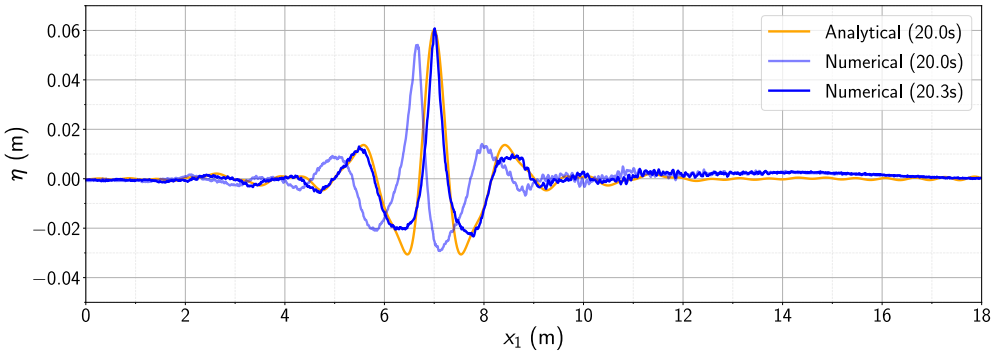
V.4.4.2 Generating the Focused Wave Numerically

From the linear wavemaker theory, it is possible to calculate the velocity needed from a piston wavemaker at a set location to reproduce a wave with the following formula for the wavemaker motion:

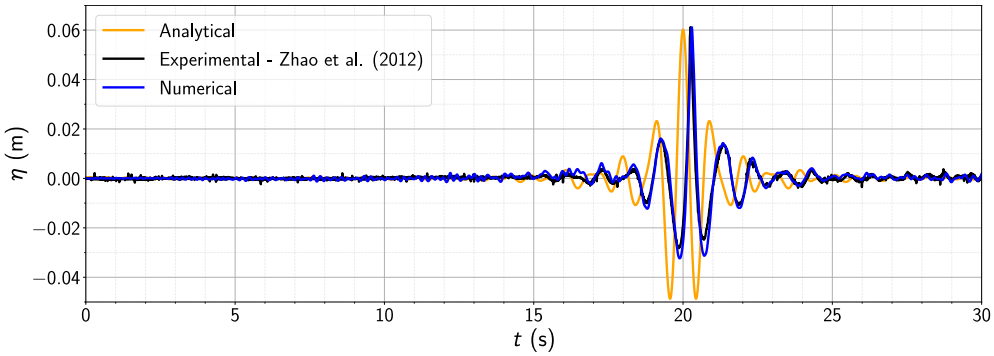
$$X^{\text{wm}}(t) = \sum_{i}^{N_f} a_i \cos\left(k_i(x_1^{\text{wm}} - x_f) - \omega_i(t - t_f)\right) \tag{V.15}$$

with X^{wm} the surge motion of the wavemaker, x_1^{wm} the original position of the wavemaker (when at rest), and N_f the number of frequencies used to generate the wave. In this case, 29 frequencies are used experimentally to generate the wave, ranging from $f_{\text{min}} = 0.6$Hz to $f_{\text{max}} = 1.6$Hz. This wavemaker technique can also be used numerically with moving boundaries, as is the case in [25] (which is also based on this experiment). However, a different numerical approach is used here: the focused wave is generated on a fixed boundary with velocity, pressure, and VOF Dirichlet boundary conditions from the fluid velocity profile calculated with the Joint North Sea Wave Project (JONSWAP) spectrum directly. The same number of frequencies and the same frequency range as the experiment are used numerically to produce the focused wave with the technique described in section III.2.1.2.

In section V.4.3 (and any case only generating monochromatic waves), the numerical domain is usually significantly smaller than the experimental domain and scaled according to the wavelength of the wave to be generated. This is because regular waves that are weakly nonlinear can be generated numerically with a technique such as Fenton Fourier transform that provides wave characteristics close to their fully developed profile. This does not apply to more complex spectral waves, including focused waves. The analytical solution for this type of waves is relatively far from a fully developed profile as nonlinear wave–wave interaction can occur when waves of different frequencies interact with each other. Therefore, it is common and even expected to observe a difference or a shift in terms of focus point experimentally or numerically with nonlinear models when compared to the theoretical or analytical focus point. This shifting issue is usually tackled using a trial-and-error approach where the theoretical



(a) Focused wave elevation over space at $t = t_f$



(b) Focused wave elevation over time at $x_1 = x_f$

Figure V.21: Free surface elevation of focused wave over time at $x_1 = x_f$ and over space at time $t = t_f$

focus point is adjusted until the wave focuses at the intended point experimentally or numerically. A wave can be considered fully focused when its profile appears to be symmetric around its peak (with troughs of approximately the same height on either side). In the case presented here, it is particularly important to focus the wave at the intended point $x_f = 7\text{m}$ in order to have the peak of the extreme wave hit the structure. Because this problem happens in nonlinear models as well as in the real world, the numerical tank used here matches exactly the dimensions of the experimental tank, the only difference being that it is a 2D slice of the 3D setup and that the height was chosen to be 0.8m (corresponding to twice the height of the mean water level). With these dimensions, a numerical trial-and-error investigation was undertaken for obtaining the right wave profile without the structure. It has been found numerically that using the corrected theoretical focus point of $x_f^c = 6.5\text{m}$ (instead of $x_f = 7\text{m}$) on the JONSWAP spectrum allows for the wave to actually focus at the intended $x_f = 7\text{m}$. To apply this correction, the following phase shift is applied on all frequency components of the JONSWAP spectrum:

$$\phi = -k(x_f - 0.5) + 2\pi f t_f \tag{V.16}$$

with k the wavenumber, and f the frequency. The numerical profile of the wave using this phase correction is shown in fig. V.21a, where it clearly appears that the wave focuses at $x_1 = 7\text{m}$. It also appears that the numerical wave, while symmetrical around its peak, has the intended peak value of $a_f = 0.06\text{m}$ but higher troughs than the analytical wave and a sharper peak. Indeed, using the signal of the analytical wave shown in the figure and corresponding to a JONSWAP spectrum with $H_s \approx 0.0365\text{m}$ creates a numerical wave with a peak that is higher than intended. In order to reach a maximum amplitude of 0.06m, the equivalent of $H_s \approx 0.0320\text{m}$ is used here for

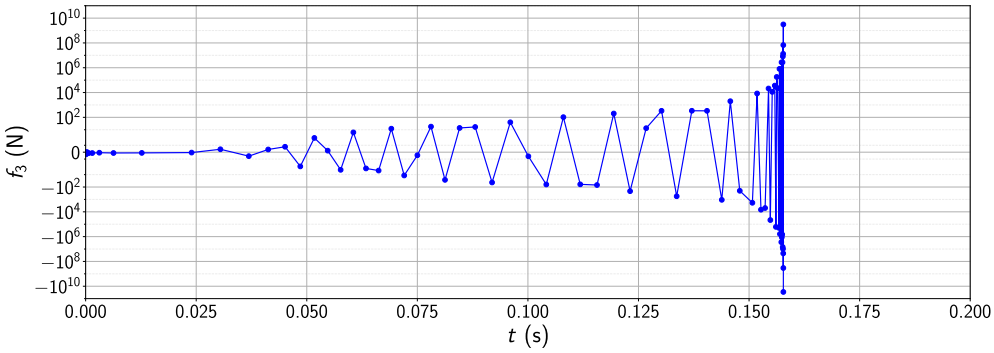


Figure V.22: Response of floating body prone to added mass effect without stabilisation scheme

producing the numerical wave. It can also be seen in fig. V.21a that the focus time is slightly shifted numerically with the wave focusing at $x_f = 7\text{m}$ at time $t_f^{\text{num}} = 20.3\text{s}$. This issue in terms of amplitude and focus time can again be explained by nonlinear wave–wave interaction. This can be accounted for by using another phase correction on each frequency component but, as can be seen in fig. V.21b, the experimental wave also experiences a temporal shift when compared to the analytical wave. It clearly appears that the numerical and experimental wave profiles match each other well, with both waves slightly trailing behind the analytical wave and focusing at $t_f^{\text{num}} = t_f^{\text{exp}} \approx 20.3\text{s}$ against the theoretical $t_f^{\text{ana}} = 20.0\text{s}$. It was therefore decided to keep this temporal shift in order to be as close as possible to the experimental conditions for validating the response of the floating body to this focused wave. Note that the experimental curve does not appear in fig. V.21a because experimental data was not available for plotting the wave profile along the tank at a given time, while the numerical profile was obtained with gauges spaced out by $\Delta x_1 = 1\text{cm}$.

V.4.4.3 Results

The particular layout (geometry and mass properties) of the floating body described above trigger the added mass effect when using a partitioned scheme. When no added mass stabilisation scheme is used, and with a temporal discretisation of $\text{CFL} = 0.1$ and spatial discretisation of $h_{e0} = 0.005\text{m}$, the simulation quickly fails after a few iterations due to instabilities leading to oscillations of the hydrodynamic forces growing in magnitude over time, as shown in fig. V.22. It can also be seen in that figure that time steps become smaller as oscillations grow, which is due to the Courant–Friedrichs–Lewy (CFL) condition that limits the possible displacement of the fluid flow over one time step within a mesh cell, combined with increasing oscillations of the body (that in turn increase the velocity of the flow). The added mass stabilisation scheme of section IV.2.3.1 has to be used here along with the added mass estimation model of section IV.2.3.2 for ensuring a stable simulation and producing the results presented below.

With a total of 490,543 mesh elements, and running in parallel on 196 cores, the stabilised simulation of the floating structure subject to extreme wave loads produced 30s of data in 57h. Numerical snapshots of the simulation around the focus time of the extreme wave and compared to experimental snapshots are shown in fig. V.23. It shows a clear qualitative similitude of free surface profile as well as floating body position and rotation between the numerical and experimental results. Furthermore, the overtopping events are well represented numerically with water hitting the superstructure at $t = 20.3\text{s}$ (which corresponds to the actual numerical and experimental focus

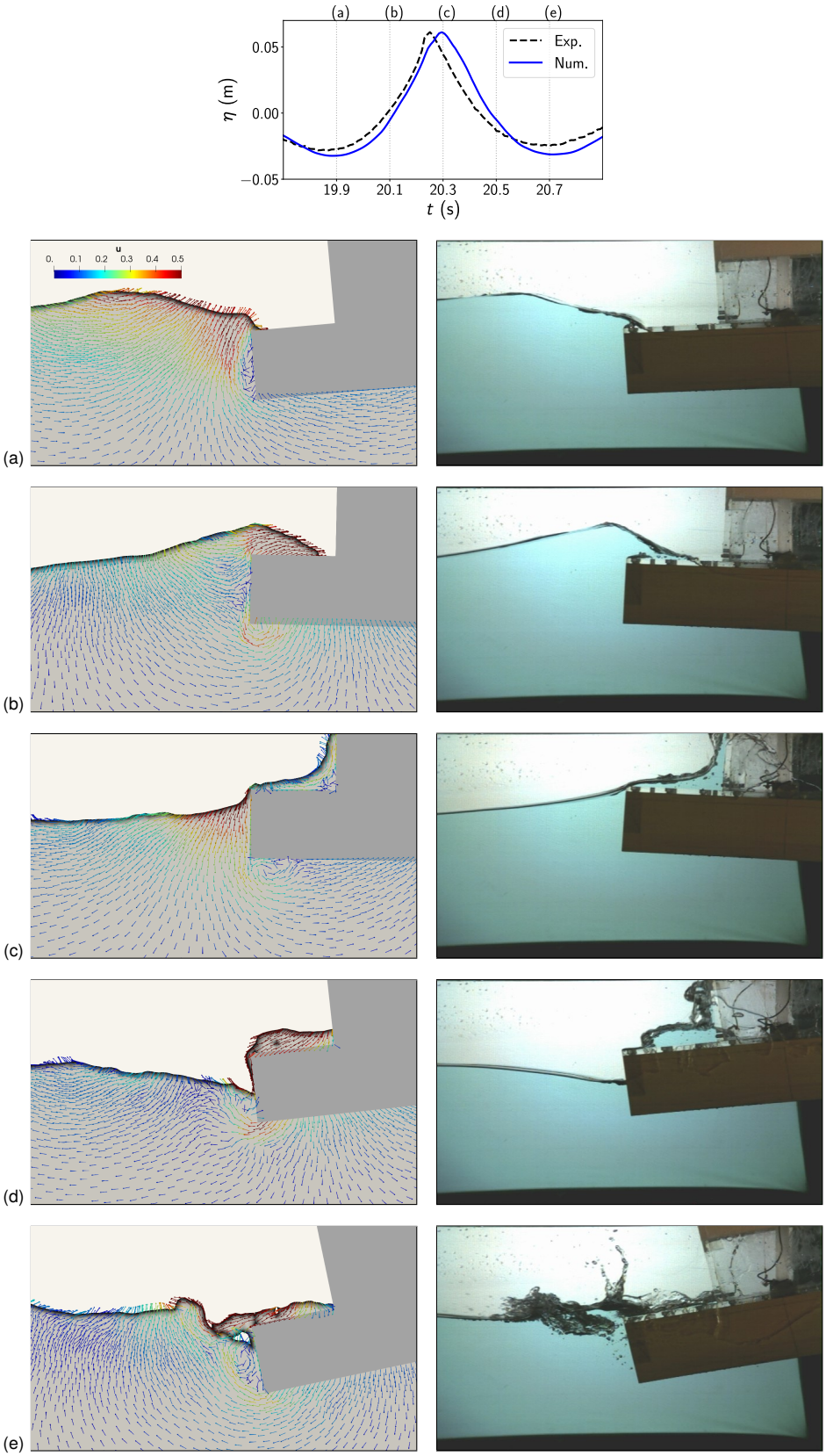
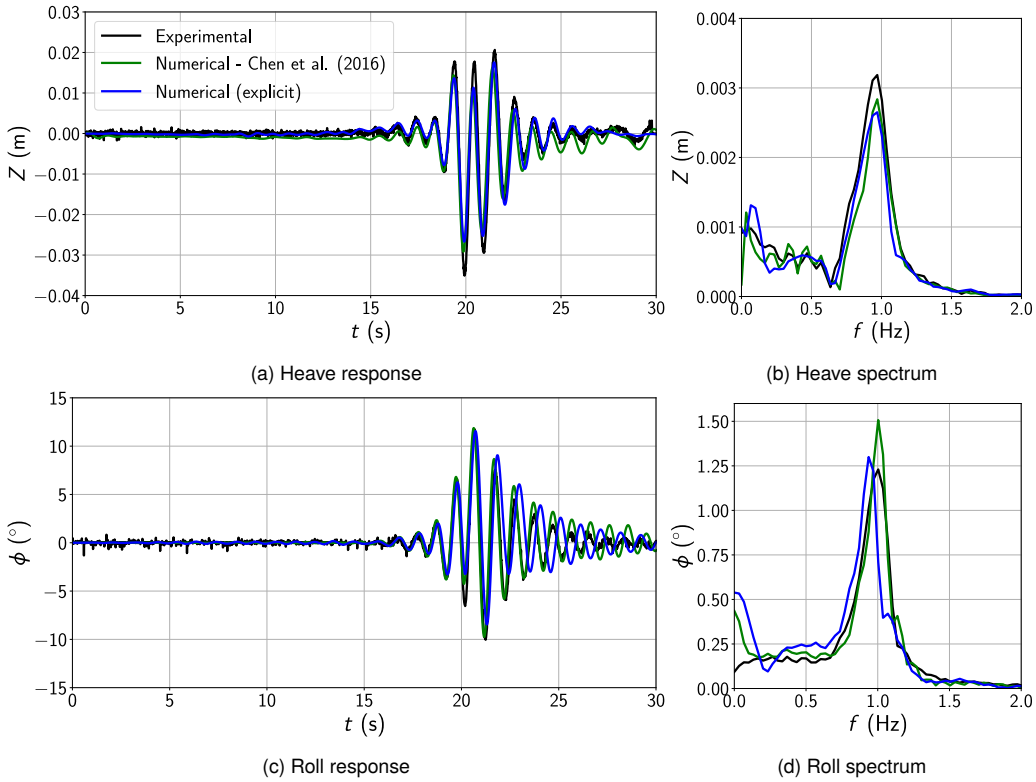


Figure V.23: Snapshots of floating body hit by focused wave ($t_f = 20$s)
Note: snapshots at 19.9s, 20.1s, 20.3s, 20.5s, and 20.7s (see top left graph for temporal map and wave elevation).
Left: numerical results, right: experimental results from [135].



(a) Heave response

(b) Heave spectrum

(c) Roll response

(d) Roll spectrum

Figure V.24: Response in heave and roll of floating body to focused wave loads
Experimental results from [135], monolithic results from [25]

time), and retreating from the structure at $t = 20.7$s. Quantitative results in terms of response of the floating body are shown in fig. V.24 for heave and roll. The results are also compared to the numerical results from [25] where a completely different approach is used for modelling the fluid known as Particle-In-Cell (PIC) and, most importantly, a monolithic scheme is used for fluid–structure coupling. Because the monolithic approach ensures a stable simulation with no added mass effect, the response obtained by [25] can be used for code-to-code validation against the added mass stabilised partitioned coupling scheme implemented here. In terms of heave, the experiment and the two numerical models predict the peak frequency accurately at $f_p^Z = 0.97$Hz which is the one obtained experimentally, and both underestimate the amplitude of the heave response of the body (16.8% for the explicit scheme and 10.9% for the monolithic scheme). In terms of roll, the peak response frequency is underestimated with the explicit scheme by 6.67%, while the monolithic scheme and experimental results are in agreement with $f_p^\phi = 1.00$Hz. The amplitude in roll is however better estimated with the explicit scheme with an overestimation of the response of 5.6%, against an overestimation of 22.5% for the monolithic scheme. Furthermore, it appears clearly on the time-series of fig. V.24c that the response frequency of the explicit scheme is initially in phase with the experimental response but that the phase shift occurs after the peak of the focused wave has passed (at $t > 21$s).

The variations of the added mass over time calculated by the added mass estimator model for sway, heave, and roll are shown in fig. V.25, corresponding to $\mathbf{A}_{22}$, $\mathbf{A}_{33}$, and $\mathbf{A}_{44}$ respectively. The initial added mass values for each DOF (which has been subtracted in the figure) are as follows: $\mathbf{A}_{22}^{(0)} = 2.478$kg, $\mathbf{A}_{33}^{(0)} = 45.271$kg, and $\mathbf{A}_{44}^{(0)} = 0.268$kg. The most important added mass contribution is in heave, where $\mathbf{A}_{33}^{(0)} = 3.12m$ (more than

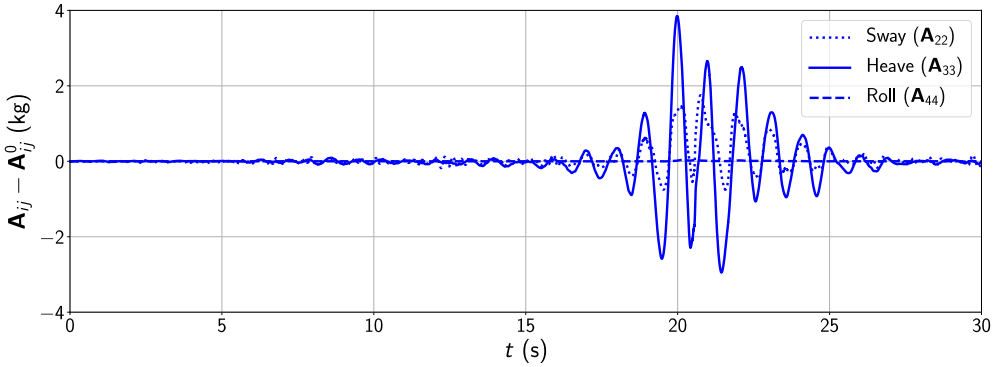


Figure V.25: Variation of the calculated added mass over time for floating body hit by focused wave (in global coordinates)

three times the mass of the body), hence the unconditional instability observed in fig. V.22 when there is no added mass stabilisation scheme.

Overall, these results further confirm that the stabilisation scheme of section IV.2.3.1 for fully explicit coupling schemes is viable as the response obtained here is close to the experimental response up to an error range comparable to the one obtained by an inherently stable monolithic numerical scheme.

V.5 Moorings

Verification and validation of the uncoupled mooring model is undertaken with the simulations presented in this section. The TPF model is not used here because these simulations do not involve wave loads, but only mooring statics and dynamics using prescribed motion of fairleads. Including the computationally expensive fluid model in the simulations would have no significant effect on the cables as the fluid is assumed to be at rest, and the effect of the fluid on the cable is taken into account but not feedback from the cable to the fluid (one-way coupling). This effect is straightforward to calculate directly when the fluid is at rest: the drag and added mass forces are simply applied to the cable with fluid velocity $\mathbf{u}_f = 0$ and fluid acceleration $\dot{\mathbf{u}}_f = 0$. The following verification and validation cases are presented in this section:

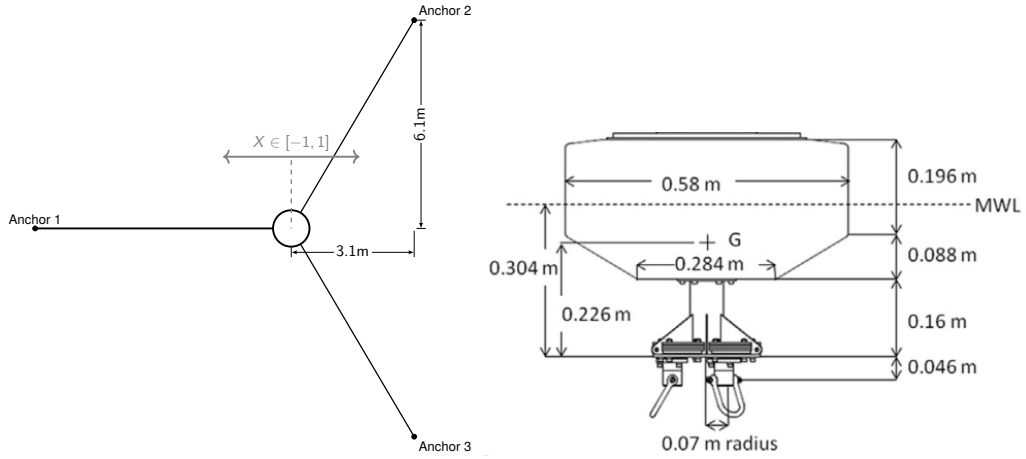
- Statics / quasi-statics model validation for multi-segmented and elastic cables in section V.5.1,
- Dynamic model validation against statics / quasi-statics model for a single catenary chain in section V.5.2,
- Dynamic model validation for mooring damping from sinusoidal top-end motion in section V.5.3.

As the computational demand of the simulations presented in this section is relatively low due to the absence of CFD, all simulations are performed in series on the laptop with characteristics introduced in table V.3.

V.5.1 Statics Model Validation

V.5.1.1 Setup

The quasi-statics model developed for this research and presented in section III.4.1 is validated against experimental data from and numerical results from OrcaFlex provided by [65]. The experiment was conducted in the Ifremer deep water wave basin in Brest (France), where a 1:5 Froude scale version of the South West Mooring Test Facility (SWMTF) was tested. This buoy and its mooring system have characteristics that are typical to offshore floating



(a) Top-down view of mooring system

(b) Dimensions of WEC (from [65])

Figure V.26: Layout of mooring system and WEC for statics model validation

point absorber Wave Energy Converters (WECs): synthetic mooring system, small device size, and highly dynamic system. The scaled WEC is moored using three catenary lines that are 120° apart, as shown in fig. V.26a. Each line is composed of four connected segments that have different length L, diameter d_0, axial stiffness EA_0, submerged weight w_0, drag coefficient C_d, and added mass coefficient C_m, as shown in table V.6 (note however that C_d and C_m are not actually used with the quasi-statics model as it solves for equilibrium). Each line is composed of three different chains and one rope, leading to a total line length of $L = 10.057\text{m}$.The horizontal distance from the anchor to the centre of the buoy is 7.083m, and the fairlead radius is 0.142m, leading to a horizontal distance between the anchors and the fairleads of $d = 6.941\text{m}$ at equilibrium. The mean water level is set as $h_{\text{mwl}} = 5.95\text{m}$ and the draft of the buoy is 0.304m without the mooring system which, when mounted, adds $0.1 - 0.078 = 0.022\text{m}$ to the draft. Taking into account that the lines are mounted on attachments that protrude at an estimated $4 \times 0.046\text{m}$ below the buoy, the fairlead height is therefore set as $h = 5.486\text{m}$ at equilibrium. This last estimation is based on the drawings of the WEC from [65] and shown in fig. V.26b, as well as the fact that this value gives a pretension that matches the experimental pretension at equilibrium (i.e. for surge offset $X = 0$).

The static experiment consists in recording the value of the tension at the fairleads for different surge offsets of the WEC from its equilibrium position. While the WEC is placed at 11 specific surge offsets in [65], the statics / quasi-statics module is used here to calculate the tension at different surge position by displacing the WEC in surge and recording tensions every 1cm in a range of $-1\text{m} < X < 1\text{m}$ in order to obtain a relatively smooth curve.

Table V.6: Characteristics of mooring lines for [65] test case

Seg. #	Seg. Type	L m	d_0 mm	w_0 kg m^{-1}	EA_0 kN	C_{d_n}	C_{d_t}	C_{m_n}	C_{m_t}
1	Chain	0.259	8.0	1.529	6.464×10^3	1.0	0.4	1.0	0.07
2	Rope	4.000	9.0	$4.25e^{-3}$	10.873	1.6	0.4	1.0	0.00
3	Chain	0.126	6.0	1.558	3.636×10^6	1.0	0.4	1.0	0.07
4	Chain	5.672	4.9	0.402	2.050×10^6	1.0	0.4	1.0	0.08

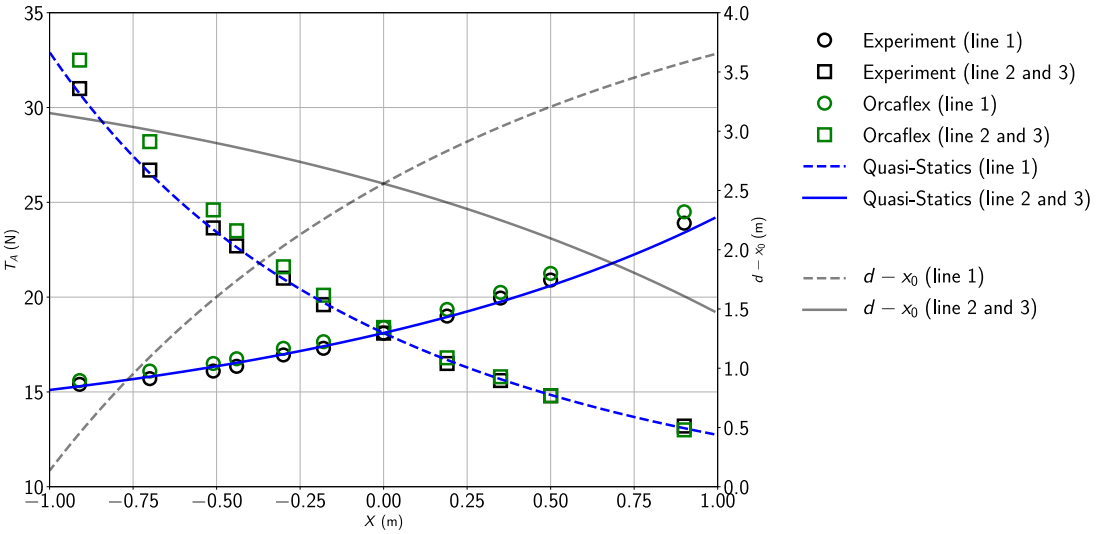


Figure V.27: Mooring static results for different surge offsets of a WEC with three multi-segmented catenary lines
Experimental data and OrcaFlex results from [65]

V.5.1.2 Results

The results for the tension in each line are plotted in fig. V.27, along with the experimental and OrcaFlex results digitised from [65]. The lengths of the lines laying on the seabed ($d - x_0$) at the different surge positions calculated by the quasi-statics model are also featured on the plot. Note that two of the mooring lines give the same results in terms of tension due to symmetry of the mooring system across the surge direction. Good agreement is observed between the quasi-statics model and the experimental data. The numerical L^2-norm error is calculated as:

$$|\mathbf{r}|_2 = \frac{1}{n_p^{\text{exp}}} \sum_{i=1}^{n_p^{\text{exp}}} \|\mathbf{T}_f^{\text{exp}} - \mathbf{T}_f^{\text{num}}\| \tag{V.17}$$

with n_p^{exp} the number of experimental points considered, $\mathbf{T}_f^{\text{exp}}$ the experimental tension at the fairlead and $\mathbf{T}_f^{\text{num}}$ the numerical tension from the statics / quasi-statics module at the fairlead. Errors of $|\mathbf{r}|_2^{[1]} = 0.107$ for line 1 and $|\mathbf{r}|_2^{[2,3]} = 0.164$ for line 2 and 3 for the quasi-statics model. This appears to be slightly better than the results from Orcaflex that give errors of $|\mathbf{r}|_2^{[1]} = 0.623$ for line 1 and $|\mathbf{r}|_2^{[2,3]} = 0.364$ for line 2 and 3 against the experimental data. Note that both the experimental and OrcaFlex data were digitised and that there could be a small bias error. Furthermore, the Orcaflex results do not show exactly the same pretension as the experiment when the system is at equilibrium (i.e. at $X = 0$).

The results here are clearly satisfactory for the statics / quasi-statics module as it is validated for multi-segmented and elastic mooring lines against experimental data. It can therefore be confidently used for static or quasi-static analysis, and for setting initial conditions of the nodal positions for the dynamic mooring model developed for this work, as will be shown in subsequent sections.

V.5.2 Static Convergence of Dynamic Model

V.5.2.1 Setup

Using the gradient deficient Absolute Nodal Coordinate Formulation (ANCF) dynamic mooring framework implemented using Chrono, the spatial convergence of the equilibrium position of a cable hanging between two points is checked here. The anchor $\mathbf{r}_a$ and fairlead $\mathbf{r}_f$ are placed on the same level horizontally so that the cable can hang symmetrically between these two points when at equilibrium. No collision detection is used as there is no seabed, and only the effect of gravity is acting on the cable so that, at equilibrium, a simple catenary shape is formed by the cable. Spatial convergence of the cable position is assessed for different levels of discretisation by varying the number of cable elements $n_\mathcal{E}$. Note that the number of cable nodes $n_\mathcal{N} = n_\mathcal{E} + 1$ in this case.

The length of the cable is set as $L = 100\text{m}$, with its anchor coordinates $\mathbf{r}_a$ and fairlead coordinates $\mathbf{r}_f$ separated by a horizontal distance of $d = \|\mathbf{r}_f - \mathbf{r}_a\| = 0.5L$. The cable of bar diameter $d_0 = 0.01\text{m}$ has a linear mass of $m_L^{\text{chain}} = 21.9 d_0^2 \text{t m}^{-1}$ (typical of a studlink chain) and axial stiffness of $EA_0^{\text{chain}} = 1.01 \times 10^8 d_0^2 \text{N m}^{-2}$. There is no bending stiffness in the cable, which is typical for chains in numerical simulations, and this is set numerically with a moment of inertia $I = 0$ (leading to $EI = 0$).

As this is a dynamic simulation, and even though we are looking for static equilibrium here, three different methods are used for reaching the equilibrium position. The first one is to fully extend the cable between an initial position of the fairlead and anchor of $d = \|\mathbf{r}_f - \mathbf{r}_a\| = L$, place the nodes of the cable equidistantly between those two points with $\Delta s = \frac{L}{n_\mathcal{N}}$ and, when the simulation starts, move the anchor and fairlead to their final position at a given rate of $\dot{\mathbf{r}} = 0.01\text{m s}^{-1}$ using prescribed motion. This process is illustrated in fig. V.28. The second technique is to already have the anchor and fairlead at their final position initially with $d = 0.5L$ and to place the nodes between the two points with $\Delta s = \frac{0.5L}{n_\mathcal{N}}$, which means that the cable is heavily compressed initially. Finally, the last technique is to use the quasi-statics module for placing the nodes in a catenary position directly. All three of those techniques yielded the exact same results that will be discussed below.

V.5.2.2 Results

The results of each simulation is compared to the solution of the most discretised simulation for which the number of elements and nodal positions at equilibrium are denoted $n_\mathcal{E}^0$ and $\mathbf{r}^0$ respectively. In all cases, and regardless of the

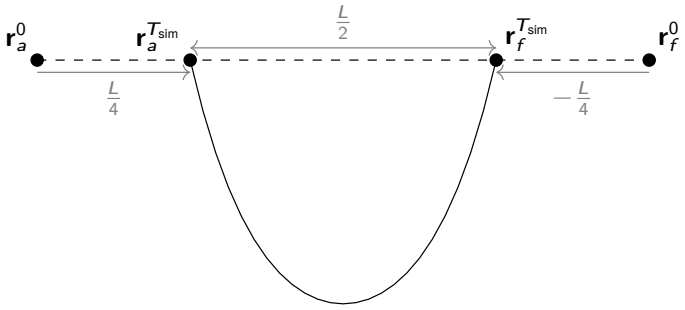
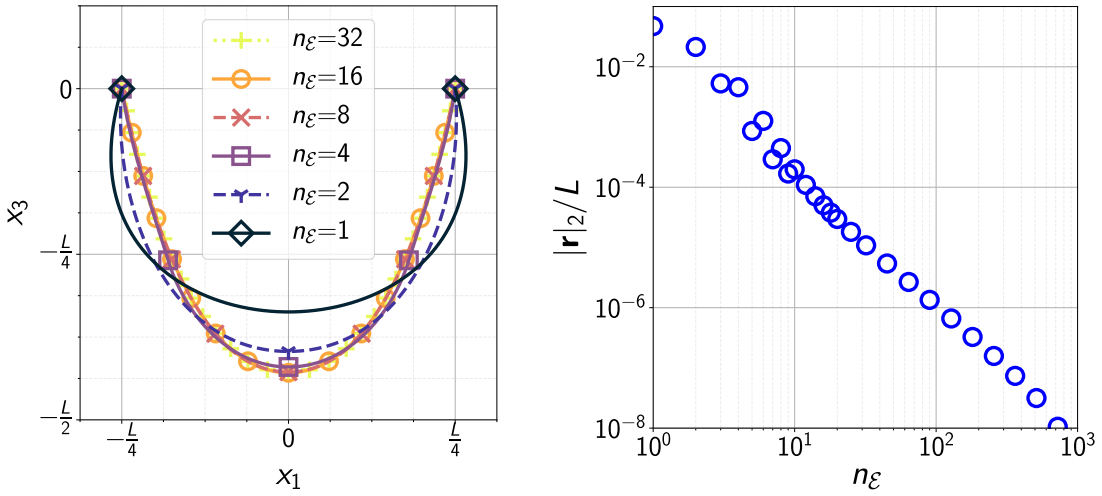


Figure V.28: Original and final position of cable for spatial convergence test of catenary position at equilibrium
Dashed line: original position; solid line: final position



(a) Shape of catenary at equilibrium for different number of elements

(b) Normalised L^2-norm error as a function of the number of elements

Figure V.29: Spatial sensibility analysis of cable position at equilibrium
Note: in fig. V.29a, simulations were run with more elements (as can be seen in fig. V.29b), but the results are not plotted here for clarity purposes, as beyond $N_\mathcal{E} = 32$, the convergence is hardly noticeable on a linear scale

number of elements $n_\mathcal{E}$ composing the cable, the position of $n_\mathcal{E}^0 + 1$ equidistant points separated by $\Delta s = \frac{L}{n_\mathcal{E}^0}$ are recorded along the cable. For the finest cable with $n_\mathcal{E} = n_\mathcal{E}^0$, this simply corresponds to the position of each of its nodes. For any of the coarser discretisation levels, recording the positions along the cable is achieved by using the shape functions defined for gradient deficient ANCF cables as described in section III.4.2.3 that allow the evaluation of the absolute position of any point along the cable. The L^2-norm is then calculated for each discretisation level as follows:

$$|\mathbf{r}|_2 = \frac{\sum_{i=1}^{n_\mathcal{E}^0+1} \left\| \mathbf{r}\left(i\Delta s\right) - \mathbf{r}^0\left(i\Delta s\right) \right\|}{n_\mathcal{E}^0 + 1} \tag{V.18}$$

Results are presented in fig. V.29a for the position of the nodes of the cable for different levels of refinement as well as the layout of the elements between the nodes through interpolation using the shape functions. Note that refinements of $n_\mathcal{E} > 32$ are not plotted for clarity purposes as the difference in position for finer cables would be qualitatively indistinguishable. In fig. V.29b, the normalised L^2-norm error $\frac{|\mathbf{r}|_2}{L}$ is featured against the spatial refinement of the cable. It appears clearly spatial convergence is achieved with order 1 and that using $n_\mathcal{E} \geq 100$ would give an error of $\frac{|\mathbf{r}|_2(n_\mathcal{E} \geq 100)}{L} < 1 \times 10^{-6}$, which is clearly an acceptable error for the engineering cases presented in the next sections.

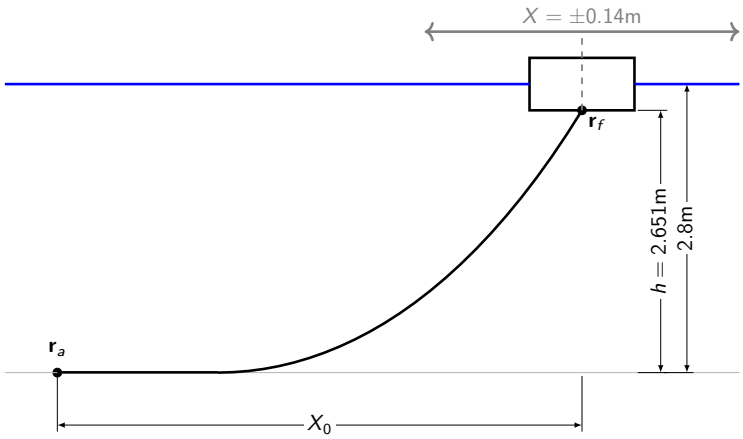


Figure V.30: Schematic representation of the numerical setup for mooring dynamic validation case

V.5.3 Mooring Dynamics Validation

V.5.3.1 Setup

The mooring simulations presented in this section aim at validating the capability of the mooring dynamics solver to evaluate static pretensions and to model nonlinearities due to dynamic motion of fairleads. The numerical simulations presented here are based on and compared to the experimental setup and results of [78, 79]. Firstly, the fairlead of a catenary mooring chain placed at a height $h = 2.651$m is held at different surge positions and the resulting static cable tensions at the fairlead are recorded. The experimental mooring line used here has a length $L = 6.980$m, bar diameter $d_0 = 2.5 \times 10^{-3}$m, submerged weight $w_0 = 1.036\,\mathrm{N\,m^{-1}}$, and axial stiffness $EA_0 = 560$kN. The characteristics of the numerical cable are the same as the experimental ones (see table V.7), with drag and added mass coefficients taken from typical studless chain values. As the water level is $h_{\mathrm{mwl}} = 2.800$m, the cable is fully submerged at all times.

To assess mooring damping according to various pretensions at the fairlead which is attached to a buoy (or WEC), prescribed and cyclic motion in surge is imposed on the structure experimentally. For different angular frequencies, the WEC is moved in sinusoidal fashion starting at different initial surge positions X^0 that correspond to different pretensions T_{A0} as shown in table V.8. Note that, as this damping is entirely due to dynamic effects depending on the speed or acceleration of the fairlead, it cannot be predicted using quasi-statics. A schematic representation of the experiment is shown in fig. V.30.

V.5.3.2 Static Pretensions

In order to ensure that the dynamic model can compute static tensions properly, mooring dynamics and quasi-statics are compared to each other for different surge positions. For the mooring dynamics model, the cable is

Table V.7: Characteristics of lines for mooring dynamics validation case

Seg. #	Seg. Type	L m	d_0 mm	w_0 $\mathrm{N\,m^{-1}}$	EA_0 kN	C_d^n	C_d^t	C_m^n	C_m^t
1	Chain	6.980	2.500	1.036	560	2.4	$1.15\pi^{-1}$	1.00	0.5

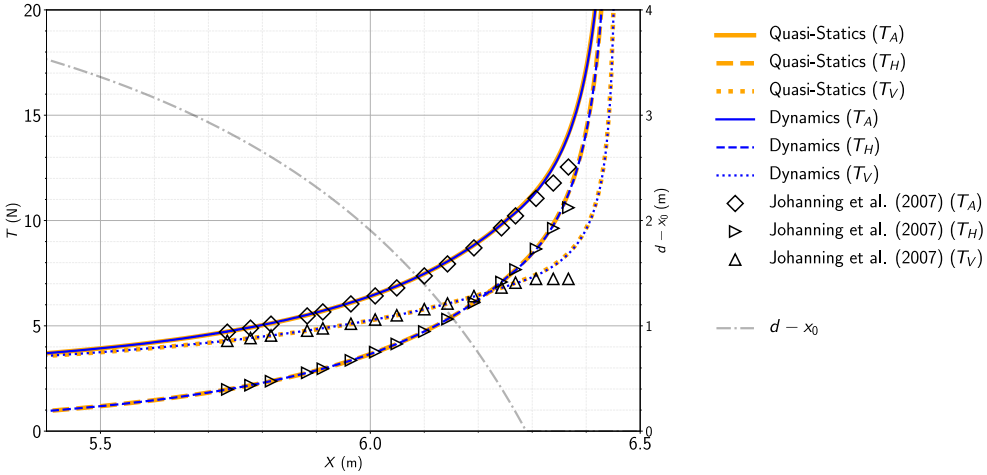


Figure V.31: Mooring statics analysis: experimental, static, and dynamic models comparison
Experimental data from [79]

highly discretised with $n_{\mathcal{E}} = 100$ and is driven from a fully stretched position ($X = \sqrt{L^2 - h^2} = 6.457\text{m}$ relative to the anchor) back to $X = 5.4\text{m}$ at a rate of $\dot{X} = -5\text{mm s}^{-1}$ with time steps of $\Delta t = 1e^{-4}\text{s}$, which is slow enough to make sure that no significant dynamic effect is introduced. This allows for producing a relatively smooth curve of static tensions along the whole surge range investigated.

The results are presented in fig. V.31, with tensions recorded every 0.01m for the dynamics and quasi-statics models and compared to 16 experimental surge points as recorded by [79]. The length of the mooring line on the seabed for the different surge positions according to the quasi-statics solver is also plotted for reference. It is clear here that excellent agreement is observed between the quasi-statics and dynamics model for the tensions at the fairlead across the whole surge range. This agreement is meaningful in terms of cross-validation because the quasi-statics and dynamics methods use two very distinct approaches: analytical catenary equations and Finite Element Method (FEM) respectively. There is also good agreement with the experimental results when the line is partly lifted, but results start diverging once the line is fully lifted, where the tensions in the fairlead are overestimated numerically when compared to the experiment. Note that there might be an experimental inconsistency as the vertical tension at the fairlead recorded from [79] does not vary experimentally once the line is fully lifted. The reason for this remains

Table V.8: Parameters and results for mooring line damping test cases

Case #	X^0 m	${T_{A0}}^{\text{exp}}$ N	${T_{A0}}^{\text{num}}$ N	$\frac{L_s}{L}$ —	ω rad/s	$\frac{{E_L}^{\text{exp}}}{w_0 ha}$ —	$\frac{{E_L}^{\text{num}}}{w_0 ha}$ —
2	5.778	4.890	4.929	0.611	0.341	0.009	0.003
6	5.964	6.040	6.106	0.704	0.429	0.004	0.011
8	6.049	6.800	6.888	0.761	0.483	0.007	0.022
10	6.143	7.940	8.053	0.837	0.561	0.081	0.049
12	6.243	9.650	9.818	0.942	0.644	0.196	0.112
13	6.269	10.220	10.371	0.974	0.779	0.157	0.154
14	6.307	11.050	11.351	1.000	0.722	0.197	0.253
15	6.339	11.780	12.530	1.000	0.787	0.452	0.428
16	6.367	12.540	14.051	1.000	0.939	0.684	0.843

Note: cases are numbered following [79]

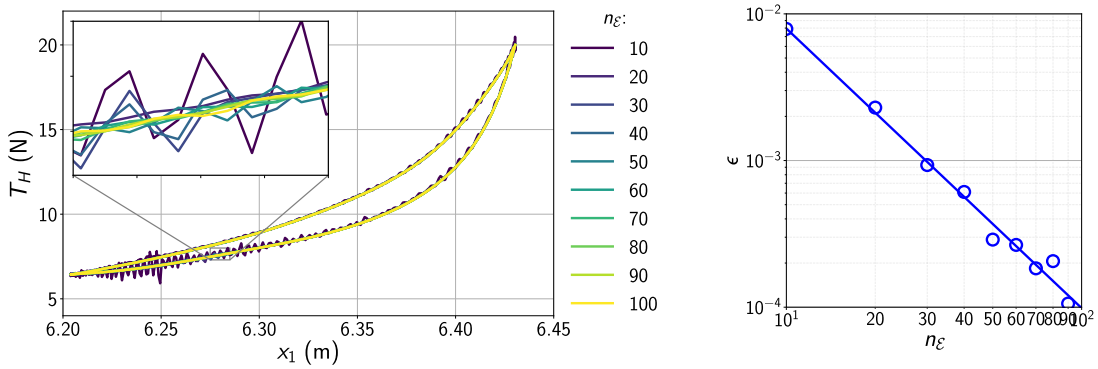


Figure V.32: Spatial sensitivity on cable dynamics with $\Delta t = 1 \times 10^{-4}$s on case 16

unknown, but this could for example be due to slight tilting of the device when the line becomes fully lifted and tensions become exponentially higher. Importantly, the horizontal tensions remain in relatively good agreement, and these are the ones we are interested in for plotting indicator diagrams in the next section.

V.5.3.3 Dynamic Line Damping

After validating pretensions over surge positions, the fairlead is now driven sinusoidally from different initial pretensions and frequencies as shown in table V.8, and with peak-to-peak amplitude $0.1h_{\text{mwl}}$. The difference in axial pretension between the experiment ${T_{A0}}^{\text{exp}}$ and the dynamics model ${T_{A0}}^{\text{num}}$ is also recorded from the results given by the previous section. For the numerical simulation, the recorded experimental surge displacement is used as input for the prescribed motion of the fairlead. From this sinusoidal motion in surge, the energy dissipation — or damping — of the horizontal tension T_H is calculated. This energy dissipation is obtained by calculating the area within the closed-loop of the X-T_H curve. Note that a total of sixteen cases were run experimentally, but only nine cases are presented here due to the experimental data that was made available. Importantly, all cases with relatively high pretensions are shown here, with cases featuring no or very little damping being cut off from the set of simulations. The most nonlinear case (number 16) is chosen to perform a numerical sensitivity analysis. The results for the level of discretisation of the cable keeping $\Delta t = 1 \times 10^{-4}$s constant is shown in fig. V.32 where spatial convergence of order 1.9 is observed as $n_\mathcal{E}$ increases. Following these results, a discretisation of $n_\mathcal{E} = 100$ elements, and time-stepping of $\Delta t = 1 \times 10^{-4}$s was selected.

The results in damping are shown in two different forms. Firstly, indicator diagrams for each case are presented in fig. V.33, showing the variation in horizontal tension over the surge range for the numerical simulation and the experiment for one cycle. Looking at fig. V.33, the quasi-statics model, which is incapable of representing nonlinear effects, is nonetheless always in good agreement with the mooring dynamics model in terms of minimum and maximum horizontal tensions. Minimum experimental tensions are also in good agreement with the numerical results across all cases. However, for case 12 and any other subsequent case with a higher pretension, numerical and quasi-static results show higher maximum tensions than the experiment. As this does not happen for cases with lower pretensions where the line is never fully lifted during the sinusoidal cycle, this discrepancy can also be associated with experimental issues discussed section V.5.3.2 when the line becomes fully lifted.

As mentioned earlier, the hysteresis of the curves formed in the indicator diagrams is due to nonlinear effects

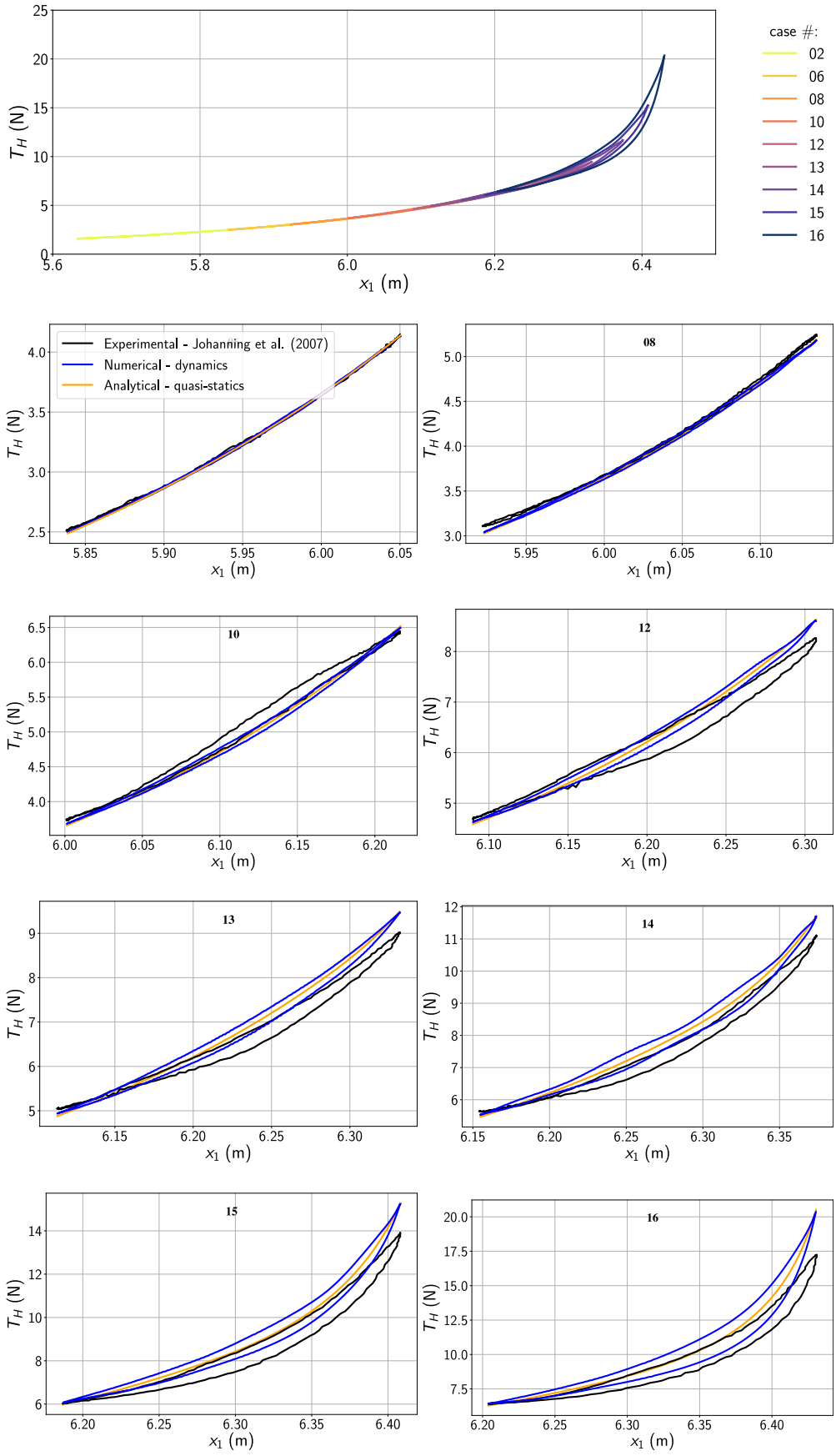


Figure V.33: Indicator diagrams of horizontal tension at fairlead
Note: case 02 not shown due to similarity with case 06

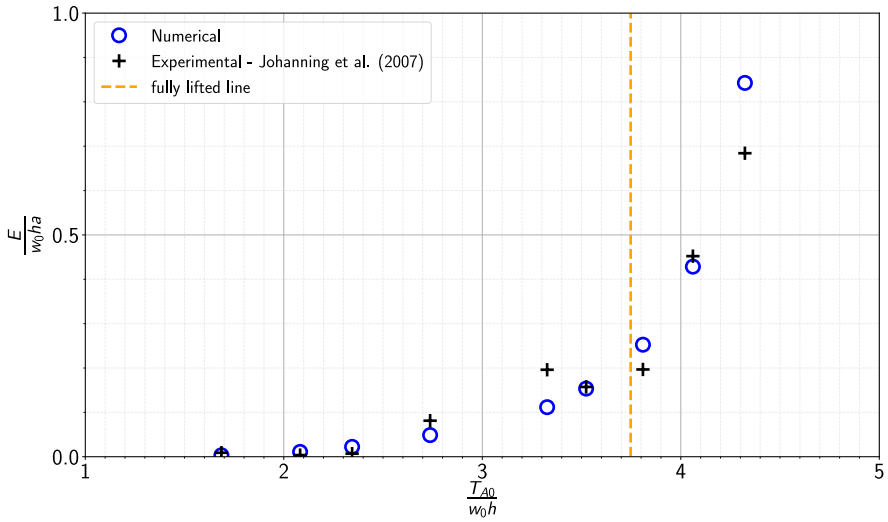


Figure V.34: Energy loss as a function of pretension

induced by the cyclic motion of the fairlead and is used to calculate the damping of the line. The energy dissipation caused by drag forces on the mooring line can be calculated as follows:

$$E_L = \int_t^{t+T} T_H \dot{X} \mathrm{d}t \tag{V.19}$$

with T the period of the sinusoidal cycle, T_H the horizontal tension and X the surge (and $\dot{X}$ the velocity). This leads to the second form of presentation of the results where the non-dimensional damping $\frac{E_L}{w_0 h a}$ is plotted for each case in section V.5.3.3. Relatively good agreement between experimental and numerical results show that nonlinear effects are well simulated, with low damping when pretensions are low, and an exponential increase in damping as pretensions become higher and the line is fully lifted. In case 2, 6, and 8, the damping is low both experimentally and numerically, meaning that there is barely any dissipated energy. This can be verified in the indicator diagrams where all curves for these cases are in relatively good agreement with the quasi-statics model (which does not simulate any nonlinear effect). For case 10 and above, damping becomes significant. There is a slight inconsistency in the experimental results, again around the point where the line becomes fully lifted, with the damping of case 12 higher than the damping of case 13 and 14 which both have a higher pretension that should lead to higher damping under forced oscillation. With the numerical model, any case with a higher pretension yields a higher damping. With the good agreement between numerical and experimental damping for different pretensions, the validation presented here shows that the dynamic mooring model with gradient deficient ANCF elements can successfully capture nonlinear behaviour from dynamics that occurs in real-world applications.

V.6 Moored Semi-Submersible Platform for Floating Wind Turbine

V.6.1 Setup

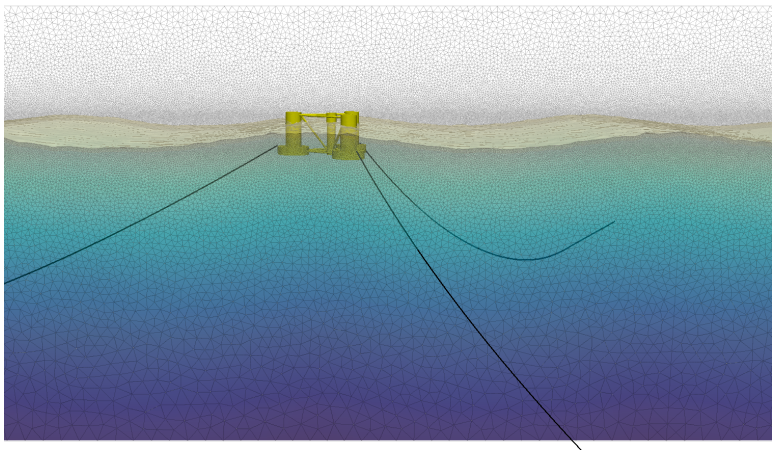


Figure V.35: Snapshot of rendered numerical domain for coupled simulation of OC4-DeepCwind platform

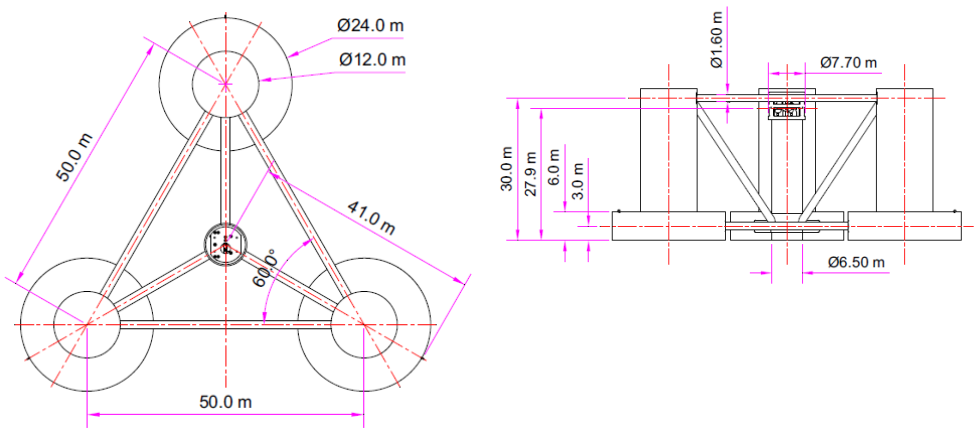


Figure V.36: Geometry of OC4-DeepCwind semi-submersible
Adapted from [84]

This validation case is based on the physical tests of the 1:50 scale model of the OC4-DeepCwind semi-submersible platform for Offshore Wind Turbine (OWT) from [111, 84]. Froude scaling was used on the physical model was tested in the wind/wave basin at the Maritime Research Institute Netherlands (MARIN) by the University of Maine DeepCwind program in 2011. The semi-submersible platform is composed of four cylinders connected to each other by trusses. The Wind Turbine (WT) is supported by a central cylinder that is surrounded by three larger cylinders, as shown in fig. V.36. The main physical properties of the semi-submersible are compiled in table V.9 from the references cited above and the MARIN report. The mooring system of the structure consists of three catenary chains with identical characteristics and length that are separated by an angle of 120°. The main mooring system properties are compiled in table V.10, with several characteristics such as drag and added mass coefficients calculated following the methodology of [64] where some of the values had been refined from previous reports due to the non-conventional geometry of the mooring chain. The experimental setup and data from this 1:50 model were

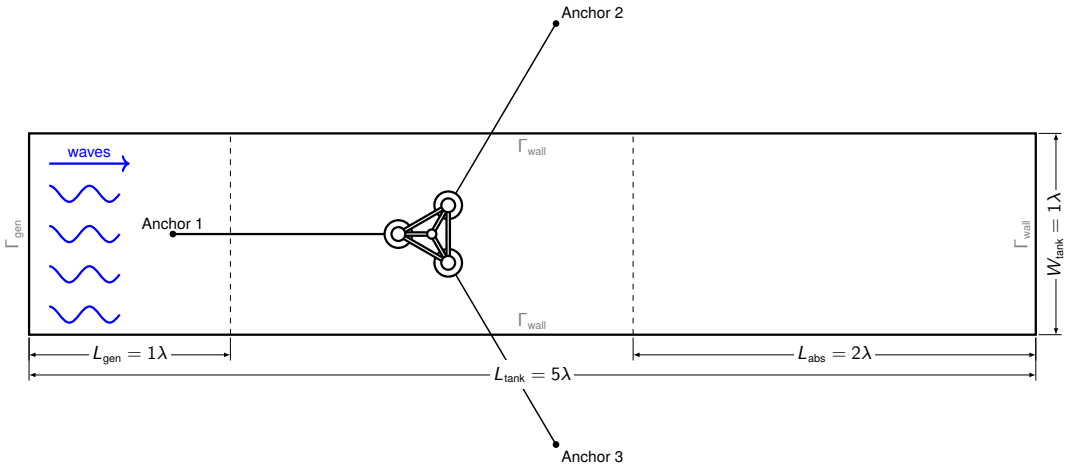


Figure V.37: Schematic representation of numerical domain (top-down view)

also used for numerical validation of a lumped-mass mooring model (MoorDyn) in [64] and floating wind turbine simulator [28]. Note that in the different references and reports cited above, all measurements are usually given in prototype scale, but that 1:50 model scale is used here in the text to reflect actual dimensions used experimentally that are also reproduced numerically.

In the numerical simulation presented below, the response of the platform is investigated, but not the response of the WT to wind loads. Therefore, the WT is not modelled numerically (i.e. only the geometry of the semi-submersible is represented) but its mass and inertia characteristics are incorporated in the model. As opposed to FSI simulations presented in section V.4 where 2D approximations could be used in order to reduce computational time, this test case must be modelled numerically in 3D because of to the geometry of the platform. Due to the sheer computational power needed for the simulation of a full-fledged 3D FSI case, only one experimental test case from [84] is selected for numerical investigation: the monochromatic wave case with height $H = 206.08\text{mm}$ and period $T = 1.71\text{s}$, giving a Fenton wavelength of $\lambda = 4.66\text{m}$. This is the most nonlinear monochromatic wave tested experimentally, making it arguably the most interesting case to investigate numerically as more dynamic effects in the response of the structure and mooring system can be expected.

For this experiment, the tank is filled with a water column of $h_{\text{mwl}} = 4\text{m}$, and the semi-submersible placed within it has a draft of 0.4m when the WT and mooring system are mounted on the structure. The size of the domain is optimised numerically to a total length of $L_{\text{tank}} = 5\lambda$, width of $W_{\text{tank}} = 1\lambda$, and height of $H_{\text{tank}} = 1.5h_{\text{mwl}} = 6\text{m}$.

	Symbol	Prototype no turbine	turbine	1:50 scale no turbine	turbine	unit
Mass	m	13,444,000	13,841,160	107.552	110.729	kg
Barycentre position above keel	—	5.600	10.110	0.112	0.202	m
Roll radius of gyration	$r_{I_{11}}$	23.910	0.478	31.610	0.623	m
Pitch radius of gyration	$r_{I_{22}}$	24.900	0.498	32.340	0.647	m
Yaw radius of gyration	$r_{I_{33}}$	32.170	0.643	32.170	0.643	m

Table V.9: OC4-DeepCwind semi-submersible platform characteristics.

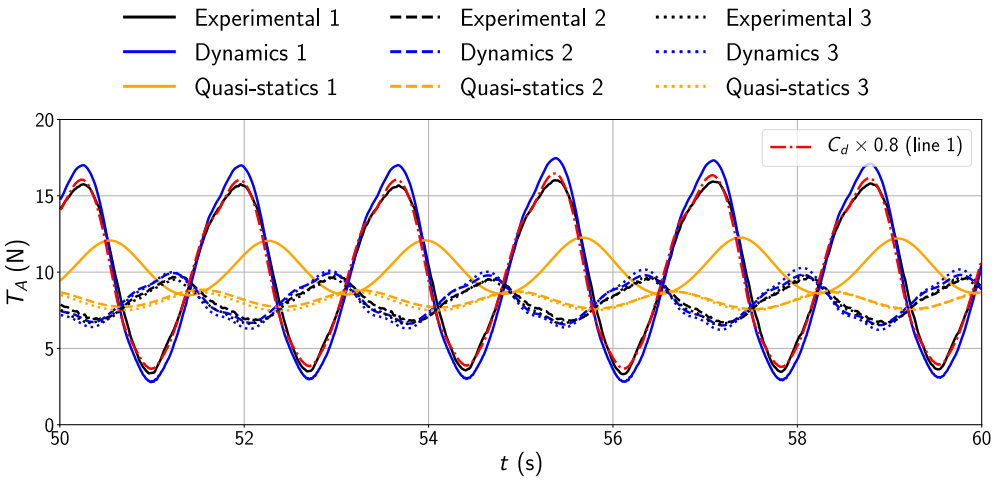
Property	Symbol	Prototype	1:50 scale	Unit
Length	L	835.500	16.710	m
Angle between lines	—	120	120	°
Fairlead radius	—	40.868	0.817	m
Fairlead depth	—	14.000	0.280	m
Anchor radius	—	837.600	16.752	m
Anchor depth	—	200.000	4.000	m
Equivalent diametre	d	133.760	2.675	mm
Nominal diametre	d_0	79.900	1.598	mm
Linear density	ρ_c	116.6	0.466	kg m^{-1}
Axial Stiffness	EA	753.600×10^6	6.029×10^3	N
Drag coefficient (normal)	$C_{d,n}$	1.080	1.080	—
Drag coefficient (tangential)	$C_{d,t}$	0.213	0.213	—
added mass coefficient (normal)	$C_{m,n}$	0.865	0.865	—
added mass coefficient (tangential)	$C_{m,t}$	0.269	0.269	—

Table V.10: Mooring system characteristics for the OC4-DeepCwind semi-submersible platform

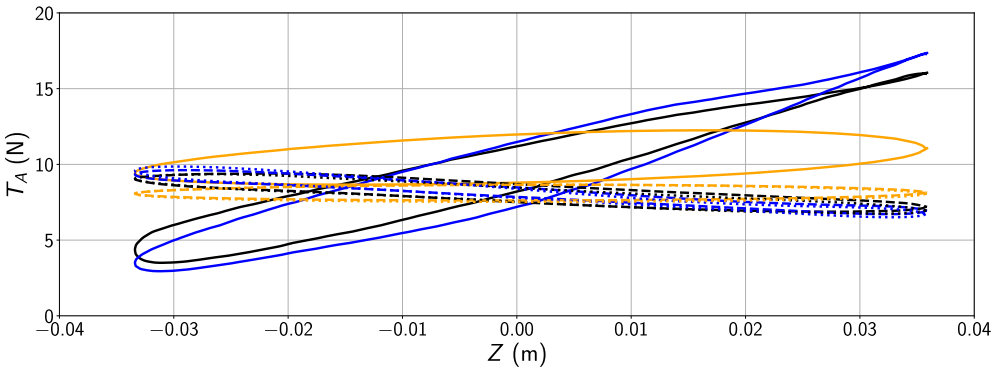
Waves are generated at the left boundary of the tank (placed at $x_1 = 0$) using the Fenton method with a generation zone spanning over 1λ from the generating boundary, while an absorption zone of 2λ is placed at the other end of the tank. The platform is positioned at $x_1 = 2\lambda$ away from the generating boundary, and in the centre across the width of the tank at $x_2 = 0.5\lambda$. Due to the cylindrical shape of the structure, minimal reflection is expected, so there is no absorption zones along the side walls of the tank. A snapshot of the simulation and schematic representation of the top-down view of the numerical layout are shown in fig. V.35 and fig. V.37, respectively.

The upstream mooring line (line 1) is placed parallel to the direction of wave propagation while the other lines (line 2 and 3) form an angle of 60° from the direction of wave propagation. Using the recorded experimental mooring line length of $L = 16.71\text{m}$, the numerical pretension calculated from the quasi-statics model is 8.33N. However, this does not match the experimental pretensions that vary according to the line considered. The line lengths are therefore adjusted numerically in order to match the recorded experimental pretensions: line 1 with a target experimental tension of 8.992N is adjusted to $L_1 = 16.674\text{m}$ (yielding 8.993N), line 2 and 3 with target pretensions of 8.540N and 8.520N are both adjusted to a length of $L_2 = L_3 = 16.699\text{m}$ (yielding 8.530N). Collision is enabled for the mooring line with the seabed. The seabed itself has a friction coefficient of 0.3, normal stiffness of $3 \times 10^6\text{Pa m}^{-1}$, and normal damping coefficient of 1.

Boundary conditions are set as free-slip on the tank bottom, front, back, and right boundaries Γ_{wall}, atmospheric on the top boundary Γ_{atm}, and unsteady two-phase flow inlet on the left boundary Γ_{gen}. No-slip conditions are applied on all boundaries of the semi-submersible $\Gamma_{\text{f}\cap\text{s}}$. The semi-submersible platform is discretised numerically with an unstructured triangular mesh of constant characteristic element size $h_e^{\text{semi}} = 0.01\text{m}$. The fluid domain is meshed using $h_{e0} = 0.02\text{m}$ up to a distance of $\pm\frac{H}{2}$ from the mean water level. This gives relatively good discretisation of the wave around the free surface with $h_{e0} < \frac{H}{10}$ and $h_{e0} < \frac{\lambda}{200}$. The mesh is coarsened gradually with a maximum increase of 10% in characteristic element size between adjacent elements (see fig. V.37 for a snapshot showing a slice of the mesh). Despite the spatial optimisation of the domain in terms of size and gradual coarsening of the mesh, this numerical setup leads to a large mesh with a total number of a total number



(a) Tensions over time (selected time window)



(b) Tensions as functions of heave (averaged over 40 periods)

Figure V.38: Tensions at fairleads for uncoupled OC4-DeepCwind semi-submersible simulation

of elements of $n_{\mathcal{E}} = 40{,}509{,}745$. Time-stepping for the fluid is controlled with CFL $= 0.5$ and, using validation results from section V.5.3.3, a conservative discretisation of 100 elements per line is used for each mooring with a fixed time step of $\Delta t_s = 1 \times 10^{-4}$s for the MBD solver. For this consequent run, 4,800 cores (i.e. processors) are used in parallel, distributed in 150 compute nodes each containing 32 cores (see Copper HPC characteristics in table V.3 for more information about the system). The fully coupled simulation presented below ran for a total 168 hours, leading to a total computational cost of 806,400 core hours.

■ V.6.2 Uncoupled Results

Before producing the fully coupled results, pre-validation of the mooring system is done using the displacement of the semi-submersible recorded experimentally. As this prestep only involves body and mooring dynamics, it is not computationally demanding and can therefore run on a simple laptop such as the one shown in table V.3.

Similarly to [64], the experimental displacement curves are filtered in order to be used reliably in the numerical setup. For this purpose, a low-pass filter with a cutoff frequency of 1Hz is used on the experimental signal (using 1:50 scale for displacements). A selected time frame of the tensions in the different lines over time is shown in fig. V.38a, and fig. V.38b shows the averaged tension over the averaged heave range over 40 periods. The quasi-statics model is also used against the experimental displacement for comparison purposes with the dynamic mooring model. The

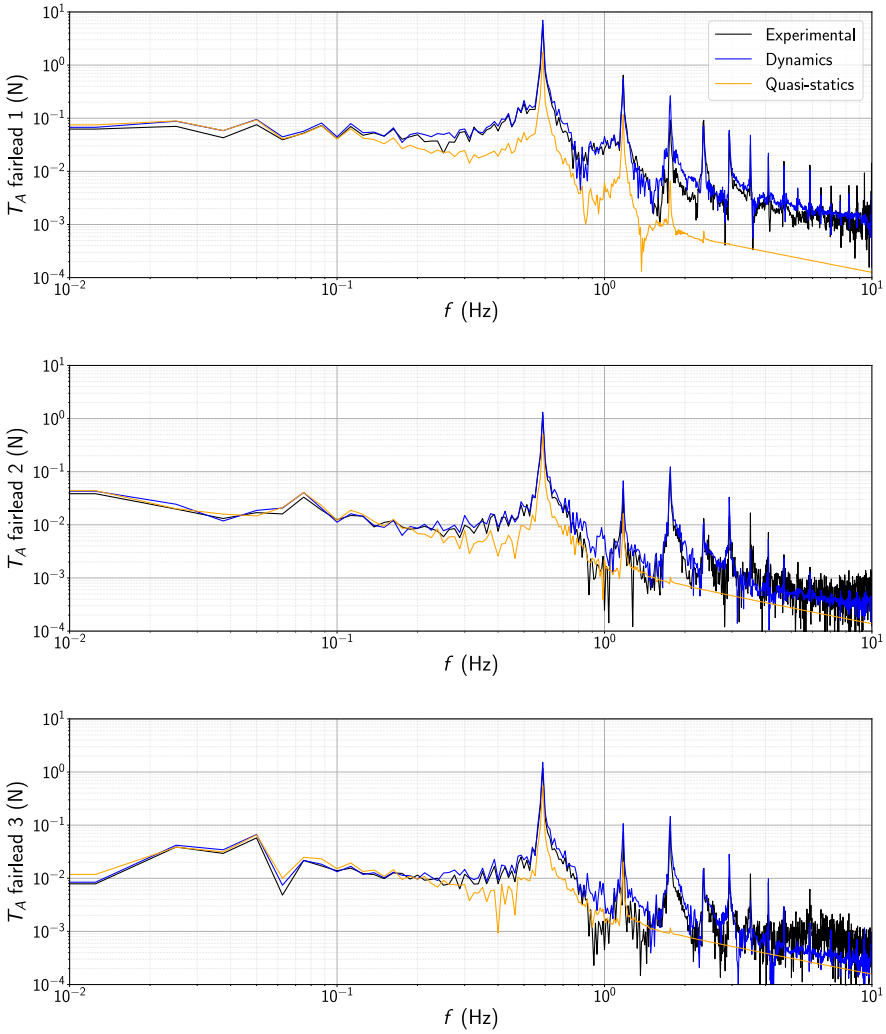
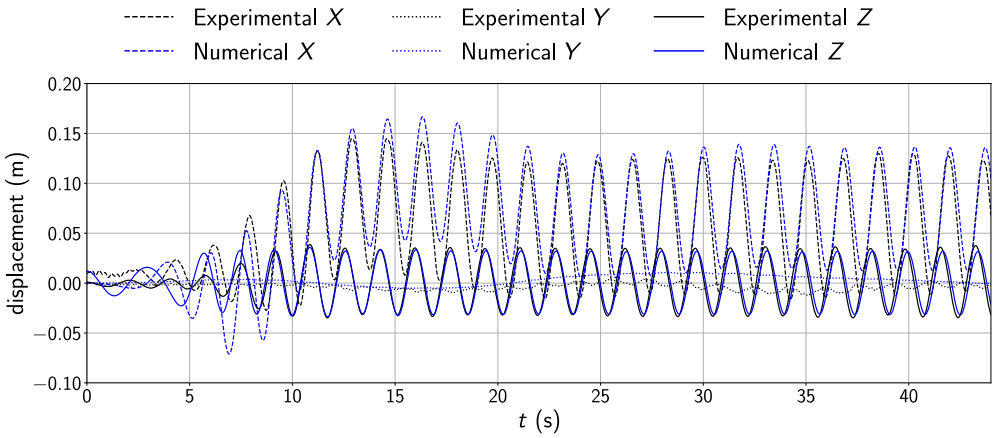
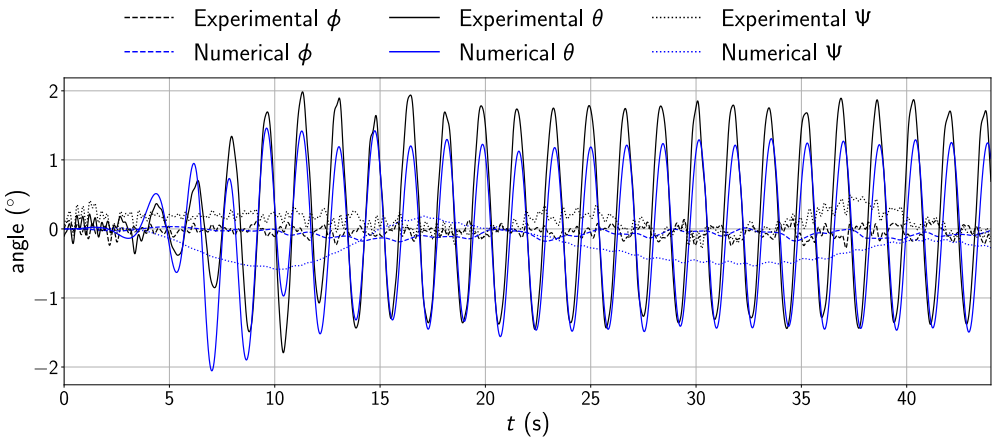


Figure V.39: Spectrum of tensions at fairleads for uncoupled semi-submersible validation case (over 40 periods)

resulting Fourier's transform spectra of each line from data for 40 wave periods are shown in fig. V.39. It clearly appears that the quasi-statics model significantly underestimates the response in tension on the mooring lines, with an averaged underestimation of 71.6% on line 1, 60.0% on line 2, and 57.8% on line 3. It is also out of phase by approximately 60° when compared to the experimental signal on every line. On the contrary, tensions from the dynamic mooring model show excellent agreement in terms of phase with the experimental tensions. However, it is more conservative than the experiment in terms of response with an overestimation of 14.9% for line 1, 12.9% on line 2, and 25.4% on line 3. This difference can be justified by uncertainties such as the empirical value of the drag coefficients, especially since the geometry of the mooring chain reported by [64] is different from conventional commercial chains. Note however that [64] obtained better agreement with experimental data on this particular uncoupled case with the drag coefficients of table V.10. Using the mooring dynamic model developed here, reducing the drag coefficient by 20% gives a better agreement with the experiment with a difference in response amplitude of 0.03% and a shift of $+0.4$N on line 1 (see fig. V.38a), 4.67% on line 2, and 6.70% on line 3. Despite this better agreement with experimental data when scaling down drag coefficients, coefficients from table V.10 are used in the



(a) Translational motion: surge X, sway Y, and heave Z



(b) Rotational motion: roll ϕ, pitch θ, and yaw Ψ

Figure V.40: Response of OC4-DeepCwind semi-submersible platform to wave loads in fully coupled simulation

fully coupled numerical simulation presented below for consistency when comparing numerical results with [64].

■ V.6.3 Fully Coupled Results

This full-fledged and coupled 3D FSI case uses all the models and tools that have been described and developed for this thesis (with the exception of the ALE mesh with monitor function): TPF module (NSEs and free surface), ALE mesh with elastostatics, rigid body motion, mooring quasi-statics and dynamics, added mass estimator model, and tools such as the particle localisation algorithm for retrieving the fluid velocity at mooring cable nodes. It is reminded here that this large simulation ran on 4,800 cores for 168h, and produced $T_{\text{sim}} = 44.5$s of numerical data. The numerical response of the OC4-DeepCwind platform is plotted against experimental response in fig. V.40 for all DOFs, where the raw experimental curves are shifted by -16.2s in order to have an equivalent distance between the structure and the first generated waves at $t = 0$ between the numerical and experimental signal. In fig. V.41, the numerical tensions at the different fairleads are plotted against experimental data.

It clearly appears that surge, heave, and pitch of the platform are in phase and of the same order of magnitude numerically and experimentally. Sway, yaw, and roll are minimal in both the experiment and the numerical simulation, due to the direction of the waves and the layout of the mooring system relative to it. The surge motion follows a

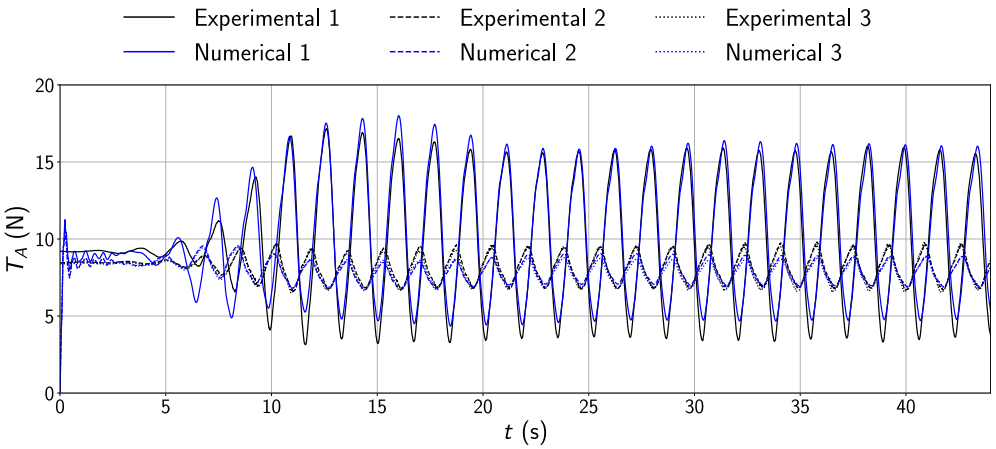


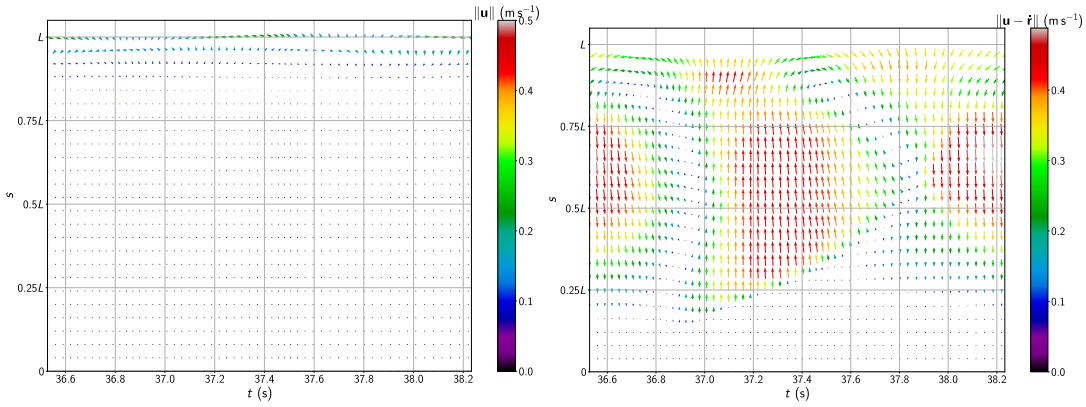
Figure V.41: Tensions at fairleads of OC4-DeepCwind semi-submersible platform in fully coupled simulation

similar trend numerically and experimentally, with the structure displaced from its static equilibrium position before restoring forces in the mooring system bring it back to a new mean position. The mean surge offset remains constant after approximately $t = 25$s in both cases with values of $\bar{X}^{\text{num}} = 0.073$m numerically and $\bar{X}^{\text{exp}} = 0.062$m experimentally. In fig. V.40a, the heave signal has been translated by $+0.012$m as the structure is not at equilibrium numerically at $t = 0$. This error of 3% in draft can be due to a difference in mass provided by [84] and the actual mass of the whole system (including moorings) to reach the theoretical draft of 0.4m, but also numerical errors from spatial discretisation of the structure that could lead to a slightly different volume.

The responses of the platform to wave loads and of the mooring system are discussed below by considering the last ten full periods, where the mean values of the signals are stabilised. In terms of hydrodynamic response of the structure itself, heave and surge are underestimated numerically by 7.55% and 10.27% respectively. For the mooring system, it appears that all numerical signals of fairlead tensions are in phase with their experimental counterparts. Mean tensions on all lines have a difference of less than 4% from the experimental data. The maximum and minimum tensions are slightly conservative numerically on the upstream line (i.e. line 1), but the amplitude of the response is underestimated by around 7.44% when compared to the experimental response. Note that this difference is of the same order of magnitude as the difference in hydrodynamic response and, because both numerical responses are slightly lower the experiment, this can potentially mean that numerical wave conditions are slightly different from their experimental counterpart. On downstream lines (i.e. lines 2 and 3), the responses in tension are underestimated by 30.70% and 34.49%. Although these seem to be large relative differences, the absolute differences are ≈ 0.9N, which is of the same order of magnitude as the absolute difference in response between the upstream line and experimental data. It is also worth mentioning that the amplitude of the responses in tension of downstream lines are approximately one order of magnitude lower than the upstream line, making the latter arguably more important to resolve accurately. Using the particle-localisation algorithm, the absolute fluid velocity retrieved along the upstream line using the particle localisation algorithm is shown in fig. V.42a over a full period of oscillation. It appears, as could be expected, that absolute fluid velocity is only significant when relatively close to the fairlead where wave kinematics are the strongest. The resulting velocity of the cable along the upstream line relative to the fluid velocity is shown in fig. V.42b for the same period of time. It appears that the line is never

fully lifted during a full cycle, with at least approximately $0.25L$ remaining on the seabed at all time. It also appears that the largest relative velocities are found between the centre of the line and a distance of $0.25L$ away from the fairlead (i.e. at $0.5L < s < 0.75L$), which consequently corresponds to the part of the line with the strongest drag forces. This implies that, in this particular case, the anchor does not experience any force induced by the mooring line as one quarter of it lies on the seabed at all times, and that largest contributions to mooring line damping are from the third quarter of the line.

While results obtained here clearly show the capability of the FSI framework developed in this thesis for simulating complex moored floating structure problems, further investigation with a finer mesh, a larger numerical domain, and a finer time discretisation could be worthwhile. Although it is likely that it would lead to an even better agreement with the experiment, this has been considered beyond the scope of this thesis due to computational resources available and time constraints. Nevertheless, it must be noted that the extensive validation of each individual and coupled components established in previous sections allowed for a reliable selection of numerical parameters. It has been consequently shown here that the FSI framework can produce numerical results that are realistic in terms of hydrodynamic response and line tensions experienced by a moored floating structure, even in the presence of highly nonlinear dynamic effects. Furthermore, these results for the fully coupled models compare better than the coupled numerical results previously obtained in the literature when compared to experimental data. In particular, the amplitude of response in heave of the semi-submersible is underestimated numerically by 7.55% with the current framework and was in phase with the experimental signal, while the coupled results from [64] show an underestimation of 26% with a leading phase of $40°$. Errors of the same order of magnitude are present in mooring line tensions, particularly on the upstream line, which is arguably the most critical line of the system as it is where the tensions are the highest. While it is not clear why this happened in [64], this relatively large difference could be due to the fact that potential flow theory was used in [64] to describe the hydrodynamics of the structure while CFD is used here, allowing for the resolution of highly nonlinear processes.



(a) Fluid velocity

(b) Cable velocity relative to fluid velocity

Figure V.42: Velocity along upstream mooring line (i.e. line 1) of the OC4-DeepCwind platform under regular wave loads with $H = 206.08$mm and period $T = 1.71$s, for one complete oscillation

Notes: velocity vectors are in X-Z plane; s is the position along unstretched cable (from anchor)

CHAPTER VI

Discussion

In this chapter, we discuss the different aspects of the high-fidelity Fluid–Structure Interaction (FSI) framework for moored floating bodies that has been developed for this work, as well as the different challenges, solutions, results, and limitations ensuing from it. This discussion is divided in three sections:

- numerical simulations, verification and validation cases, their findings, and details about the main challenges encountered are reviewed in section VI.1,
- general remarks about software development focusing on details about communication between models and parallelisation of the code are presented in section VI.2,
- and the main limitations of the framework are discussed in section VI.3.

For a global conclusion, see chapter VII.

VI.1 On Simulation Results and Numerical Challenges

The different simulations of chapter V provide a thorough verification and validation of all of the main models and tools introduced in chapter III and chapter IV, and highlight the capabilities and limitations of the FSI framework developed here. These results for individual models and for coupled problems show the capabilities and limitations of the framework and allow the numerical modeller to confidently use it in a variety of scenarios.

Firstly, the fluid dynamics model encompassing the Navier–Stokes Equations (NSEs) and free surface tracking models showed that nonlinearities of the flow that were present experimentally were successfully captured. In section V.4.3, vortices around sharp corners were well defined and their influence on the response of the floating structure was well represented when compared to experimental data and other nonlinear numerical models. In section V.4.4, nonlinear wave–wave interactions from spectral waves — specifically focused waves using the Joint North Sea Wave Project (JONSWAP) spectrum — that led to a phase difference between the analytical and experimental signals were also present in the Computational Fluid Dynamics (CFD) model, achieving a better agreement with the experimental data rather than the analytical solution. Furthermore, highly nonlinear green water events such as overtopping on a moving floating structure were represented successfully, as can be seen qualitatively in fig. V.23. All these complex Two-Phase Flow (TPF) features were robustly represented when using moving domains (e.g. for structures with mesh-conforming boundaries moving within the fluid domain). In terms of FSI, good agreement

was obtained over all the numerical simulations presented in section V.4 against experimental data. The numerical response of floating bodies using fully explicit, partitioned Arbitrary Lagrangian–Eulerian (ALE) coupling was also in good agreement with numerical results from fully implicit, monolithic models. The validation cases covered a relatively wide range of scenarios including: free oscillation of a floating body in heave (section V.4.1), free oscillation in roll (section V.4.2), wave-induced oscillation across a range of wave frequencies (section V.4.3), and the response under extreme wave loads (section V.4.4). This variety of applications clearly showed the extent of the versatility of the framework for the robust simulation of FSI for floating structures. Therefore, coupling the TPF, mesh motion, and body dynamics models together with a fully partitioned ALE scheme achieved both an accurate response of floating bodies under hydrodynamic loads as well as a realistic description of the fluid dynamics.

For simulations involving moving floating structures, numerical challenges were encountered with the ALE formulation and mesh deformation. When relatively large mesh motion occurred, the framework was generally robust and retained good accuracy. Investigation on moving domain techniques and mesh motion approaches was nonetheless an important aspect of building this high-fidelity framework. The elastostatics ALE mesh model, indispensable in the current framework for including the motion of structures with mesh-conforming boundaries, was suitable for all the applications presented here: no mesh entanglement or simulation-breaking mesh deterioration was encountered. Even though it was not featured in the simulations of chapter V, it is worth mentioning that on significantly longer FSI simulations involving nonlinear motion of a floating body, it was observed that the mesh can become highly deteriorated or tangled over time. This happens mostly on FSI simulations of floating bodies with several Degrees of Freedom (DOFs) (e.g. combined heave, sway, roll) and under wave-induced oscillation after a relatively high number of periods (e.g. hundreds of wave periods). In the method implemented here, the mesh is deformed to its new configuration from its configuration at the previous time step, and this can lead to uneven deformation of the mesh over time. When this occurs, the exact original configuration (i.e. at $t = 0$) is generally not recoverable through mesh elastostatics from a deformed position, even after placing the body back to its original position. Note however that when there is only one DOF, the mesh can usually recover its original configuration as any movement is then reversible. A way to deal with mesh deterioration from nonlinear motion of conforming mesh boundaries is to store the original (i.e. initial) configuration of the mesh and deform the mesh from this original configuration to the configuration of the current time step, instead of deforming it from the configuration from the previous time step. Deforming from the initial configuration is known as total ALE, as opposed to updated ALE [5]. The total ALE approach however leads to i) the need for more memory in order to store the original configuration of the mesh and ii) mapping quality issues due to large differences between reference and spatial domain [5]. Another more general remedy to this mesh deterioration problem is to perform mesh smoothing operations such as periodical Laplacian smoothing. However, when the mesh uses non-uniform refinement (such as in fig. V.1), Laplacian smoothing steps tend to render the mesh more uniform globally (i.e. highly refined areas will become coarser, and vice-versa), which is undesirable in some cases (such as when intending to keep a highly refined mesh around the free surface). This homogenization of the mesh is also irreversible when using the elastostatics mesh motion on its own. This is where combining the elastostatics and monitor function models show a clear advantage, as it allows for retaining a highly refined mesh around mesh-conforming boundaries that can also be adjusted dynamically around

non-conforming boundaries. Even if Laplacian smoothing is used periodically to recover the quality of the mesh, the monitor function model ensures that refinement is kept as intended by re-adjusting the volume of the smoothed elements to the desired value. It is important to add that, when mesh-conforming boundaries in motion are featured in a simulation, the monitor function model cannot handle the mesh deformation by itself in a reliable manner and that, therefore, the elastostatics model has to be used beforehand in any case. The monitor function model could however be safely used on its own when moving boundaries are implicit, such as for moving solids using Immersed Boundary Method (IBM). This investigation on mesh motion techniques is ongoing work as it could improve results overall, on top of allowing for longer simulations to retain a high-quality mesh.

Other numerical challenges arose from the added mass effect that was encountered in several simulations using partitioned fluid–structure coupling. The FSI simulations of section V.4 were all fully stable through the use of a stabilisation scheme for explicitly partitioned coupling scheme that was developed for the purposes of this thesis. As explained in previous chapters, instability from the added mass effect is commonly encountered when dealing with floating bodies that have a low density relative to the fluid density, and/or that have elongated geometries (see section V.3.1 for relevant examples). Unconditional instability of the explicit coupling between the TPF and the Multi-body Dynamics (MBD) models due to the added mass effect was triggered in several of the simulations presented here, notably in section V.4.4 and section V.6. The main challenge was to stabilise the coupling while keeping a high computational efficiency and without affecting the overall response of the floating body in any significant way (i.e. by retaining acceptable accuracy). In order to achieve this with a fully explicit coupling scheme, the added mass of the rigid body was first estimated by solving the Partial Differential Equation (PDE) system described in section IV.2.3.2 (and verified in section V.3.2) to then be used as a penalty term acting on the acceleration of the body as described in section IV.2.3.1. It has been shown conceptually in section V.3.3 that, unless a relatively coarse temporal discretisation was used, unconditional stability of simulations prone to the added mass effect could be recovered with minimal effects on the response of the body, and this was further demonstrated in section V.4.1. When coupled with the added mass estimator model, the stabilisation scheme was successfully validated on realistic cases that produced numerical responses closely matching experimental data. In section V.4.4, due to the geometry and density of the floating body, the simulation was highly sensitive to the added mass effect and failed only after a few time steps when no stabilisation scheme was used. Accurately estimating the added mass and using this estimation in the non-iterative stabilisation scheme led to a stable simulation and resulted in a numerical response with a level of accuracy comparable to an inherently stable monolithic scheme. The non-iterative stabilisation scheme was also successfully applied to the 3D case of a floating semi-submersible platform for an Offshore Wind Turbine (OWT) presented in section V.6. When compared to iterative coupling schemes that require solving the fluid–solid problem several times in order to converge to a stable solution, the non-iterative added mass stabilised coupling scheme presented here dramatically increases computational efficiency and code modularity. These stabilised results clearly showed that explicitly partitioned coupling schemes can be used in a stable and accurate manner for the numerical simulation of engineering applications involving floating structures moving within a high-density fluid.

In terms of mooring simulations, the static / quasi-static and dynamic models developed and used for this work were in excellent agreement regarding tension in mooring lines at static equilibrium. This cross-validation of two

completely different approaches confirmed that static tensions are accurately estimated by both models, as was further validated by comparisons with experimental data in section V.5.3.2. It also allowed the static / quasi-static model to calculate the nodal positions and direction gradients necessary for the initial conditions of the dynamic mooring model using gradient deficient Absolute Nodal Coordinate Formulation (ANCF) elements. From the mooring damping investigation of section V.5.3.3 and the full-fledged 3D case of the OWT platform in section V.6, it clearly appeared that nonlinear effects were well captured by the dynamic model. It also showed that the quasi-static model clearly underestimated the response in tension (e.g. see fig. V.38a) due to its inability to model damping of the mooring line. This further confirmed that a fully dynamic mooring model is necessary for estimating tension in the mooring system of non-static simulations in a more accurate manner. The velocity of the fluid along mooring cables was also retrieved successfully by tracking the cable nodes in the fluid mesh through the particle localisation algorithm of section IV.4.3. This was then used to accurately apply drag and inertia forces along the cable.

The moored floating semi-submersible case of section V.6 showed that the FSI framework works robustly as a whole when all models are coupled together. It also showcased the scalability of the numerical implementation as it was heavily parallelised. By achieving a good agreement between experimental and numerical data, the framework proved its viability in realistically simulating the response of complex systems in 3D. The results obtained with the uncoupled approach (mooring dynamics model with prescribed motion from experimental data) for the particular wave conditions studied were slightly more conservative than the experimental data and the numerical results from a lumped mass model developed and used by [64]. However, the fully coupled results using the framework developed here showed significantly better agreement with experiment than the coupled results of [64]. This realistic validation case with multiple mooring lines and floating structure with all 6 DOFs further highlights the suitability of the framework to simulate complex multiphysics problems.

Overall, the various numerical simulations and results have shown that the numerical framework developed here has the ability to simulate highly nonlinear processes of fluid flow, floating structure dynamics, mooring dynamics, and all of them coupled together. Its main drawback remains the computational power needed for performing large scale, highly refined 3D simulations. While this is currently the price to pay for producing reliable high-fidelity FSI results, it is expected that, with the continuing trend in increase of computational power availability, such simulations will become more affordable in the close future.

■ VI.2 On Software Development

We discuss briefly some general thoughts on software development aspects of the FSI framework, without going into the finer details. As mentioned before, the TPF capabilities of the framework were implemented using the Proteus toolkit for solving PDEs with Finite Element Method (FEM), while the MBD and mooring dynamics capabilities were implemented using the Chrono library. Even though the FSI framework uses both of these libraries, it is entirely built upon the Proteus toolkit, with Chrono being a dependency when compiling the source code. At the time of writing, the code repository of Proteus is available online at `https://github.com/erdc/proteus`, where contributions made by the author for building the FSI framework presented here are recorded. The majority of the software development work that has been done for this research is therefore available online and can be used to reproduce

any of the simulations from chapter V.

When building this FSI framework, the Chrono library was mostly accessible using C++ (with relatively limited high-level Python access), while Proteus was accessible in Python for high-level access and C++ for low-level access and efficiency. Both Python and C++ have therefore been extensively used for the implementation of the different models and tools presented in this thesis. On top of developing the different models themselves, interfacing between these two programming languages and libraries was a major aspect of software development. The main inter-model communication can be seen conceptually in the workflow diagram of fig. IV.1. Python-C++ communication was mostly achieved with Cython, sharing objects or pointers between the two languages. When using Cython, only one object needs to be created and stored in memory to be directly accessed from either Python or C++. As a concrete example, retrieving fluid velocity solutions at mooring cable nodes to calculate external forces along the cable was achieved with inter-model and inter-language communication. A single fluid velocity array of dimensions $n_{\mathcal{N}} \times 3$ (with $n_{\mathcal{N}}$ the number of nodes of the cable) can be created using Cython (thus making it accessible from Python), populated by retrieving the solution for the fluid velocity solution from Proteus at the known coordinates of the cable nodes, to then be passed in a function or accessed directly through pointers in C++ in order to calculate the drag and inertia forces in Chrono. Cython was also used for the work presented in this thesis to increase the computational efficiency of several parts of the code in Proteus that were originally designed in Python, such as for applying typical boundary conditions efficiently and for relaxation zones where looping over a large number of mesh elements can significantly affect performance when using Python instead of C++. Note that, at the time of writing, Cython was the main way used in Proteus for communication between high level (Python) and low level (C++) code internally, while Chrono started using SWIG to wrap some of its C++ code and make it accessible from Python. Pointers of SWIG objects can also be accessed and manipulated in Cython, making it possible to use the best of both worlds.

Another major recurring aspect that had to be considered for the implementation of the methods presented here was parallelisation of the code. Most of the cases presented in chapter V had to run in parallel with a relatively large number of processors. This is due to raw computational power needed to solve large FSI problems, and the memory needed to store problem variables. In simple terms, solving a large problem in parallel is done by breaking it down into smaller pieces in order to reduce both the power and the memory needed to solve each subproblem. This can be achieved on a large scale when using High Performance Computing (HPC), where the resolution of large problems is achieved through distributed memory. In an HPC architecture, or computer cluster, nodes (or computers) are all connected and can communicate with each other in order to perform a global task in parallel. Within each node, there is a certain number of cores (or processors) that are also able to communicate with each other, further dividing the global problem. Each core can be referred to as a rank, which corresponds to its unique identification number within a global computer cluster. When running a CFD simulation in parallel, the fluid mesh is divided into as many subdomains as there are ranks, each ideally owning approximately the same number of mesh elements. Running a large CFD simulation on a single rank or only a few ranks (such as on a laptop or typical desktop computer) can lead to a very large number of mesh elements per rank, which renders the simulation prohibitively slow to run in the best case, or simply throws an Out of Memory (OOM) error in the worst case. All models presented here take advantage of parallelisation to scale the global problem as efficiently as possible, with the exception of the MBD

solver that, at the time of writing, was not parallelised when using FEM and collision detection. Therefore, body and mooring dynamics were always solved on a single rank, regardless of the total number of ranks used for the simulation (which can be several thousand as in section V.6). Solving body dynamics in serial was not a major issue as it is usually one or several orders of magnitude faster to solve than the parallelised fluid problem. Note however that for more demanding MBD problems (e.g. highly discretised mooring cables or collision detection for a large number of bodies), this can become an issue as all ranks must wait for the the MBD problem to finish its calculation in series before carrying on with the simulation. Furthermore, while Proteus allows for automatically parallelising PDE problems, several of the models presented in this thesis still had to be designed carefully in order to work as intended in parallel. For example, the moving mesh model with monitor function needs to perform a global domain integral of the area function to scale the monitor function accordingly, and this must be achieved by gathering the local sum of each rank as a pre-step and broadcasting the resulting global sum back to all ranks before solving the PDE. Additionally, rank-to-rank communication can become intense when transforming the mesh because each mesh node is owned by a unique rank but, when it is displaced during pseudo-time stepping, it can leave the spatial domain owned by this rank and therefore needs to retrieve the PDE solution from the spatial subdomain owned by another rank. This information must then be communicated back to the original rank when pseudo-time stepping is over in order to displace the mesh node from the owning rank. This also applies to mooring nodes moving within the fluid mesh (as they also use the particle-localisation algorithm), but this is usually less intense as the number of cable nodes is typically several orders of magnitude less than the number of fluid mesh nodes. Another example is Laplacian smoothing of the mesh where, if done several times in a row, communication between ranks becomes necessary to update the position of shared nodes from overlapping parallel layers.

In summary, the core software development work undertaken by the author in terms of implementation and improvement of raw capabilities of the FSI framework include: the coupling of fluid (Proteus) and body (Chrono) dynamics, wave generation and absorption with relaxation zones, models for ALE mesh motion, model for estimating the added mass of arbitrarily shaped structures, models for mooring statics and dynamics, as well as general software engineering to respond to the underlying aims of making the framework more computationally efficient and making the code more accessible and navigable from both a user and developer perspective. Additionally, it is worth noting that, at the time of writing, new approaches and capabilities beyond the scope of this thesis were being developed within Proteus by various developers, including: IBM, mesh adaptivity, sedimentation, turbulence, shallow water equations, and improvement over free surface tracking methods. Due to its open-source nature, the freely accessible high-fidelity FSI framework that has been developed and presented here is also likely to undergo further development and improvement.

VI.3 On Limitations of the FSI Framework

One of the main limitations of the framework presented here (which is common to all high-fidelity FSI models) is the relatively high computational cost of large simulations. This was particularly apparent in section V.6 where 806,400 core hours were used to complete a single simulation. This limitation is still prominent despite the careful selection of the techniques presented here according to their computational efficiency when compared to their alternatives.

For example, the non-iterative added mass stabilisation scheme between the fluid and the solid solvers avoids any additional iteration of either solver per time step, inducing significant computational savings when compared to implicit schemes. Computational efficiency is also the reason for using gradual mesh refinement on most simulations presented here. Many other aspects of the code can be optimised, sometimes simply by converting code written in high-level interpreted programming languages to low-level compiled languages (as mentioned in the previous section), or by improving the efficiency of the algorithms themselves. For the latter, an example that has been implemented in the work presented here is the particle localisation algorithm by element boundary intersection presented in section IV.4.3, allowing a computationally efficient way to look for elements containing a set of global coordinates.

Another limitation is that the added mass estimator model of section IV.2.3.2 is technically limited to rigid bodies and cannot be applied as it stands to flexible bodies. This limitation is explained by the fact that a PDE is solved for each DOF of the structure by applying the same unit acceleration on all boundaries of the body. In other words, all points on the boundaries of the body are assumed to accelerate at the same rate when solving the PDE. This is not necessarily the case when boundaries deform relative to each other over time, as is the case for flexible bodies. An additional limitation occurs when several moving rigid bodies are close to each other, as the effect of one body on the other would not be properly taken into account with the current implementation of the added mass estimator. Even though it is possible to estimate the added mass for each body separately by considering that the other bodies are fixed in space when solving for the added mass model, the actual value of the added mass can change as bodies move in different directions at the same time. Simply considering all bodies in the domain to be moving at the same unit acceleration for each DOF when solving the PDE is also not an appropriate solution regarding the effect of the bodies on each other's added mass as, again, they can technically move in any direction independently. For example, if two bodies are moving towards each other, this has the effect of increasing the value of their respective added mass due to the fluid being essentially compressed between the two bodies. A potential solution is to approximate the effect of the acceleration of other bodies on the added mass of a given body by using their last known acceleration for the boundary conditions along their boundaries (assuming that the change from the last time step to the new time step is negligible). This is considered as further work to be investigated to improve the added mass estimator model to more general cases, but it is beyond the scope of this thesis as the current added mass stabilisation scheme has proven to be satisfactory for all cases presented here featuring single floating bodies. Note also that this limitation of the added mass estimator model can be circumvented when using the non-iterative coupling scheme of section IV.2.3.1 by introducing an overestimated added mass value. Furthermore, using the added mass estimator model to calculate the added mass of flexible bodies as if they were rigid bodies technically leads to a conservative added mass value. This is due to the fact that flexible bodies "absorb" some of the instabilities from the added mass by deforming in a direction opposite to the hydrodynamic loads. As mentioned previously, the added mass value can be overestimated in the non-iterative stabilisation scheme and still yield accurate results.

CHAPTER VII

Conclusion

The design, development, and numerical implementation of a complete framework for high-fidelity Fluid–Structure Interaction (FSI) simulations of moored floating bodies have been detailed in this thesis. Extensive verification and validation on all aspects of this computational framework also showed that it can reliably approximate the responses of real-world floating structures and their mooring systems under realistic environmental loads. The main models and tools that have been implemented and coupled together for building the framework are listed as follows:

- Partitioned Arbitrary Lagrangian–Eulerian (ALE) coupling schemes between for FSI, with the highly modular and computationally efficient explicit Conventional Serial Staggered (CSS) scheme as the main approach used;
- Mesh motion techniques for conforming (elastostatics) and non-conforming (monitor function) boundaries returning mesh velocity necessary for the ALE formulation of the governing equations applied on the fluid mesh;
- Wave generation tools for regular (linear and nonlinear) and spectral waves (including focused waves);
- Relaxation zones for the absorption of transmitted and reflected waves from obstacles;
- Added mass estimator model for arbitrary geometries using pressure differential field, allowing for the dynamic estimation of the added mass of a moving structure;
- Added mass stabilisation scheme for partitioned, explicit coupling schemes that are otherwise prone to unconditional instability due to strong added mass effect;
- Mooring statics / quasi-statics module for multi-segmented and elastic cables that can be used for preliminary design stages of mooring systems and for initial conditions of dynamic mooring models;
- Mooring dynamics model based on the gradient deficient Absolute Nodal Coordinate Formulation (ANCF) for beam elements, using two-way (full) coupling with the structures and one-way coupling with the fluid;
- Efficient particle localisation algorithm for retrieving solutions from models using the fluid mesh, such as fluid velocity for applying drag forces at mooring cable nodes non-conforming to the fluid mesh.

Altogether, the above can be used for highly nonlinear FSI simulations of moored floating structures. The suitability of FSI framework was clearly established through the several engineering applications considered in this thesis, successfully fulfilling the aims and objectives set for this research effort. Furthermore, the framework has shown: its robustness through its capacity in effectively handling highly unstable cases that are prone to strong added mass effects, as well as handling relatively large mesh deformations; its versatility with the variety of FSI

simulations ranging from simple free oscillations of floating structures up to their response to a range of nonlinear monochromatic wave frequencies and extreme wave loads; its scalability with the successful simulation of large 3D cases combining all aspects of the framework and parallelised with several thousands of cores (leading to simulations lasting for hundreds of thousands of core hours); its modularity with the selection and use of appropriate methods such as fully partitioned non-iterative fluid–structure coupling schemes; and, in relative terms, its computational efficiency through the selection and development of an efficient set of computational tools and techniques. The reader is reminded here that all research and simulations presented in this thesis are fully reproducible after building the Proteus source code that is available online at `https://github.com/erdc/proteus`, where software development contributions made by the author also appear. Published work arising from this thesis is listed after this conclusion chapter, and more articles related to this research are in preparation.

The quest for ever greater efficiency and accuracy for high-fidelity FSI models is however boundless, as many aspects can always be improved upon. For example, computational efficiency for the fluid problem can be increased by allowing for gradual coarsening of the mesh along the free surface in areas where the description of the fluid does not need to be accurate, such as in absorption zones. This requires further work on the free surface description model as such coarsening along the air-water interface is not possible without introducing spurious fluid velocities due to the dependence on the mesh element size to smooth the Volume of Fluid (VOF) field in the current implementation. Furthermore, mesh motion models such as the monitor function model that was featured and numerically implemented here can be used to dynamically refine around the free surface instead of using a large refined band covering all areas where the free surface is expected to evolve. Concerning the added mass estimator model, further work is needed to extend it to multi-body problems or flexible bodies. In terms of mooring modelling, various additional aspects can be implemented such as nonlinear stiffness along the cable (varying with tension), line breaking mechanisms, and anchor movement or dislodgement mechanisms. The verification and validation of the current mooring model could be also extended for highly nonlinear events such as shocks in cables. Note that, as the Multiphase Flow (MPF) and Multibody Dynamics (MBD) solvers are fully partitioned, it is also possible to extend the capabilities of the existing framework by modifying or replacing one of the solvers without affecting the coupling strategy. The non-iterative added mass stabilisation scheme for explicitly partitioned fluid–structure coupling can also be applied completely independently of the chosen solvers.

The FSI framework has shown its reliability for simulating complex floating body simulations using a fully partitioned ALE coupling strategy. Mooring dynamics using gradient deficient ANCF elements has also shown its suitability for describing dynamic effects accurately in the applications considered here. Coupled together, the models have produced numerical responses comparable to real-world experiments and to inherently stable numerical models. Due to the variety of successfully validated applications presented in chapter V, the framework developed here can be robustly used for the simulations and investigation of engineering cases involving moored floating bodies under typical or extreme environmental conditions. Its main limitation – computational intensity – is likely to become gradually less of an issue with ongoing advances in computer science. This high-fidelity FSI framework for the numerical simulation of moored floating bodies is already an economical and generally more flexible alternative to physical testing for the production valuable data in the context of realistic offshore engineering applications.

Published Work

[1] Tristan de Lataillade, Aggelos S. Dimakopoulos, and Chris E. Kees. “A Coupling Strategy for Modelling Moored Floating Structures”. In: *Advanced Numerical Modelling of Wave Structure Interaction*. CRC Press, 2020 (in press).

[2] Tristan de Lataillade, Aggelos S. Dimakopoulos, Chris E. Kees, Lars Johanning, David Ingram, and Tezdogan Tezdogan. “CFD Modelling Coupled with Floating Structures and Mooring Dynamics for Offshore Renewable Energy Devices using the Proteus Simulation Toolkit”. In: *Proceedings of the 12th European Wave and Tidal Energy Conference*. 2017.

[3] Aggelos S. Dimakopoulos, Tristan de Lataillade, and Chris E. Kees. “Fast random wave generation in numerical tanks”. In: *Engineering and Computational Mechanics* 172.1 (2019), pp. 1–11.

[4] Giovanni Cozzuto, Aggelos S. Dimakopoulos, Tristan de Lataillade, Pedro Otinar Morillas, and Chris E. Kees. “Simulating Oscillatory and Sliding Displacements of Caisson Breakwaters Using a Coupled Approach”. In: *Journal of Waterway, Port, Coastal, and Ocean Engineering* 145.3 (2019).

[5] Aggelos S. Dimakopoulos, Tristan de Lataillade, Branoc Richards, and Chris E. Kees. “Advanced tools for modelling fluid interaction with coastal and marine structures”. In: *Structures in the Marine Environment (SIME)*. 2019.

[6] Aggelos S. Dimakopoulos, M. Sklia, Jason H. Collins, Chris E. Kees, and Tristan de Lataillade. “Advanced Wave Generation Systems for Numerical Modelling of Coastal Structures”. In: *Coastal Structures*. 2019, pp. 712–722.

[7] Chris E. Kees, Giovanni Cozzuto, Aggelos S. Dimakopoulos, and Tristan de Lataillade. “Application of the Proteus Toolkit to Marine, Coastal, and Riverine Engineering Problems”. In: *Computational Methods in Water Resources XXII*. 2018.

[8] Aggelos S. Dimakopoulos, Giovanni Cozzuto, Tristan de Lataillade, Chris E. Kees, and Giovanni Cuomo. “Numerical modelling of caisson breakwaters under sliding and overturning motion using the Proteus toolkit”. In: *Coasts, Marine Structures and Breakwaters 2017*. 2018, pp. 1039–1050.

[9] Aggelos S. Dimakopoulos, Giovanni Cozzuto, Tristan de Lataillade, Chris E. Kees, A. Alderson, and S. Richardson. “Modelling fully coupled fluid structure interaction using the computational toolkit Proteus”. In: *Australasian Coasts & Ports 2017: Working with Nature*. 2017, p. 917.

[10] Aggelos S. Dimakopoulos, Giovanni Cozzuto, Tristan de Lataillade, and Chris E. Kees. “Towards High Fidelity Numerical Wave Tanks for Modelling Coastal and Ocean Engineering Processes”. In: *American Geophysical Union (AGU)*. 2017.

[11] Chris E. Kees, Manuel Quezada de Luna, Alvin Zhang, Milad Rakhsha, Tristan de Lataillade, and Aggelos S. Dimakopoulos. “Open Source Tools for Adaptive Simulation of Fluid-Structure Interaction Processes”. In: *American Geophysical Union (AGU)*. 2017.

Bibliography

[1] A. M. Afonso, M. S. N. Oliveira, P. J. Oliveira, M. A. Alves, and F. T. Pinho. “The Finite Volume Method in Computational Rheology”. In: *Finite Volume Method - Powerful Means of Engineering Design*. Vol. 1. 2012, pp. 141–170.

[2] Hyung Taek Ahn, Larisa Branets, and Graham F Carey. “Moving boundary simulations with dynamic mesh smoothing”. In: *International Journal for Numerical Methods in Fluids* 64.8 (2010), pp. 887–907.

[3] Mark Ainsworth and J. Tinsley Oden. “A unified approach to a posteriori error estimation using element residual methods”. In: *Numerische Mathematik* 65.1 (1993), pp. 23–50.

[4] Khyruddin Akbar Ansari. “Mooring Dynamics of Offshore Vessels”. In: *Developments in Offshore Engineering*. Gulf Professional Publishing, 1999. Chap. 5, pp. 195–255.

[5] Harm Askes, Ellen Kuhl, and Paul Steinmann. “An ALE formulation based on spatial and material settings of continuum mechanics. Part 2: Classification and applications”. In: *Computer Methods in Applied Mechanics and Engineering* 193.39-41 SPEC. ISS. (2004), pp. 4223–4245.

[6] Matteo Astorino, Franz Chouly, and Miguel A Fernández. “An added-mass free semi-implicit coupling scheme for fluid–structure interaction”. In: *Comptes Rendus de l’Académie des Sciences* 347.1-2 (2009), pp. 99–104.

[7] J W Banks, W D Henshaw, D W Schwendeman, and Qi Tang. “A stable partitioned FSI algorithm for rigid bodies and incompressible flow. Part I: Model problem analysis”. In: *Journal of Computational Physics* 343 (2017), pp. 432–468.

[8] J W Banks, W D Henshaw, D W Schwendeman, and Qi Tang. “A stable partitioned FSI algorithm for rigid bodies and incompressible flow. Part II: General formulation”. In: *Journal of Computational Physics* 343 (2017), pp. 469–500.

[9] R.J. Barthelmie, M.S. Courtney, J. Højstrup, and S.E. Larsen. “Meteorological aspects of offshore wind energy: Observations from the Vindeby wind farm”. In: *Journal of Wind Engineering and Industrial Aerodynamics* 62.2-3 (1996), pp. 191–211.

[10] John T. Batina. “Unsteady Euler airfoil solutions using unstructured dynamic meshes”. In: *AIAA Journal* 28.8 (1990), pp. 1381–1388.

[11] Lars Bergdahl, Johannes Palm, Claes Eskilsson, and Jan Lindahl. “Dynamically Scaled Model Experiment of a Mooring Cable”. In: *Journal of Marine Science and Engineering* 4.1 (2016), p. 5.

[12] M. Berzeri and A. A. Shabana. “Development of simple models for the elastic forces in the absolute nodal co-ordinate formulation”. In: *Journal of Sound and Vibration* 235.4 (2000), pp. 539–565.

[13] H Bihs and A Kamath. “A combined level set/ghost cell immersed boundary representation for floating body simulations”. In: *International Journal for Numerical Methods in Fluids* 83.12 (2017), pp. 905–916.

[14] A de Boer, M. S. van der Schoot, and H Bijl. “Mesh deformation based on radial basis function interpolation”. In: *Computers and Structures* 85.11-14 (2007), pp. 784–795.

[15] J. U. Brackbill and H. M. Ruppel. “FLIP: A method for adaptively zoned, particle-in-cell calculations of fluid flows in two dimensions”. In: *Journal of Computational Physics* 65.2 (1986), pp. 314–343.

[16] Larisa Branets and G F Carey. “A local cell quality metric and variational grid smoothing algorithm”. In: *Engineering with Computers* 21.1 (2005), pp. 19–28.

[17] M. Breuer, G. De Nayer, M. Münsch, T. Gallinger, and R. Wüchner. “Fluid-structure interaction using a partitioned semi-implicit predictor-corrector coupling scheme for the application of large-eddy simulation”. In: *Journal of Fluids and Structures* 29 (2012), pp. 107–130.

[18] D. T. Brown and S. Mavrakos. “Comparative study on mooring line dynamic loading”. In: *Marine Structures* 12.3 (1999), pp. 131–151.

[19] Erik Burman and Miguel A. Fernández. “Explicit strategies for incompressible fluid-structure interaction problems: Nitsche type mortaring versus Robin-Robin coupling”. In: *International Journal for Numerical Methods in Engineering* 97.10 (2014), pp. 739–758.

[20] Erik Burman and Miguel A. Fernández. “Stabilization of explicit coupling in fluid-structure interaction involving fluid incompressibility”. In: *Computer Methods in Applied Mechanics and Engineering* 198.5-8 (2009), pp. 766–784.

[21] Erik Burman and Miguel A. Fernández. “Stabilized explicit coupling for fluid-structure interaction using Nitsche’s method”. In: *Comptes Rendus Mathematique* 345.8 (2007), pp. 467–472.

[22] Antoni Calderer, Seokkoo Kang, and Fotis Sotiropoulos. “Level set immersed boundary method for coupled simulation of air/water interaction with complex floating structures”. In: *Journal of Computational Physics* 277 (2014), pp. 201–227.

[23] P Causin, J F Gerbeau, and Fabio Nobile. “Added-mass effect in the design of partitioned algorithms for fluid-structure problems”. In: *Computer Methods in Applied Mechanics and Engineering* 194.42-44 (2005), pp. 4506–4527.

[24] Limin Chen, Guanghua He, Atilla Incecik, and Dazheng Wang. “Comparative study on steady wave-making problem using viscous and potential-flow methods”. In: *Ocean Engineering* 154 (2018), pp. 143–152.

[25] Qiang Chen, Jun Zang, Aggelos S. Dimakopoulos, David M. Kelly, and Chris J K Williams. “A Cartesian cut cell based two-way strong fluid-solid coupling algorithm for 2D floating bodies”. In: *Journal of Fluids and Structures* 62 (2016), pp. 252–271.

[26] Alexandre Joel Chorin. “Numerical Solution of the Navier-Stokes Equations”. In: *Mathematics of Computation* 22.104 (1968), pp. 745–762.

[27] Alain Clément, Pat McCullen, António Falcão, Antonio Fiorentino, Fred Gardner, Karin Hammarlund, George Lemonis, Tony Lewis, Kim Nielsen, Simona Petroncini, M. Teresa Pontes, Phillippe Schild, Bengt Olov Sjöström, Hans Christian Sørensen, and Tom Thorpe. *Wave energy in Europe: Current status and perspectives*. 2002.

[28] Alexander J. Coulling, Andrew J. Goupee, Amy N. Robertson, Jason M. Jonkman, and Habib J. Dagher. “Validation of a FAST semi-submersible floating wind turbine numerical model with DeepCwind test data”. In: *Journal of Renewable and Sustainable Energy* 5.2 (2013).

[29] R. Courant, K. Friedrichs, and H. Lewy. “Über die partiellen Differenzengleiehungen der”. In: *Mathematische Annalen* 100.1 (1928), pp. 32–74.

[30] Giovanni Cuomo, William Allsop, and Shigeo Takahashi. “Scaling wave impact pressures on vertical walls”. In: *Coastal Engineering* 57.6 (2010), pp. 604–609.

[31] Olivier Darrigol. “Between hydrodynamics and elasticity theory: The first five births of the Navier-Stokes equation”. In: *Archive for History of Exact Sciences* 56.2 (2002), pp. 95–150.

[32] Det Norske Veritas. *DNV-OS-E301 - Position Mooring*. Tech. rep. 2004.

[33] Det Norske Veritas. *DNV-OS-E301 - Position Mooring*. Tech. rep. 2010.

[34] Karen Devine, Erik Boman, Robert Heaphy, Bruce Hendrickson, and Courtenay Vaughan. “Zoltan Data Management Services for Parallel Dynamic Applications”. In: *Computing in Science and Engineering*. Vol. 4. 2. IEEE, 2002, pp. 90–97.

[35] Brecht Devolder, Pál Schmitt, Pieter Rauwoens, Björn Elsaesser, and Peter Troch. “A review of the implicit motion solver algorithm in OpenFOAM to simulate a heaving Buoy”. In: *18th Numerical Towing Tank Symposium* (2015), pp. 1–6.

[36] Aggelos S. Dimakopoulos, Mark Cooker, Encarni Medina-Lopez, Daniele Longo, and Romain Pinguet. “Flow characterisation and numerical modelling of OWC wave energy converters”. In: *Proceedings of the 11th European Wave and Tidal Energy Conference*. 2015.

[37] Aggelos S. Dimakopoulos, Tristan de Lataillade, and Chris E. Kees. “Fast random wave generation in numerical tanks”. In: *Engineering and Computational Mechanics* 172.1 (2019), pp. 1–11.

[38] DNV GL. *DNVGL-OS-E301 - Position Mooring*. 2015.

[39] DNV GL. *DNVGL-OS-E301 - Position Mooring*. Tech. rep. 2018.

[40] Guillaume Ducrozet, Félicien Bonnefoy, and Yves Perignon. “Applicability and limitations of highly non-linear potential flow solvers in the context of water waves”. In: *Ocean Engineering* 142.April (2017), pp. 233–244.

[41] Richard P Dwight. “Robust Mesh Deformation using the Linear Elasticity Equations”. In: *Computational Fluid Dynamics 2006*. 2009, pp. 401–406.

[42] Leonhard Euler. “Principes Généraux du Mouvement des Fluides”. PhD thesis. Mémoires de l’Académie des Sciences de Berlin, 1755.

[43] C. Farhat and M. Lesoinne. “Two efficient staggered algorithms for the serial and parallel solution of three-dimensional nonlinear transient aeroelastic problems”. In: *Computer Methods in Applied Mechanics and Engineering* 182.3-4 (2000), pp. 499–515.

[44] C. Farhat, A. Rallu, K. Wang, and T. Belytschko. “Robust and provably second-order explicit–explicit and implicit–explicit staggered time-integrators for highly non-linear compressible fluid–structure interaction problems”. In: *International Journal for Numerical Methods in Engineering* 84 (2010), pp. 73–107.

[45] Charbel Farhat, Kristoffer G. van der Zee, and Philippe Geuzaine. “Provably second-order time-accurate loosely-coupled solution algorithms for transient nonlinear computational aeroelasticity”. In: *Computer Methods in Applied Mechanics and Engineering* 195.17-18 (2006), pp. 1973–2001.

[46] J. D. Fenton. “Nonlinear wave theories”. In: *The Sea - Ocean Engineering Science* 9 (1990), pp. 3–25.

[47] J. D. Fenton. “The numerical solution of steady water wave problems”. In: *Computers and Geosciences* 14.3 (1988), pp. 357–368.

[48] Miguel A. Fernández and Jean Frédéric Gerbeau. “Algorithms for fluid-structure interaction problems”. In: *Modeling, Simulation and Applications* 1 (2009), pp. 307–346.

[49] Miguel A Fernández, Jean Frederic Gerbeau, and Ceremade Grandmont. “A projection semi-implicit scheme for the coupling of an elastic structure with an incompressible fluid”. In: *International Journal for Numerical Methods in Engineering* 69.4 (2007), pp. 794–821.

[50] Miguel A Fernández, Jimmy Mullaert, and Marina Vidrascu. “Explicit Robin–Neumann schemes for the coupling of incompressible fluids with thin-walled structures”. In: *Computer Methods in Applied Mechanics and Engineering* 267 (2013), pp. 566–593.

[51] Miguel Angel Fernández. “Coupling schemes for incompressible fluid-structure interaction: implicit, semi-implicit and explicit”. In: *SeMA Journal* 55.1 (2011), pp. 59–108.

[52] Richard P. Feynman, Robert B. Leighton, and Matthew Sands. *The Feynman Lectures on Physics*. Vol. 2. 1963. Chap. 40-41.

[53] John Fitzgerald and Lars Bergdahl. “Including moorings in the assessment of a generic offshore wave energy converter: A frequency domain approach”. In: *Marine Structures* 21.1 (2008), pp. 23–46.

[54] Christiane Förster, Wolfgang A Wall, and Ekkehard Ramm. “Artificial added mass instabilities in sequential staggered coupling of nonlinear structures and incompressible viscous flows”. In: *Computer Methods in Applied Mechanics and Engineering* 196.7 (2007), pp. 1278–1293.

[55] Johannes Gerstmayr and Ahmed A Shabana. “Analysis of thin beams and cables using the absolute nodal co-ordinate formulation”. In: *Nonlinear Dynamics* 45.1-2 (2006), pp. 109–130.

[56] Christophe Geuzaine and Jean-Francois Remacle. “Gmsh: A 3-D finite element mesh generator with built-in pre- and post-processing facilities”. In: *International Journal for Numerical Methods in Engineering* 79.1 (2009), pp. 1309–1331.

[57] R. A. Gingold and J. J. Monaghan. “Smoothed Particle Hydrodynamics: Theory and Application to Non-Spherical Stars”. In: *Monthly Notices of the Royal Astronomical Society* 181 (1977), pp. 375–389.

[58] Y. Goda. *Random Seas and Design of Maritime Structures*. World Scientific, 2000.

[59] Matthias Grajewski, Michael Köster, Susanne Kilian, and Stefan Turek. “Numerical Analysis and Practical Aspects of a Robust and Efficient Grid Deformation Method in the Finite Element Context”. In: *SIAM Journal on Scientific Computing* 31 (2005), p. 1539.

[60] Matthias Grajewski, Michael Köster, and Stefan Turek. “A new multilevel grid deformation method”. In: (2008), pp. 1–23.

[61] Matthias Grajewski, Michael Köster, and Stefan Turek. “Numerical analysis and implementational aspects of a new multilevel grid deformation method”. In: *Applied Numerical Mathematics* 60.8 (2010), pp. 767–781.

[62] Matthias Grajewskl, Michael Köster, and Stefan Turek. “Mathematical and numerical an alysis of a robust and efficient grid deformation m ethod in the finite element context”. In: *SIAM Journal on Scientific Computing* 31.2 (2008), pp. 1539–1557.

[63] J. L. Guermond, P. Minev, and Jie Shen. “An overview of projection methods for incompressible flows”. In: *Computer Methods in Applied Mechanics and Engineering* 195.44-47 (2006), pp. 6011–6045.

[64] Matthew Hall and Andrew Goupee. “Validation of a lumped-mass mooring line model with DeepCwind semisubmersible model test data”. In: *Ocean Engineering* 104 (2015), pp. 590–603.

[65] V. Harnois, S. D. Weller, L. Johanning, P. R. Thies, M. Le Boulluec, D. Le Roux, V. Soulé, and J. Ohana. “Numerical model validation for mooring systems: Method and application for wave energy converters”. In: *Renewable Energy* 75 (2015), pp. 869–887.

[66] V. Harnois, S. D. Weller, Lars Johanning, P. R. Thies, M. Le Boulluec, D. Le Roux, V. Soulé, and J. Ohana. “Numerical model validation for mooring systems: Method and application for wave energy converters”. In: *Renewable Energy* 75.4 (2015), pp. 869–887.

[67] A Haselbacher, F. M. Najjar, and J. P. Ferry. “An efficient and robust particle-localization algorithm for unstructured grids”. In: *Journal of Computational Physics* 225.2 (2007), pp. 2198–2213.

[68] K. Hasselmann, T.P. Barnett, E. Bouws, H. Carlson, D.E. Cartwright, K. Enke, J.A. Ewing, H. Gienapp, D.E. Hasselmann, P. Kruseman, A. Meerburg, P. Müller, D.J. Olbers, K. Richter, W. Sell, and H. Walden. *Measurements of Wind-Wave Growth and Swell Decay during the Joint North Sea Wave Project (JONSWAP)*. Deutches Hydrographisches Institut, 1973.

[69] C. W. Hirt, A. A. Amsden, and J. L. Cook. “An arbitrary Lagrangian-Eulerian computing method for all flow speeds”. In: *Journal of Computational Physics* 14.3 (1974), pp. 227–253.

[70] C.W Hirt and B.D Nichols. “Volume of fluid (VOF) method for the dynamics of free boundaries”. In: *Journal of Computational Physics* 39.1 (1981), pp. 201–225.

[71] A. Huerta and F. Casadei. “New ALE applications in non-linear fast-transient solid dynamics”. In: *Engineering Computations* 11.4 (1994), pp. 317–345.

[72] E. Huse. “Influence of mooring line damping upon rig motions.” In: *Offshore Technology Conference*. 1986.

[73] E. Huse. “New developments in prediction of mooring system damping”. In: *Proceedings of the Annual Offshore Technology Conference* 1989 (1991), pp. 291–298.

[74] S R Idelsohn, F Del Pin, R Rossi, and E. Onate. “Fluid–structure interaction problems with strong added-mass effect”. In: *International Journal for Numerical Methods in Engineering* 80 (2013), pp. 1261–1294.

[75] Soichi Ito. *Study of the transient heave oscillation of a floating cylinder*. Tech. rep. Massachusetts Institute of Technology, 1977.

[76] Niels G. Jacobsen, David R. Fuhrman, and Jørgen Fredsøe. “A wave generation toolbox for the open-source CFD library: OpenFoam”. In: *International Journal for Numerical Methods in Fluids* 70 (2012), pp. 1073–1088.

[77] L. Johanning, G. H. Smith, and J. Wolfram. “Mooring design approach for wave energy converters”. In: *Proceedings of the Institution of Mechanical Engineers Part M: Journal of Engineering for the Maritime Environment* 220.4 (2006), pp. 159–174.

[78] Lars Johanning, George H Smith, and Julian Wolfram. “Interaction between mooring line damping and response frequency as a result of stiffness alteration in surge”. In: *Proceedings of the International Conference on Offshore Mechanics and Arctic Engineering - OMAE*. Vol. 2006. 2006, pp. 1–10.

[79] Lars Johanning, George H Smith, and Julian Wolfram. “Measurements of static and dynamic mooring line damping and their importance for floating WEC devices”. In: *Ocean Engineering* 34.14-15 (2007), pp. 1918–1934.

[80] Lars Johanning and Julian Wolfram. “Challenging tasks on moorings for floating WECs”. In: *Proc. of the Int. Symposium on Fluid Machinery for Wave and Tidal Energy: State of the Art and New Developments, London, UK*. Vol. 19. 2005, pp. 1–9.

[81] Kwang Hyo Jung, Kuang-an Chang, and Hyo Jae Jo. “Viscous Effect on the Roll Motion of a Rectangular Structure”. In: *Journal of engineering mechanics* 132.2 (2006), pp. 190–200.

[82] Chris E. Kees, I. Akkerman, M. W. Farthing, and Y. Bazilevs. “A conservative level set method suitable for variable-order approximations and unstructured meshes”. In: *Journal of Computational Physics* 230 (2011), pp. 4536–4558.

[83] Jeffrey Ketchman and Y. K. Lou. “Application of the finite element method to towed cable dynamics.” In: *OCEAN 75 Conference* (1975), pp. 98–107.

[84] Bonjun J. Koo, Andrew J. Goupee, Richard W. Kimball, and Kostas F. Lambrakos. “Model Tests for a Floating Wind Turbine on Three Different Floaters”. In: *Journal of Offshore Mechanics and Arctic Engineering* 136.2 (2014), p. 021904.

[85] Weoncheol Koo and Moo Hyun Kim. “Freely floating-body simulation by a 2D fully nonlinear numerical wave tank”. In: *Ocean Engineering* 31.16 (2004), pp. 2011–2046.

[86] Ellen Kuhl, Harm Askes, and Paul Steinmann. “An ALE formulation based on spatial and material settings of continuum mechanics. Part 1: Generic hyperelastic formulation”. In: *Computer Methods in Applied Mechanics and Engineering* 193.39-41 SPEC. ISS. (2004), pp. 4207–4222.

[87] Tristan de Lataillade, Aggelos S. Dimakopoulos, Chris E. Kees, Lars Johanning, David Ingram, and Tezdogan Tezdogan. “CFD Modelling Coupled with Floating Structures and Mooring Dynamics for Offshore Renewable Energy Devices using the Proteus Simulation Toolkit”. In: *Proceedings of the 12th European Wave and Tidal Energy Conference*. 2017.

[88] Bernard Le Méhauté. *An Introduction to Hydrodynamics and Water Waves*. 1976.

[89] M. Lesoinne and C. Farhat. “Higher-order subiteration-free staggered algorithm for nonlinear transient aeroelastic problems”. In: *AIAA Journal* 36.9 (1998), pp. 1754–1757.

[90] L. B. Lucy. “A Numerical Approach to the Testing of the Fission Hypothesis”. In: *The Astronomical Journal* 82 (1977), pp. 1013–1024.

[91] Hermann G Matthies, Rainer Niekamp, and Jan Steindorf. “Algorithms for strong coupling procedures”. In: *Computer Methods in Applied Mechanics and Engineering* 195.17-18 (2006), pp. 2028–2049.

[92] Rajat Mittal and Gianluca Iaccarino. “Immersed Boundary Methods”. In: *Annual Review of Fluid Mechanics* 37.1 (2005), pp. 239–261.

[93] Todd Munson. “Mesh shape-quality optimization using the inverse mean-ratio metric”. In: *Mathematical Programming* 110.3 (2007), pp. 561–590.

[94] Alireza Naseri, Oriol Lehmkuhl, Ignacio Gonzalez, Eduard Bartrons, Carlos David Pérez-Segarra, and Assensi Oliva. “A semi-implicit coupling technique for fluid–structure interaction problems with strong added-mass effect”. In: *Journal of Fluids and Structures* 80 (2018), pp. 94–112.

[95] Claude Louis Navier. “Memorie sur les lois du lois du mouvement des fluides”. In: *Mem. Acad. Sci Inst. France* 6 (1827), pp. 298–440.

[96] Ali Nematbakhsh, Erin E. Bachynski, Zhen Gao, and Torgeir Moan. “Comparison of wave load effects on a TLP wind turbine by using computational fluid dynamics and potential flow theory approaches”. In: *Applied Ocean Research* 53 (2015), pp. 142–154.

[97] J. N. Newman. *Marine Hydrodynamics*. 40th ed. The MIT Press, 2018, pp. 151–153.

[98] Rachel Nicholls-Lee, Adam Walker, Simon Hindley, and Richard Argall. “Coupled multi-phase CFD and transient mooring analysis of the floating wave energy converter OWEL”. In: *Proceedings of the International Conference on Offshore Mechanics and Arctic Engineering - OMAE*. Vol. 8. 2013.

[99] O G Nwogu and Z Demirbilek. “BOUSS - 2D: A Boussinesq Wave Model for Coastal Regions and Harbors.” In: *Technical Report ERDC/CHL TR-01-25, U.S.* September (2001).

[100] R. O’Reilly. “Liste des Brevets d’inventions”. In: *Annales des Arts et Manufactures*. Paris: Imprimerie des Annales, 1806, pp. 291–292.

[101] D. Ocaña-Blanco, I. Castañeda-Sabadell, and A. Souto-Iglesias. “CFD and potential flow assessment of the hydrodynamics of a kitefoil”. In: *Ocean Engineering* 146.September (2017), pp. 388–400.

[102] J. T. Oden and S. Prudhomme. “Goal-oriented error estimation and adaptivity for the finite element method”. In: *Computers and Mathematics with Applications* 41.5-6 (2001), pp. 735–756.

[103] H Ormberg and K Larsen. “Coupled analysis of floater motion and mooring dynamics for a turret-moored ship”. In: *Applied Ocean Research* 20.1-2 (1998), pp. 55–67.

[104] Stanley Osher and James a. Sethian. “Fronts propagating with curvature dependent speed: Algorithms based on Hamilton-Jacobi formulations”. In: *Journal of Computational Physics* 1988 (1988), pp. 12–49.

[105] Johannes Palm, Claes Eskilsson, and Lars Bergdahl. “An hp-adaptive discontinuous Galerkin method for modelling snap loads in mooring cables”. In: *Ocean Engineering* 144.August (2017), pp. 266–276.

[106] Johannes Palm, Claes Eskilsson, Guilherme Moura Paredes, and Lars Bergdahl. “Coupled mooring analysis of floating wave energy converters using CFD: Formulation and Validation”. In: *International Journal of Marine Energy* 16 (2016), pp. 83–99.

[107] S. Piperno and C. Farhat. “Partitioned procedures for the transient solution of coupled aeroelastic problems - Part II: energy transfer analysis and three-dimensional applications”. In: *Computer Methods in Applied Mechanics and Engineering* 190.24-25 (2001), pp. 3147–3170.

[108] Serge Piperno, Charbel Farhat, and Bernard Larrouturou. “Partitioned procedures for the transient solution of coupled aroelastic problems Part I: Model problem, theory and two-dimensional application”. In: *Computer methods in applied mechanics and engineering* 124.1-2 (1995), pp. 79–112.

[109] L Qian, D M Causon, D M Ingram, and C G Mingham. “Cartesian cut cell two-fluid solver for hydraulic flow problems”. In: *Journal of Hydraulic Engineering* 129.9 (2003), pp. 688–696.

[110] Patrick J. Roache. *Verification and Validation in Computational Science and Engineering*. Hermosa Publishers, 1998.

[111] A. Robertson, J. Jonkman, and M. Masciola. *Definition of the Semisubmersible Floating System for Phase II of OC4*. Tech. rep. September. 2014, pp. 1–38.

[112] Marcos Vinícius Rodrigues, Fabrício Nogueira Correa, and Breno Pinheiro Jacob. “Implicit domain decomposition methods for coupled analysis of offshore platforms”. In: *Communications in Numerical Methods in Engineering*. Vol. 23. 6. 2007, pp. 599–621.

[113] MM Selim and RP Koomullil. “Mesh Deformation Approaches – A Survey”. In: *Journal of Physical Mathematics* 7.2 (2016).

[114] Jonathan Richard Shewchuk. “Delaunay refinement algorithms for triangular mesh generation”. In: *Computational Geometry* 22 (2002), pp. 21–74.

[115] Hang Si. “TetGen, a delaunay-based quality tetrahedral mesh generator”. In: *ACM Transactions on Mathematical Software* 41.2 (2015).

[116] Heinrich Söding. “How to Integrate Free Motions of Solids in Fluids”. In: *4th Numerical Towing Tank Symposium*. 2001, pp. 23–25.

[117] G. G. Stokes. “On the effect of the internal friction of fluids on the motion of pendulums”. In: *Transactions of the Cambridge Philosophical Society* 9 (1851).

[118] Mark Sussman, Peter Smereka, and Stanley Osher. *A Level Set Approach for Computing Solutions to Incompressible Two-Phase Flow*. 1994.

[119] Iradj Tadjbakhsh and Joseph B Keller. “Standing surface waves of finite amplitude”. In: *Journal of Fluid Mechanics* 8.3 (1960), pp. 442–451.

[120] R Temam. “Sur l’approximation de la solution des équations de Navier-Stokes par la méthode des pas fractionnaires (II)”. In: *Archive for Rational Mechanics and Analysis* 33 (1969), pp. 377–385.

[121] R. Temam. “Sur l’approximation de la solution des équations de Navier-Stokes par la méthode des pas fractionnaires (I)”. In: *Archive for Rational Mechanics and Analysis* 32 (1969), pp. 135–153.

[122] Tayfun E Tezduyar. “Finite Element Methods for Flow Problems with Moving Boundaries and Interfaces”. In: *Archives of Computational Methods in Engineering* 8.2 (2001), pp. 83–130.

[123] Vidar Thomée. “From finite differences to finite elements”. In: *Journal of Computational and Applied Mathematics* 128.1-2 (2001), pp. 1–54.

[124] Onno Ubbink. “Numerical prediction of two fluid systems with sharp interfaces”. PhD thesis. University of London, 1997.

[125] J. Vollmer, R. Mencl, and H. Muller. “Improved Laplacian Smoothing of Noisy Surface Meshes”. In: *Computer Graphics Forum* 18.3 (1999), pp. 131–138.

[126] Thomas S Walton and Harry Polachek. “Calculation of Transient Motion of Submerged Cables”. In: *Mathematics of Computation* 14.69 (1960), p. 27.

[127] Shan Wang, Sheng Xu, Gong Xiang, and C Guedes Soares. “An overview of synthetic mooring cables in marine applications”. In: *Advances in Renewable Energies Offshore - Proceedings of the 3rd International Conference on Renewable Energies Offshore, RENEW 2018*. 2019, pp. 853–863.

[128] W. C. Webster. “Mooring-induced damping”. In: *Ocean Engineering* 22.6 (1995), pp. 571–591.

[129] S D Weller, L Johanning, P Davies, and S. J. Banfield. *Synthetic mooring ropes for marine renewable energy applications*. 2015.

[130] Christian Windt, Josh Davidson, and John V Ringwood. *High-fidelity numerical modelling of ocean wave energy systems: A review of computational fluid dynamics-based numerical wave tanks*. 2018.

[131] Hugh A Wolgamot and Colm J Fitzgerald. “Nonlinear hydrodynamic and real fluid effects on wave energy converters”. In: *Proceedings of the Institution of Mechanical Engineers, Part A: Journal of Power and Energy*. Vol. 229. 7. 2015, pp. 772–794.

[132] Zhengqiang Xu and Shan Huang. “Numerical investigation of mooring line damping and the drag coefficients of studless chain links”. In: *Journal of Marine Science and Application* 13.1 (2014), pp. 76–84.

[133] C Yvin, A Leroyer, E Guilmineau, P Queutey, and M Visonneau. “Couplage de Codes pour l’Etude d’Interactions Fluide-Structure de Corps Rigides dans le Domain de l’Hydrodynamique Navale”. In: *14èmes Journées de l’Hydrodynamique*. 2014.

[134] Camille Yvin, Alban Leroyer, and Michel Visonneau. “Co-simulation in fluid-structure interaction problem with rigid bodies”. In: *16th Numerical Towing Tank Symposium* (2013).

[135] Xizeng Zhao and Changhong Hu. “Numerical and experimental study on a 2-D floating body under extreme wave conditions”. In: *Applied Ocean Research* 35 (2012), pp. 1–13.

[136] O. C. Zienkiewicz and J. Z. Zhu. “The superconvergent patch recovery and a posteriori error estimates. Part 1: The recovery technique”. In: *Numerical Methods for Engineering* 33 (1992), pp. 1331–1364.