



PONTIFICIA
UNIVERSIDAD
CATÓLICA
DE CHILE



THE UNIVERSITY
of EDINBURGH

UNSTEADY THREE-DIMENSIONAL FLOWS DOMINATED BY FREE-SURFACE DYNAMICS: NUMERICAL INSIGHTS FOR FLUVIAL AND COASTAL APPLICATIONS

JORGE SANDOVAL ULLOA

Thesis submitted to the Pontificia Universidad Católica de Chile in partial fulfilment of the requirements for the degree of Doctor in Engineering Sciences

Thesis submitted to the University of Edinburgh in partial fulfilment of the requirements for the degree of Doctor of Philosophy conferred in the College of Science and Engineering

Supervisors:

CRISTIÁN ESCAURIAZA

DAVID INGRAM

Santiago de Chile, December, 2025

© MMXXV, JORGE SANDOVAL ULLOA



PONTIFICIA
UNIVERSIDAD
CATÓLICA
DE CHILE



THE UNIVERSITY
of EDINBURGH

**UNSTEADY THREE-DIMENSIONAL
FLOWS DOMINATED BY FREE-SURFACE
DYNAMICS: NUMERICAL INSIGHTS FOR
FLUVIAL AND COASTAL APPLICATIONS**

JORGE SANDOVAL ULLOA

Members of the Committee:

CRISTIÁN ESCAURIAZA
DAVID INGRAM
YADRAN ETEROVIC
WERNHER BREVIS
RODRIGO CIENFUEGOS
THOMAS BRUCE
YASUNORI WATANABE

Santiago de Chile, December, 2025

ACKNOWLEDGEMENTS

I would like to thank all the people who supported me throughout these years. I would also like to thank ANID for the Doctoral Scholarship that funded my studies (Beca Doctorado Nacional ANID 2020-21200563). This research was partially supported by the supercomputing infrastructure of the NLHPC (ECM-02).

CONTENTS

ACKNOWLEDGEMENTS	iii
LIST OF FIGURES	vii
LIST OF TABLES	xii
ABSTRACT	xiii
LAY SUMMARY	xv
RESUMEN	xvii
1. Introduction	2
1.1. Incompressible flow equations and numerical solution strategies	3
1.1.1. Pressure-based methods	4
1.1.2. Density-based methods	6
1.2. Free-surface modelling	8
1.3. Mass loss and reinitialisation approach in Level-Set method	10
1.4. Physically-consistent free surface boundary conditions	13
1.5. Free surface dynamics in fluid structure interaction in coastal and fluvial environments	15
1.6. Perspectives for coastal and fluvial modelling in renewable energy	17
1.7. Objectives and Thesis Structure	23
2. Computational Fluid Dynamics Model	26
2.1. Governing equations of incompressible free-surface flows	26
2.1.1. Pressure-splitting	28
2.2. Dynamic free-surface boundary conditions	29
2.2.1. Tangential Dynamic Boundary Condition (TDBC)	29
2.2.2. Normal Dynamic Boundary Condition (NDBC)	36
2.3. Slope limiters for gradient reconstruction	39

2.4.	Turbulence Model	41
2.5.	Artificial Compressibility numerical model	42
2.5.1.	Block-Diagonal algorithm and numerical schemes	44
3.	Free surface evolution model	54
3.1.	Free Surface Advection: Level Set Method	54
3.2.	Reinitialisation of the level-set function	56
3.2.1.	Free surface reconstruction and geometric distance computation	57
3.2.2.	Local mass correction	59
3.2.3.	Global mass correction	60
4.	Validation of the Numerical Model	64
4.1.	Dynamic Boundary Conditions	64
4.1.1.	Tangential Dynamic Boundary Condition	66
4.1.2.	Normal Dynamic Boundary Condition	73
4.2.	Geometric Reinitialisation	75
4.3.	Full model	78
4.3.1.	Low amplitude sloshing tank	79
4.3.2.	Solitary wave propagation	83
5.	Solitary Wave Interactions with In-Stream Structures	88
5.1.	Introduction	88
5.2.	Computational Setup	89
5.3.	Run-up	91
5.4.	Wave hydrodynamic loading	92
5.5.	Bed shear stresses	95
5.6.	Vortical structures	97
5.7.	Discussion	103
6.	Conclusions and future work	107
	REFERENCES	114

APPENDIX A. Transformation of Navier-Stokes equations into generalised curvilinear coordinates	127
A.1. Non-dimensional Navier-Stokes equations for an incompressible fluid with a source term	127
A.1.1. Flow scales	127
A.1.2. Mass conservation	127
A.1.3. Momentum conservation	128
A.2. Governing Equations in Strong Conservation Form	130
A.3. Transformation of the Governing Equations to Generalised Curvilinear Coordinates	132
A.3.1. Linear Fluxes	133
A.3.2. Non-Linear Fluxes	134
A.3.3. Viscous Fluxes	135
APPENDIX B. Block-Diagonal Algorithm	138
B.1. Boundary conditions	144
B.2. Tridiagonal matrix system	146
B.3. Free-surface boundary conditions	148
APPENDIX C. Dynamic Boundary Condition at the free surface	151
C.1. Tangential Boundary Condition	151
C.2. Normal Boundary Condition	152
APPENDIX D. Simpson rule in Generalised Curvilinear Coordinates	154

LIST OF FIGURES

1.1	Illustration of a control volume intersected by the free surface. The phase interface causes abrupt changes in density and viscosity, which affect the discretisation of mass and momentum fluxes.	14
1.2	Examples of recent studies of free-surface flows interacting with structures. (a) Channel flow interacting with a bridge-abutment (extracted from Kara et al. (2015), where SW corresponds to standing wave, WSD to the water-surface dip, and WS to the water surface position). (b) Solitary wave interacting with a rectangular structure (extracted from Bagherizadeh et al. (2021), where the colour corresponds to the velocity magnitude at the free surface.)	18
1.3	The flow environment where tidal turbines operate and sources of important flow phenomena experienced by devices. Extracted from Adcock et al. (2021).	19
1.4	Instantaneous coherent structures visualized with by using the q -criterion (Hunt et al., 1988) coloured by the streamwise velocity for ADM, BEM, and ALM approach (Sandoval et al., 2021).	21
1.5	Top view of the location of the minimum velocity at the centreline of the turbine. Empty orange circles mark the instantaneous location. Filled black circles highlight one of the instantaneous solutions. Black centreline is the time-average location of the velocity minimum. Black dashed-lines is the envelopment of the wake (Sandoval et al., 2021).	22
2.1	Local water depth.	28
2.2	Ghost nodes scheme. (a) Nodes in the vicinity of the free surface used in the extrapolation algorithm. Only water nodes (blue empty circles) are considered for extrapolation. (b) Zoom on the free-surface vicinity.	31
2.3	Ghost nodes scheme for pressure extrapolation.	37

3.1	Level set representation of a the zero level-set (blue lines) of a three-dimensional function (red surface). Image obtained from Gibou et al. (2018).	54
3.2	a) Example of a triangulation \mathcal{T}_h . The zero-level set of ϕ is entirely contained within \mathcal{T}_h . b) Example of an element (tetrahedron) \mathcal{K} within the triangulation \mathcal{T}_h	58
3.3	Local mass correction steps. a) Zero-level set of ϕ after the advection step described in section 3.1. b) Zero-level set of ϕ after the geometric distance calculation step described in subsection 3.2.1. c) Zero-level set of ϕ after the local mass correction step.	59
	$_{\text{TOTAL}}$. Subfigure b) shows de decomposition of the total volume into the contribution of single-phase cells (SPC), single-phase tetrahedrons (SPT) and changing-phase tetrahedrons (CPT). Subfigure c) depicts the volume after correcting the nodes belonging to changing-phase tetrahedrons.	61
4.1	figure.caption.12 Flow chart of the Artificial-Compressibility Level Set method. The red block shows the steps comprised within the Navier-Stokes solver. The blue block depicts the ones considered within the free-surface evolution model.	65
4.2	Rotating cylinder scheme. Filled contour levels of ϕ on the tested curvilinear grid is presented on the plane $y = 0$. \mathcal{S} is the isosurface corresponding to $\phi = 0$	67
4.3	Prescribed (blue arrows) and extrapolated (red arrows) velocity field in the rotating cylinder test case. Panel (a) shows the tangential error in the t -direction, while panel (b) shows the error in the s -direction.	68
4.4	Rotating sphere scheme. The rotations α , β , and γ around the Cartesian axis is showed on the upper right location of the figure. Filled contour levels of ϕ on the tested curvilinear grid is presented on the plane normal to the resulting tilting vector. \mathcal{S} is the isosurface corresponding to $\phi = 0$	70
4.5	Prescribed (blue arrows) and extrapolated (red arrows) velocity field in the rotating sphere test case. Panel (a) shows the tangential error in the t -direction, while panel (b) shows the error in the s -direction.	71

4.6	Position of the tested wave theory in the Le Méhauté diagram. H , h , and T are the wave height, water depth and wave period, respectively (Le Méhauté, 2013).	72
4.7	Prescribed (blue arrows) and extrapolated velocities for the tested Stokes second-order wave over a wavelength in the x direction.	73
4.8	Rigid-body rotating cylinder scheme. The water forms a paraboloid, where the free surface corresponds to the isosurface of $p = 0$. H represents the still water level	73
4.9	Extrapolated pressure field. Panel (a) shows the pressure distribution above the free surface in the vicinity of three adjacent nodes. Panel (b) presents a zoomed-in view of a grid element to compare the locations of the isosurfaces $\phi = 0$ and $p = 0$	74
4.10	Zalesak's disc scheme. The slotted disc is rotated in a circular rigid-body motion around the centre of the domain located at (x_C, y_C)	76
4.11	Final shape after one revolution using two different reinitialisation schemes. The red line is the final shape obtained by using the reinitialisation method of Sussman et al. (1994), whereas the blue line is the final shape of the proposed geometric reinitialisation scheme adapted from Ausas et al. (2011).	78
4.12	Volume error for different scenarios: no reinitialisation, Sussman & Fatemi method, and four different geometric reinitialisation tests (every 1, 2, 5 and 10 time steps). The blue band shows the 5% volume error range, whereas the red band corresponds to the 10% error one.	79
4.13	Sloshing tank dimensions and initial condition	80
4.14	Velocity and dynamic pressure field for dimensionless times (a) $t = 1.74$ and (b) $t = 3.53$	82
4.15	Evolution of the water level at the left boundary using the current method (blue line with star markers), the method proposed by Hergibo et al. (2025) (red dashed line), and the analytical solution of Wu et al. (2001) (yellow solid line).	83

4.16 Schematic diagram of a three-dimensional solitary wave of height A_c propagating over the computational domain. (a) Three-dimensional view of the initial condition. (b) Side view.	85
4.17 Instantaneous profiles of the solitary wave at various times (with $t = 0$ set after 6 simulation time units	86
5.1 Initial condition of the computational domain for the study of the interaction between a solitary wave and two square obstacles. (a) Isometric view. (b) Top view.	90
5.2 Snapshots of the wave elevation for the tested obstacle separations at four different times: $t_1 = 3.0$, $t_2 = 5.0$, $t_3 = 6.5$, and $t_4 = 8.5$. Panels (a)–(d) correspond to $s = 0.5$, (e)–(h) to $s = 1.0$, and (i)–(l) to $s = 2.0$	91
5.3 Time series of the wave run-up at the centre of the front face of the obstacles for the three tested separations. The dashed line at $h = 1.1$ corresponds to the undisturbed wave amplitude. The period in which the run-up exceeds this level defines the wave run-up window, used to characterize the peak interaction phase.	92
5.4 Pressure-induced force on the obstacles: front face (solid line) and rear face (dashed line).	94
5.5 Friction-induced force on the obstacles: inner face (solid line) and outer face (dashed line).	95
5.6 Snapshots of filled contour maps of the non-dimensional shear velocity (u_τ) for the tested obstacle separations at four different times: $t_1 = 3.0$, $t_2 = 5.0$, $t_3 = 6.5$, and $t_4 = 8.5$. Panels (a)–(d) correspond to $s = 0.5$, (e)–(h) to $s = 1.0$, and (i)–(l) to $s = 2.0$. A schematic sketch of the wave propagation stage is shown in the upper-right corner of each panel.	96
5.7 Snapshot of the u -velocity component around the obstacle at $t = 6.5$ for $s = 0.5$. The dashed-line rectangle in front of the obstacle highlights the absence of a Horseshoe Vortex (HSV) system.	98

5.8 Time series of u_τ for a point located at a distance of 0.25 from the obstacle surface, measured outwards (dashed line), and inwards (solid line) for the three tested separations.	99
5.9 Computed z -vorticity at $z = 0.5$ for $s = 0.5$ at times $t_2 = 5.0$, $t_3 = 6.5$, and $t_4 = 8.5$. Panels (b), (d), and (e) are zoomed-in views of panels (a), (c), and (f), respectively. The stagnation point on the front face of the obstacle is indicated by SP	100
5.10 Computed z -vorticity at $z = 0.5$ for $s = 1.0$ at times $t_2 = 5.0$, $t_3 = 6.5$, and $t_4 = 8.5$. Panels (b), (d), and (e) are zoomed-in views of panels (a), (c), and (f), respectively. The stagnation point on the front face of the obstacle is indicated by SP	101
5.11 Computed z -vorticity at $z = 0.5$ for $s = 2.0$ at times $t_2 = 5.0$, $t_3 = 6.5$, and $t_4 = 8.5$. Panels (b), (d), and (e) are zoomed-in views of panels (a), (c), and (f), respectively. The stagnation point on the front face of the obstacle is indicated by SP	102
B.1 ξ^1 -sweep for a j, k position in the computational domain	146
B.2 ξ^1 -sweep for a j, k position in the computational domain	149
C.1 Traction vector at an arbitrary point for a plane defined by normal direction \underline{n}	151

LIST OF TABLES

4.1	Zalesak's disc characteristics	77
4.2	Sloshing tank parameters	80
5.1	Computational setup and simulation parameters.	89
5.2	Deviation of the stagnation point (<i>SP</i>) from the centre of the front face of the obstacle.	103

ABSTRACT

Unsteady, three-dimensional free-surface flows are ubiquitous in coastal and fluvial environments. They are characterised by a dynamic free-surface that evolves and modifies its shape in response to local flow hydrodynamics, and a bulk flow typically in the turbulent regime. The conservation laws that govern the evolution of this system are the incompressible Navier-Stokes equations. When the free-surface evolution is considered, a set of normal and tangential dynamic boundary conditions, derived from the local stress balance at the free surface, must be incorporated to define a physically-consistent problem. Disregarding this fundamental aspect of the phenomenon may lead to inaccurate solutions when the problem is addressed numerically, as evidenced by the recurring issues with mass conservation and poorly captured vortical dynamics reported in the literature.

In this work, a fully-unsteady single-phase three-dimensional free-surface viscous numerical model based on an Artificial Compressibility Navier-Stokes solver and the Level-Set method for capturing the free-surface evolution, is presented. Special emphasis is placed on the correct implementation and coupling of physically-consistent free-surface boundary conditions, which is achieved by the combination of the Ghost-Fluid method with a solution procedure based on a modified version of the Weighted Least Squares method.

The evolution of the free surface is captured using the Level-Set method. Its advection equation is advanced in time with an explicit third-order accurate Total Variation Diminishing Runge–Kutta scheme, while the convective term is discretised using a third-order accurate Weighted Non-Oscillatory scheme. Although these high-order methods ensure an accurate solution procedure, they cannot fully prevent degradation of mass conservation properties over time. This issue is addressed by a mass-preserving geometric redistancing procedure, adapted from that presented by Ausas et al. (2011), and modified to enforce strong mass conservation throughout the entire simulation period. This combination of methods has proven to be a robust approach for obtaining accurate solutions of unsteady free-surface flows.

The solver is thoroughly tested and validated, and then applied to study the interaction between a solitary wave and two identical in-stream obstacles at varying separations. While the separation had minor effects on the wave run-up and pressure-induced loadings on them, it showed to have a strong influence on lateral friction forces and inter-obstacle bed shear stresses, due to the enhanced streaming observed at smaller spacings. Unlike in-stream structures subjected to shear currents, the wave–obstacle interaction did not lead to the formation of a self-sustained Horseshoe Vortex in front of the obstacle. Instead, higher bed shear stresses were observed at the corners of the structure. The vortical structures that emerged from the interaction between the wave and the obstacles were also studied. They remained confined near the obstacles, not leading to the formation of a self-sustained shear layer. Despite these observations, in order to elucidate the underlying physics of this behaviour and identify conditions that promote stronger vortical activity, extending the study to a wider range of conditions is proposed as future work.

Keywords: Free-surface flows, Artificial Compressibility method, free-surface dynamic boundary conditions, Ghost-Fluid method, Level-Set method, geometric mass-preserving reinitialisation method, wave-structure interaction, vortical structures

LAY SUMMARY

Flows with a moving water surface, such as those found in rivers, estuaries, and coastal areas, are very common in nature. In these environments, the water surface constantly changes shape as waves propagate and interact with the underlying flow and with natural or engineering structures. Correctly predicting this behaviour is important for many practical applications, including coastal protection, flood risk assessment, and the design of hydraulic structures.

This work presents a new computer model designed to simulate these complex water flows in three dimensions. The model focuses on accurately representing how the water surface moves and how it interacts with the flow beneath it. Particular care is taken to ensure that the physical laws governing the motion of water are respected at the water surface. Ignoring these surface conditions can lead to unrealistic results, such as artificial loss or gain of water or incorrect representation of swirling motions in the flow.

To track the evolving water surface, the model uses a mathematical technique that follows the surface as it moves through space and time. Advanced numerical methods are employed to achieve high accuracy. However, such methods can gradually introduce small errors in the total amount of water. To overcome this limitation, a special correction procedure is included to preserve the correct water volume throughout the entire simulation.

After being carefully tested and validated, the model is used to study how a single large wave interacts with two obstacles placed on the bed of a channel. The results show that the distance between the obstacles has little effect on how high the wave climbs or on the forces pushing directly against the obstacles. However, it strongly affects sideways forces and stresses acting on the bed between them, especially when the obstacles are close together. The study also shows that, unlike steady currents, wave-driven flows do not generate long-lasting swirling structures in front of the obstacles. Instead, intense stresses tend to concentrate near their corners.

These findings help improve our understanding of wave–structure interactions and highlight the importance of using physically consistent and accurate models when studying free-surface flows. Further studies covering a wider range of conditions are proposed to better understand when stronger flow structures may develop.

Keywords: water waves, rivers and coastal flows, computer modelling, moving water surface, wave–structure interaction

RESUMEN

Los flujos tridimensionales no estacionarios con superficie libre son ubicos en ambientes costeros y fluviales. Se caracterizan por una superficie libre dinámica que evoluciona y modifica su forma en respuesta a la hidrodinámica local del flujo, y un flujo principal típicamente en régimen turbulento. Las leyes de conservación que gobiernan la evolución de este sistema son las ecuaciones de Navier-Stokes incompresibles. Al considerar la evolución de la superficie libre y para definir un problema físicamente consistente, es necesario incorporar un conjunto de condiciones de borde dinámicas normales y tangenciales, derivadas del balance local de esfuerzos en la superficie libre. Ignorar este aspecto fundamental del fenómeno puede conducir a soluciones numéricas inexactas, como lo evidencian los problemas recurrentes de conservación de masa y la débil captura de la dinámica vortical reportados en la literatura.

En este trabajo, se presenta un modelo numérico viscoso tridimensional, totalmente no estacionario y monofásico para superficies libres, basado en las ecuaciones de Navier-Stokes, mediante un método de Compresibilidad Srtificial acoplado con el método Level-Set para capturar la evolución de la superficie libre. Se pone especial énfasis en la correcta implementación y acoplamiento de condiciones de borde físicamente consistentes en la superficie libre, lo cual se logra mediante la combinación del método de Fluido Fantasma con un procedimiento de solución basado en una versión modificada del método de Mínimos Cuadrados Ponderados.

La evolución de la superficie libre se captura mediante el método Level-Set. Su ecuación de advección se resuelve en el tiempo mediante un esquema explícito de Runge-Kutta de tercer orden con la propiedad de *Total Variation Diminishing*, mientras que el término convectivo se discretiza usando un esquema ponderado no oscilatorio de tercer orden. Aunque estos métodos de alto orden aseguran un procedimiento numérico preciso, no pueden prevenir completamente la degradación de las propiedades de conservación de masa del método a lo largo del tiempo. Este problema se aborda mediante un procedimiento

geométrico de redistanciamiento que preserva la masa, adaptado del presentado por Ausas et al. (2011) y modificado para asegurar una conservación robusta durante todo el periodo de simulación. Esta combinación de métodos ha demostrado ser un enfoque eficaz para obtener soluciones precisas en flujos no estacionarios con superficie libre.

El metodo es exhaustivamente probado y validado, para luego aplicarse al estudio de la interacción entre una ola solitaria y dos obstáculos idénticos, con separaciones variables. Si bien la separación tuvo efectos menores en la elevación de la ola y en las cargas inducidas por presión sobre los obstáculos, mostró una influencia significativa en las fuerzas de fricción laterales y los esfuerzos de corte en el lecho entre los obstáculos, debido al aumento del flujo observado a menores separaciones. A diferencia de lo observado en estructuras que interactúan con corrientes estacionarias, la interacción ola–obstáculo no condujo a la formación de un vórtice de herradura frente al obstáculo. En cambio, los mayores esfuerzos de corte de fondo se observaron en las esquinas de la estructura. También se estudiaron los vórtices surgidos de la interacción entre la ola y los obstáculos, las cuales permanecieron confinadas cerca de los mismos, sin conducir a la formación de una capa de corte persistente. A pesar de estas observaciones, para esclarecer la física subyacente a este comportamiento e identificar las condiciones que promueven una actividad de estructuras coherentes más intensa, se propone extender el estudio a un rango más amplio de condiciones como trabajo futuro.

Palabras Claves: Flujos en superficie libre, método de Compresibilidad Artificial, condiciones de borde dinámicas en la superficie libre, método de fluido fantasma, método Level-Set, método de reinicialización geométrica conservativo, interacción ola-estructura, estructuras coherentes del flujo

CHAPTER 1

INTRODUCTION

1. INTRODUCTION

The accurate simulation of unsteady three-dimensional free-surface flows is of fundamental importance in fluvial and coastal engineering. From waves in coastal waters to the complex interactions between currents and topography in riverine environments, the dynamics of the free surface govern key transport processes, energy exchanges, and morphodynamic evolution.

Modelling these flows remains challenging. Strong non-linearities, large-scale turbulent structures, and complex boundary interactions give rise to highly unsteady dynamics that are difficult to capture both theoretically and numerically. In particular, the behaviour of the free surface introduces strong requirements on the numerical representation of interface deformation, pressure–velocity coupling, and mass conservation. Accurately resolving wave steepening, surface curvature, and the momentum exchange across the interface often demands fine spatial and temporal resolutions, together with numerical schemes that remain stable and consistent over long integration times.

These requirements are further compounded in three-dimensional settings, where anisotropic turbulence, secondary circulations, and three-dimensional vortical structures interact with the free surface in a strongly coupled manner. Traditional depth-averaged or quasi-two-dimensional models cannot capture these effects, while fully three-dimensional solvers must balance accuracy with computational efficiency. Interface-capturing methods, such as Level-Set, Volume-of-Fluid, or hybrid formulations, each exhibit specific advantages and limitations in representing the interface’s geometry, its motion, and its influence on the surrounding flow. Alongside this, the pressure–velocity coupling, inherent to incompressible flows, imposes additional numerical constraints, especially when the interface undergoes rapid deformation (Christou et al., 2021; McSherry et al., 2017).

These challenges highlight the need for high-fidelity numerical tools able to robustly track the interface while preserving physically consistent flow fields, which are crucial for advancing predictive capabilities in environmental fluid mechanics. Such tools must integrate accurate interface representation, stable pressure–velocity coupling, and strict mass

conservation, all while remaining applicable to large-scale and long-duration simulations relevant to real-world hydrodynamic problems. These considerations motivate the development of improved numerical strategies that can capture, in a unified framework, both the global evolution of the free surface and the detailed hydrodynamic processes occurring within the bulk flow.

In this thesis, we investigate unsteady incompressible three-dimensional flows dominated by free-surface dynamics using a numerical framework based on the Artificial Compressibility method coupled with a Level-Set interface capturing technique and a geometric reinitialisation method. This combination aims to address some of the central numerical difficulties encountered in free-surface simulations, including the accurate enforcement of mass conservation, and the physically consistent representation of pressure and velocity fields near the interface. Our hypothesis is that, by addressing these numerical challenges in a physically consistent manner, it is possible to accurately represent the complex hydrodynamics arising from the interaction between the free surface and the bulk flow, while avoiding common numerical artifacts observed in the literature for fully unsteady free-surface flows, such as poor overall mass conservation. Our focus is on the development, validation, and application of this method to scenarios relevant to fluvial and coastal systems, with perspectives towards renewable energy applications.

1.1. Incompressible flow equations and numerical solution strategies

The unsteady incompressible flow motion of a viscous fluid is described by the mass and momentum balance equations

$$\nabla \cdot \mathbf{u} = 0 \quad (1.1)$$

$$\frac{\partial \mathbf{u}}{\partial t} + \mathbf{C}(\mathbf{u}) - \mathbf{D}(\mathbf{u}) + \mathbf{G}(p) + \mathbf{S}_g = 0 \quad (1.2)$$

Where $\mathbf{u} = [u, v, w]^T$, p is the pressure divided by the density, $\mathbf{S}_g = [0, 0, -1/Fr^2]^T$, and \mathbf{C} , \mathbf{D} , and \mathbf{G} denote convection, diffusion and gradient operators, respectively, i.e

$$\mathbf{C}(\mathbf{u}) = (\mathbf{u} \cdot \nabla) \mathbf{u}, \quad \mathbf{D}(\mathbf{u}) = \frac{1}{Re} \nabla^2 \mathbf{u}, \quad \mathbf{G}(p) = \nabla p \quad (1.3)$$

Where Re , and Fr are the Reynolds and Froude numbers, respectively. These two flow parameters are based on characteristic length \mathcal{L} and velocity scales (\mathcal{U}) of the flow, such that

$$Re = \frac{\mathcal{U}\mathcal{L}}{\nu}, \quad Fr = \frac{\mathcal{U}}{\sqrt{g\mathcal{L}}} \quad (1.4)$$

The main difficulty with the solution of these equations is the decoupling of the continuity and momentum equations due to the absence of an evolution term of the pressure in the former (Drikakis & Rider, 2005). Over the years, a variety of numerical methods have been developed to compute time-accurate solutions of the governing equations. These methods can generally be categorised into two main classes: pressure-based methods and density-based methods.

1.1.1. Pressure-based methods

The main idea of this group of methods is to solve the pressure field such that a divergence-free velocity field computed every time step (Kwak & Kiris, 2010). Method in this category include pressure Poisson method and pressure projection methods

1.1.1.1. Pressure Poisson Method

This method was introduced in 1965 by Harlow, Welch, et al. (1965) and is referred as the MAC (marker and cell) method. The momentum equations are discretised in time with a first-order Euler explicit scheme. The convective and diffusive terms are evaluated explicitly, while the pressure gradient and the velocity field within the continuity equation are treated implicitly. To obtain a pressure field that ensures a divergence-free velocity field at the new time level, the discrete momentum equations are substituted in the continuity

equation to derive a Poisson explicit equation for the pressure field. The semi-discrete equations read

$$\nabla^2 p^{n+1} = \nabla \cdot \left[\frac{1}{\Delta t} \mathbf{u}^n - \mathbf{C}(\mathbf{u}^n) + \mathbf{D}(\mathbf{u}^n) \right] \quad (1.5)$$

$$\frac{\mathbf{u}^{n+1} - \mathbf{u}^n}{\Delta t} = - [\mathbf{C}(\mathbf{u}^n) - \mathbf{D}(\mathbf{u}^n) + \mathbf{G}(p^{n+1}) + \mathbf{S}_g] \quad (1.6)$$

Given the velocity field at the n time level, the Poisson equations 1.5 is first solved to determine the pressure field at the new time step $n + 1$. The computed pressure is then incorporated into the discrete momentum equation 1.6 to advance the velocity field in time. It can be easily demonstrated that the velocity field obtained from equation 1.6 is divergence free.

1.1.1.2. Projection or fractional step method

The idea of a fractional step method was independently proposed by Chorin (1968) and Temam (1969). This method encompasses similarities with the pressure Poisson approach as it ultimately produces a Poisson equation to be solved. The projection method relies on the Hodge (or Helmholtz) decomposition, which states that any vector function \mathbf{u}^* can be decomposed into a divergence free (solenoidal) part \mathbf{u} and curl-free one, which in turn can be expressed as the gradient of a potential φ (Chorin et al., 1990; Drikakis & Rider, 2005), such that

$$\mathbf{u}^* = \mathbf{u} + \nabla \varphi \quad (1.7)$$

The basic goal of the projection methods is to obtain an intermediate velocity field \mathbf{u}^* , disregarding its solenoidal nature in a so-called prediction step. Then, this velocity field is projected into a divergence-free space by adding to it the gradient of a pressure field obtained by solving a Poisson equation in the so-called correction step ($\mathbf{u} = \mathbf{u}^* - \nabla \varphi$).

In its simplest form, the method can be presented as follows:

Prediction Step:

The intermediate velocity field (\mathbf{u}^*) is obtained as

$$\frac{\mathbf{u}^* - \mathbf{u}^n}{\Delta t} = -[\mathbf{C}(\mathbf{u}^n) - \mathbf{D}(\mathbf{u}^n) + \mathbf{S}_g] \quad (1.8)$$

Correction Step:

A Poisson equation is solved to obtain p^{n+1}

$$\nabla^2 p^{n+1} = \frac{1}{\Delta t} \nabla \cdot \mathbf{u}^* \quad (1.9)$$

And the velocity field at the new time step \mathbf{u}^{n+1} is then obtained as

$$\mathbf{u}^{n+1} = \mathbf{u}^* - \Delta t \mathbf{G}(p^{n+1}) \quad (1.10)$$

This algorithm is nearly identical to the MAC formulation, which can be recovered by re-combining equations 1.8 and 1.10 to eliminate \mathbf{u}^* . The principal advantage of the projection method is the flexibility it offers for constructing second-order accurate schemes and for handling certain velocity terms implicitly (Tang & Sotiropoulos, 2007).

An important issue in the projection method is the prescription of suitable boundary conditions for the intermediate velocity field. Since the pressure gradient does not appear in the prediction equation, the intermediate velocity \mathbf{u}^* does not approximate \mathbf{u}^{n+1} . Consequently, implicit relations between \mathbf{u}^* and \mathbf{u}^{n+1} must be derived at the boundaries in order to maintain the accuracy of the method (Kim & Moin, 1985).

1.1.2. Density-based methods

An alternative strategy is the Artificial Compressibility (AC) method, was first introduced by Chorin (1967), which augments the incompressible equations with a pseudo-time derivative of pressure. This method was extended to solve the unsteady Navier-Stokes

equations by Merkle (1987) and Peyret (1976). The mass and momentum balance equations, with this dual-time approach reads

$$\frac{1}{\beta} \frac{\partial p}{\partial \tau} + \nabla \cdot \mathbf{u} = 0 \quad (1.11)$$

$$\frac{\partial \mathbf{u}}{\partial \tau} + \frac{\partial \mathbf{u}}{\partial t} + \mathbf{C}(\mathbf{u}) - \mathbf{D}(\mathbf{u}) + \mathbf{G}(p) + \mathbf{S}_g = 0 \quad (1.12)$$

Where τ is the pseudo-time and β is the compressibility coefficient. The parameter β is a disposable parameter, analogous to a relaxation parameter, which allows the equations to converge to a solution that satisfies the incompressibility condition. As the value of β increases, the system of equations becomes numerically stiffer, posing greater challenges for stability and convergence (Ferziger et al., 2019).

The system formed by equations 1.11 and 1.12 also has similarities with the equations of motion for a compressible fluid at low Mach numbers (Drikakis & Rider, 2005). By using this analogy, we can relate the artificial compressibility parameter to an artificial speed of sound, which represent the propagation speed of hydrodynamic pressure waves (Lee et al., 2006), such that

$$c = \sqrt{\beta} \quad (1.13)$$

In a truly incompressible flow, pressure disturbances propagate instantaneously throughout the domain. However, in the artificial compressibility method, the introduction of a pseudo-time derivative in the continuity equation leads to a finite propagation speed, resulting in a temporal lag between flow disturbances and the corresponding pressure response. To approximate incompressibility effectively, the artificial compressibility parameter should be chosen as large as the numerical algorithm allows, ensuring rapid recovery of the incompressible behaviour while maintaining both the accuracy and stability of the numerical scheme. By adding the time derivative of the hydrodynamic pressure into 1.11,

the physical meaning of the original continuity equation is modified, but it is recovered when numerical solutions converge to steady state in pseudo-time.

This approach transforms the problem into an hyperbolic and hyperbolic-parabolic system for inviscid and viscous incompressible flows, respectively, which can be solved by standard, implicit, time-marching methods, providing a unified time-marching solution of pressure and velocity (Kwak & Kiris, 2010; Madsen & Schäffer, 2006).

In this work, we adopt the AC method due to its simplicity, computational efficiency, and the fact that boundary conditions can be imposed consistently and simultaneously for all the primitive variables.

1.2. Free-surface modelling

In addition to the flow solver, a modelling approach to capture the dynamic evolution of the free surface is required. The free surface imposes additional kinematic and dynamic boundary conditions to the simulation and its correct representation has shown to be crucial to achieving accurate solutions in free-surface flows (Nangia et al., 2019; Saruwatari et al., 2009; Watanabe et al., 2008). The kinematic boundary condition is hyperbolic in nature and states that since there is no advective mass transfer across the air-water interface, the normal velocity component at the free surface must be equal to the velocity of the surface itself (McSherry et al., 2017). The dynamic boundary condition consists of a jump condition for momentum conservation across the interface, which leads to normal and tangential boundary conditions at points in the vicinity of the free surface (Watanabe et al., 2008). Methods to capture the free-surface dynamics in three-dimensional flow simulations can be classified into three broad categories: (1) arbitrary Lagrangian–Eulerian (ALE) methods; (2) front tracking methods; and (3) interface capturing methods (Kang & Sotiropoulos, 2012; Tryggvason et al., 2011).

In ALE methods, the flow is solved only in the water phase. The free-surface geometry is captured using a boundary-fitted moving curvilinear mesh, which conforms to the air-water interface every time step (Hirt et al., 1974). These methods are limited to situations

where the interface does not break, and the deformation remains regular enough to ensure connectivity of the surface-fitted grid (Kang & Sotiropoulos, 2012). Hence, they are not suitable for capturing the free surface in most of the previously described problems, where large deformation of the free surface can occur, or interactions with large-scale turbulent structures might induce irregular deformation patterns.

Front tracking methods are an approach where a single set of governing equations is solved for both phases. The free surface, for its part, is tracked explicitly by connected marker points which are advected in a Lagrangian manner by the flow field (Tryggvason et al., 2001; Unverdi & Tryggvason, 1992). This method is capable of capturing the breakup and merging of interfaces (Shin & Juric, 2002), but it can be very expensive computationally as a re-mesh step is needed to track the marker points (Kang & Sotiropoulos, 2012).

In contrast to the previously described methods, where the interface is captured explicitly, interface-capturing methods adopt an implicit formulation to represent the evolution of the free-surface (Gibou et al., 2018). These methods also simulate both phases, but the method for capturing the free surface is inherently Eulerian (Kang & Sotiropoulos, 2012). The simulated phases are discretised on a fixed background grid, and the interface position is captured by a scalar marker function defined as a continuous field over the computational domain. The main advantage of these methods is that no assumption is needed about the connectivity of the phase interface. Hence, topological interface transitions, such as wave breaking, air entrainment, or droplet formation, can be modelled in a straightforward manner. The most widely used methods in this category are the Level-Set (LS) method of Osher and Sethian (1988) and the volume-of-fluid (VOF) method of Hirt and Nichols (1981).

The level-set method consists in representing the interface as the zero level-set of a higher dimensional function. This function is defined as the signed distance from the interface, being positive in the liquid, negative in the gas, and zero at the interface itself (Dianat et al., 2017). The main advantage of this representation is its ability to naturally handle changes in topology (Gibou et al., 2018), because by this definition, geometrical features such as normal directions and curvatures can be easily defined by well-known tools from

differential geometry theory. VOF methods use the volume fraction of one phase (usually the liquid one) in each computational cell as an Eulerian marker function. These methods have the advantage of conserving the total volume by construction. However, they are more complicated than level-set methods in three spatial dimensions and it is difficult to compute smooth geometric properties accurately, such as curvatures from the volume fraction only. A combination of high-accuracy flow solvers, coherent-structure resolving turbulence models and interface-capturing methods have been combined to generate advanced numerical models to simulate real free-surface flows both in natural processes and engineering applications.

In this thesis, a level-set method is adopted to capture the evolution of the free surface in time. The transport equation for the signed distance function ϕ reads

$$\frac{D\phi}{Dt} \equiv \frac{\partial\phi}{\partial t} + u_i \frac{\partial\phi}{\partial x_i} = 0 \quad (1.14)$$

Where u_i is the background velocity field and ϕ is the signed distance function, whose zero-level set represents the free surface.

1.3. Mass loss and reinitialisation approach in Level-Set method

One of the major limitations of the Level-Set method is the mass loss, primarily due to numerical dissipation in simulations. The hyperbolic nature of the level set equation 1.14 makes necessary considerable numerical dissipation in the solution process (Yuan et al., 2018). Since this method was first proposed by Osher and Sethian (1988), substantial efforts have been done to overcome this drawback.

One straightforward idea is to improve spatial discretisation approaches by increasing the order of the schemes and refining the mesh. For instance, the use high order Weighted Essentially Non-Oscillatory (WENO) schemes (Liu et al., 1994) have demonstrated to reduce mass loss effectively, in addition to less diffusive capturing of the free-surface topology in comparison with lower order schemes (Gibou et al., 2018; Yuan et al., 2018). Another examples of strategies are Adaptative Mesh Refinement (AMR) methods (Tryggvason

et al., 2011), which locally refine the mesh near the interface, improving the accuracy of the solution where is needed. However, it has been noted that while these methods can reduce the impact of numerical dissipation by minimising numerical error, they cannot completely prevent mass loss and, more importantly, result in a substantial increase in computational cost.

Yuan et al. (2018) estimated the evolution of mass loss by considering the most common approach for representing two-phase flows, which involves smearing the material properties across the interface using a Heaviside function based on the distance to the free surface. In the case of air and water, with densities ρ_a and ρ_w respectively, the density field of the flow is described by

$$\rho(\phi) = \rho_a + (\rho_w - \rho_a)H(\phi) \quad (1.15)$$

Where H , is a Heaviside-type function defined as

$$H(\phi) = \begin{cases} 0 & \phi < -\epsilon \\ \frac{1}{2} \left[1 + \frac{\phi}{\epsilon} + \frac{1}{\pi} \sin \left(\frac{\pi\phi}{\epsilon} \right) \right] & |\phi| \leq \epsilon \\ 1 & \phi > \epsilon \end{cases} \quad (1.16)$$

Where ϵ is a parameter which is $\mathcal{O}(\Delta x)$ (Kang & Sotiropoulos, 2012). Given this formulation, the mass loss equation derived by Yuan et al. (2018) reads

$$\frac{dM}{dt} = - \int_{\Omega} \rho \frac{\partial H(\phi)}{\partial \phi} \varepsilon_{\text{dif}} d\Omega - \int_{\Omega} \nabla \cdot (H(\phi) \rho \mathbf{u}) d\Omega \quad (1.17)$$

Where Ω is the analysed control volume, and the variable ε_{dif} corresponds to the numerical diffusion associated to the chosen scheme for the convective terms of the level-set equations. This equation shows that the mass loss is then proportional to the numerical diffusion of the solution scheme and the density difference (second integral) of the considered fluids. The density difference in the air and water phase is significant, which makes the construction of mass preserving schemes for free-surface flows particularly challenging.

To improve the mass-preserving properties of the level-set method, a further step named “reinitialisation” or “redistancing” is needed. As its names indicates, this method seeks to recover the property of the level-set function as a signed distance function, which cannot be ensured after the advection of it, and it’s crucial for global mass conservation (Gibou et al., 2018; Osher & Fedkiw, 2003).

Broadly speaking, reinitialisation techniques can be divided into two major families: those based on solving a Hamilton-Jacobi (H-J) type equation and those relying on geometrical reconstruction of the free surface. The H-J approach formulates reinitialisation as a partial differential equation that seeks to modify the Eulerian field of ϕ in order to achieve the signed distance property $|\phi| = 1$. This formulations is typically written as (Osher & Fedkiw, 2003)

$$\frac{\partial \phi}{\partial \tau} + S(\phi_0)(|\nabla \phi| - 1) = 0 \quad (1.18)$$

where ϕ is the level-set function, τ is a fictitious time, and $S(\phi_0)$ is a smoothed sign function, which has been defined in various ways by different authors. For example Kang and Sotiropoulos (2012), defined it as

$$S(\phi_0) = \begin{cases} 1, & \phi_0 \geq \epsilon, \\ -1, & \phi_0 \leq -\epsilon, \\ \frac{\phi_0}{\epsilon} - \frac{1}{\pi} \sin\left(\frac{\pi\phi_0}{\epsilon}\right), & \text{otherwise.} \end{cases} \quad (1.19)$$

Where ϵ is the same parameter used in equation 1.16. This PDE is evolved until a steady state is reached, effectively re-shaping the level-set function into a signed distance function without significantly altering the zero level-set. Notable implementations of this method are described in works by Sussman et al. (1994) and Russo and Smereka (2000), which also introduced high-order schemes and entropy fixes to improve accuracy near the interface.

Alternatively, geometrical reinitialisation methods use explicit reconstruction of the interface. These approaches do not rely on solving an evolution equation but instead use geometric algorithms to compute the minimal distance from grid points to the reconstructed interface. The local reconstruction of the free surface is carried out by using methods for isosurfaces computation and extraction. The most common methods are the Marching Cubes and Marching Tetrahedra algorithms (Foley, 1996), which, by using the nodal values of the signed distance function in cells where a phase change is present, allow for the reconstruction of an isosurface by means of linear interpolation.

One such approach is presented by Ausas et al. (2011), who introduced a geometric, mass-preserving redistancing method. The zero level set is first reconstructed as a triangulated surface from the Eulerian field of ϕ . Then, for each computational node, the signed distance to this surface is computed explicitly, ensuring that the reinitialised function reflects the correct distance while preserving the interface location. Importantly, the scheme includes an additional correction step that enforces global mass conservation without altering the interface position.

The method of Ausas et al. (2011) is used in this thesis, as it allows for an accurate representation of the free surface while offering robust mass-preserving properties, regardless of the local mesh configuration and without introducing any adjustable parameters into the equations.

1.4. Physically-consistent free surface boundary conditions

In Finite Volume methods, the system variables are advanced in time by solving the discrete conservation laws for mass and momentum. The numerical fluxes are computed over local control volumes defined by the computational grid, based on the balance of conserved quantities (Ferziger et al., 2019; LeVeque, 2002). In two-phase flow formulations, near the free surface, inconsistencies in the discrete mass and momentum fluxes may arise due to the heterogeneous distribution of material properties (e.g., density and viscosity) within the control volume (see Figure 1.1). These discrepancies can introduce mass loss

and spurious shear stresses and pressure gradients, which may propagate into the bulk flow, degrade the accuracy of the solution, and hinder the correct representation of interfacial dynamics.

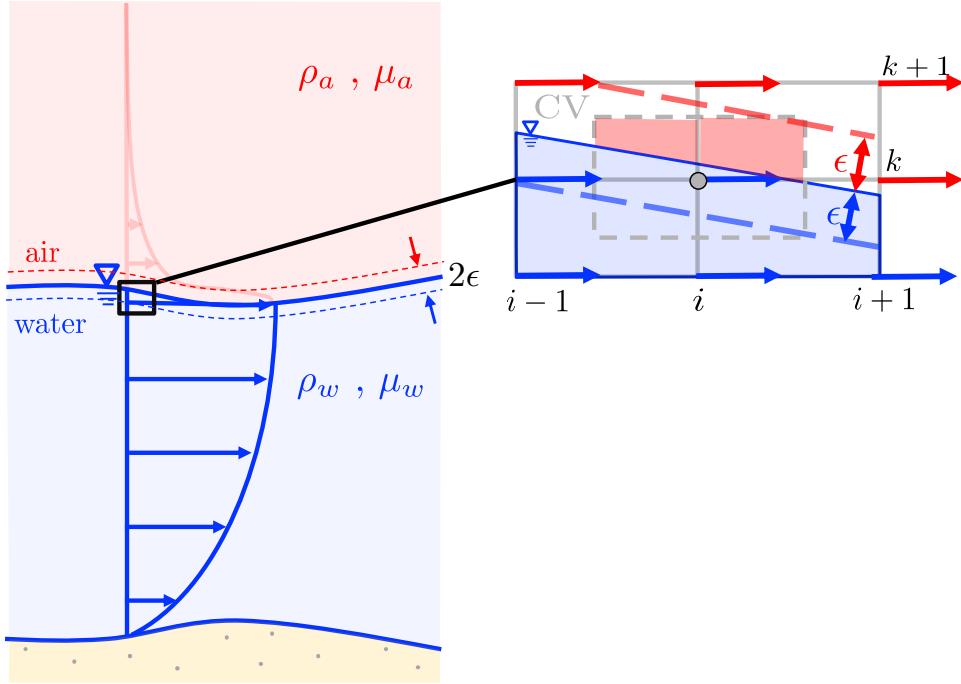


FIGURE 1.1. Illustration of a control volume intersected by the free surface. The phase interface causes abrupt changes in density and viscosity, which affect the discretisation of mass and momentum fluxes.

To mitigate such artifacts, it is crucial to enforce appropriate dynamic boundary conditions at the free surface. In the context of incompressible multiphase or free-surface flows, these boundary conditions typically require the continuity of normal and tangential stress at the interface. For flows involving air and water, this translates into enforcing a zero-gauge pressure at the interface (neglecting surface tension) and a shear-free condition when the tangential stress from the gas phase is negligible (Popinet, 2018; Tryggvason et al., 2011).

Several approaches have been proposed to enforce these conditions numerically. Interface-capturing methods such as the Volume-of-Fluid (VOF) and Level-Set methods rely on smoothing techniques or geometric reconstructions to approximate interface normals and curvatures, which are used in turn to apply interfacial forces and impose stress continuity

(Hirt & Nichols, 1981; Osher & Sethian, 1988; Scardovelli & Zaleski, 1999). However, many models implicitly assume a sharp interface without ensuring the correct transmission of shear and pressure across it. In such cases, the numerical method may inadequately resolve the stress discontinuity, particularly in under-resolved regions or when low-order schemes are used near the interface.

Recent developments (e.g. Carrica et al., 2007; Watanabe, 2022; Watanabe et al., 2008) have incorporated ghost fluid methods (Fedkiw et al., 1999) or modified finite difference/finite volume stencils to better account for the discontinuities in pressure and velocity gradients across the interface. These techniques allow for the imposition of jump conditions in a conservative framework while maintaining the accuracy of the numerical scheme.

Despite these advances, some models still rely on simplistic treatments that neglect the dynamic boundary conditions altogether or use overly diffusive interface representations, resulting in inaccurate predictions of wave propagation, free-surface patterns , or near-surface vorticity (Lin, 2008; Watanabe, 2022).

To overcome these challenges, this work adopts a single-phase flow formulation combined with explicit dynamic boundary conditions at the free surface (Carrica et al., 2007; Watanabe et al., 2008). The aim is to ensure physically consistent momentum transfer near the interface and to minimise spurious numerical effects that could compromise the fidelity of the simulation.

1.5. Free surface dynamics in fluid structure interaction in coastal and fluvial environments

In coastal and fluvial environments, the interaction between the flow and in-stream structures encompasses a wide range of physical processes where surface waves or riverine flows interact with natural or engineered structures such as breakwaters, marine energy devices, bridge piers, levees, and vegetation (Chen et al., 2014). A critical feature of these interactions is the dynamic response of the free surface, which undergoes substantial deformation due to both the incoming flow conditions and the geometry of the obstacle. In

many cases, the resulting free-surface patterns, which may include wave run-up, plunging jets, and local splashing, are directly responsible for generating turbulence, entraining air, and producing strong unsteady forces on the structures (Lin, 2008).

Accurately capturing the evolution of the free surface in three dimensions is therefore essential for understanding the hydrodynamic loads and the local flow physics. For instance, in the case of wave breaking against a vertical wall or the local acceleration of flow around bridge foundations during flooding, the shape and motion of the free surface can govern the intensity and structure of the turbulent wake (Ingram et al., 2009; Kara et al., 2015; Khosronejad et al., 2020b).

This, in turn, affects sediment transport, scour, and potential structural damage. Turbulence generated by wave–structure interaction is often highly anisotropic and unsteady, with coherent vortical structures forming near the interface and propagating downstream. These effects are particularly important in fluvial settings, where shallow depths and complex topographies amplify the coupling between surface dynamics and near-bed shear layers (Flora & Khosronejad, 2021; Khosronejad et al., 2019).

Free-surface three-dimensional Computational Fluid Dynamics (CFD) solvers have become a key tool for studying these phenomena, offering the ability to resolve the full spatial variability of both the flow field and the free surface (Huang et al., 2022). Unlike depth-averaged or two-dimensional models, 3D approaches can simulate oblique wave attack, lateral flows, and vertical accelerations, all of which are significant in real-world configurations. A number of studies have employed such solvers to investigate wave loading (Chen et al., 2014; Hu et al., 2016), run-up and overtopping (Ha et al., 2014; Ingram et al., 2009; Orszaghova et al., 2014), and flow separation (Kara et al., 2015; Khosronejad et al., 2019) around structures (see Figure 1.2), with particular emphasis on resolving the interface and its role in turbulence production. In riverine contexts, 3D solvers have also been applied to model local scour, complex eddy formation, and flow–vegetation interaction (Flora & Khosronejad, 2021; Khosronejad et al., 2020a; Khosronejad et al., 2020b),

demonstrating the necessity of capturing free-surface variability in order to predict near-field hydrodynamics with fidelity.

Nevertheless, the use of 3D free-surface flow solvers in coastal and fluvial environments is still limited. The requirement for high grid resolution near the free surface and within boundary layers results in high computational costs, especially when simulating large domains or long-duration events. In addition, turbulence modelling introduces further uncertainty, as common approaches such as RANS may underpredict localised turbulent energy, while LES or hybrid schemes remain computationally expensive for practical applications (Bagherizadeh et al., 2021). As such, careful validation against laboratory or field measurements is indispensable, particularly when assessing quantities such as pressure impulses, overtopping volumes, or sediment transport rates.

Within this framework, the present 3D solver is developed to address the interaction between the flow and rigid structures problems in both coastal and fluvial domains, with a particular focus on the interaction between free-surface deformation and turbulent structures. By resolving the evolution of the interface and the flow field simultaneously, the model aims to capture key mechanisms such as flow separation, vortex formation, and hydrodynamic loading on structures. This capability is essential for advancing both scientific understanding and engineering design in environments where the coupling between surface dynamics and hydrodynamic forces governs the system response.

1.6. Perspectives for coastal and fluvial modelling in renewable energy

Hydrokinetic turbines are devices that generate electricity by harnessing the kinetic energy of flowing water without the need for a dam or significant head difference (Khan et al., 2009). They operate in coastal and fluvial environments with complex hydrodynamics, characterised by a wide range of temporal and spatial scales and a variety of processes (see Figure 1.3). All these phenomena take place in a domain vertically bounded by the bottom boundary (i.e., sea or river bed) and the free surface, whose dynamics may significantly

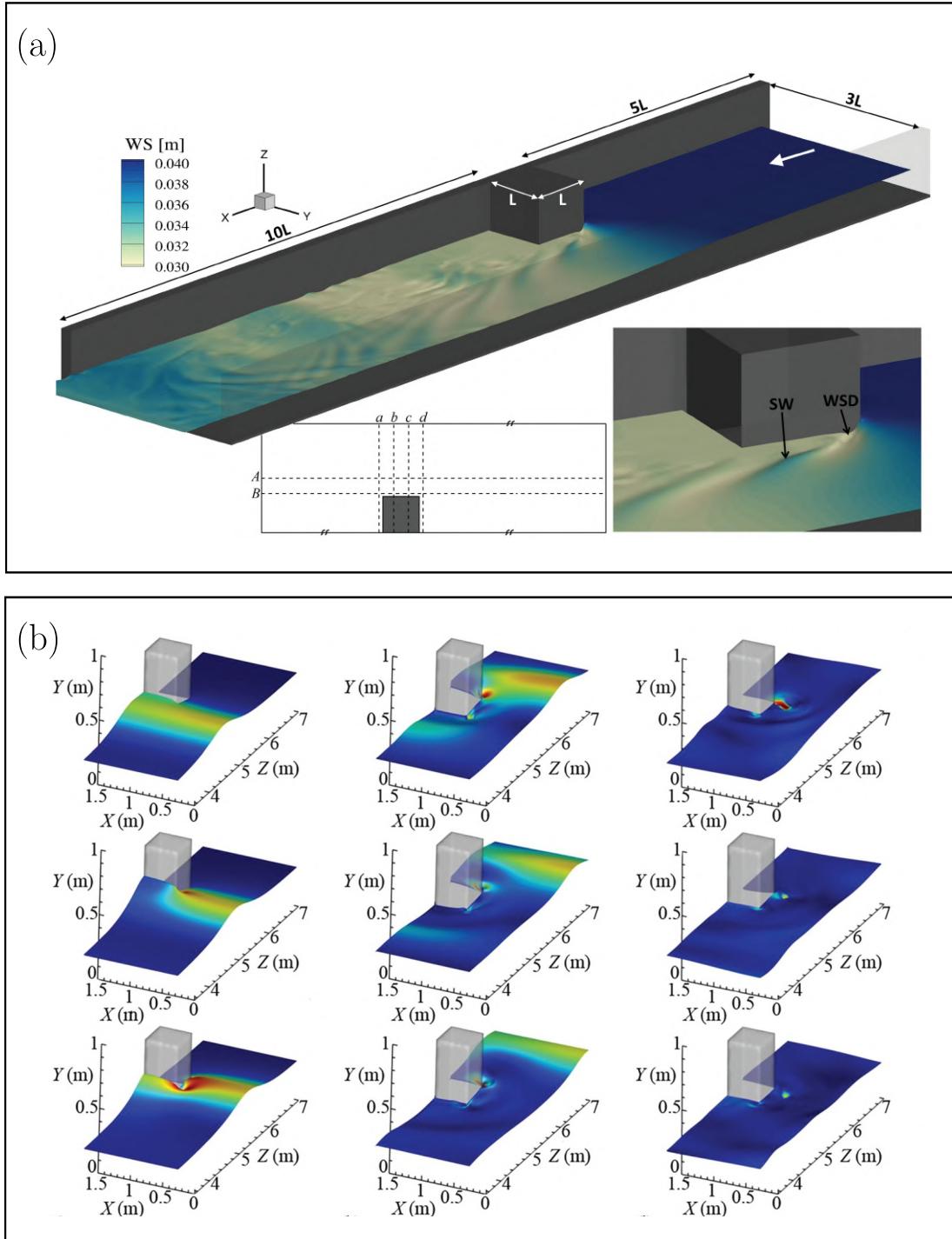


FIGURE 1.2. Examples of recent studies of free-surface flows interacting with structures. (a) Channel flow interacting with a bridge-abutment (extracted from Kara et al. (2015), where SW corresponds to standing wave, WSD to the water-surface dip, and WS to the water surface position). (b) Solitary wave interacting with a rectangular structure (extracted from Bagherizadeh et al. (2021), where the colour corresponds to the velocity magnitude at the free surface.)

influence both the inflow conditions encountered by the devices and the evolution of their downstream wakes (Li et al., 2021; Ouro et al., 2024).

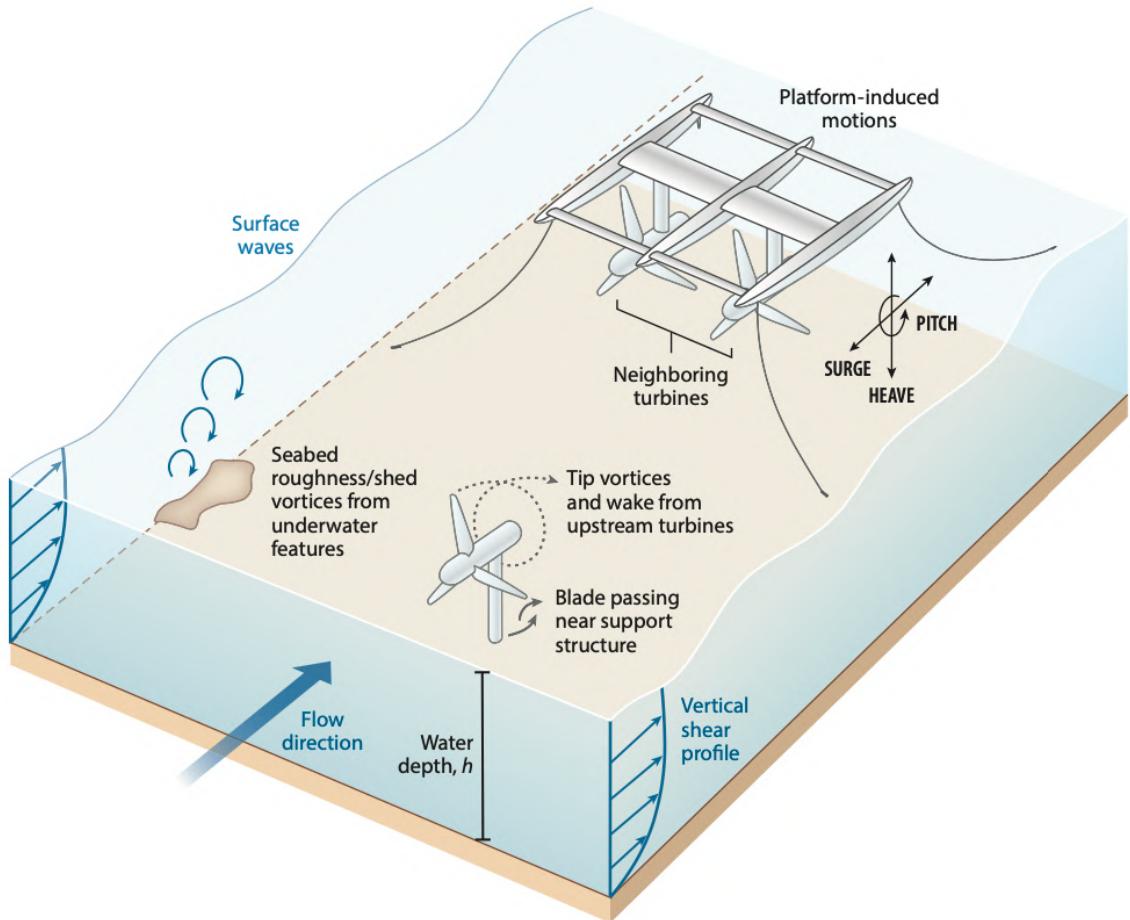


FIGURE 1.3. The flow environment where tidal turbines operate and sources of important flow phenomena experienced by devices. Extracted from Adcock et al. (2021).

Turbulent wakes behind the devices, exhibit large-scale coherent motions induced by the interaction of the rotor with the incoming flow. These features include complex phenomena such as rotational tip-vortex advection, hub vortex instabilities and wake meandering (Chamorro et al., 2013; Chamorro et al., 2015; Howard et al., 2015; Kang et al., 2014; Li et al., 2021; Ouro et al., 2017; Ouro & Stoesser, 2019; Sandoval et al., 2021).

The complex interaction between the flow and the turbine rotor can be modelled using the so-called momentum sink approach (Gajardo et al., 2019). In this method, turbines

are represented by introducing external forces into the momentum conservation equations, calculated through actuator models based on the geometry and characteristics of the turbines. This simplification captures the essential dynamics of flow–turbine interaction without fully resolving the turbine geometry, thereby reducing computational cost and complexity (Sandoval et al., 2021; Soto-Rivas et al., 2019). The sink term is incorporated directly into the momentum equation as follows:

$$\frac{\partial \mathbf{u}}{\partial t} + \mathbf{C}(\mathbf{u}) - \mathbf{D}(\mathbf{u}) + \mathbf{G}(p) + \mathbf{S}_g + \mathbf{S}_T = 0 \quad (1.20)$$

where \mathbf{S}_T is the non-dimensional local force per unit volume introduced into the flow by the actuator model. In this PhD project, we used the incompressible flow solver without the free-surface extension to investigate the influence of three different actuator models on the wake dynamics of a benchmark turbine case presented in Stallard et al. (2013). The actuator models considered were the Actuator Disc Model (ADM), a model based on Blade Element Momentum (BEM) theory, and the Actuator Line Model (ALM).

The simplest is the ADM (Burton et al., 2011; Soto-Rivas et al., 2019) that represents the turbines by incorporating a horizontal force per unit volume within the disk region, which encompasses the entire volume occupied by the turbine. In this approach, the devices are represented by a thrust force and the geometrical details of the turbine and blades are not considered. A more complex model is BEM approach (Creech et al., 2017; Gajardo et al., 2019), which implicitly incorporates the turbine rotation defined by the tip speed ratio (TSR) and the blade geometry, to compute the local instantaneous forces in the actuator volume as a function of the disk radius. Finally, the most sophisticated tested approach is the ALM (Sørensen & Shen, 2002), which represents the individual blades by distributed forces along lines that rotates at a defined TSR. In ALM, the forces exerted on the flow field are parametrised by using data from the blade geometry and the local hydrodynamics.

A complete analysis is provided in Sandoval et al. (2021), but general findings of this study can be summarised as follows: (i) All three models accurately reproduce the

mean flow and velocity deficit beyond $4D$ downstream, with errors under 20 %. (ii) ADM produces a wider wake due to large arch-shaped coherent vortices formed behind the disk, driven by the resolved instantaneous velocity (see Figure 1.4). (iii) BEM captures the mean wake features more effectively (Gajardo et al., 2019; Gotelli et al., 2019), though some rotational effects remain unresolved. (iv) The well-known wake asymmetry is only captured by the ALM, that is to say, incorporating the turbine rotation explicitly (see Figure 1.5)

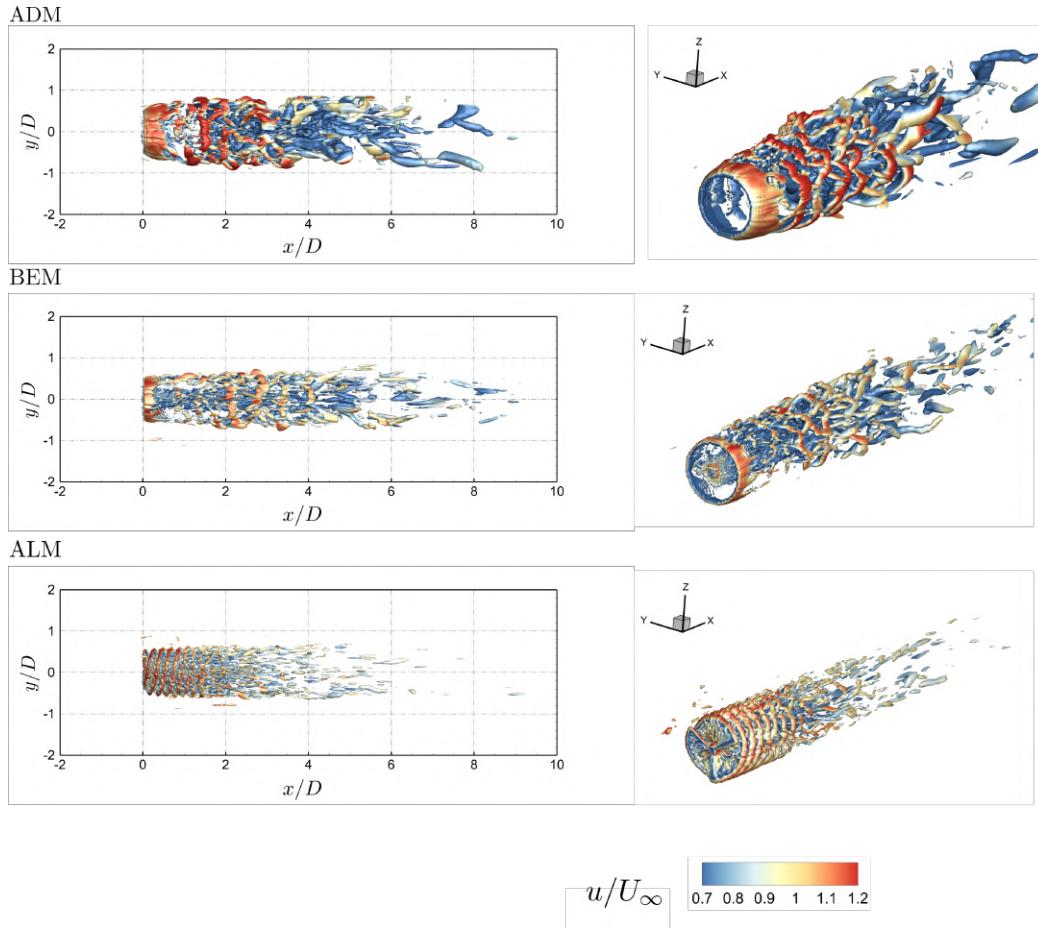


FIGURE 1.4. Instantaneous coherent structures visualized with by using the q -criterion (Hunt et al., 1988) coloured by the streamwise velocity for ADM, BEM, and ALM approach (Sandoval et al., 2021).

As recent studies have shown (Li et al., 2021; Ouro et al., 2024), the wake dynamics described above can be significantly influenced by transient free-surface deformations (e.g.,

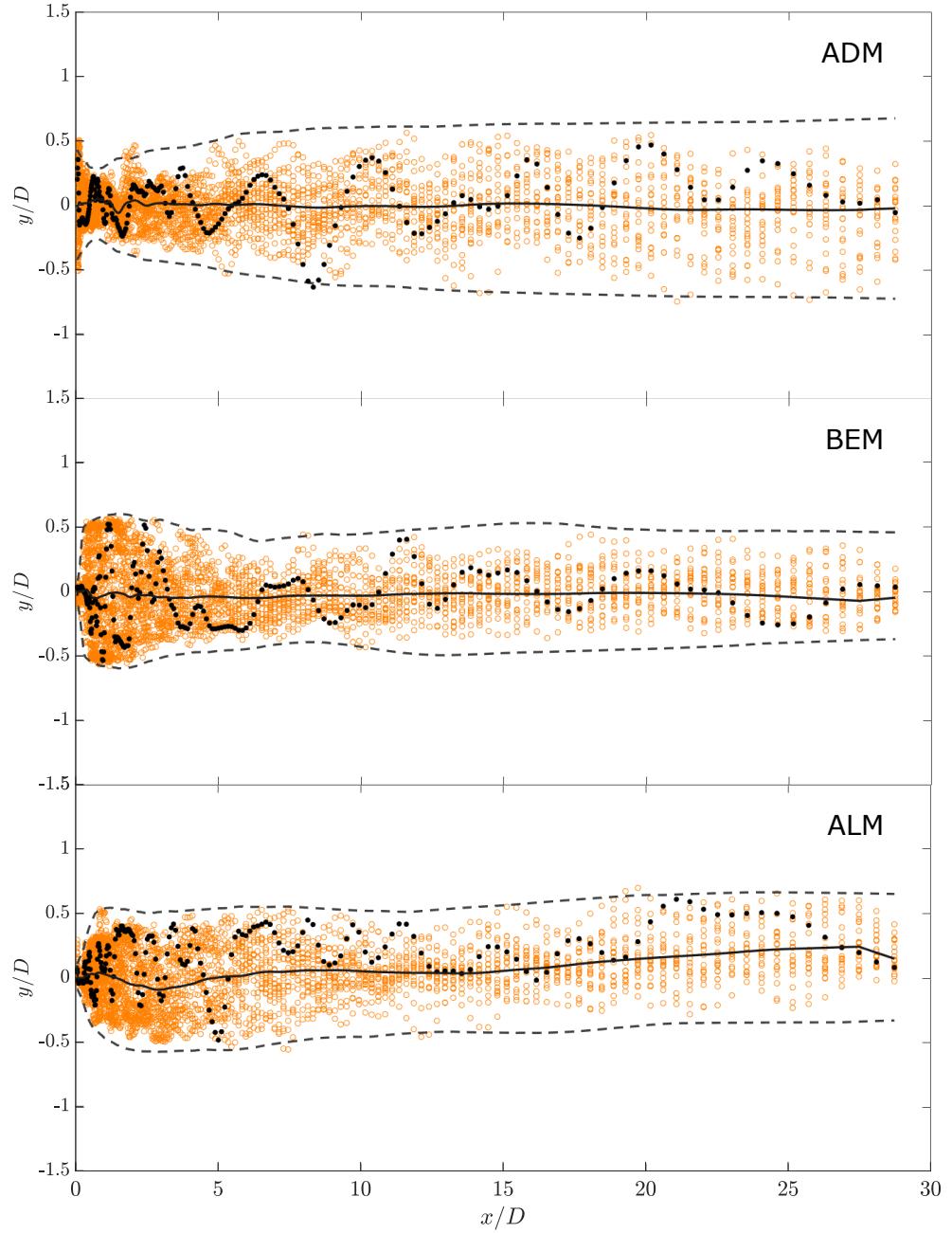


FIGURE 1.5. Top view of the location of the minimum velocity at the centreline of the turbine. Empty orange circles mark the instantaneous location. Filled black circles highlight one of the instantaneous solutions. Black centreline is the time-average location of the velocity minimum. Black dashed-lines is the envelopment of the wake (Sandoval et al., 2021).

waves or local variations in hydraulic head). Extending the incompressible flow solver used in Sandoval et al. (2021) to simulate the operation of tidal turbines within free-surface flows represents a key step towards providing a robust numerical framework for modelling renewable energy devices under fully realistic conditions. In particular, the physically-consistent free-surface boundary condition approach proposed in this thesis will allow to accurately study the interaction of the tip vortices with the free-surface and the free-surface response to local pressure variations.

1.7. Objectives and Thesis Structure

The objective of this thesis is to develop a novel numerical framework to simulate three-dimensional free-surface flows. The specific objectives consist of (i) implementing a robust numerical method to impose consistent normal and tangential boundary conditions at the free surface, (ii) preserving global volume throughout the simulation by using a geometric-based reinitialisation method for the level-set function, and (iii) studying the interaction of free-surface flows with in-stream obstacles, highlighting the vortical dynamics that arises from the interaction. The thesis is structured as follows:

In Chapter 2, we present the computational fluid dynamics (CFD) model used to solve the mass, momentum, and turbulence equations within a single-phase flow framework. The numerical algorithms, schemes, and techniques employed to compute the fluxes, solve the resulting systems, and advance the solution in time are described in detail. Special emphasis is placed on the numerical treatment near the free surface to satisfy the dynamic boundary conditions while preserving the high-order accuracy of the methods.

In Chapter 3, we present the model used to capture and compute the evolution of the free surface. The level set method is described, along with the numerical techniques employed to achieve high-order accuracy in both time and space. Subsequently, the mass-preserving geometric redistancing method is detailed, with particular emphasis on the free-surface reconstruction algorithm and the procedures for ensuring both local and global mass conservation.

In Chapter 4, we present an overview of how the methods introduced in Chapters 2 and 3 are integrated to construct the complete solver. We then present individual validation cases for the components related to free-surface flows, followed by cases that assess the performance of the full numerical solver.

Then, in Chapter 5, we present an analysis of the flow around two obstacles with variable separation distances, subjected to the action of a solitary wave. Flow and free-surface patterns are discussed, along with their effects on run-up, bed shear stress, and the hydrodynamic loading exerted on the obstacles for the different separation values considered.

Finally, in Chapter 6, we outline the main conclusions of this work and discuss perspectives and directions for future research.

In addition, four Appendix sections with the transformation of the Navier-Stokes equations into generalised curvilinear coordinates, the Block-Diagonal algorithm to solve the Navier-Stokes equations, the derivation of the Dynamic Boundary conditions, and the derivation of the three-dimensional Simpson's rule for numerical integration in generalised curvilinear coordinates, are provided.

CHAPTER 2

COMPUTATIONAL FLUID DYNAMICS MODEL

2. COMPUTATIONAL FLUID DYNAMICS MODEL

In this section, we describe the mathematical model used to reproduce three-dimensional free-surface flows, as well as the numerical algorithms developed and implemented to carry out high-fidelity computational simulations of them.

2.1. Governing equations of incompressible free-surface flows

The equations to be solved are the single-phase three-dimensional spatially-filtered continuity and Navier-Stokes equations for instantaneous and incompressible flow (Carrica et al., 2007; Rodi et al., 2013) using Large-Eddy Simulation (LES) based on the dynamic Smagorinsky model (Germano et al., 1991).

$$\frac{\partial u_i}{\partial x_j} = 0 \quad (2.1)$$

$$\frac{\partial u_i}{\partial t} + \frac{\partial (u_i u_j)}{\partial x_j} = -\frac{\partial p}{\partial x_i} + \frac{1}{Re} \frac{\partial}{\partial x_j} \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right) - \frac{\partial \tau_{ij}}{\partial x_j} + S_{g_i} \quad (2.2)$$

$$\tau_{ij} - \frac{1}{3} \delta_{ij} \tau_{kk} = -2\nu_t \mathcal{S}_{ij}, \quad \nu_t = C_s \Delta^2 |\mathbf{S}| \quad (2.3)$$

Where u_i and p are the filtered velocity and pressure, respectively, S_{g_i} is the gravitational acceleration, and ν_t is the turbulent viscosity. For the LES model, C_s corresponds to the Smagorinsky constant, \mathcal{S}_{ij} is the strain rate tensor, Δ is the grid filter and $|\mathbf{S}| = \sqrt{2\mathcal{S}_{ij}\mathcal{S}_{ij}}$. This approach is able to provide accurate descriptions of turbulent flow fields, yielding detailed solutions by resolving all the scales larger than the size of the computational grid. This produces information on the flow unsteadiness, turbulence characteristics and vortical structures.

To solve the free-surface flow, we adopt the single-phase level-set method presented by Carrica et al. (2007). In contrast to the standard level-set method for incompressible flows, the single-phase level-set method is applied only to solve the flow field in the water phase.

Some advantages of this approach are that the interface remains sharp, the computation is performed within a fluid with uniform properties and minor computations are needed in the air, where only extrapolated velocities are needed. All these characteristics make the problem considerably easier to converge. This method considers suitable interpolations at the water/air interface, which enforce appropriate momentum jump conditions (Watanabe et al., 2008)

The equations are transformed into a generalized curvilinear coordinate system to handle straightforwardly multi-directional flows and simulate complex bathymetries (a detail description of the transformation procedure is presented in Appendix A). They can be written in vector format, in a generalised curvilinear coordinate system and in strong conservation form as follows (tensors are denoted using boldface straight symbols and vector boldface italic ones),

$$\Gamma \frac{\partial \mathbf{Q}}{\partial t} + J \frac{\partial}{\partial \xi^j} (\mathbf{F}_L^j + \mathbf{F}_N^j + \mathbf{F}_v^j) + \mathbf{S}_g = 0 \quad (2.4)$$

where,

$$\Gamma = \text{diag} [0 \ 1 \ 1 \ 1]$$

$$\mathbf{Q} = [P, u_1, u_2, u_3]^T \quad (2.5)$$

$$\begin{aligned} \mathbf{F}_L^j &= \frac{1}{J} \left[U^j, P \frac{\partial \xi^j}{\partial x_1}, P \frac{\partial \xi^j}{\partial x_2}, P \frac{\partial \xi^j}{\partial x_3} \right]^T \\ \mathbf{F}_N^j &= \frac{1}{J} [0, u_1 U^j, u_2 U^j, u_3 U^j]^T \\ \mathbf{F}_v^j &= \frac{1}{J} \left(\frac{1}{Re} + \nu_{sgs} \right) \left[0, \frac{\partial u_1}{\partial \xi^m} g^{mj} + R_{m1} \frac{\partial \xi^j}{\partial x_m}, \frac{\partial u_2}{\partial \xi^m} g^{mj} + R_{m2} \frac{\partial \xi^j}{\partial x_m}, \frac{\partial u_3}{\partial \xi^m} g^{mj} + R_{m3} \frac{\partial \xi^j}{\partial x_m} \right]^T \\ \mathbf{S}_g &= \left(-\frac{1}{Fr^2} \right) [0, 0, 0, 1]^T \end{aligned}$$

In these equations, $P = p + 2/3k$, where k is the resolved turbulent kinetic energy and p is the pressure divided by the density. u_i ($i = 1, 2, 3$) are the Cartesian velocity components, x_i are the Cartesian coordinates, $\xi_{x_i}^j = \partial \xi^j / \partial x_i$ are the metrics of the geometric transformation, $J = |\partial \xi^j / \partial x_i|$ is the Jacobian of the transformation, $U^j = u_i \xi_{x_i}^j$ are the

contravariant velocity components, $g^{ij} = \xi_{x_k}^i \xi_{x_k}^j$ are the components of contravariant metric tensor, Re is the Reynolds number, ν_{SGS} is the sub-grid scale viscosity and tensor R_{ij} is defined as $R_{ij} = (\partial u_j / \partial \xi^k) \xi_j^k$. S is the source term which includes the gravity effects.

2.1.1. Pressure-splitting

In order to incorporate the effect of the gravity into the momentum conservation equations, the pressure-split formulation of Casulli (1999) and Fringer et al. (2006) is adopted. This formulation allows the flow to be locally forced by horizontal gradients of hydrostatic pressure.

Following the work of the aforementioned authors, the total pressure p is split into its hydrostatic (p_h) and dynamic (p_d) components, such that $p = p_h + p_d$. The hydrostatic pressure in non-dimensional form is defined as

$$p_h = \frac{h}{Fr^2} \quad (2.6)$$

Where h is the water depth defined as the vertical distance to the free-surface, as shown in Figure 2.1.

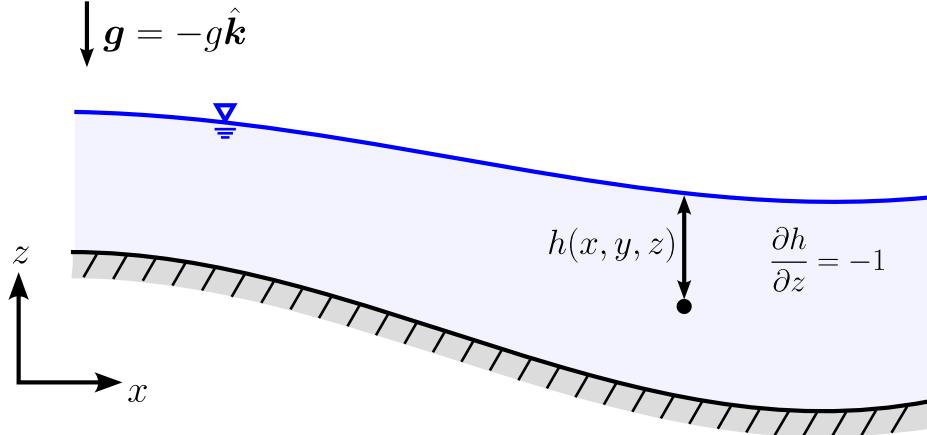


FIGURE 2.1. Local water depth.

If we introduce this decomposition into the equation 2.2 and consider that the gravity is aligned with the z direction, we obtain

$$\frac{\partial u}{\partial t} + \frac{\partial(uu_j)}{\partial x_j} = -\frac{\partial p_d}{\partial x} + \frac{1}{Re} \frac{\partial}{\partial x_j} \left(\frac{\partial u}{\partial x_j} + \frac{\partial u_j}{\partial x} \right) - \frac{\partial \tau_{1j}}{\partial x_j} - \frac{1}{Fr^2} \frac{\partial h}{\partial x}, \quad (2.7)$$

$$\frac{\partial v}{\partial t} + \frac{\partial(vu_j)}{\partial x_j} = -\frac{\partial p_d}{\partial y} + \frac{1}{Re} \frac{\partial}{\partial x_j} \left(\frac{\partial v}{\partial x_j} + \frac{\partial u_j}{\partial y} \right) - \frac{\partial \tau_{2j}}{\partial x_j} - \frac{1}{Fr^2} \frac{\partial h}{\partial y}, \quad (2.8)$$

$$\frac{\partial w}{\partial t} + \frac{\partial(wu_j)}{\partial x_j} = -\frac{\partial p_d}{\partial z} + \frac{1}{Re} \frac{\partial}{\partial x_j} \left(\frac{\partial w}{\partial x_j} + \frac{\partial u_j}{\partial z} \right) - \frac{\partial \tau_{3j}}{\partial x_j}. \quad (2.9)$$

Where $\partial h/\partial z = -1$. With this formulation, the distribution of the free surface is explicitly incorporated into the governing equations.

2.2. Dynamic free-surface boundary conditions

In this subsection, we present the extension to generalised curvilinear coordinates of the method proposed by Watanabe et al., 2008 to satisfy tangential and normal dynamic boundary conditions for free-surface flows.

2.2.1. Tangential Dynamic Boundary Condition (TDBC)

The shear-free tangential dynamic boundary condition for a free surface reads

$$\left\{ \left(\frac{\partial u_i}{\partial x_j} \right) + \left(\frac{\partial u_j}{\partial x_i} \right)^S \right\} n_j t_i = 0 \quad (2.10)$$

Where n_i and t_i represent normal and tangential directions respectively (is analogous for s tangential direction). The normal vector pointing towards the air phase is defined as $\mathbf{n} = -\nabla\phi$. The level-set function gradient is calculated using the second-order ENO scheme presented by Sussman et al. (1998) and adapted to curvilinear coordinates by Kang and Sotiropoulos (2012).

$$\frac{\partial\phi}{\partial x_i} = \frac{\partial\phi}{\partial\xi^m} \frac{\partial\xi^m}{\partial x_i} \quad (2.11)$$

Where $\partial\xi^m/\partial x_i$ correspond to the metrics of the transformation between the Cartesian and the curvilinear coordinates, and the spatial derivative of ϕ in the curvilinear directions are given by

$$\frac{\partial\phi}{\partial\xi^1} = \begin{cases} \frac{\partial\phi^+}{\partial\xi^1} & \text{sgn}(\phi_{i,j,k})(\phi_{i+1,j,k} - \phi_{i,j,k}) < 0 \quad \text{and} \\ \frac{\partial\phi^-}{\partial\xi^1} & \text{sgn}(\phi_{i,j,k})(\phi_{i,j,k} - \phi_{i-1,j,k}) < -\text{sgn}(\phi_{i,j,k})(\phi_{i+1,j,k} - \phi_{i,j,k}), \\ & \text{sgn}(\phi_{i,j,k})(\phi_{i,j,k} - \phi_{i-1,j,k}) > 0 \quad \text{and} \\ \frac{1}{2} \left(\frac{\partial\phi^+}{\partial\xi^1} + \frac{\partial\phi^-}{\partial\xi^1} \right) & \text{sgn}(\phi_{i,j,k})(\phi_{i+1,j,k} - \phi_{i,j,k}) > -\text{sgn}(\phi_{i,j,k})(\phi_{i,j,k} - \phi_{i-1,j,k}), \\ & \text{otherwise,} \end{cases} \quad (2.12)$$

The sign function sgn is given by:

$$\text{sgn}(\phi) = \begin{cases} -1 & , \quad \phi < 0 \\ 0 & , \quad \phi = 0 \\ 1 & , \quad \phi > 0 \end{cases} \quad (2.13)$$

And $\frac{\partial\phi^+}{\partial\xi^1}, \frac{\partial\phi^-}{\partial\xi^1}$ are defined as

$$\begin{aligned} \left(\frac{\partial\phi}{\partial\xi^1} \right)^+ &= (\phi_{i+1,j,k} - \phi_{i,j,k}) \\ &\quad - \frac{1}{2} M (\phi_{i+1,j,k} - 2\phi_{i,j,k} + \phi_{i-1,j,k}, \phi_{i+2,j,k} - 2\phi_{i+1,j,k} + \phi_{i,j,k}), \end{aligned} \quad (2.14)$$

$$\begin{aligned} \left(\frac{\partial\phi}{\partial\xi^1} \right)^- &= (\phi_{i,j,k} - \phi_{i-1,j,k}) \\ &\quad - \frac{1}{2} M (\phi_{i+1,j,k} - 2\phi_{i,j,k} + \phi_{i-1,j,k}, \phi_{i,j,k} - 2\phi_{i-1,j,k} + \phi_{i-2,j,k}), \end{aligned} \quad (2.15)$$

Where M is defined as

$$M(a, b) = \begin{cases} a & |a| < |b|, \\ b & |b| \leq |a|. \end{cases} \quad (2.16)$$

To determine the tangential vector system, we follow the procedure proposed by Albin et al. (2016), which allows to obtain tangential vectors aligned with the principal curvature directions without computing an eigenvalue problem. This approach provided a robust criterion based on the local geometry of the free surface with low computational cost. For the velocity gradient, we adopted the method proposed by Watanabe et al., 2008

$$\frac{\partial u_i}{\partial x_j} (\boldsymbol{x}_s + \boldsymbol{\alpha}) = \frac{\partial u_i}{\partial x_j} (\boldsymbol{x}_s) + \mathbf{f}_{ij}(\boldsymbol{\alpha}) \quad (2.17)$$

Where \mathbf{f}_{ij} is a linear function to approximate the (i, j) component of the velocity gradient tensor and $\boldsymbol{\alpha} = (\alpha, \beta, \gamma)$ is the position vector from the free-surface (\boldsymbol{x}_s). An schematic representation of the method is presented in 2.2

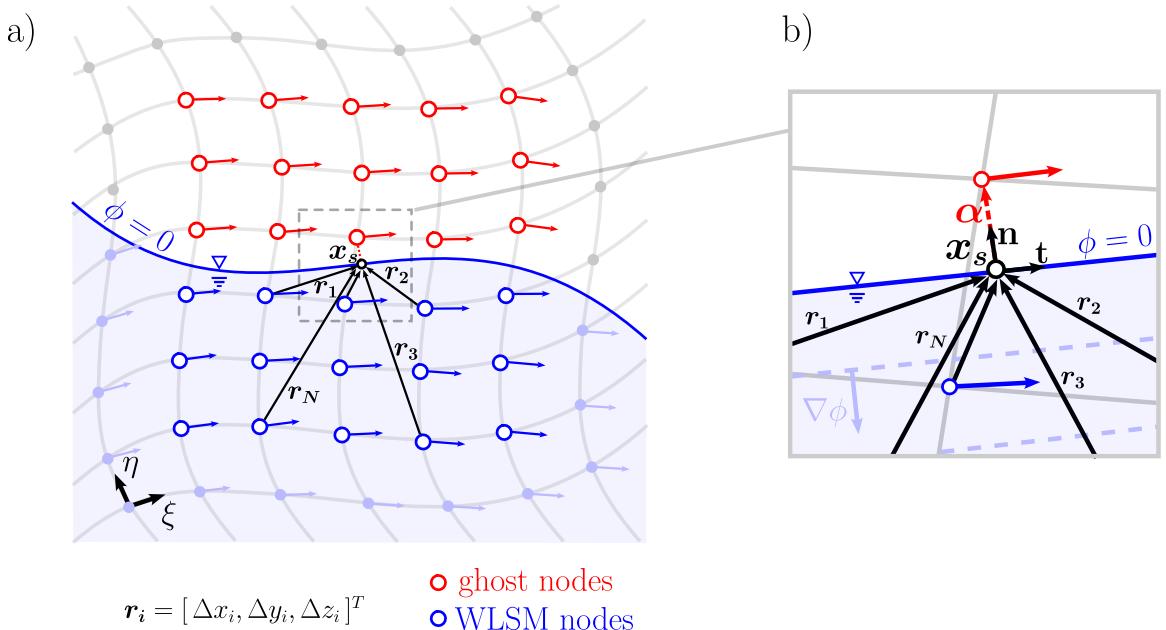


FIGURE 2.2. Ghost nodes scheme. (a) Nodes in the vicinity of the free surface used in the extrapolation algorithm. Only water nodes (blue empty circles) are considered for extrapolation. (b) Zoom on the free-surface vicinity.

The linear correction function \mathbf{f}_{ij} reads.

$$\mathbf{f}_{ij}(\boldsymbol{\alpha}) = a_{ij} + b_{ij}\alpha + c_{ij}\beta + d_{ij}\gamma \quad (2.18)$$

Replacing the velocity gradient at the free surface with the proposed approximation, we obtain

$$\left\{ \left(\frac{\partial u_i}{\partial x_j} (\mathbf{x}_s + \boldsymbol{\alpha}) - \mathbf{f}_{ij}(\boldsymbol{\alpha}) \right) + \left(\frac{\partial u_j}{\partial x_i} (\mathbf{x}_s + \boldsymbol{\alpha}) - \mathbf{f}_{ji}(\boldsymbol{\alpha}) \right) \right\} n_j t_i = 0 \quad (2.19)$$

The local velocity gradient at $(\mathbf{x}_s + \boldsymbol{\alpha})$ is computed in generalised curvilinear coordinates as follows.

$$\frac{\partial u_i}{\partial x_j} (\mathbf{x}_s + \boldsymbol{\alpha}) = \frac{\partial u_i}{\partial \xi^m} \frac{\partial \xi^m}{\partial x_j} (\mathbf{x}_s + \boldsymbol{\alpha}) \quad (2.20)$$

Where $(\partial u_i / \partial \xi^m)$ and $(\partial \xi^m / \partial x_j)$ can be directly computed over the curvilinear grid nodes. Rearranging terms, we can express the system as follows.

$$\mathbf{f}_{ij}(\boldsymbol{\alpha}) n_j t_i + \mathbf{f}_{ji}(\boldsymbol{\alpha}) n_j t_i = \left\{ \left(\frac{\partial u_i}{\partial \xi^m} \frac{\partial \xi^m}{\partial x_j} (\mathbf{x}_s + \boldsymbol{\alpha}) \right) + \left(\frac{\partial u_j}{\partial \xi^m} \frac{\partial \xi^m}{\partial x_i} (\mathbf{x}_s + \boldsymbol{\alpha}) \right) \right\} n_j t_i \quad (2.21)$$

Which can be expressed in matrix form

$$\mathbf{T}_t \cdot \mathbf{A} = \mathbf{B}_t \quad (2.22)$$

Where (summation symbols were added for a better understanding)

$$\begin{aligned} \mathbf{T}_t = & [2n_1 t_1, 2n_1 t_1 \alpha, 2n_1 t_1 \beta, 2n_1 t_1 \gamma \dots (n_i t_j + n_j t_i) + (n_i t_j + n_j t_i) \alpha, \\ & (n_i t_j + n_j t_i) \beta, (n_i t_j + n_j t_i) \gamma \dots 2n_3 t_3, 2n_3 t_3 \alpha, 2n_3 t_3 \beta, 2n_3 t_3 \gamma] \end{aligned} \quad (2.23)$$

$$\mathbf{A} = [a_{11}, b_{11}, c_{11}, d_{11}, a_{21}, b_{21}, c_{21}, d_{21}, \dots, a_{ij}, b_{ij}, c_{ij}, d_{ij}, \dots, a_{33}, b_{33}, c_{33}, d_{33}]^T \quad (2.24)$$

$$\mathbf{B}_t = \sum_{m,j,i=1}^3 \left\{ \left(\frac{\partial u_i}{\partial \xi^m} \frac{\partial \xi^m}{\partial x_j} (\mathbf{x}_s + \boldsymbol{\alpha}) \right) + \left(\frac{\partial u_j}{\partial \xi^m} \frac{\partial \xi^m}{\partial x_i} (\mathbf{x}_s + \boldsymbol{\alpha}) \right) \right\} n_j t_i \quad (2.25)$$

These expressions are analogous for \underline{s} tangential direction. We can approximate the coefficient matrix \mathbf{A} using the Least-Square Method, using N velocity gradients on the nearest N grid nodes to the free surface location. The formulation of the final matrix system reads

$$\mathbf{T} \cdot \mathbf{A} = \mathbf{B} \quad (2.26)$$

Where

$$\mathbf{T} = \begin{bmatrix} \sum_{n=1}^N (\mathbf{T}_t(1))_n^2 & \sum_{n=1}^N (\mathbf{T}_t(1))_n (\mathbf{T}_t(2))_n & \cdots & \sum_{n=1}^N (\mathbf{T}_t(1))_n (\mathbf{T}_t(36))_n \\ \sum_{n=1}^N (\mathbf{T}_t(2))_n (\mathbf{T}_t(1))_n & \ddots & & \vdots \\ \vdots & & \sum_{n=1}^N (\mathbf{T}_t(I))_n (\mathbf{T}_t(J))_n & \vdots \\ \sum_{n=1}^N (\mathbf{T}_t(36))_n (\mathbf{T}_t(1))_n & \cdots & \cdots & \sum_{n=1}^N (\mathbf{T}_t(36))^2_n \\ \sum_{n=1}^N (\mathbf{T}_s(1))_n^2 & \sum_{n=1}^N (\mathbf{T}_s(1))_n (\mathbf{T}_s(2))_n & \cdots & \sum_{n=1}^N (\mathbf{T}_s(1))_n (\mathbf{T}_s(36))_n \\ \sum_{n=1}^N (\mathbf{T}_s(2))_n (\mathbf{T}_s(1))_n & \ddots & & \vdots \\ \vdots & & \sum_{n=1}^N (\mathbf{T}_s(I))_n (\mathbf{T}_s(J))_n & \vdots \\ \sum_{n=1}^N (\mathbf{T}_s(36))_n (\mathbf{T}_s(1))_n & \cdots & \cdots & \sum_{n=1}^N (\mathbf{T}_s(36))^2_n \end{bmatrix} \quad (2.27)$$

$$\begin{aligned}
\mathbf{B} = & \left[\sum_{n=1}^N \sum_{m,j,i=1}^3 \left\{ \left(\frac{\partial u_i}{\partial \xi^m} \frac{\partial \xi^m}{\partial x_j} (\mathbf{x}_s + \boldsymbol{\alpha}_n) \right) + \left(\frac{\partial u_j}{\partial \xi^m} \frac{\partial \xi^m}{\partial x_i} (\mathbf{x}_s + \boldsymbol{\alpha}_n) \right) \right\} n_j t_i (\mathbf{T}_t(1))_n , \dots \right. \\
& \dots \sum_{n=1}^N \sum_{m,j,i=1}^3 \left\{ \left(\frac{\partial u_i}{\partial \xi^m} \frac{\partial \xi^m}{\partial x_j} (\mathbf{x}_s + \boldsymbol{\alpha}_n) \right) + \left(\frac{\partial u_j}{\partial \xi^m} \frac{\partial \xi^m}{\partial x_i} (\mathbf{x}_s + \boldsymbol{\alpha}_n) \right) \right\} n_j t_i (\mathbf{T}_t(I))_n , \dots \\
& \dots \sum_{n=1}^N \sum_{m,j,i=1}^3 \left\{ \left(\frac{\partial u_i}{\partial \xi^m} \frac{\partial \xi^m}{\partial x_j} (\mathbf{x}_s + \boldsymbol{\alpha}_n) \right) + \left(\frac{\partial u_j}{\partial \xi^m} \frac{\partial \xi^m}{\partial x_i} (\mathbf{x}_s + \boldsymbol{\alpha}_n) \right) \right\} n_j t_i (\mathbf{T}_t(36))_n , \dots \\
& \dots \sum_{n=1}^N \sum_{m,j,i=1}^3 \left\{ \left(\frac{\partial u_i}{\partial \xi^m} \frac{\partial \xi^m}{\partial x_j} (\mathbf{x}_s + \boldsymbol{\alpha}_n) \right) + \left(\frac{\partial u_j}{\partial \xi^m} \frac{\partial \xi^m}{\partial x_i} (\mathbf{x}_s + \boldsymbol{\alpha}_n) \right) \right\} n_j s_i (\mathbf{T}_s(1))_n , \dots \\
& \dots \sum_{n=1}^N \sum_{m,j,i=1}^3 \left\{ \left(\frac{\partial u_i}{\partial \xi^m} \frac{\partial \xi^m}{\partial x_j} (\mathbf{x}_s + \boldsymbol{\alpha}_n) \right) + \left(\frac{\partial u_j}{\partial \xi^m} \frac{\partial \xi^m}{\partial x_i} (\mathbf{x}_s + \boldsymbol{\alpha}_n) \right) \right\} n_j s_i (\mathbf{T}_s(J))_n , \dots \\
& \left. \dots \sum_{n=1}^N \sum_{m,j,i=1}^3 \left\{ \left(\frac{\partial u_i}{\partial \xi^m} \frac{\partial \xi^m}{\partial x_j} (\mathbf{x}_s + \boldsymbol{\alpha}_n) \right) + \left(\frac{\partial u_j}{\partial \xi^m} \frac{\partial \xi^m}{\partial x_i} (\mathbf{x}_s + \boldsymbol{\alpha}_n) \right) \right\} n_j s_i (\mathbf{T}_s(36))_n \right]^T
\end{aligned} \tag{2.28}$$

The accuracy of the gradient estimation can be degraded in the presence of non-uniform or spatially biased stencils. This is a potential issue in the present formulation, as the stencil is determined by the instantaneous position of the free surface. To overcome these difficulties, the weighted least squares method is adopted, as it assigns higher importance to data points that are more relevant, typically those closer to the evaluation point. This controls the sensitivity of the approximation to local variations, leading to a more reliable and physically consistent gradient (White et al., 2019). In this work, a hybrid weighting function is constructed based on a combination of the inverse of the distance to the free surface and the alignment with the free-surface normal, such that nodes more aligned with it contribute more significantly to the system. The local weight is calculated as

$$\mathcal{W}_n = |\hat{\mathbf{r}}_n \cdot \mathbf{n}| e^{-\left(\frac{r_n}{\Delta s_{\max}}\right)^2} \tag{2.29}$$

Here, \mathbf{r}_n is the position vector from node n to the free surface (with $\hat{\mathbf{r}}_n$ its unit vector), \mathbf{n} is the unit normal vector at the free surface, and Δs_{\max} is the distance between the free surface and the furthest node within the considered neighbourhood. This formulation, using exponential modulation instead of the classical inverse-distance approach, avoids potential division by zero when the free surface is too close to background grid nodes (Wu & Fu, 2025). Finally, each entry of the matrix \mathbf{T} and the vector \mathbf{B} is multiplied its corresponding \mathcal{W}_n , and the resulting system conforms the weighted one.

The spare-matrix system is solved using the Singular Value Decomposition (SVD) method. The solution of the system provides with the coefficients of the approximation linear function \mathbf{f}_{ij} , which allows to compute the velocity gradients at the free surface. The approximated function satisfies the zero-shear condition imposed by (2.10). The velocity gradient at the free surface is then computed as follows.

$$\frac{\partial u_i}{\partial x_j}(\mathbf{x}_s) = \frac{\partial u_i}{\partial \xi^m} \frac{\partial \xi^m}{\partial x_j}(\mathbf{x}_s + \boldsymbol{\alpha}_*) - \mathbf{f}_{ij} \quad (2.30)$$

Where $(\mathbf{x}_s + \boldsymbol{\alpha}_*)$ is the position of the nearest node to the free surface in the water phase. The previous expression can be evaluated with the computed values of \mathbf{f}_{ij} , nodal velocity and metric transformation values. The velocity is computed at the free surface using a second-order Taylor expansion.

$$u_i(\mathbf{x}_s) = u_i(\mathbf{x}_s + \boldsymbol{\alpha}) - \frac{\partial u_i}{\partial x}(\mathbf{x}_s) \cdot \alpha - \frac{\partial u_i}{\partial y}(\mathbf{x}_s) \cdot \beta - \frac{\partial u_i}{\partial z}(\mathbf{x}_s) \cdot \gamma \quad (2.31)$$

Then, the velocity is extrapolated to the nodes in the air phase at the position $(\mathbf{x}_s + \boldsymbol{\alpha}_e)$ using a second-order Taylor approximation again.

$$u_i(\mathbf{x}_s + \boldsymbol{\alpha}_e) = u_i(\mathbf{x}_s) + \frac{\partial u_i}{\partial x}(\mathbf{x}_s) \cdot \alpha_e + \frac{\partial u_i}{\partial y}(\mathbf{x}_s) \cdot \beta_e + \frac{\partial u_i}{\partial z}(\mathbf{x}_s) \cdot \gamma_e \quad (2.32)$$

2.2.2. Normal Dynamic Boundary Condition (NDBC)

On the free surface, the normal dynamic boundary condition imposes a Dirichlet boundary condition for the pressure. For a zero atmospheric pressure, in non-dimensional form, this condition reads (Robertson et al., 2004):

$$p_{fs} n_i n_i + \left(\frac{1}{Re} + \nu_{SGS} \right) \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right) n_i n_j = 0 \quad (2.33)$$

Where p_{fs} is the dynamic component of the pressure as the hydrostatic one is zero at the free surface. The velocity gradient, and the normal direction at the free surface were previously calculated according to the tangential dynamic boundary condition, so the pressure at the free surface can be computed from it in order to satisfy the NDBC. To extrapolate pressure values to the ghost-fluid nodes, a procedure based on the Weighted Least Square Method to compute the pressure gradient at the free surface is applied (see below Figure 2.3).

With the pressure value at the free surface and the computed pressure at the nodes in the water nodes surrounding the free surface location, it is possible to formulate a set of equations as follows

$$\begin{aligned} \mathbf{r}_1 \cdot \nabla p \Big|_{fs} &= p_{fs} - p_1, \\ \mathbf{r}_2 \cdot \nabla p \Big|_{fs} &= p_{fs} - p_2, \\ &\vdots \\ \mathbf{r}_N \cdot \nabla p \Big|_{fs} &= p_{fs} - p_N. \end{aligned} \quad (2.34)$$

This system, in matrix form reads

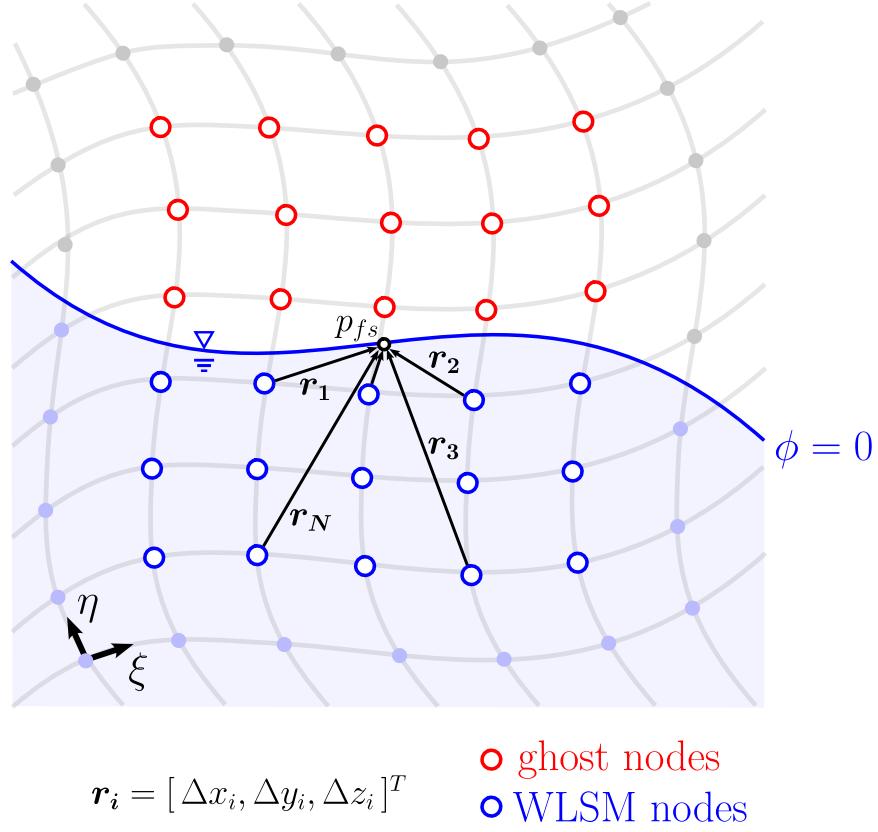


FIGURE 2.3. Ghost nodes scheme for pressure extrapolation.

$$\begin{bmatrix}
 \Delta x_1 & \Delta y_1 & \Delta z_1 \\
 \Delta x_2 & \Delta y_2 & \Delta z_2 \\
 \Delta x_3 & \Delta y_3 & \Delta z_3 \\
 \Delta x_4 & \Delta y_4 & \Delta z_4 \\
 \Delta x_5 & \Delta y_5 & \Delta z_5 \\
 \vdots & \vdots & \vdots \\
 \Delta x_N & \Delta y_N & \Delta z_N
 \end{bmatrix}
 \begin{bmatrix}
 \frac{\partial p}{\partial x} \\
 \frac{\partial p}{\partial y} \\
 \frac{\partial p}{\partial z}
 \end{bmatrix}
 =
 \begin{bmatrix}
 p_{fs} - p_1 \\
 p_{fs} - p_2 \\
 p_{fs} - p_3 \\
 p_{fs} - p_4 \\
 p_{fs} - p_5 \\
 \vdots \\
 p_{fs} - p_N
 \end{bmatrix} \quad (2.35)$$

The system above can be expressed as

$$\mathbf{R} \cdot \nabla p_{fs} = \mathbf{b} \quad (2.36)$$

Which corresponds to the Least Square Approximation of the system. We can easily move to a Weighted Least Square Method, where each weight represents the importance of the sample. We use the same weight used for the TDBC defined by the alignment of the vector \mathbf{r} with the normal direction of the free surface and its module. The weighted system becomes

$$\mathbf{R}_w \cdot \nabla p_{fs} = \mathbf{b}_w \quad (2.37)$$

Where

$$\mathbf{R}_w = \begin{bmatrix} w_1\Delta x_1 & w_1\Delta y_1 & w_1\Delta z_1 \\ w_2\Delta x_2 & w_2\Delta y_2 & w_2\Delta z_2 \\ w_3\Delta x_3 & w_3\Delta y_3 & w_3\Delta z_3 \\ w_4\Delta x_4 & w_4\Delta y_4 & w_4\Delta z_4 \\ w_5\Delta x_5 & w_5\Delta y_5 & w_5\Delta z_5 \\ \vdots & \vdots & \vdots \\ w_N\Delta x_N & w_N\Delta y_N & w_N\Delta z_N \end{bmatrix} \quad (2.38)$$

$$\mathbf{b}_w = \begin{bmatrix} w_1(p_{fs} - p_1) \\ w_2(p_{fs} - p_2) \\ w_3(p_{fs} - p_3) \\ w_4(p_{fs} - p_4) \\ w_5(p_{fs} - p_5) \\ \vdots \\ w_N(p_{fs} - p_N) \end{bmatrix} \quad (2.39)$$

The system $\mathbf{R}_w \cdot \nabla p_{fs} = \mathbf{b}_w$ can be solved using a Singular Value Decomposition method. With the pressure gradient computed at the free surface in this fashion, pressure values can be extrapolated to the ghost nodes. These pressure values meet the NDBC and

are used for the computation of the pressure gradient and pressure-based artificial dissipation terms in the vicinity of the free surface.

2.3. Slope limiters for gradient reconstruction

When reconstructing gradients using the Least Squares Method in finite volume algorithms, it is common to encounter non-physical oscillations near steep gradients or discontinuities (Nishikawa, 2014; White et al., 2019). These oscillations arise because the Least Squares reconstruction is inherently linear and does not account for local extrema or solution boundedness. To prevent the emergence of new extrema and ensure monotonicity in the vicinity of the free-surface, a slope limiter is introduced. The limiter modifies the reconstructed velocity and pressure gradients in a way that preserves second-order accuracy in smooth regions while reducing it near sharp transitions, thereby maintaining numerical stability and preventing spurious oscillations.

To limit the reconstructed gradients, the Venkatakrishnan's (Venkatakrishnan, 1993, 1995) limiter is chosen due to its robust convergence properties and its relatively easy implementation (Blazek, 2015).

Consider a linear reconstruction of a scalar quantity φ (which can be p, u, v or w) within a neighbourhood \mathcal{N}_s , at a point \mathbf{x}_s located on the free surface. Using its gradient at the same point, the reconstructed variable φ can be expressed as:

$$\varphi(\mathbf{x}) = \varphi_s + \nabla\varphi_s \cdot (\mathbf{x} - \mathbf{x}_s). \quad (2.40)$$

To prevent overshoots and undershoots, the reconstructed values are limited using the Venkatakrishnan's limiter, which applies a smooth function to modify the gradient:

$$\tilde{\nabla}\varphi = \psi \nabla\varphi, \quad (2.41)$$

where the limiter function ψ is defined as:

$$\psi = \min_{j \in \mathcal{N}_s} \begin{cases} \frac{1}{\Delta_j} \left[\frac{(\Delta_{s,\max}^2 + \epsilon^2)\Delta_j + 2\Delta_j^2\Delta_{s,\max}}{\Delta_{s,\max}^2 + 2\Delta_j^2 + \Delta_{s,\max}\Delta_j + \epsilon^2} \right] & \text{if } \Delta_j > 0, \\ \frac{1}{\Delta_j} \left[\frac{(\Delta_{s,\min}^2 + \epsilon^2)\Delta_j + 2\Delta_j^2\Delta_{s,\min}}{\Delta_{s,\min}^2 + 2\Delta_j^2 + \Delta_{s,\min}\Delta_j + \epsilon^2} \right] & \text{if } \Delta_j < 0, \\ 1 & \text{if } \Delta_j = 0, \end{cases} \quad (2.42)$$

where:

$$\Delta_j = \nabla \varphi_s \cdot \mathbf{r}_j \quad (2.43)$$

$$\Delta_{s,\min} = \varphi_{\min} - \varphi_s \quad (2.44)$$

$$\Delta_{s,\max} = \varphi_{\max} - \varphi_s \quad (2.45)$$

The vector \mathbf{r}_j is the position vector from the free surface to the computational node j within \mathcal{N}_s , and the values φ_{\min} and φ_{\max} correspond to the minimum and maximum values of φ within the same neighbourhood. The parameter ϵ^2 controls the degree of limiting applied. Setting ϵ^2 to zero enforces full limiting, which can hinder convergence. Conversely, choosing a large ϵ^2 causes the limiter function to approach unity, effectively removing the limiting and potentially allowing oscillations in the solution. In practice, it has been found that ϵ^2 should scale proportionally with a local length scale, namely,

$$\epsilon^2 = (K\Delta h)^3 \quad (2.46)$$

Here, K is a constant of order unity, and Δh is, for example, the average local cell edge length within \mathcal{N}_s . Blazek (2015) shows that a value of $K = 5$ achieves good convergence rates while preserving the limiting properties of the function. In addition, the

Venkatakrishnan limiter does not completely nullify the gradient at smooth extrema, preserving second-order accuracy in those regions.

The limited gradient $\tilde{\nabla}\varphi$ is used to extrapolate the flow field onto the ghost fluid nodes ensuring that no new extrema are introduced in the vicinity of the free surface.

2.4. Turbulence Model

The filtered Navier-Stokes equations presented in 2.1 and 2.2 are derived by decomposing the velocity and pressure field into resolved and unresolved components and then averaging them over a spatial filter (Sagaut, 2005). This process introduces sub-grid stress terms in the momentum equations 2.2, which are modelled using the dynamic Smagorinsky sub-grid scale (sgs) model (Germano et al., 1991; Smagorinsky, 1963). In this model, the sub-grid stress tensor is approximated using a Boussinesq-type formulation as follows

$$\tau_{ij} - \frac{1}{3}\delta_{ij}\tau_{kk} = -2\nu_{sgs}\mathcal{S}_{ij} \quad (2.47)$$

Where the tensor \mathcal{S}_{ij} is the strain-rate tensor of the resolved scaled, defined as

$$\mathcal{S}_{ij} \equiv \frac{1}{2} \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right) \quad (2.48)$$

The sub-grid scale viscosity is calculated as

$$\nu_{sgs} = C_s \Delta^2 |\mathbf{S}| \quad (2.49)$$

Where C_s is the dynamic Smagorinsky constant, Δ is the filter size, and $|\mathbf{S}| = \sqrt{2\mathcal{S}_{ij}\mathcal{S}_{ij}}$. The box filter (Sagaut, 2005) is employed in this implementation, and its size is given by the grid size (grid filter)

$$\Delta = J^{-1/3} \equiv \sqrt[3]{\Delta x \Delta y \Delta z} \quad (2.50)$$

The dynamic Smagorinsky constant is a factor that locally scales the sub-grid scale viscosity, and therefore it evolves in space and time. The optimal value of C_s is computed to minimize the mean square error between the resolved stress at the grid filter and the test filter as follows (Germano et al., 1991; Lilly, 1992)

$$C_s = \frac{\langle L_{ij} M_{ij} \rangle}{\langle M_{kl} M_{kl} \rangle}, \quad (2.51)$$

Where

$$L_{ij} = \widehat{u_i u_j} - \widehat{u}_i \widehat{u}_j, \quad (2.52)$$

$$M_{ij} = 2\Delta^2 \widehat{S_{ij} | S|} - 2\widehat{\Delta}^2 \widehat{S_{ij}} \widehat{| S|}, \quad (2.53)$$

In equations 2.52 and 2.53, the operator $(\widehat{\cdot})$ corresponds to the average over the test filter, which is twice its size ($\widehat{\Delta} = 2\Delta$). In three dimensions, the average over the test filter involves 27 nodes surrounding the given grid node. The average is performed using a Simpsons rule integration (Fujii, 2010), which details are described in Appendix D.

For generalised curvilinear coordinates, Armenio and Piomelli (2000) pointed out that equation 2.51 is not invariant with respect to a rotation of the frame of reference. To correct this, the following invariant formulation is used in this implementation

$$C_s = \frac{\langle L_{il} M_{im} g_{lm} \rangle}{\langle M_{kp} M_{kq} g_{pq} \rangle}, \quad (2.54)$$

Where g_{ij} is the covariant metric tensor defined as $g_{ij} = (\partial x_m / \partial \xi_i)(\partial x_m / \partial \xi_j)$.

2.5. Artificial Compressibility numerical model

We develop a new version of the Artificial Compressibility (AC) approach with pressure-based residual smoothing (Sotiropoulos & Constantinescu, 1997), adapting the model to the work with dynamic free surfaces and free-surface dynamic boundary conditions.

In the AC method, the continuity equation is modified by adding a pseudo-time (denoted by τ) derivative of the pressure term (Chorin, 1967; Kwak & Kiris, 2010). For open channel flows, Lee et al. (2006) stated that, for a pressure-split formulation, the pressure term that has to be added to the continuity equation is the dynamic component. Hence, the AC continuity equation reads

$$\frac{1}{\beta} \frac{\partial p_d}{\partial \tau} + \frac{\partial u_i}{\partial x_i} = 0 \quad (2.55)$$

Where β is the artificial compressibility parameter, which is proportional to the propagation speed of pressure pseudo-waves, such that large values of β produce that artificial waves spread quickly throughout the computational domain (when $\beta \rightarrow \infty$, the incompressibility condition is recovered). The parameter β is used as a preconditioner of the system (Ahn & Kallinderis, 2006)

Adding pseudo-time derivatives of the velocities to the momentum equations, equation 2.55 forms a hyperbolic-parabolic type of time-dependent system of equations and implicit schemes developed for compressible flows can be implemented (Kwak & Kiris, 2010). With the addition of the pseudo-time derivative, the momentum equations become

$$\frac{\partial u}{\partial \tau} + \frac{\partial u}{\partial t} + \frac{\partial(uu_j)}{\partial x_j} = -\frac{\partial p_d}{\partial x} + \frac{\partial}{\partial x_j} \left[\left(\frac{1}{Re} + \nu_{sgs} \right) \left(\frac{\partial u}{\partial x_j} + \frac{\partial u_j}{\partial x} \right) \right] - \frac{1}{Fr^2} \frac{\partial h}{\partial x} \quad (2.56)$$

$$\frac{\partial v}{\partial \tau} + \frac{\partial v}{\partial t} + \frac{\partial(vu_j)}{\partial x_j} = -\frac{\partial p_d}{\partial y} + \frac{\partial}{\partial x_j} \left[\left(\frac{1}{Re} + \nu_{sgs} \right) \left(\frac{\partial v}{\partial x_j} + \frac{\partial u_j}{\partial y} \right) \right] - \frac{1}{Fr^2} \frac{\partial h}{\partial y} \quad (2.57)$$

$$\frac{\partial w}{\partial \tau} + \frac{\partial w}{\partial t} + \frac{\partial(wu_j)}{\partial x_j} = -\frac{\partial p_d}{\partial z} + \frac{\partial}{\partial x_j} \left[\left(\frac{1}{Re} + \nu_{sgs} \right) \left(\frac{\partial w}{\partial x_j} + \frac{\partial u_j}{\partial z} \right) \right] \quad (2.58)$$

Recasting equations (2.55) and (2.56) - (2.58) into strong conservative form in generalised curvilinear coordinates (the details of the transformation are provided in Appendix A) as (2.4)

$$\tilde{\boldsymbol{\Gamma}} \frac{\partial \mathbf{Q}}{\partial \tau} + \boldsymbol{\Gamma} \frac{\partial \mathbf{Q}}{\partial t} + J \frac{\partial}{\partial \xi^j} (\mathbf{F}_L^j + \mathbf{F}_N^j + \mathbf{F}_v^j) + \mathbf{S}_g = 0 \quad (2.59)$$

Where the tensor $\tilde{\boldsymbol{\Gamma}} = \text{diag}[1/\beta, 1, 1, 1]^T$.

2.5.1. Block-Diagonal algorithm and numerical schemes

The curvilinear coordinate system is discretised in a non-staggered computational grid. A second-order three-point backwards and first-order Euler implicit discretisation are used for the physical and pseudo time derivatives, respectively. In pressure-based methods, the mass conservation equation is always satisfied at the implicit time level and the pressure field in the momentum equations plays the role of an implicit Lagrange multiplier (Gresho & Sani, 1987; Sotiropoulos & Constantinescu, 1997). Thus, the non-linear, viscous fluxes and source term are incorporated in explicit fashion, whilst the linear term is included in an implicit form. The equation discretised in time (ℓ for pseudo-time and n for physical time) reads

$$\begin{aligned} \tilde{\boldsymbol{\Gamma}} \frac{\mathbf{Q}^{\ell+1,n+1} - \mathbf{Q}^{\ell,n+1}}{\Delta \tau} + \boldsymbol{\Gamma} \frac{3\mathbf{Q}^{\ell+1,n+1} - 4\mathbf{Q}^n + \mathbf{Q}^{n-1}}{2\Delta t} + J \frac{\partial}{\partial \xi^m} \left(\mathbf{F}_L^m{}^{\ell+1,n+1} \right) \\ + J \frac{\partial}{\partial \xi^m} \left(\mathbf{F}_N^m{}^{\ell,n+1} + \mathbf{F}_v^m{}^{\ell,n+1} \right) + \mathbf{S}_g{}^{\ell,n+1} = 0 \end{aligned} \quad (2.60)$$

The linear flux divergence $\partial \mathbf{F}_L^m / \partial \xi^m$ is expressed in non-conservative form as

$$\frac{\partial \mathbf{F}_L^m}{\partial \xi^m} = \frac{\partial \mathbf{F}_L^m}{\partial \mathbf{Q}} \frac{\partial \mathbf{Q}}{\partial \xi^m} \quad (2.61)$$

Where $\partial \mathbf{F}_L^m / \partial \mathbf{Q}$ is the Jacobian matrix of the linear convective flux, which depends on the transformation metrics only (constant for a non-moving grid).

$$\frac{\partial \mathbf{F}_L^m}{\partial \mathbf{Q}} = \mathbf{A}_L^m = \frac{1}{J} \begin{bmatrix} 0 & \frac{\partial \xi^m}{\partial x_1} & \frac{\partial \xi^m}{\partial x_2} & \frac{\partial \xi^m}{\partial x_3} \\ \frac{\partial \xi^m}{\partial x_1} & 0 & 0 & 0 \\ \frac{\partial \xi^m}{\partial x_2} & 0 & 0 & 0 \\ \frac{\partial \xi^m}{\partial x_3} & 0 & 0 & 0 \end{bmatrix} \quad (2.62)$$

Recasting (2.60)

$$\begin{aligned} \tilde{\Gamma} \frac{\mathbf{Q}^{\ell+1,n+1} - \mathbf{Q}^{\ell,n+1}}{\Delta \tau} + \Gamma \frac{3\mathbf{Q}^{\ell+1,n+1} - 4\mathbf{Q}^n + \mathbf{Q}^{n-1}}{2\Delta t} + J \mathbf{A}_L^m \frac{\partial}{\partial \xi^m} \left(\mathbf{Q}^{\ell+1,n+1} \right) \\ + J \frac{\partial}{\partial \xi^m} \left(\mathbf{F}_N^m{}^{\ell,n+1} + \mathbf{F}_v^m{}^{\ell,n+1} \right) + \mathbf{S}_g{}^{\ell,n+1} = 0 \end{aligned} \quad (2.63)$$

We define $\Delta \mathbf{Q}^{\ell+1} = \mathbf{Q}^{\ell+1,n+1} - \mathbf{Q}^{\ell,n+1}$ and we manipulate the physical time derivative and the linear flux term in such a way that $3\mathbf{Q}^{\ell+1,n+1} - 4\mathbf{Q}^n + \mathbf{Q}^{n-1} = 3\mathbf{Q}^{\ell+1,n+1} - 3\mathbf{Q}^{\ell,n+1} + 3\mathbf{Q}^{\ell,n+1} - 4\mathbf{Q}^n + \mathbf{Q}^{n-1} = 3\Delta \mathbf{Q}^{\ell+1} + 3\mathbf{Q}^{\ell,n+1} - 4\mathbf{Q}^n + \mathbf{Q}^{n-1}$, and $\mathbf{Q}^{\ell+1,n+1} = \mathbf{Q}^{\ell+1,n+1} - \mathbf{Q}^{\ell,n+1} + \mathbf{Q}^{\ell,n+1} = \Delta \mathbf{Q}^{\ell+1} - \mathbf{Q}^{\ell,n+1}$. Introducing this into (2.63), we obtain

$$\begin{aligned} \tilde{\Gamma} \frac{\Delta \mathbf{Q}^{\ell+1}}{\Delta \tau} + \Gamma \frac{3 \Delta \mathbf{Q}^{\ell+1}}{2 \Delta t} + J \mathbf{A}_L^m \frac{\partial}{\partial \xi^m} (\Delta \mathbf{Q}^{\ell+1}) + \Gamma \frac{3\mathbf{Q}^{\ell,n+1} - 4\mathbf{Q}^n + \mathbf{Q}^{n-1}}{2\Delta t} \\ + J \mathbf{A}_L^m \frac{\partial}{\partial \xi^m} \left(\mathbf{Q}^{\ell,n+1} \right) + J \frac{\partial}{\partial \xi^m} \left(\mathbf{F}_N^m{}^{\ell,n+1} + \mathbf{F}_v^m{}^{\ell,n+1} \right) + \mathbf{S}_g{}^{\ell,n+1} = 0 \end{aligned} \quad (2.64)$$

The remaining explicit linear flux term is re-introduced in conservative form in the sum of the flux divergences terms

$$\begin{aligned} \tilde{\Gamma} \frac{\Delta \mathbf{Q}^{\ell+1}}{\Delta \tau} + \Gamma \frac{3}{2} \frac{\Delta \mathbf{Q}^{\ell+1}}{\Delta t} + J \mathbf{A}_L^m \frac{\partial}{\partial \xi^m} (\Delta \mathbf{Q}^{\ell+1}) + \Gamma \frac{3\mathbf{Q}^{\ell,n+1} - 4\mathbf{Q}^n + \mathbf{Q}^{n-1}}{2\Delta t} \\ + J \frac{\partial}{\partial \xi^m} \left(\mathbf{F}_L^m{}^{\ell,n+1} + \mathbf{F}_N^m{}^{\ell,n+1} + \mathbf{F}_v^m{}^{\ell,n+1} \right) + \mathbf{S}_g{}^{\ell,n+1} = 0 \quad (2.65) \end{aligned}$$

The solution is achieved by linearising the implicit system of equations, adopting the pressure-based implicit preconditioner of Sotiropoulos and Constantinescu (1997) as the approximate factorisation method. The implicit treatment of the velocity divergence and the pressure gradient terms result in a coupled, block-implicit system with linear coefficients that depend only on the metrics of the geometric transformation. Since the system is linear, we do not need to recalculate the matrix coefficients during every dual-time iteration which improves the efficiency of the solution algorithm (Paik et al., 2005; Sotiropoulos & Constantinescu, 1997). Recasting the equation, it can be written as follow:

$$\left(\tilde{\Gamma} + \frac{3\Delta\tau}{2\Delta t} \Gamma + J\Delta\tau \mathbf{A}_L^m \frac{\partial}{\partial \xi^m} \right) \Delta \mathbf{Q}^{\ell+1} = -\Delta\tau \cdot J \cdot \hat{\mathbf{R}}^{\ell,n+1} \quad (2.66)$$

Where

$$\hat{\mathbf{R}}^{\ell,n+1} = \Gamma \frac{3\mathbf{Q}^{\ell,n+1} - 4\mathbf{Q}^n + \mathbf{Q}^{n-1}}{2\Delta t J} + \mathbf{R}^{\ell,n+1} \quad (2.67)$$

And

$$\mathbf{R}^{\ell,n+1} = \frac{\partial}{\partial \xi^m} (\mathbf{F}_L^m + \mathbf{F}_N^m + \mathbf{F}_v^m)^{\ell,n+1} + \frac{\mathbf{S}_g{}^{\ell,n+1}}{J} \quad (2.68)$$

Resulting in

$$\hat{\mathbf{R}}^{\ell,n+1} = \Gamma \frac{3\mathbf{Q}^{\ell,n+1} - 4\mathbf{Q}^n + \mathbf{Q}^{n-1}}{2\Delta t J} + \frac{\partial}{\partial \xi^m} (\mathbf{F}_L^m + \mathbf{F}_N^m + \mathbf{F}_v^m)^{\ell,n+1} + \frac{\mathbf{S}_g{}^{\ell,n+1}}{J} \quad (2.69)$$

We define matrix \mathbf{E}

$$\mathbf{E} = \tilde{\Gamma} + \frac{3\Delta\tau}{2\Delta t}\Gamma = \tilde{\Gamma} + \alpha\Gamma = \begin{bmatrix} 1/\beta & 0 & 0 & 0 \\ 0 & \alpha & 0 & 0 \\ 0 & 0 & \alpha & 0 \\ 0 & 0 & 0 & \alpha \end{bmatrix} \quad (2.70)$$

So the system reads.

$$\left(\mathbf{E} + J\Delta\tau \mathbf{A}_L^m \frac{\partial}{\partial \xi^m} \right) \Delta \mathbf{Q}^{\ell+1} = -\Delta\tau \cdot J \cdot \hat{\mathbf{R}}^{\ell,n+1} \quad (2.71)$$

Which leads to

$$\left(\mathbf{I} + \Delta\tau \widetilde{\mathbf{A}_L^m} \frac{\partial}{\partial \xi^m} \right) \Delta \mathbf{Q}^{\ell+1} = -\Delta\tau \cdot J \cdot \mathbf{E}^{-1} \cdot \hat{\mathbf{R}}^{\ell,n+1} \quad (2.72)$$

Where

$$\widetilde{\mathbf{A}_L^m} = J\mathbf{E}^{-1}\mathbf{A}_L^m \quad (2.73)$$

We define the following second-order accurate first derivative spatially-discrete operator (where i, j, k are the 3-D computational indexes)

$$\frac{\partial}{\partial \xi^1} (\) \approx \delta_{\xi^1} (\)_{i,j,k} = \frac{(\)_{i+1,j,k} - (\)_{i-1,j,k}}{2\Delta\xi^1} \quad (2.74)$$

Expressing the equation in spatial-discrete form

$$\left(I + \Delta\tau_{i,j,k} \widetilde{\mathbf{A}_L^m}_{i,j,k} \delta_{\xi^m} (\) \right) \Delta \mathbf{Q}_{i,j,k}^{\ell+1} = -\Delta\tau_{i,j,k} \cdot J_{i,j,k} \cdot \mathbf{E}_{i,j,k}^{-1} \cdot \hat{\mathbf{R}}_{i,j,k}^{\ell,n+1} \quad (2.75)$$

Expanding the summation index m

$$\begin{aligned} & \left(I + \Delta\tau_{i,j,k} \left(\widetilde{\mathbf{A}}_{L,i,j,k}^1 \delta_{\xi^1}(\) + \widetilde{\mathbf{A}}_{L,i,j,k}^2 \delta_{\xi^2}(\) + \widetilde{\mathbf{A}}_{L,i,j,k}^3 \delta_{\xi^3}(\) \right) \right) \Delta\mathbf{Q}_{i,j,k}^{\ell+1} \\ &= -\Delta\tau_{i,j,k} \cdot J_{i,j,k} \cdot \mathbf{E}_{i,j,k}^{-1} \cdot \widehat{\mathbf{R}}_{i,j,k}^{\ell,n+1} \end{aligned} \quad (2.76)$$

Equation (2.76) can be factorised using Standard Beam and Warming method (Beam & Warming, 1976):

$$\begin{aligned} & \left(\mathbf{I} + \Delta\tau_{i,j,k} \widetilde{\mathbf{A}}_{L,i,j,k}^1 \delta_{\xi^1}(\) \right) \left(\mathbf{I} + \Delta\tau_{i,j,k} \widetilde{\mathbf{A}}_{L,i,j,k}^2 \delta_{\xi^2}(\) \right) \left(\mathbf{I} + \Delta\tau_{i,j,k} \widetilde{\mathbf{A}}_{L,i,j,k}^3 \delta_{\xi^3}(\) \right) \Delta\mathbf{Q}_{i,j,k}^{\ell+1} \\ &= -\Delta\tau_{i,j,k} \cdot J_{i,j,k} \cdot \mathbf{E}_{i,j,k}^{-1} \cdot \widehat{\mathbf{R}}_{i,j,k}^{\ell,n+1} \end{aligned} \quad (2.77)$$

The matrix $\widetilde{\mathbf{A}}_L^m$ can be re-expressed using a similarity transformation

$$\widetilde{\mathbf{A}}_L^m = \mathbf{M}_m \boldsymbol{\Lambda}_m \mathbf{M}_m^{-1} \quad (2.78)$$

Where

$$\boldsymbol{\Lambda}_m = \text{diag} [0, 0, \theta\sqrt{g^{mm}}, -\theta\sqrt{g^{mm}}] \quad (2.79)$$

$$\theta = \sqrt{\beta/\alpha} \quad (2.80)$$

Where \mathbf{M}_m are the modal matrices of the transformation, defined as

$$\mathbf{M}_m = \begin{bmatrix} 0 & 0 & \sqrt{\alpha\beta} & \sqrt{\alpha\beta} \\ -\frac{\left(\frac{\partial\xi^m}{\partial x_2} + \frac{\partial\xi^m}{\partial x_3}\right)}{s^m} & \frac{\frac{\partial\xi^m}{\partial x_1} \left(\frac{\partial\xi^m}{\partial x_3} - \frac{\partial\xi^m}{\partial x_2}\right)}{s^m \sqrt{g^{mm}}} & \frac{\frac{\partial\xi^m}{\partial x_1}}{\sqrt{g^{mm}}} & -\frac{\frac{\partial\xi^m}{\partial x_1}}{\sqrt{g^{mm}}} \\ \frac{\frac{\partial\xi^m}{\partial x_1}}{s^m} & \frac{g^{mm} + \frac{\partial\xi^m}{\partial x_2} \left(\frac{\partial\xi^m}{\partial x_3} - \frac{\partial\xi^m}{\partial x_2}\right)}{s^m \sqrt{g^{mm}}} & \frac{\frac{\partial\xi^m}{\partial x_2}}{\sqrt{g^{mm}}} & -\frac{\frac{\partial\xi^m}{\partial x_2}}{\sqrt{g^{mm}}} \\ \frac{\frac{\partial\xi^m}{\partial x_1}}{s^m} & -\frac{g^{mm} - \frac{\partial\xi^m}{\partial x_3} \left(\frac{\partial\xi^m}{\partial x_3} - \frac{\partial\xi^m}{\partial x_2}\right)}{s^m \sqrt{g^{mm}}} & \frac{\frac{\partial\xi^m}{\partial x_3}}{\sqrt{g^{mm}}} & -\frac{\frac{\partial\xi^m}{\partial x_3}}{\sqrt{g^{mm}}} \end{bmatrix} \quad (2.81)$$

With

$$s^m = \sqrt{2g^{mm} - \left(\frac{\partial\xi^m}{\partial x_2} - \frac{\partial\xi^m}{\partial x_3}\right)^2} \quad (2.82)$$

This leads to

$$\begin{aligned} \mathbf{M}_{1_{i,j,k}} (\mathbf{I} + \Delta\tau_{i,j,k} \Lambda_{1_{i,j,k}} \delta_{\xi^1}(\cdot)) \mathbf{M}_{1_{i,j,k}}^{-1} \mathbf{M}_{2_{i,j,k}} (\mathbf{I} + \Delta\tau_{i,j,k} \Lambda_{2_{i,j,k}} \delta_{\xi^2}(\cdot)) \mathbf{M}_{2_{i,j,k}}^{-1} \\ \mathbf{M}_{3_{i,j,k}} (\mathbf{I} + \Delta\tau_{i,j,k} \Lambda_{3_{i,j,k}} \delta_{\xi^3}(\cdot)) \mathbf{M}_{3_{i,j,k}}^{-1} \Delta \mathbf{Q}_{i,j,k}^{\ell+1} = -\Delta\tau_{i,j,k} \cdot J_{i,j,k} \cdot \mathbf{E}_{i,j,k}^{-1} \cdot \widehat{\mathbf{R}}_{i,j,k}^{\ell,n+1} \end{aligned} \quad (2.83)$$

The operator $(\mathbf{I} + \Delta\tau \Lambda_m \partial/\partial\xi^m(\cdot))$ is the implicit diagonal operator. An Implicit Residual Smoothing (IRS) second-difference operator, $\varepsilon^m \wp(\widetilde{\mathbf{A}}_L^m) \mathbf{I} \frac{\partial^2}{\partial\xi^m \partial\xi^m}(\cdot)$ is added to further enhance the stability, robustness and damping properties of the algorithm (Sotiropoulos & Constantinescu, 1997). Hence, the resultant implicit diagonal operator reads

$$\mathbf{T}_{\xi^m}(\cdot) = \left(\mathbf{I} + \Delta\tau \left(\Lambda_m \frac{\partial}{\partial\xi^m}(\cdot) - \varepsilon^m \wp(\widetilde{\mathbf{A}}_L^m) \mathbf{I} \frac{\partial^2}{\partial\xi^m \partial\xi^m}(\cdot) \right) \right) \quad (2.84)$$

Where ε^m is the implicit residual smoothing coefficient in m direction and $\wp(\widetilde{\mathbf{A}}_L^m)$ is the spectral radius of $\widetilde{\mathbf{A}}_L^m$, the maximum absolute value of its eigenvalues. It reads, in this case.

$$\wp \left(\tilde{\mathbf{A}}_L^m \right) = \theta \cdot \sqrt{g^{mm}} \quad (2.85)$$

In spatially-discretised form

$$\mathbf{T}_{\xi^m}(\)_{i,j,k} = \left(\mathbf{I} + \Delta \tau_{i,j,k} \left(\Lambda_{m_{i,j,k}} \delta_{\xi^m}(\)_{i,j,k} - \varepsilon^m \wp \left(\tilde{\mathbf{A}}_{L_{i,j,k}}^m \right) \mathbf{I} \cdot \delta_{\xi^m \xi^m}(\)_{i,j,k} \right) \right) \quad (2.86)$$

Where

$$\frac{\partial^2}{\partial \xi^1 \partial \xi^1}(\) \approx \delta_{\xi^1 \xi^1}(\)_{i,j,k} = \frac{(\)_{i+1,j,k} - 2(\)_{i,j,k} + (\)_{i-1,j,k}}{(\Delta \xi^1)^2} \quad (2.87)$$

Which leads to

$$\begin{aligned} & \left(\mathbf{M}_{1_{i,j,k}} \mathbf{T}_{\xi^1}(\)_{i,j,k} \mathbf{M}_{1_{i,j,k}}^{-1} \right) \left(\mathbf{M}_{2_{i,j,k}} \mathbf{T}_{\xi^2}(\)_{i,j,k} \mathbf{M}_{2_{i,j,k}}^{-1} \right) \left(\mathbf{M}_{3_{i,j,k}} \mathbf{T}_{\xi^3}(\)_{i,j,k} \mathbf{M}_{3_{i,j,k}}^{-1} \right) \Delta \mathbf{Q}_{i,j,k}^{\ell+1} \\ &= -\Delta \tau_{i,j,k} \cdot J_{i,j,k} \cdot \mathbf{E}_{i,j,k}^{-1} \cdot \hat{\mathbf{R}}_{i,j,k}^{\ell,n+1} \end{aligned} \quad (2.88)$$

The vector $\hat{\mathbf{R}}_{i,j,k}^{\ell,n+1}$ is expressed in spatially discrete fashion as:

$$\begin{aligned} \hat{\mathbf{R}}_{i,j,k}^{\ell,n+1} &= \Gamma \frac{3\mathbf{Q}_{i,j,k}^{\ell,n+1} - 4\mathbf{Q}_{i,j,k}^n + \mathbf{Q}_{i,j,k}^{n-1}}{2\Delta t J_{i,j,k}} + \\ & \quad \widetilde{\delta}_{\xi^m} (\mathbf{F}_L^m)_{i,j,k}^{\ell,n+1} + \widetilde{\delta}_{\xi^m} (\mathbf{F}_N^m)_{i,j,k}^{\ell,n+1} + \widetilde{\delta}_{\xi^m} (\mathbf{F}_v^m)_{i,j,k}^{\ell,n+1} + \frac{\mathbf{S}_{g_{i,j,k}}^{\ell,n+1}}{J_{i,j,k}}. \end{aligned} \quad (2.89)$$

Where the operator $\widetilde{\delta}_{\xi^m}(\)_{i,j,k}$ is defined as

$$\tilde{\delta}_{\xi^1}(\)_{i,j,k} = \frac{(\)_{i+1/2,j,k} - (\)_{i-1/2,j,k}}{\Delta\xi^1} \quad (2.90)$$

The reconstruction of the linear-convective and viscous fluxes at $i \pm 1/2, j \pm 1/2, k \pm 1/2$ is performed by using centred differences. The reconstruction of the non-linear convective fluxes is carried out by using the QUICK scheme (Leonard, 1979; Nishikawa, 2021).

The solution is advanced by successively inverting the system in each direction (ξ^1, ξ^2 and ξ^3):

$$\mathbf{T}_{\xi^1} \left(\overline{\Delta Q^*}_{i,j,k} \right) = \mathbf{M}_{1,i,j,k}^{-1} \left(-\Delta\tau_{i,j,k} J_{i,j,k} \mathbf{E}_{i,j,k}^{-1} \widehat{\mathbf{R}}_{i,j,k}^{\ell,n+1} \right) = \mathbf{M}_{1,i,j,k}^{-1} \mathbf{R}_{i,j,k}^{*,\ell,n+1} \quad (2.91)$$

$$\Delta Q_{i,j,k}^* = \mathbf{M}_{1,i,j,k} \overline{\Delta Q^*}_{i,j,k} \quad (2.92)$$

$$\mathbf{T}_{\xi^2} \left(\overline{\Delta Q^{**}}_{i,j,k} \right) = \mathbf{M}_{2,i,j,k}^{-1} \Delta Q_{i,j,k}^* = \mathbf{M}_{2,i,j,k}^{-1} \mathbf{R}_{i,j,k}^{**\ell,n+1} \quad (2.93)$$

$$\Delta Q_{i,j,k}^{**} = \mathbf{M}_{2,i,j,k} \overline{\Delta Q^{**}}_{i,j,k} \quad (2.94)$$

$$\mathbf{T}_{\xi^3} \left(\overline{\Delta Q^{\ell+1,n+1}}_{i,j,k} \right) = \mathbf{M}_{3,i,j,k}^{-1} \Delta Q_{i,j,k}^{**} = \mathbf{M}_{3,i,j,k}^{-1} \mathbf{R}_{i,j,k}^{***\ell,n+1} \quad (2.95)$$

$$\Delta Q_{i,j,k}^{\ell+1,n+1} = \mathbf{M}_{3,i,j,k} \overline{\Delta Q^{\ell+1,n+1}}_{i,j,k} \quad (2.96)$$

The application of the second-difference implicit diagonal operator to the unknowns, $\mathbf{T}_{\xi^1} \left(\overline{\Delta Q^*}_{i,j,k} \right)$, $\mathbf{T}_{\xi^2} \left(\overline{\Delta Q^{**}}_{i,j,k} \right)$, and $\mathbf{T}_{\xi^3} \left(\overline{\Delta Q^{\ell+1,n+1}}_{i,j,k} \right)$, yields tridiagonal systems along each coordinate direction on the left-hand side of the discretised equations. These systems are solved using the PaScal algorithm developed by Kim et al. (2021), which is specifically designed for massive tridiagonal systems. The algorithm uses the Message Passing Interface (MPI) (Gropp et al., 1999) to perform efficient matrix inversion across arbitrary processor topologies, enabling the solver to handle grids with $\mathcal{O}(10^6-10^7)$ nodes efficiently on high-performance computing platforms.

The notation of $\mathbf{R}_{i,j,k}^{*,\ell,n+1}$, $\mathbf{R}_{i,j,k}^{**\ell,n+1}$ and $\mathbf{R}_{i,j,k}^{***\ell,n+1}$ remarks that the right-hand side of the equation is updated every sub-step. It is important to keep this in mind because it will

become relevant to set proper boundary conditions for the method at domain boundaries and nodes next to the free surface.

A detailed explanation of the solution of the Block Diagonal System and the treatment of the domain and free-surface boundary conditions is presented in Appendix B.

CHAPTER 3

FREE SURFACE EVOLUTION MODEL

3. FREE SURFACE EVOLUTION MODEL

3.1. Free Surface Advection: Level Set Method

The free-surface evolution will be captured using the Level-Set Method (LSM) proposed by Osher and Sethian (1988), which adopts an implicit description of the interface. A continuous scalar signed distance function is introduced in the domain, whose zero-level defines the interface between the two immiscible fluids (see Figure 3.1).

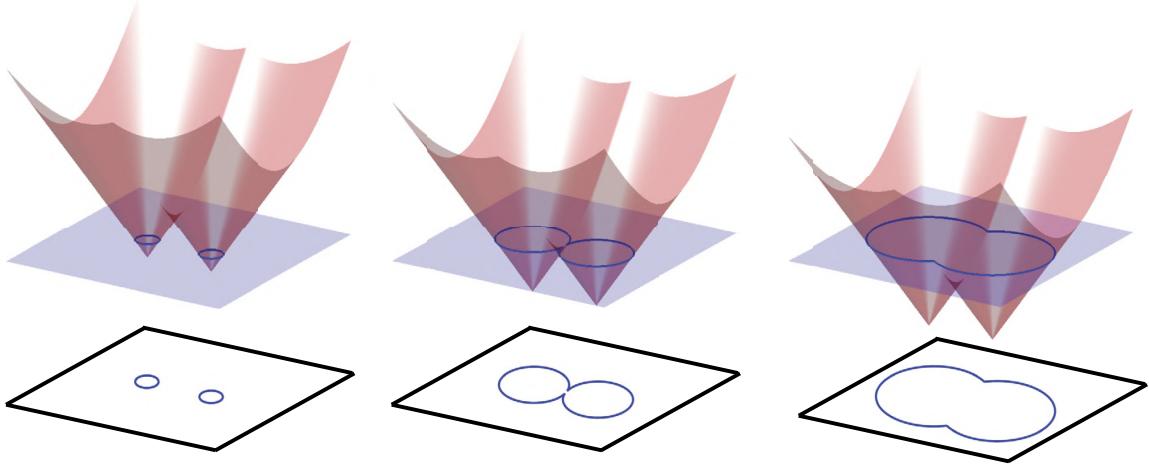


FIGURE 3.1. Level set representation of a the zero level-set (blue lines) of a three-dimensional function (red surface). Image obtained from Gibou et al. (2018).

This function is propagated by an advection equation, which is solved in conjunction with the conservation of mass and momentum for both fluids. The Level-Set advection equation in generalised curvilinear coordinates reads

$$\frac{\partial \phi}{\partial t} + J \left(\frac{U^m}{J} \frac{\partial \phi}{\partial \xi^m} \right) = 0 \quad (3.1)$$

The function ϕ is initialised as a signed distance function which satisfies $|\nabla \phi| = 1$.

The level-set formulation given by equation 3.1 is a non-linear hyperbolic equation, where discontinuities in derivatives may arise even for smooth initial conditions. To obtain accurate and stable solutions, we need suitable numerical schemes as total variation diminishing (TVD) schemes and shock-capturing high-order accurate methods for the convective kinematics (Shu & Osher, 1989). If we recast the equation in semi-discrete form, we get

$$\frac{d\phi_{i,j,k}}{dt} = -J L(\phi) \quad (3.2)$$

Where $L(\phi)$ is the spatial operator $L(\phi) = \frac{U^m}{J} \frac{\partial \phi}{\partial \xi^m}$. We also define the discrete operator $\mathcal{L}(\phi)_{i,j,k} = L(\phi) + \mathcal{O}(\Delta x^r)$ as an r -th order approximation of $L(\phi)$ at $(x_{i,j,k}, y_{i,j,k}, z_{i,j,k})$ (Drikakis & Rider, 2005).

To evolve the discrete equation in time, we use a third-order accurate TVD Runge-Kutta scheme for the time derivative (Shu & Osher, 1989):

$$\phi_{i,j,k}^{(1)} = \phi_{i,j,k}^n - J_{i,j,k} \Delta t \mathcal{L}(\phi^n)_{i,j,k} \quad (3.3)$$

$$\phi_{i,j,k}^{(2)} = \frac{3}{4} \phi_{i,j,k}^n + \frac{1}{4} \phi_{i,j,k}^{(1)} - \frac{1}{4} J_{i,j,k} \Delta t \mathcal{L}(\phi^{(1)})_{i,j,k} \quad (3.4)$$

$$\phi_{i,j,k}^{n+1} = \frac{1}{3} \phi_{i,j,k}^n + \frac{2}{3} \phi_{i,j,k}^{(2)} - \frac{2}{3} J_{i,j,k} \Delta t \mathcal{L}(\phi^{(2)})_{i,j,k} \quad (3.5)$$

Where

$$\mathcal{L}(\phi)_{i,j,k} = \left(\frac{U^1}{J} \right)_{i,j,k} \frac{\partial \phi}{\partial \xi^1} \Big|_{i,j,k} + \left(\frac{U^2}{J} \right)_{i,j,k} \frac{\partial \phi}{\partial \xi^2} \Big|_{i,j,k} + \left(\frac{U^3}{J} \right)_{i,j,k} \frac{\partial \phi}{\partial \xi^3} \Big|_{i,j,k} \quad (3.6)$$

The spatial derivative in the ξ^1 coordinate direction (analogous for the others) is discretised as:

$$\frac{\partial \phi}{\partial \xi_1} \Big|_{i,j,k} = \frac{\phi_{i+\frac{1}{2},j,k} - \phi_{i-\frac{1}{2},j,k}}{\Delta \xi_1} \quad (3.7)$$

The variable ϕ is reconstructed at $i \pm 1/2, j, k$ using a third-order WENO scheme (Jiang & Shu, 1996). For instance, $\phi_{i \pm 1/2, j, k}$ is computed as follows

$$\phi_{i+\frac{1}{2},j,k} = \frac{\alpha_1}{\alpha_1 + \alpha_2} \left(\frac{\phi_C}{2} + \frac{\phi_R}{2} \right) + \frac{\alpha_2}{\alpha_1 + \alpha_2} \left(-\frac{\phi_L}{2} + \frac{3\phi_C}{2} \right) \quad (3.8)$$

Were α_1 and α_2 are the non-linear weights, obtained as

$$\alpha_1 = \frac{2}{3} \frac{1}{((\phi_C - \phi_R)^2 + \epsilon_0)^2} \quad (3.9)$$

$$\alpha_2 = \frac{1}{3} \frac{1}{((\phi_L - \phi_C)^2 + \epsilon_0)^2} \quad (3.10)$$

The parameter ϵ_0 is introduced into the denominator to avoid divisions by zero, and is set to 10^{-6} (Kang & Sotiropoulos, 2012). The selection of the stencil (ϕ_L, ϕ_C, ϕ_R) is upwinded, based on the contravariant velocity:

$$(\phi_L, \phi_C, \phi_R)_{i+\frac{1}{2},j,k} = \begin{cases} (\phi_{i-1,j,k}, \phi_{i,j,k}, \phi_{i+1,j,k}) & \text{if } U_1 > 0 \\ (\phi_{i+2,j,k}, \phi_{i+1,j,k}, \phi_{i,j,k}) & \text{if } U_1 \leq 0 \end{cases} \quad (3.11)$$

3.2. Reinitialisation of the level-set function

The function ϕ is initialised as a signed distance function which satisfies $|\nabla \phi| = 1$. While equation (3.2) satisfies an accurate advection of the zero level set according to the given velocity field, it does not guarantee that ϕ remains as a distance function after the advection step. Keeping the level-set function as a distance function is crucial in the LSM. This is because many geometrical characteristics, which define topological aspects of the free-surface evolution (e.g., normal and tangential directions, curvature), require this property to be preserved (Osher & Fedkiw, 2003).

To provide a correct representation of the level-set function as a signed distance function, and to ensure the mass conservation of the method, we adapted the geometric mass-preserving redistancing algorithm presented by (Ausas et al., 2011). This method has three steps: (i) the free-surface reconstruction via linear interpolation and geometric distance computation, (ii) a local mass correction via a piecewise correction function, and (iii) a global mass correction via a continuous correction function.

3.2.1. Free surface reconstruction and geometric distance computation

The method proposed by Ausas et al. (2011) was originally devised to be used in a finite element framework by Mut et al. (2006). In that formulation, the level-set function is linearly interpolated over each element of the discretisation. To combine this approach with the present finite-volume structured mesh, a partition or “triangulation” around the free-surface is needed (“triangulation” is used hereinafter as the term to denote this step, but in three dimensions, the elements are actually tetrahedrons).

Let’s define the arbitrary triangulation $\mathcal{T}_h \subseteq \Omega$, where Ω is the whole computational domain. \mathcal{T}_h is the region of the space composed of the set of all the tetrahedrons (denoted by \mathcal{K}) whose vertices have ϕ nodal values of different sign or at least one ϕ nodal value equal to zero. The position of the vertices of the tetrahedrons in \mathcal{K} is a set of nodes denoted by \mathcal{P} , and correspond to nodes of the background structured grid of the computational domain (see Figure 3.2.a). Hereinafter, we refer to the nodes in \mathcal{P} as the *first neighbours* of the free surface. Let V_h be the space of continuous functions that are linear within each tetrahedron of \mathcal{T}_h , and let $\phi_h \in V_h$ be a function, such that $\phi_h(\mathbf{x}_p) = \phi(\mathbf{x}_p)$, $\forall \mathbf{x}_p \in \mathcal{P}$. The zero-level set of ϕ_h can be denoted by \mathcal{S}_h , so the aim of the geometric reinitialisation method of Ausas et al. (2011) is to find a function $\tilde{\phi}_h \in V_h$, that approximates the signed distance function d_s , such that

$$d_s(\mathbf{x}) = \text{sign}(\phi_h) \min_{\mathbf{y} \in \mathcal{S}_h} \|\mathbf{x} - \mathbf{y}\| \quad (3.12)$$

Let’s denote the local approximation of ϕ_h within K as ϕ_K , and the surface that corresponds to its zero-level set within K as $\mathcal{S}_K \subset \mathcal{S}_h$ (see Figure 3.2.b). The linear function ϕ_K can be determined within every tetrahedron $K \in \mathcal{K}$ using a Marching-Tetrahedron algorithm (Foley, 1996; Garcia-Villalba et al., 2025; Ingram et al., 2003).

With the local linear reconstruction of the free surface on every element K , we can compute the geometric distance from the nodes in \mathcal{P} to the planes \mathcal{S}_K . The result is the signed distance function to the reconstructed surface ϕ_h^*

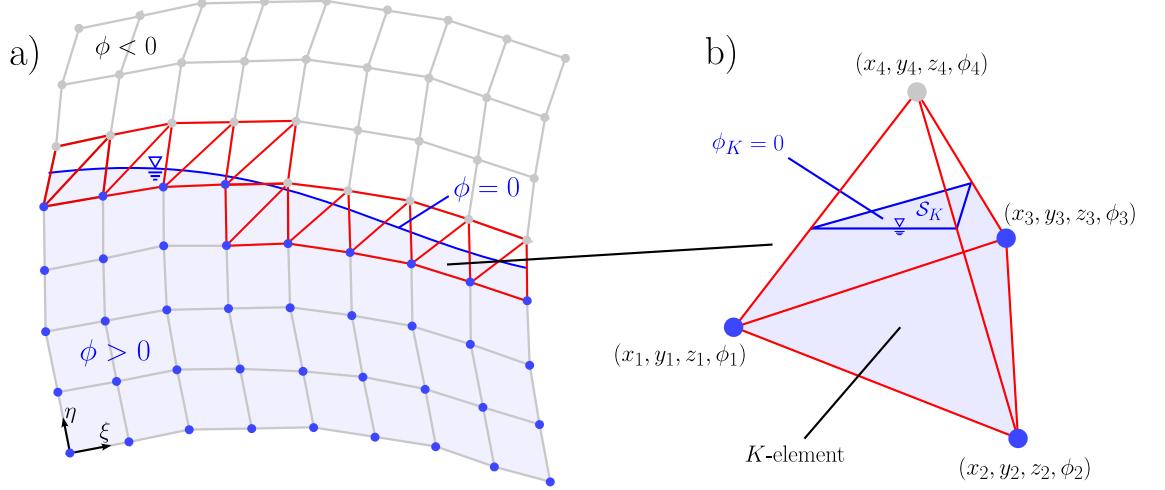


FIGURE 3.2. a) Example of a triangulation \mathcal{T}_h . The zero-level set of ϕ is entirely contained within \mathcal{T}_h . b) Example of an element (tetrahedron) \mathcal{K} within the triangulation \mathcal{T}_h

$$\phi_h^*(\mathbf{x}_I) = \text{sign}(\phi_h(\mathbf{x}_I)) \min_{\mathbf{y} \in S_{K_J}} \|\mathbf{x}_I - \mathbf{y}\|, \forall \mathbf{x}_I \in \mathcal{P} \quad (3.13)$$

Where $K_J \subset \mathcal{K}$ is a an arbitrary subset of elements in the vicinity of \mathbf{x}_I . In the present computational implementation, tetrahedron associated to nodes three-grid-nodes away showed to be a conservative, and therefore sufficient range for J .

As the zero-level set defined by ϕ_h is different from the one defined by ϕ_h^* , there is a volume discrepancy that has to be corrected. Ausas et al. (2011), proposed to correct this by implementing a two-step correction procedure. Let's define the correction function ψ_h , such that the reinitialised level-set field $\tilde{\phi}_h$ reads

$$\tilde{\phi}_h = \phi_h^* + \psi_h \quad (3.14)$$

The function ψ_h is such that the volume enclosed by $\tilde{\phi}_h$ is the same as that of ϕ_h . In the next two subsections, the procedure to determine $\tilde{\phi}_h$ is described.

The difference in the volume enclosed by the zero-level set of ϕ_h and ϕ_h^* for a given triangulation \mathcal{T}_h is given by

$$\Delta V(\phi_h, \phi_h^*) = \int_{\mathcal{T}_h} \left(\mathcal{H}(\phi_h(\mathbf{x})) - \mathcal{H}(\phi_h^*(\mathbf{x})) \right) d\mathbf{x} \quad (3.15)$$

Where \mathcal{H} is the Heaviside step function, such that $\mathcal{H}(s) = 1$ if $s > 0$, and $\mathcal{H} = 0$ otherwise. This equation can be expressed based on the contribution of every element to the volume difference as follows

$$\Delta V(\phi_h, \phi_h^*) = \sum_{K \in \mathcal{K}} \Delta V_K(\phi_h, \phi_h^*) = \sum_{K \in \mathcal{K}} \int_K \left(\mathcal{H}(\phi_h(\mathbf{x})) - \mathcal{H}(\phi_h^*(\mathbf{x})) \right) d\mathbf{x} \quad (3.16)$$

3.2.2. Local mass correction

As the nodal values of ϕ after the advection step and the ones computed as the geometric distance to the reconstructed surface (ϕ_h^*) are different, the volume within every element is also modified. We are interested in finding a piecewise-constant function σ_K , such that (Figure 3.3)

$$\Delta V_K(\phi_h, \phi_h^* + \sigma_K) = 0 \quad (3.17)$$

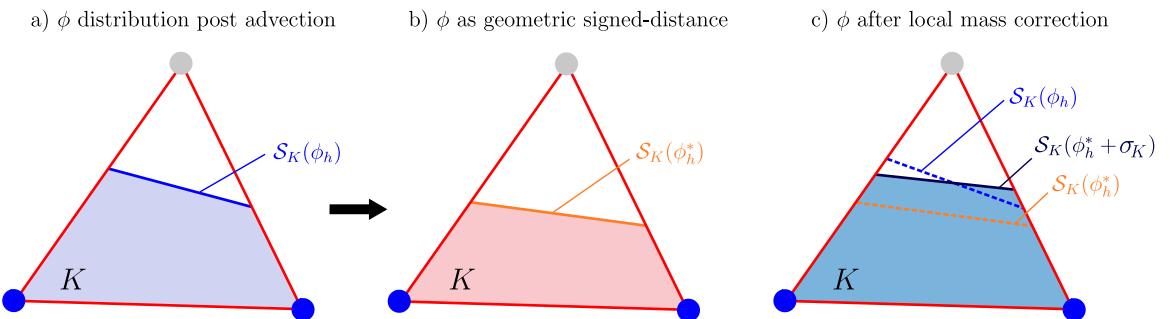


FIGURE 3.3. Local mass correction steps. a) Zero-level set of ϕ after the advection step described in section 3.1. b) Zero-level set of ϕ after the geometric distance calculation step described in subsection 3.2.1. c) Zero-level set of ϕ after the local mass correction step.

Equation 3.17 is a non-linear equation for σ_K , which is solved within every element employing an iterative secant method (Press, 1996).

As the function σ_K is calculated based on the element enclosed volume, it doesn't necessarily describe a continuous ϕ function over the nodes in \mathcal{P} . To overcome this issue, a new function nodal function is computed, which corresponds to the average of σ_K over the elements that has a vertex in that node. Let's define N_I as the number of elements in \mathcal{K} that contains the node I as one of the vertices. We then can define $\omega_I \equiv \omega_h(\mathbf{x}_I)$ as

$$\omega_h(\mathbf{x}_I) = \frac{1}{N_I} \sum_{\substack{K \in \mathcal{K} \\ I \in K}} \sigma_K \quad (3.18)$$

3.2.3. Global mass correction

In our implementation, the global mass correction step slightly differs from the one presented in Ausas et al. (2011). In their work, the global volume that is forced to be preserved, is the one after the advection step. This implementation does not take into account mass loss during the advection step, which can accumulate throughout the simulation and have a negative impact on the accuracy of the results. In the present work, the total volume of the water phase is calculated before the advection step, and then is used to calculate the global correction function (see Figure 3.4).

In regards to the water phase, we can state that

$$V_{\substack{\text{Before} \\ \text{Advection}}}^{\text{TOTAL}} = V_{\substack{\text{After} \\ \text{Advection}}}^{\text{TOTAL}} + \Delta V \quad (3.19)$$

The volume after the advection of ϕ can be calculated using the triangulation \mathcal{T}_h . Given the triangulation, the total volume is the contribution from the single phase cells (SPC), single phase tetrahedrons (SPT), and changing phase tetrahedrons (CPT) (see Figure 3.4.b). This implies that the right-hand side of the equation can be rewritten as

$$V_{\substack{\text{Before} \\ \text{Advection}}}^{\text{TOTAL}} = V_{\substack{\text{After} \\ \text{Advection}}}^{\text{SPC}} + V_{\substack{\text{After} \\ \text{Advection}}}^{\text{SPT}} + V_{\substack{\text{After} \\ \text{Advection}}}^{\text{CPT}} + \Delta V \quad (3.20)$$

Then, finally

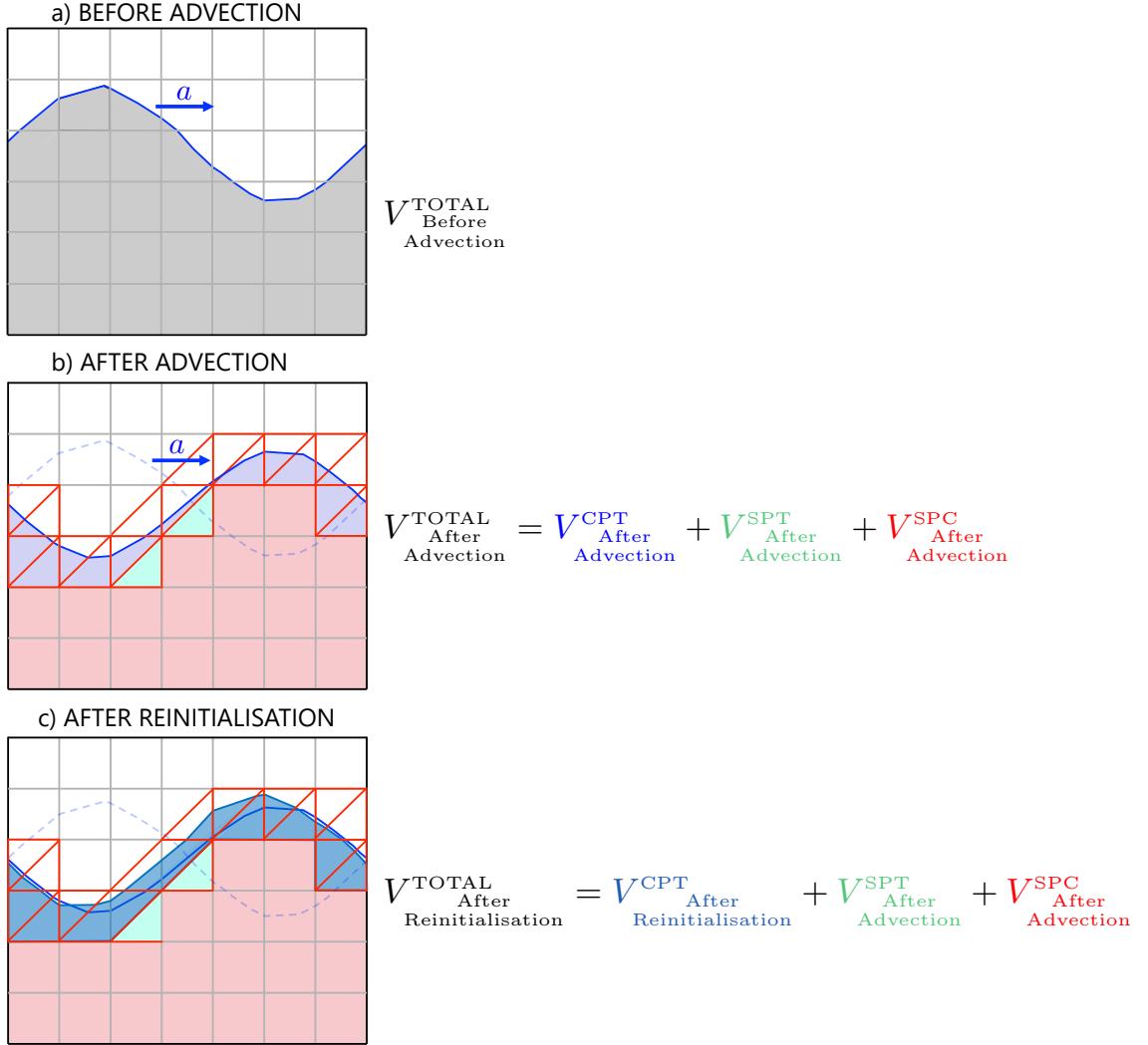


FIGURE 3.4. Schematic representation of the global mass correction. Subfigure a) shows a travelling wave and the liquid phase volume before advection $V^{\text{TOTAL}}_{\text{Before Advection}}$. Subfigure b) shows de decomposition of the total volume into the contribution of single-phase cells (SPC), single-phase tetrahedrons (SPT) and changing-phase tetrahedrons (CPT). Subfigure c) depicts the volume after correcting the nodes belonging to changing-phase tetrahedrons.

$$\Delta V = V^{\text{TOTAL}}_{\text{Before Advection}} - \left(V^{\text{SPC}}_{\text{After Advection}} + V^{\text{SPT}}_{\text{After Advection}} + V^{\text{CPT}}_{\text{After Advection}} \right) \quad (3.21)$$

In order to avoid excessive distortions of the free-surface, the only term of the right-hand side of the equation that is modified is the one from the phase-changing tetrahedrons.

The aim of the global mass correction step is then to reinitialise ϕ in order to determine the volume $V_{\text{After Reinitialisation}}^{\text{CPT}}$ that minimises ΔV (Figure 3.4.c).

From the local mass correction step, the value of ω_h was computed on every node, the global correction continuous function ψ_h can be computed as

$$\phi_h(\mathbf{x}_I) = C \omega_h(\mathbf{x}_I) \quad (3.22)$$

Where C is a constant that preserves the volume globally, such that

$$\Delta V(\phi_h, \phi_h^* + C\omega_h) = 0 \quad (3.23)$$

Equation 3.23 is a non-linear equation for C . This equation is solved using a globally convergent Newton-Raphson method (Press, 1996).

The computational implementation of the methods described in this section is based on the derived variables capability of Fortran 90 (Lahey & Ellis, 1994). Using this feature, it was possible to create nodes and tetrahedrons as data structures and associate them to the background grid of the flow simulation using pointer-structure methods, which allowed to avoid memory overheads and led to a computationally-efficient implementation of the algorithm.

To reinitialise the nodes that do not belong to \mathcal{P} , we calculate the geometric distance to the reconstructed planar surfaces determined in the first step of the method. The information of these surfaces was defined and stored as an attribute of each tetrahedron in \mathcal{K} . They are defined by either three or four points (in the Marching Tetrahedra algorithm, there are $2^4 = 16$ possible combinations of reconstructed surfaces, based on the nodal values of ϕ (Foley, 1996)), and a node in \mathcal{P} may have one or more of them, depending on how many associated tetrahedrons it has. A range of three nodes around the nodes in \mathcal{P} showed to be sufficient for the advection scheme to correctly calculate the temporal evolution of the free surface.

CHAPTER 4

VALIDATION OF THE NUMERICAL MODEL

4. VALIDATION OF THE NUMERICAL MODEL

In this chapter, the mathematical and numerical models introduced in Chapters 2 and 3 are subjected to validation and testing. A flowchart with the various procedures executed by the code is presented in Figure 4.1. It should be noted that not all procedures depicted in the flowchart were subjected to testing; only those enclosed in grey boxes were considered. The procedures associated with the Navier–Stokes solver in generalised curvilinear coordinates correspond to pre-existing functionalities of the solver and had been thoroughly tested and validated in previous studies (see, for example, Paik et al. (2005), Sotiropoulos and Constantinescu (1997), and Sotiropoulos and Abdallah (1992)). The Level-Set advection step is also excluded, as it is a widely adopted approach for this type of method (see Gibou et al. (2018)), and the specific formulation implemented herein corresponds to the one presented in Kang and Sotiropoulos (2012).

The procedures highlighted in grey boxes correspond to the tangential and normal dynamic boundary condition extrapolations described in Section 2.2, and the geometric redistancing method discussed in Section 3.2. These cases are analysed in detail, as they represent novel functionalities introduced into the code. In particular, the extension of the dynamic boundary condition method to generalised curvilinear coordinates constitutes an additional capability beyond the original formulation presented in Watanabe et al. (2008).

Finally, to test the full model, gravity–viscosity-driven waves are simulated. Two cases are considered: (i) the free oscillation of a small-amplitude sloshing tank (a standing wave), and (ii) the propagation of a solitary wave (a travelling wave). These cases are tested using the same setup adopted in the literature and are therefore carried out on regular Cartesian grids.

4.1. Dynamic Boundary Conditions

In this section, the cases considered to test the ability of the method to fulfil the free-surface dynamic boundary conditions described in 2.2, are presented.

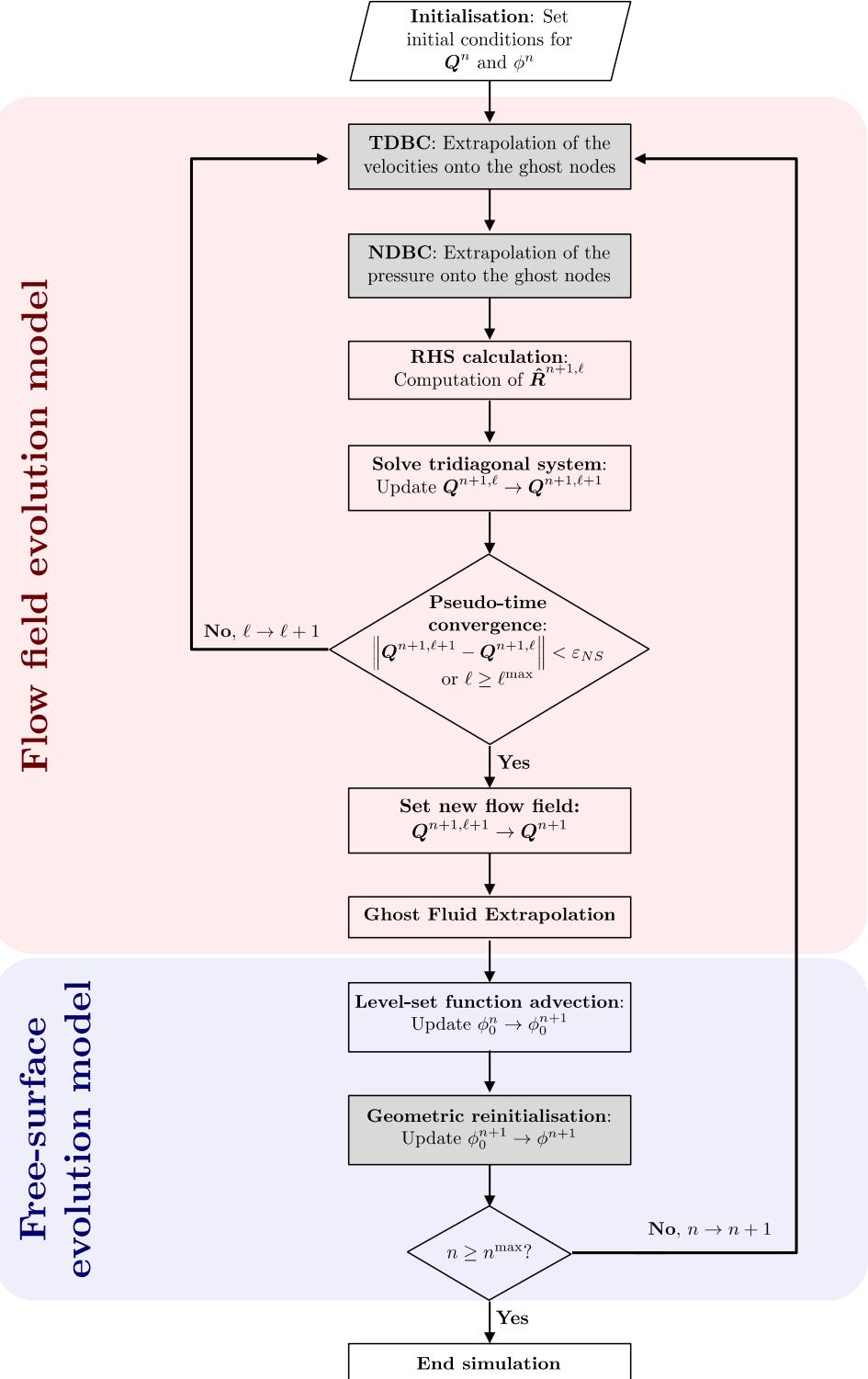


FIGURE 4.1. Flow chart of the Artificial-Compressibility Level Set method. The red block shows the steps comprised within the Navier-Stokes solver. The blue block depicts the ones considered within the free-surface evolution model.

4.1.1. Tangential Dynamic Boundary Condition

To validate the robustness of the method proposed in 2.2, we analyse the extrapolated velocity field and the deviation from the zero-shear condition in three cases:

- (i) Rotating cylinder
- (ii) Rotating sphere
- (iii) Non-linear travelling wave

4.1.1.1. Rotating cylinder

A solid-body rotation with constant angular velocity Ω as the one presented in Watanabe et al. (2008) is tested. The rotating flow without relative distortion of the fluid elements ensures a zero tangential stress at the fluid boundary. This simple test case aims to prove the capability of the ghost-fluid method presented in 2.2 to extrapolate a velocity field that keeps the zero-tangential stress condition at the boundary in a flow with a dominant flow direction (tangential velocities). The present test also aims to prove the accuracy of the method in generalised curvilinear coordinates, which is an additional capability compared to the one presented and tested in Watanabe et al. (2008). The domain is a cube of edge $2D$ and the cylinder of diameter D is aligned with the y axis at the centre. The diameter is set to $D = 1$, and a computational domain of $51 \times 51 \times 51$ is used. The curvilinear grid is presented in 4.2 at the $y = 0$ plane. The dimensionless angular velocity is set to $\Omega = 2.0$.

The prescribed velocity field for the region where $\phi > 0$ is defined as

$$u = -\Omega z \quad (4.1)$$

$$w = \Omega x \quad (4.2)$$

To analyse the performance of the method, we analyse the magnitude of the error in both tangential directions. The error is defined as the deviation from the zero-shear condition, such that

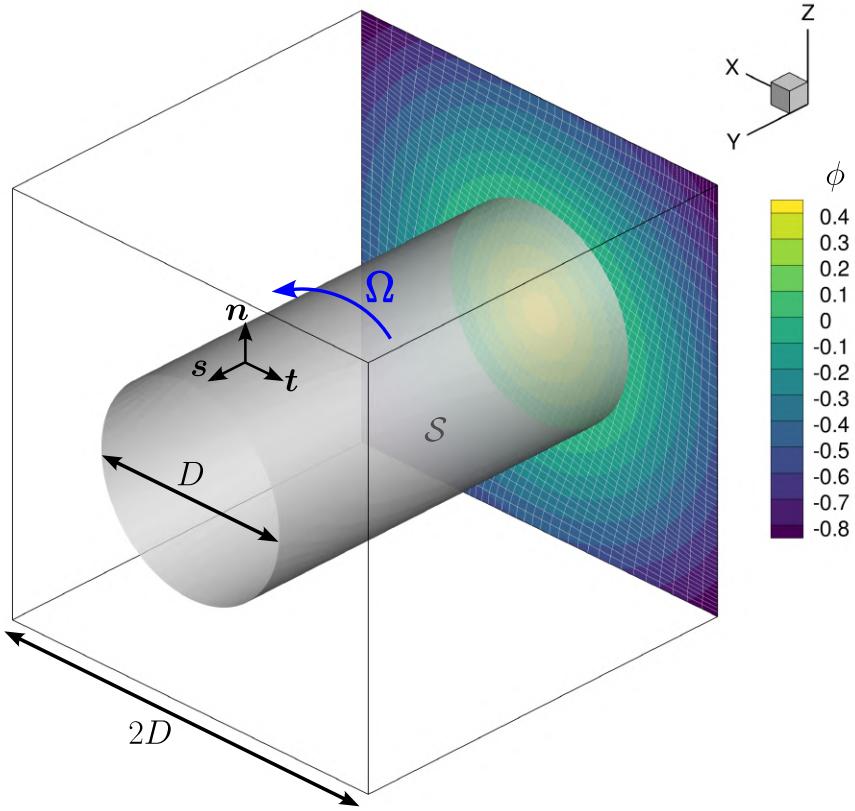


FIGURE 4.2. Rotating cylinder scheme. Filled contour levels of ϕ on the tested curvilinear grid is presented on the plane $y = 0$. \mathcal{S} is the isosurface corresponding to $\phi = 0$.

$$\varepsilon_t = \left| \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right) n_i t_j \right| \quad (4.3)$$

$$\varepsilon_s = \left| \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right) n_i s_j \right| \quad (4.4)$$

In Figure 4.3, the prescribed (blue vectors) and the extrapolated (red vectors) velocities, along with contour maps of ε_t and ε_s .

The maximum error values are $\max_{x_s \in \mathcal{S}} \varepsilon_t(\mathbf{x}_s) = 7.4 \times 10^{-3} \sim \Delta x^2$ and $\max_{x_s \in \mathcal{S}} \varepsilon_s(\mathbf{x}_s) = 8.9 \times 10^{-3} \sim \Delta x^2$. The results show that even with a heavily distorted curvilinear grid, the proposed method is capable of extrapolating a physically consistent velocity field while preserving the second-order accuracy of the numerical solver.

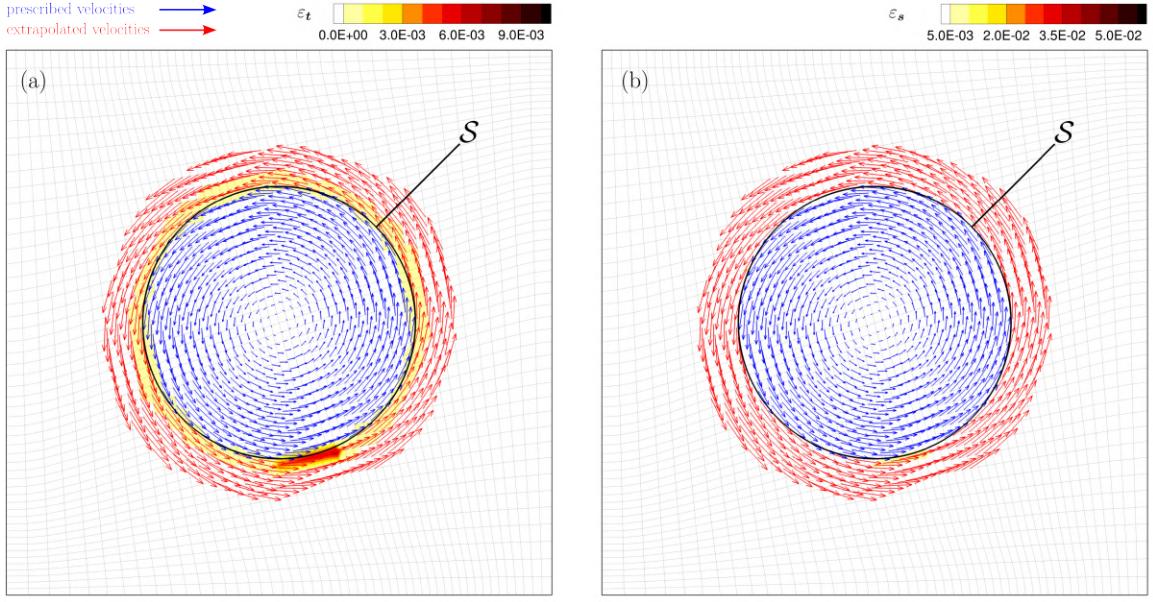


FIGURE 4.3. Prescribed (blue arrows) and extrapolated (red arrows) velocity field in the rotating cylinder test case. Panel (a) shows the tangential error in the t -direction, while panel (b) shows the error in the s -direction.

4.1.1.2. Rotating sphere

A second test case is carried out to demonstrate the capability of the method in dealing with a spatially varying velocity fields. A tilted rotating sphere of diameter D with the following prescribed velocity field is tested

$$\mathbf{u} = \mathbf{R} (\boldsymbol{\Omega} \times \mathbf{r}) \quad (4.5)$$

Where \mathbf{r} is the position vector from the centre of the sphere, and \mathbf{R} is a rotation matrix which is composed of three rotations around the Cartesian axis, such that

$$\mathbf{R} = \mathbf{R}_z(\gamma) \mathbf{R}_y(\beta) \mathbf{R}_x(\alpha) \quad (4.6)$$

With

$$\mathbf{R}_x(\alpha) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \alpha & -\sin \alpha \\ 0 & \sin \alpha & \cos \alpha \end{bmatrix} \quad (4.7)$$

$$\mathbf{R}_y(\beta) = \begin{bmatrix} \cos \beta & 0 & \sin \beta \\ 0 & 1 & 0 \\ -\sin \beta & 0 & \cos \beta \end{bmatrix} \quad (4.8)$$

$$\mathbf{R}_z(\gamma) = \begin{bmatrix} \cos \gamma & -\sin \gamma & 0 \\ \sin \gamma & \cos \gamma & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (4.9)$$

An schematic representation of the tested case is presented in Figure 4.4

The domain and curvilinear grid used in this simulation are the same as those employed in the rotating cylinder test presented above. The extrapolated velocities, and contour maps of ε_t and ε_s are presented in Figure 4.5.

The extrapolated velocity field demonstrates the versatility of the proposed method in handling complex geometries and fully three-dimensional flow structures. The maximum error values are $\max_{\mathbf{x}_s \in \mathcal{S}} \varepsilon_t(\mathbf{x}_s) = 1.7 \times 10^{-3} \sim \Delta x^2$ and $\max_{\mathbf{x}_s \in \mathcal{S}} \varepsilon_s(\mathbf{x}_s) = 2.7 \times 10^{-3} \sim \Delta x^2$. These results confirm the method's ability to extrapolate a physically consistent velocity field while preserving the second-order accuracy of the numerical solver, even in complex, spatially varying flow fields.

4.1.1.3. Non-linear travelling wave

To test the proposed method on a more physically-realistic case, the capability of the model in extrapolating the velocity field from a second-order Stokes wave is analysed. The selection of this type of wave theory is based on the fact that the velocity field presents a rich orbital dynamics near the free surface, which in turn exhibits a geometry sharper and higher in amplitude than sinusoidal linear waves (Lin, 2008). The equations for the surface elevation, horizontal, and vertical velocity, are given by

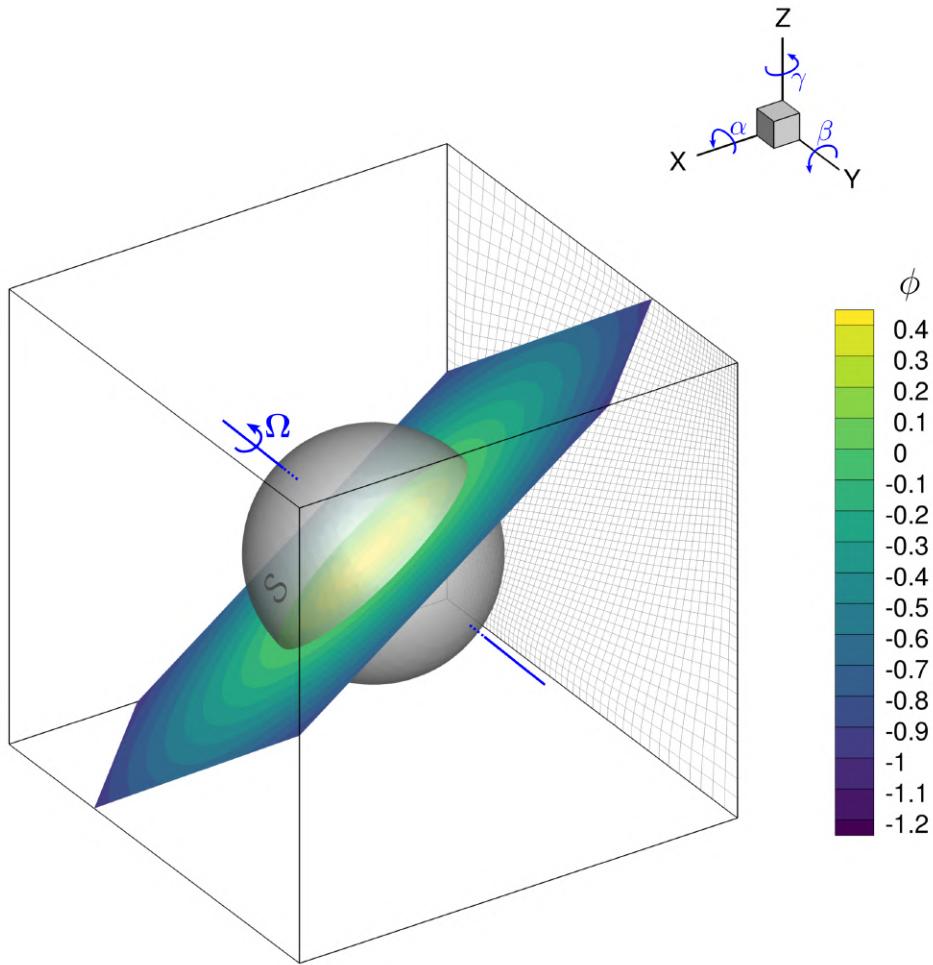


FIGURE 4.4. Rotating sphere scheme. The rotations α , β , and γ around the Cartesian axis is showed on the upper right location of the figure. Filled contour levels of ϕ on the tested curvilinear grid is presented on the plane normal to the resulting tilting vector. S is the isosurface corresponding to $\phi = 0$.

$$\begin{aligned} \eta(x, t) &= a \cos(kx - \omega t) \\ &+ \frac{a^2 k}{4} \frac{\cosh(kh)}{\sinh^3(kh)} (2 + \cosh(2kh)) \cos(2kx - 2\omega t), \end{aligned} \quad (4.10)$$

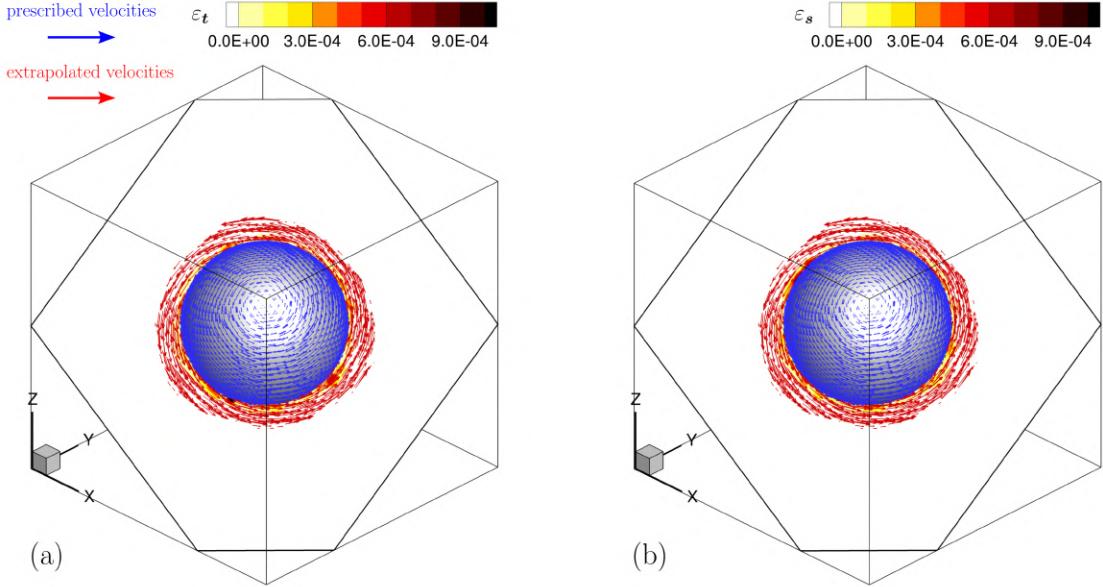


FIGURE 4.5. Prescribed (blue arrows) and extrapolated (red arrows) velocity field in the rotating sphere test case. Panel (a) shows the tangential error in the t -direction, while panel (b) shows the error in the s -direction.

$$u(x, z, t) = \frac{agk}{\omega} \frac{\cosh(k(z + h))}{\cosh(kh)} \cos(kx - \omega t) + \frac{3a^2\omega k}{4} \frac{\cosh(2k(z + h))}{\sinh^4(kh)} \cos(2kx - 2\omega t), \quad (4.11)$$

$$w(x, z, t) = \frac{agk}{\omega} \frac{\sinh(k(z + h))}{\cosh(kh)} \sin(kx - \omega t) + \frac{3a^2\omega k}{4} \frac{\sinh(2k(z + h))}{\sinh^4(kh)} \sin(2kx - 2\omega t). \quad (4.12)$$

The chosen parameters for the equations are $a = H/2 = 0.02$ m, $T = 2$ s, and $h = 0.5$ m, where H is wave height, and h is the water depth of the still water level. The location of this wave regime in the Le Méhauté diagram (Le Méhauté, 2013) is presented in figure 4.6.

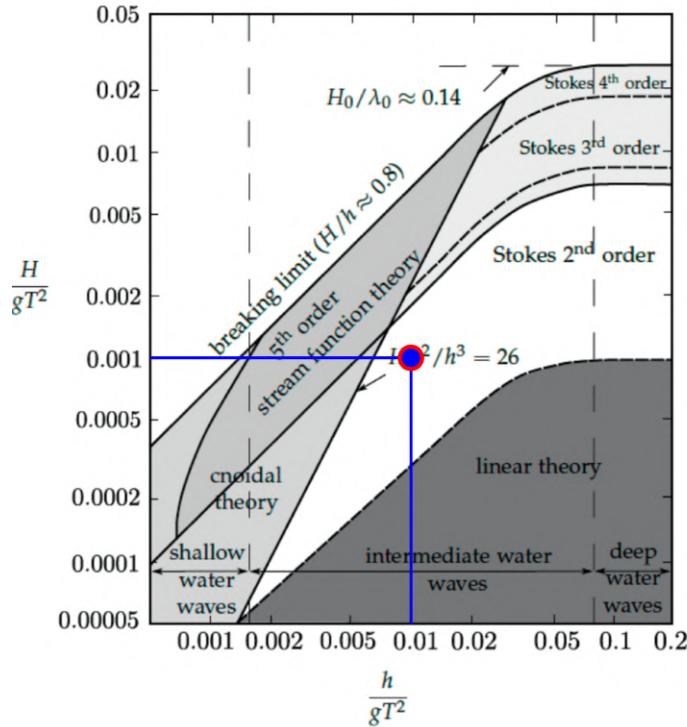


FIGURE 4.6. Position of the tested wave theory in the Le Méhauté diagram. H , h , and T are the wave height, water depth and wave period, respectively (Le Méhauté, 2013).

The extrapolation velocity field for $t = 0$ and a wavelenght extension in the x -direction is presented in figure 4.7

On the one hand, it can be seen that the proposed method is capable of extrapolating velocity fields with complex spatial structures, resembling, in this case, the orbital patterns observed in the prescribed velocities of the water phase. On the other hand, Stokes second-order wave theory, like linear wave theory, assumes ideal flow. This ensures that the free-shear condition at the free surface is satisfied by construction, and that the metrics analysed in previous cases to demonstrate the method's performance are applicable herein. The maximum error values are given by $\max_{\mathbf{x}_s \in \mathcal{S}} \varepsilon_t(\mathbf{x}_s) = 8.0 \times 10^{-3} \sim \Delta x^2$ and $\max_{\mathbf{x}_s \in \mathcal{S}} \varepsilon_s(\mathbf{x}_s) = 8.4 \times 10^{-3} \sim \Delta x^2$, which demonstrate the second-order accuracy of the method in this physically-consistent case.

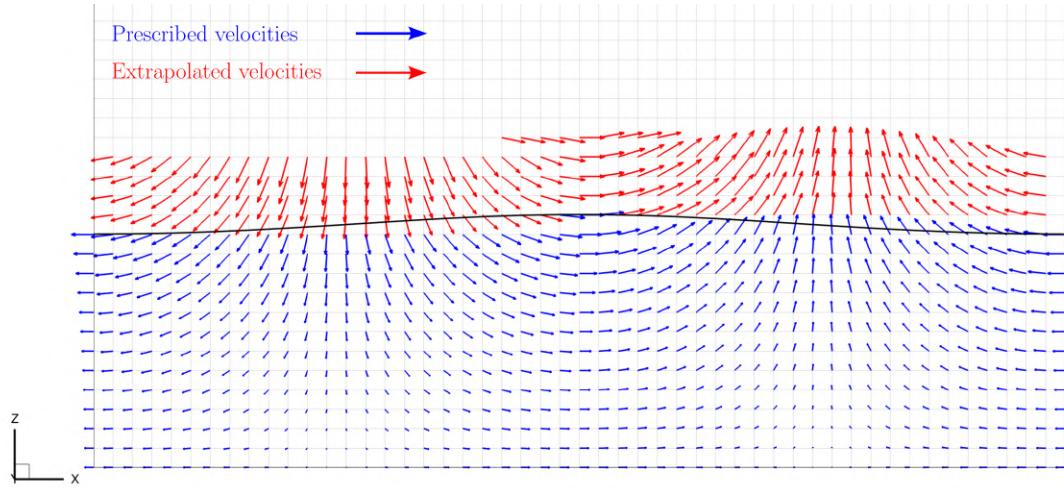


FIGURE 4.7. Prescribed (blue arrows) and extrapolated velocities for the tested Stokes second-order wave over a wavelength in the x direction.

4.1.2. Normal Dynamic Boundary Condition

To test the performance of the method in extrapolating pressure values in order to meet the normal dynamic boundary condition 2.33, we analyse the well-known rigid-body rotating cylinder (White, 1994).

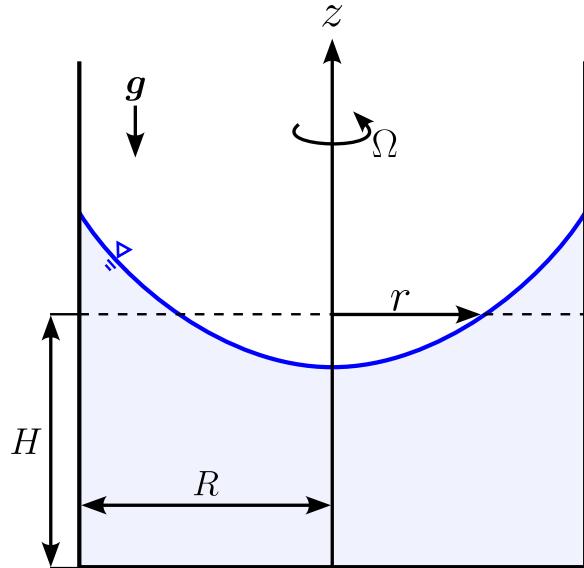


FIGURE 4.8. Rigid-body rotating cylinder scheme. The water forms a paraboloid, where the free surface corresponds to the isosurface of $p = 0$. H represents the still water level

This test case has an analytical solution for the pressure field obtained from the hydrostatic law (White, 1994). The three-dimensional pressure field in non-dimensional form reads

$$p(x, y, z) = \frac{-z}{Fr^2} + \frac{(x^2 + y^2)}{2}\Omega^2 \quad (4.13)$$

Where $Fr = \Omega R / \sqrt{gH}$. For this study case, we set $\Omega = 1$, $H = 1$ and $R = 1$. In this rigid-body rotation test case, the free-surface corresponds to zero-level set of p . This fact allows us to test the method in its ability to extrapolate a pressure field that keeps the locus of $p = 0$ coincident with the free surface (zero-level set of ϕ). The extrapolated pressure field is shown in Figure 4.9.

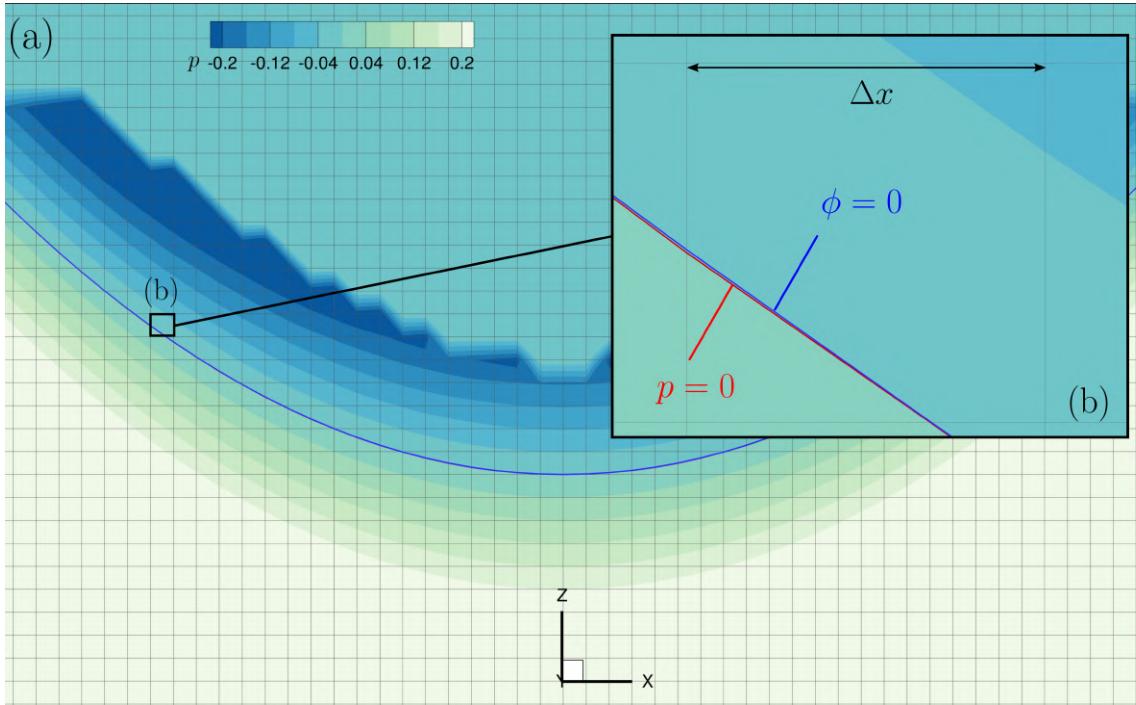


FIGURE 4.9. Extrapolated pressure field. Panel (a) shows the pressure distribution above the free surface in the vicinity of three adjacent nodes. Panel (b) presents a zoomed-in view of a grid element to compare the locations of the isosurfaces $\phi = 0$ and $p = 0$

The results depicted in Figure 4.9 show the ability of the method to extrapolate a pressure field that is consistent with the geometry of the free surface surface. The zoomed-in

view presented in 4.9(b) shows the accuracy of the method to preserve the location of $p = 0$ coincident with the free surface.

4.2. Geometric Reinitialisation

To test the ability to preserve mass accurately of the proposed reinitialisation, we carried out a test based on the rigid-body advection of the Zalesak disc (Zalesak, 1979). This is one of the canonical test cases for interface-advection algorithms (Van der Pijl et al., 2005). The test corresponds to a slotted disc rotated throughout one revolution around the centre of the computational domain. The prescribed velocity field, (u_Z, v_Z) in two dimensions is:

$$u_Z(y) = y_C - y \quad (4.14)$$

$$v_Z(x) = x - x_C \quad (4.15)$$

Where (x_C, y_C) is the initial position of the Zalesak's disc. And schematic representation of the problem is presented in Figure 4.10.

To test the performance of the proposed geometric reinitialisation method in preserving mass accurately, the results after one revolution of the Zalesak's disc test-case are compared with the method proposed by Sussman et al. (1994), which is one of the most extended reinitialisation methods in literature (Gibou et al., 2018). The equation proposed by Sussman et al. (1994) reads

$$\frac{\partial \phi}{\partial \tau} + S(\phi_0) (\|\nabla \phi\| - 1) = 0 \quad (4.16)$$

Where S and $\tilde{\delta}$ are two sigmoid-type functions defined as (Kang & Sotiropoulos, 2012)

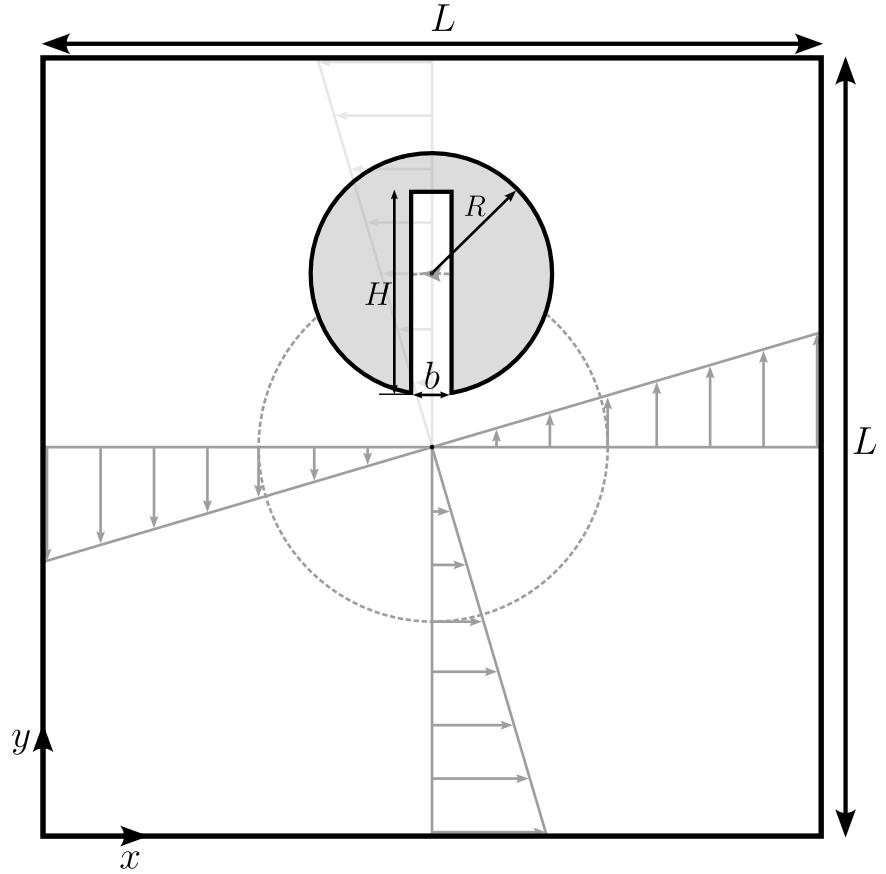


FIGURE 4.10. Zalesak's disc scheme. The slotted disc is rotated in a circular rigid-body motion around the centre of the domain located at (x_C, y_C)

$$S(\phi_0) = \begin{cases} 1, & \phi_0 \geq \epsilon, \\ -1, & \phi_0 \leq -\epsilon, \\ \frac{\phi_0}{\epsilon} - \frac{1}{\pi} \sin\left(\frac{\pi\phi_0}{\epsilon}\right), & \text{otherwise.} \end{cases} \quad (4.17)$$

$$\tilde{\delta}(\phi) = \begin{cases} \frac{1}{2\epsilon} \left(1 + \cos\left(\frac{\pi\phi}{\epsilon}\right) \right), & |\phi| \leq \epsilon, \\ 0, & \text{otherwise.} \end{cases} \quad (4.18)$$

The pseudo-time τ is used to converge 4.16 to steady state, which ensures that ϕ is a signed distance function, and therefore $|\nabla\phi| = 1$. As this formulation does not ensure mass conservation, Sussman and Fatemi (1999) introduced a mass correction term of the right-hand side

$$\frac{\partial\phi}{\partial\tau} + S(\phi_0)(|\nabla\phi| - 1) = \lambda \tilde{\delta}(\phi) |\nabla\phi| \quad (4.19)$$

Where λ is defined at each node as (Osher & Fedkiw, 2003)

$$\lambda_{i,j,k} = -\frac{\int_{\Omega_{i,j,k}} \tilde{\delta}(\phi) S(\phi_o)(1 - |\nabla\phi|) d\Omega}{\int_{\Omega_{i,j,k}} \tilde{\delta}^2(\phi) |\nabla\phi| d\Omega} \quad (4.20)$$

A test case with the following characteristics was set up

L	H	R	b	x_C	y_C
1	0.25	0.15	0.05	0.5	0.75

TABLE 4.1. Zalesak's disc characteristics

The test case was simulated on a uniform grid of 51×51 ($\Delta x = 0.01$) with a time step for the advection equation of $\Delta t = 1 \times 10^{-4}$. In Figure 4.11, the final shape of the Zalesak's disc after one revolution is presented for both reinitialisation methods.

It can be seen that the geometric reinitialisation produced a less smeared shape compared to the one by Sussman and Fatemi (1999), but some bias in the flow direction is introduced. To test the mass-conservation performance, the percentage of volume change is presented in figure 4.12. The base case with no reinitialisation is compared to the ones obtained by the application of the method by Sussman and Fatemi (1999) and the proposed geometric ones.

From figure 4.12, it can be seen that applying the geometric reinitialisation at every time step yields a solution that preserves the global volume with almost no variation (0.02 %). This represents a significant improvement over the original geometric reinitialisation

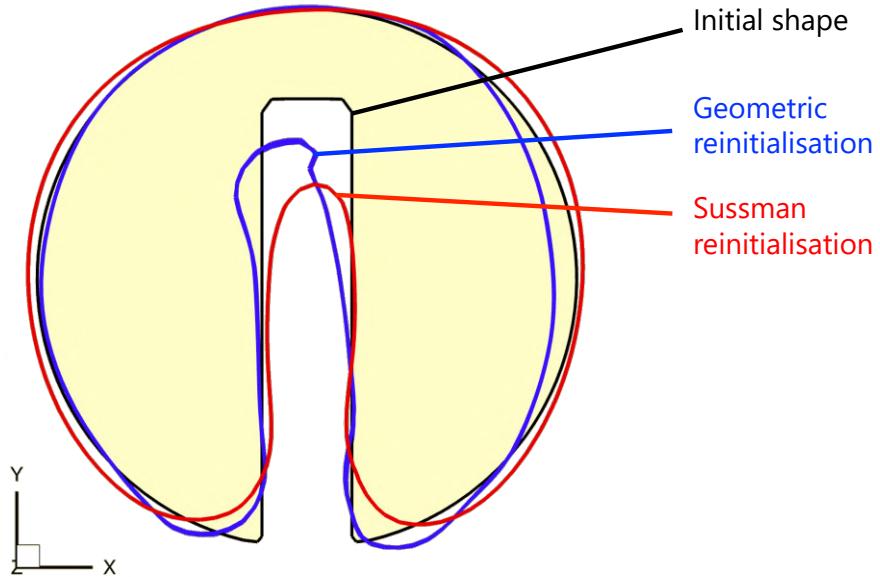


FIGURE 4.11. Final shape after one revolution using two different reinitialisation schemes. The red line is the final shape obtained by using the reinitialisation method of Sussman et al. (1994), whereas the blue line is the final shape of the proposed geometric reinitialisation scheme adapted from Ausas et al. (2011).

proposed by Ausas et al. (2011), which resulted in a volume error of approximately 5% for the same case. Furthermore, the method outperforms that of Sussman and Fatemi (1999), a widely used and standard approach in free-surface solvers, even when the reinitialisation is not applied at every time step, making it highly computationally efficient.

4.3. Full model

Two cases are analysed to test the ability of the model to simulate complex free-surface flows. The first one is the low amplitude sloshing tank first introduced by Tadjbakhsh and Keller (1960), and thoroughly analysed by later authors (Hergibo et al., 2025; Yue et al., 2005). The second one is a fully non-linear solitary wave case presented by Yue et al. (2005), and later used as a benchmark case in different works (e.g. Christou et al., 2021; Lin et al., 2005)

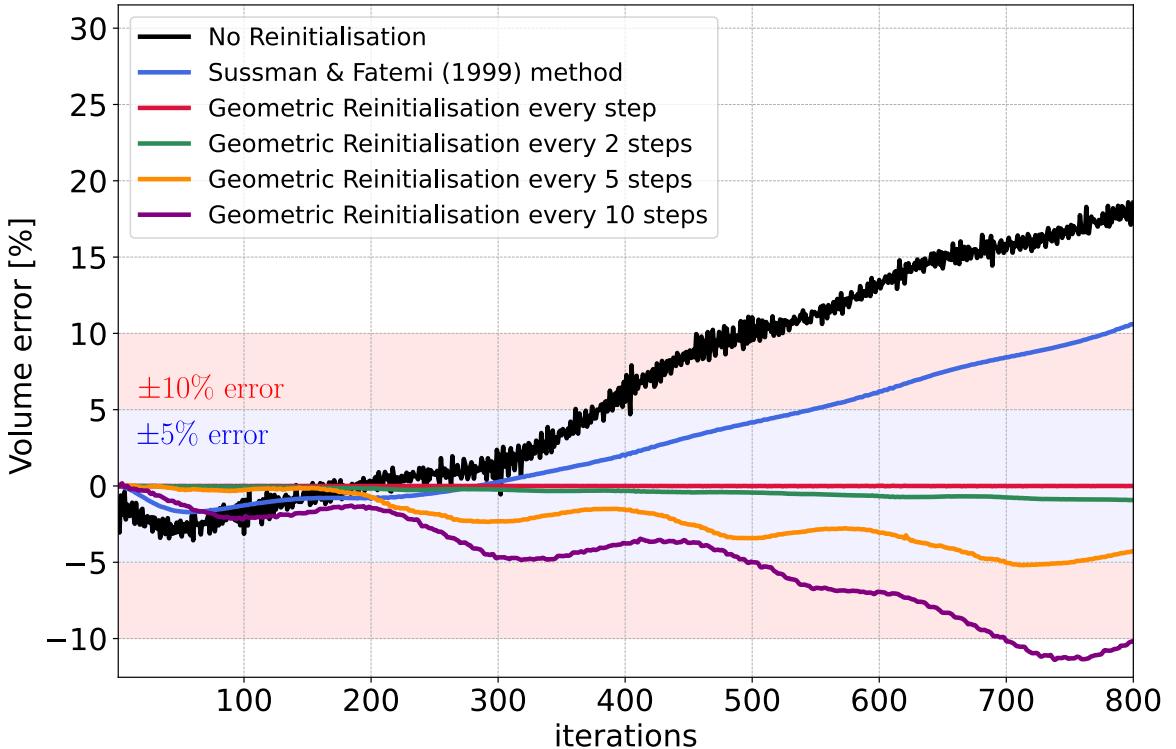


FIGURE 4.12. Volume error for different scenarios: no reinitialisation, Sussman & Fatemi method, and four different geometric reinitialisation tests (every 1, 2, 5 and 10 time steps). The blue band shows the 5% volume error range, whereas the red band corresponds to the 10% error one.

4.3.1. Low amplitude sloshing tank

The performance of the model is tested by analysing the free oscillation of a viscous liquid contained in a tank of width L . The initial profile is given by the following sinusoidal distribution of the water depth

$$h(x) = H + a \cos \left(kx + \frac{kL}{2} \right) \quad (4.21)$$

Where H is the still water depth, a is the amplitude of the wave, and k is the wavenumber, defined as $k = 2\pi/\lambda$, with λ as the wavelength (see Figure 4.13). The initial pressure field is purely hydrostatic, so $p_d = 0$ over the whole domain.

The initial wave is a full wavelength the values of its parameters are summarised in Table 4.2.

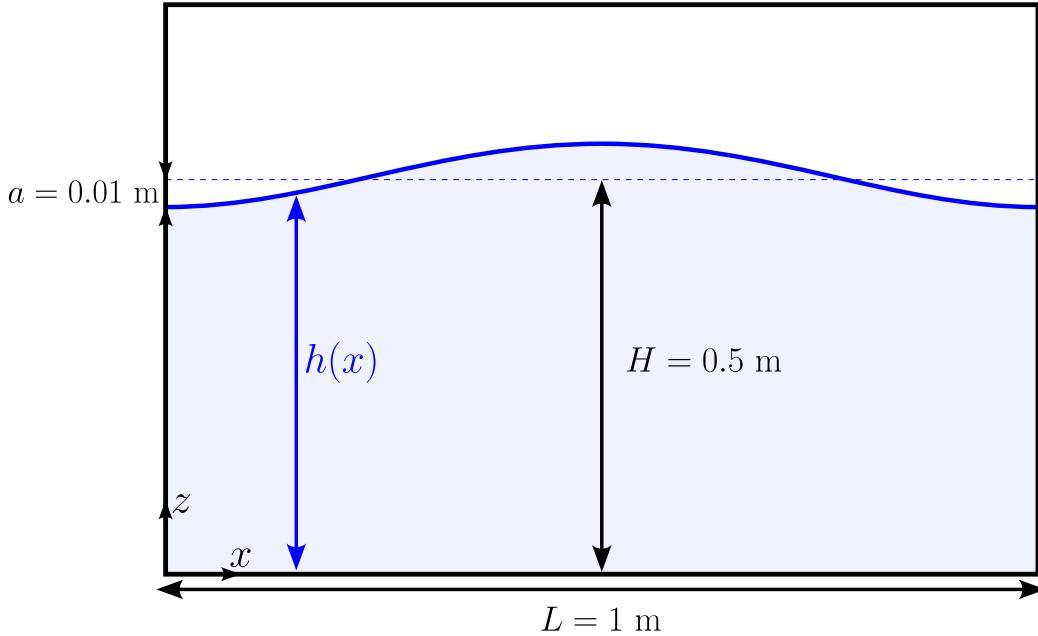


FIGURE 4.13. Sloshing tank dimensions and initial condition

L	H	a	k
1 m	0.5 m	0.01 m	2π

TABLE 4.2. Sloshing tank parameters

For this case, the length scale is defined by the water depth H , and the velocity scale by the shallow water wave celerity \sqrt{gH} . This scales define the Reynolds number as follows

$$Re = \frac{H\sqrt{gH}}{\nu} \quad (4.22)$$

With this choice of the velocity scale, the Froude number is constrained to be equal to 1. The simulation is carried out on computational domain of dimensions $[0, L] \times [0, 1.2H]$ in the $x - z$ plane and $0.05L$ in the y direction. The side walls are assumed to be slip walls, whilst the bottom boundary is a no-slip wall (Marchandise & Remacle, 2006). The flow was simulated using an uniform grid of 31×36 and a non-dimensional physical time step of $\Delta t = 0.01$ is chosen. The parameters for the Artificial Compressibility method are $CFL = 1.0$, $VN = 0.1$, and $\beta = 1.0$.

The numerical results are compared to the analytical solution based on the linearised Navier-Stokes equations proposed by Wu et al. (2001)

$$\frac{\eta(t)}{\eta_0} = 1 - \frac{1}{1 + 4\nu^2 k^3/g} \left[1 - e^{-2\nu k^2 t} \left(\cos(\sqrt{kg} t) + 2\nu k^2 \frac{\sin(\sqrt{kg} t)}{\sqrt{kg}} \right) \right] \quad (4.23)$$

Where $\eta(t)$ corresponds to the elevation with respect to the still water level ($h = 1$). The physical parameters for the analytical equation are the wavenumber k and the viscosity $\nu = gh^{3/2}/Re$ (Marchandise & Remacle, 2006).

Two snapshots of the free surface, alongside the velocity and pressure fields, are shown in Figure 4.14. The chosen times correspond to the crest and trough at the left-hand side wall, respectively (see Figure 4.15).

Figure 4.14 shows a counter-rotating vortex pair near the free surface, which directs the flow downward at the domain centre, generating a local dynamic pressure peak. Half a period later, the opposite behaviour occurs: the vortex strength dissipates due to viscosity, reverses its rotation direction, and shifts toward the walls. The flow periodically oscillates with smaller velocities each times due to the viscous damping of the velocity field, until the velocity is zero everywhere and the free-surface remain still.

The free surface location at the left-hand side wall is presented in Figure 4.14. In the plot, the results are compared to the analytical solution by Wu et al. (2001) and the numerical results obtained by Hergibo et al. (2025) for $Re = 200$.

The results show the damped behaviour previously described. The time series show to be in good agreement with the analytical solution proposed by Wu et al. (2001) and the numerical simulations performed by Hergibo et al. (2025), showing a slightly less dissipative free-surface evolution compared to these authors.

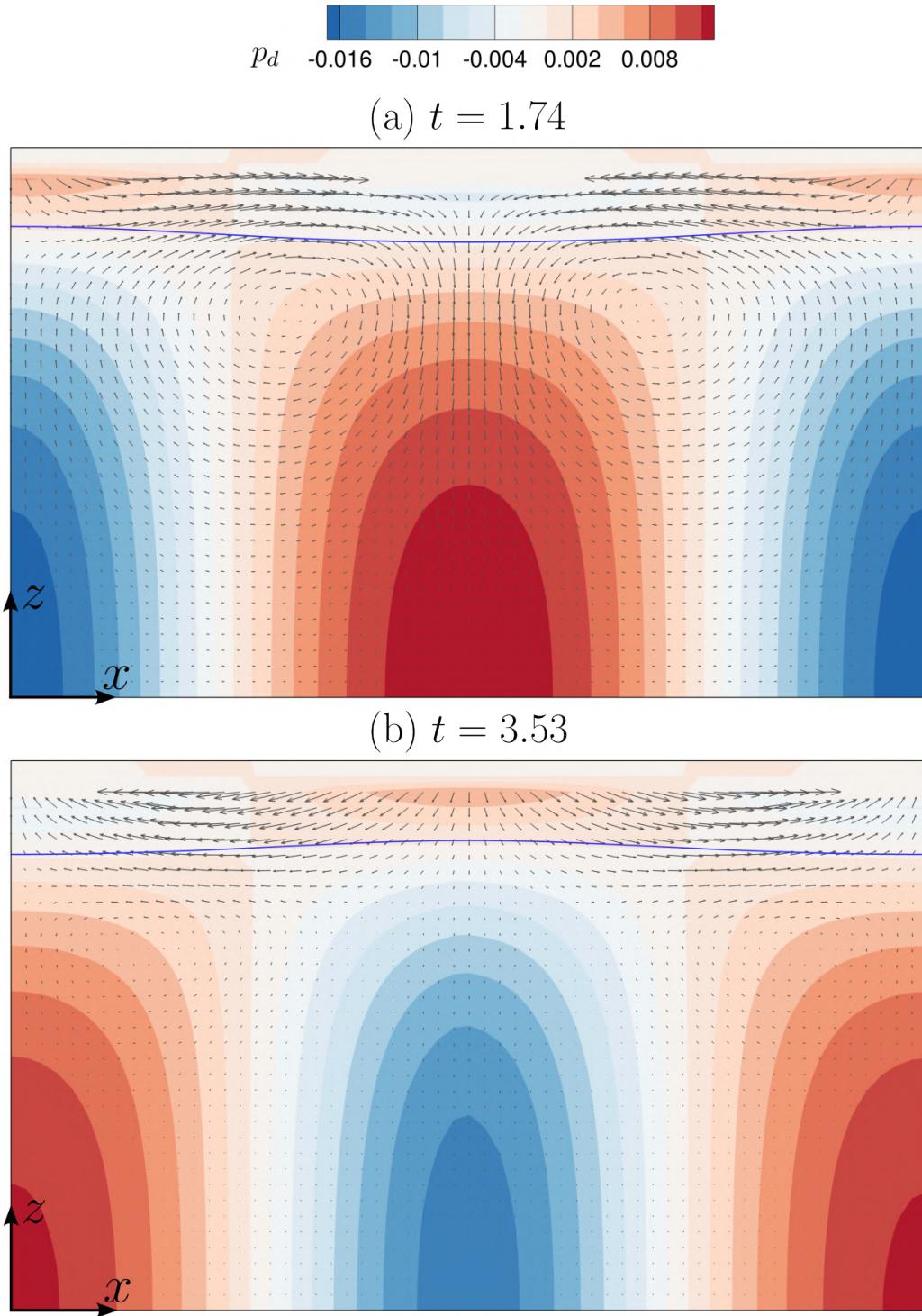


FIGURE 4.14. Velocity and dynamic pressure field for dimensionless times (a) $t = 1.74$ and (b) $t = 3.53$

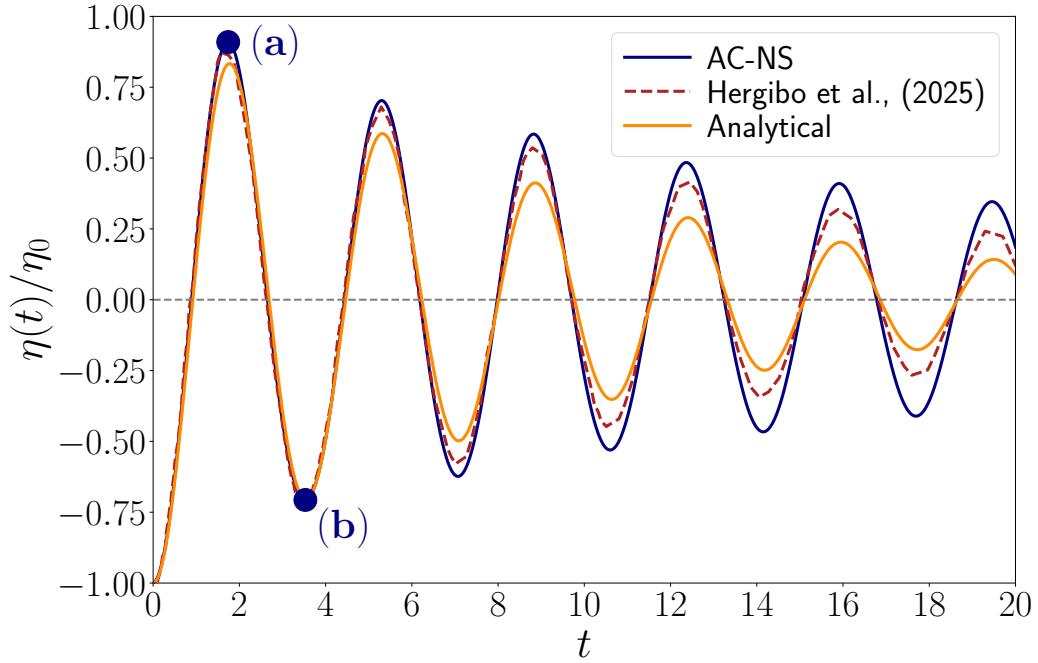


FIGURE 4.15. Evolution of the water level at the left boundary using the current method (blue line with star markers), the method proposed by Hergibo et al. (2025) (red dashed line), and the analytical solution of Wu et al. (2001) (yellow solid line).

4.3.2. Solitary wave propagation

To validate the solver for a travelling wave case, a solitary wave propagation along a channel is simulated. The chosen case is presented in Ramaswamy (1990), and corresponds to a channel of size $20L \times L \times 1.6L$ (x , y , and z directions respectively), where L is the still water depth. The theoretical wave celerity $c_g = \sqrt{gL}$ is set to 1.0 m/s. The Reynolds number, defined as $Re = c_g L / \nu$ is set to 50,000, and the Froude number is set to 1.0.

The solitary wave is generated using the Laitone's analytical approximation (Ramaswamy, 1990). In this formulation, a still water surface with a Boussinesq profile, initially in hydrostatic balance, is suddenly released from the left vertical wall (Yue et al., 2005). The initial profile is set as,

$$A(x, 0) = \frac{A_0}{\cosh^2 \left(\frac{\sqrt{3A_0}}{2} x \right)} \quad (4.24)$$

The non-dimensional initial amplitude A_0 is set to 0.4. Water starts moving due to the hydrostatic pressure gradient in the horizontal direction and a propagating wave driven by gravity and viscosity is formed. After $t = 6$, the influence of the left wall is negligible and the wave may be considered as a solitary wave (Yue et al., 2005). The solitary wave propagates in the positive x direction until it reaches the right vertical wall and climbs up to height of $L + A_{\text{run-up}}$. The instantaneous amplitude is denoted as A_c and the whole profile is described by $A(x, t)$ (see Figure 4.16)

The computational grid has $801 \times 11 \times 81$ nodes and the parameters for the artificial compressibility solver are $CFL = 0.1$, $VN = 0.1$, $\beta = 0.1$, $\varepsilon^\xi = \varepsilon^\eta = \varepsilon^\zeta = 2.5$. The time step is set to $\Delta t = 0.001$, and a minimum of 11 pseudo-iterations is imposed to the solver to ensure convergence.

In Figure 4.17 the progression and run-up of the solitary wave is depicted. The profiles indicate that the wave shape remains stable and well-defined as it propagates. A slight variation in the crest is observed between $t = 2$, $t = 4$, and $t = 6$, which reveals the presence of a weak second mode. This behaviour is expected, as the solitary wave is inherently non-linear (Wang & Liu, 2022).

The run-up obtained in this simulation is $A_{\text{run-up}} = 0.39$, which differs by approximately 2% from the analytical value ($A_{\text{run-up}} = 0.40$). This level of agreement is consistent with the error margins typically reported in validated free-surface CFD studies, where deviations of 2–5% are generally considered acceptable for wave propagation and run-up predictions (Katsidoniaki et al., 2023). On this basis, the combined viscous and numerical damping introduced by the solver remains within acceptable limits. The wave speed, measured from the wave crests in Figure 4.17, is 1.05, which is close to the theoretical value and matches exactly the result reported by previous numerical studies (Yue et al., 2005).

Finally, the mass variation between the initial and final profiles corresponds to 0.02 %, which demonstrates the solver's ability to achieve strong mass conservation.

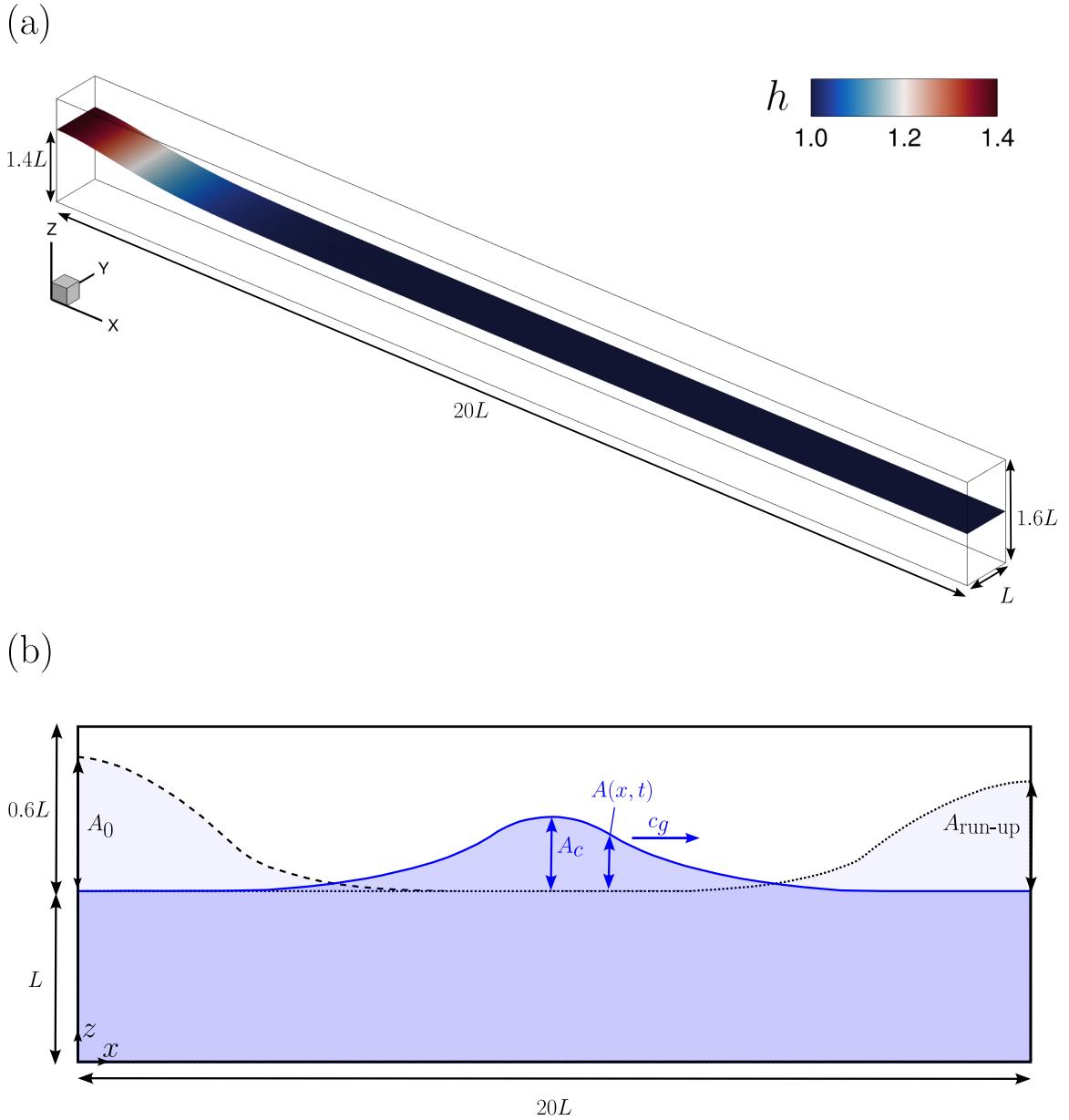


FIGURE 4.16. Schematic diagram of a three-dimensional solitary wave of height A_c propagating over the computational domain. (a) Three-dimensional view of the initial condition. (b) Side view.

The results presented in this chapter demonstrate that the coupled methods form a suitable numerical framework for accurately simulating complex free-surface flows. Using the well-established natural sloshing and solitary wave benchmarks, the full model achieves excellent agreement with analytical solutions and existing literature data.

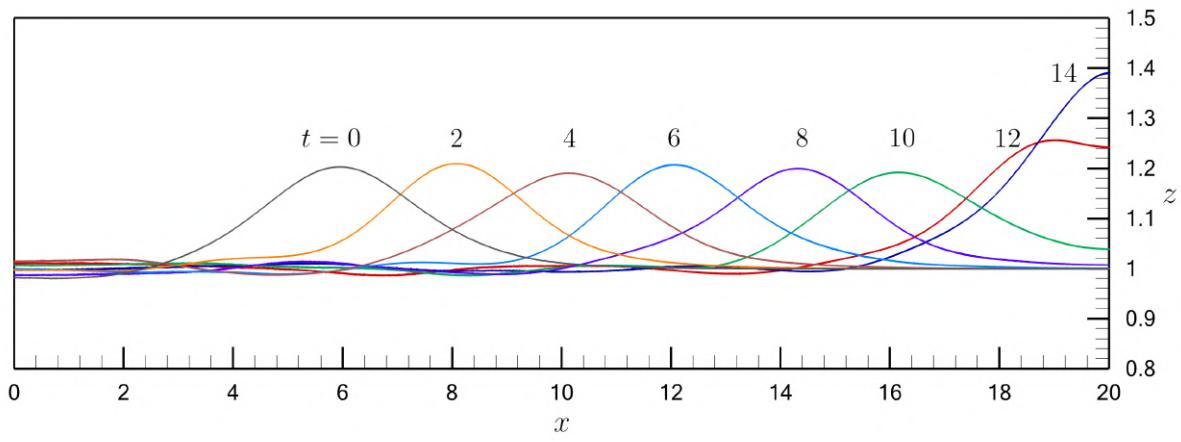


FIGURE 4.17. Instantaneous profiles of the solitary wave at various times (with $t = 0$ set after 6 simulation time units)

CHAPTER 5

SOLITARY WAVE INTERACTIONS WITH IN-STREAM STRUCTURES

5. SOLITARY WAVE INTERACTIONS WITH IN-STREAM STRUCTURES

5.1. Introduction

Solitary waves are often used to study long waves such as tsunamis (Madsen et al., 2008; Winckler & Liu, 2015). The choice of solitary waves in this regard, is motivated by the absence of dominant orbital motions, which means they do not generate strong oscillatory flows. This simplifies the identification of the primary mechanisms responsible for the observed reverse flow around in-stream obstacles.

Bagherizadeh et al. (2021) presented a high-resolution numerical model, which used LES and the Immersed Boundary Method, to study the interaction of a solitary wave with a single rectangular structure. Their results revealed the formation of coherent vortex tubes near the edges of the obstacle, which are advected primarily by the velocity field and induce relatively large deformations on the free surface.

Previous experimental and numerical investigations of solitary-wave interaction with submerged sharp-edged obstacles have shown that flow separation at corners can generate confined vortical structures even in predominantly inviscid or transitional regimes (Chang et al., 2001; Lin & Huang, 2010). Their findings provide a relevant physical context, demonstrating that vortex generation in solitary-wave–structure interactions is highly sensitive to local geometry and obstacle length. Typically, such vortices remain confined unless the boundary layer development is sufficient to sustain larger-scale shear layers.

Building upon this, the present chapter extends the analysis to two identical obstacles with varying separation distances, aiming to investigate how enhanced streaming between them influences wave loading and the resulting vortical dynamics. This configuration is also relevant from a practical standpoint, as during tsunami events propagating upstream in rivers, bridge piers typically act as multiple aligned obstacles perpendicular to the wave direction, potentially leading to complex vortex interactions and intensified localised loading.

5.2. Computational Setup

To numerically generate a stable travelling solitary wave, the method proposed by Yue et al. (2005), as described in the previous chapter, is employed. As before, the flow scales are defined as $\mathcal{L} = L$ and $\mathcal{U} = c_g = \sqrt{gL}$, where L is the still-water depth. This leads to a Reynolds and Froude number of 50,000 and 1.0, respectively, placing the present configuration within the turbulent range reported in previous experimental studies of solitary-wave propagation (Higuera et al., 2018). For numerical stability purposes, an initial amplitude of $A_0 = 0.2$ is used, which leads to a solitary wave of amplitude 0.1, when it travels downwave.

A computational domain of size $16L \times 8L \times 1.4L$ is set up for the simulation. The obstacles are located 7 length units away from the generation boundary, with separation distances of $s = 0.5, 1.0$, and 2.0 tested in different cases. A structured grid comprising $N_x \times N_y \times N_z = 641 \times 321 \times 71$ nodes is employed, resulting in a total of 14,609,031 computational nodes, and spacings of $\Delta x = 0.025$, $\Delta y = 0.025$, and $\Delta z = 0.02$.

The obstacles are represented within the flow as solid wall boundaries and simulated using the *I*-Blanking technique (Ge, 2004), which was adapted for the purposes of this code. The non-dimensional time step for the simulations was set to $\Delta t = 0.001$.

The simulation parameters are summarised in table 5.1

Parameter	Values
$N_x \times N_y \times N_z$	$641 \times 321 \times 71$
Total number of nodes	14,609,031
Δx	0.025
Δy	0.025
Δz	0.02
Δt	0.001
s	0.5, 1.0, 2.0
Obstacle edge	1.0

TABLE 5.1. Computational setup and simulation parameters.

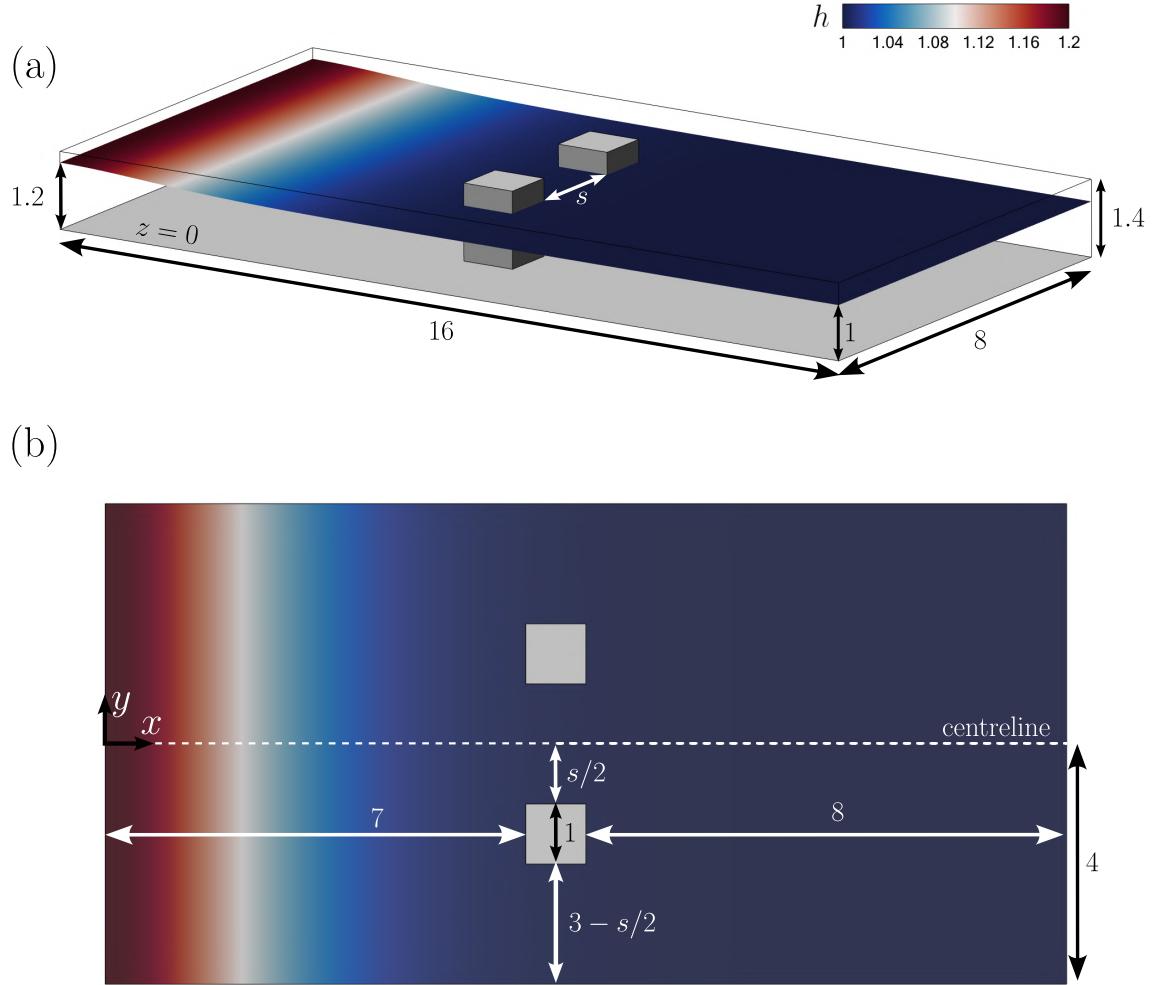


FIGURE 5.1. Initial condition of the computational domain for the study of the interaction between a solitary wave and two square obstacles. (a) Isometric view. (b) Top view.

This computational setup allows for a total simulation time of 8.5 time units, covering the period during which the solitary wave interacts with the obstacles. Beyond this point, the solution becomes unstable, and the simulation cannot proceed. The underlying causes, implications, and potential strategies to mitigate the onset of numerical instabilities are discussed in section 5.7 of this chapter. Nevertheless, this simulation window enables a comprehensive analysis of several key phenomena arising from the interaction between the solitary wave and the obstacles, which are explored throughout this chapter.

5.3. Run-up

To evaluate the effect of separation on the interaction between the solitary wave and the obstacles, a 3D isometric view of the free surface is shown in Figure 5.2.

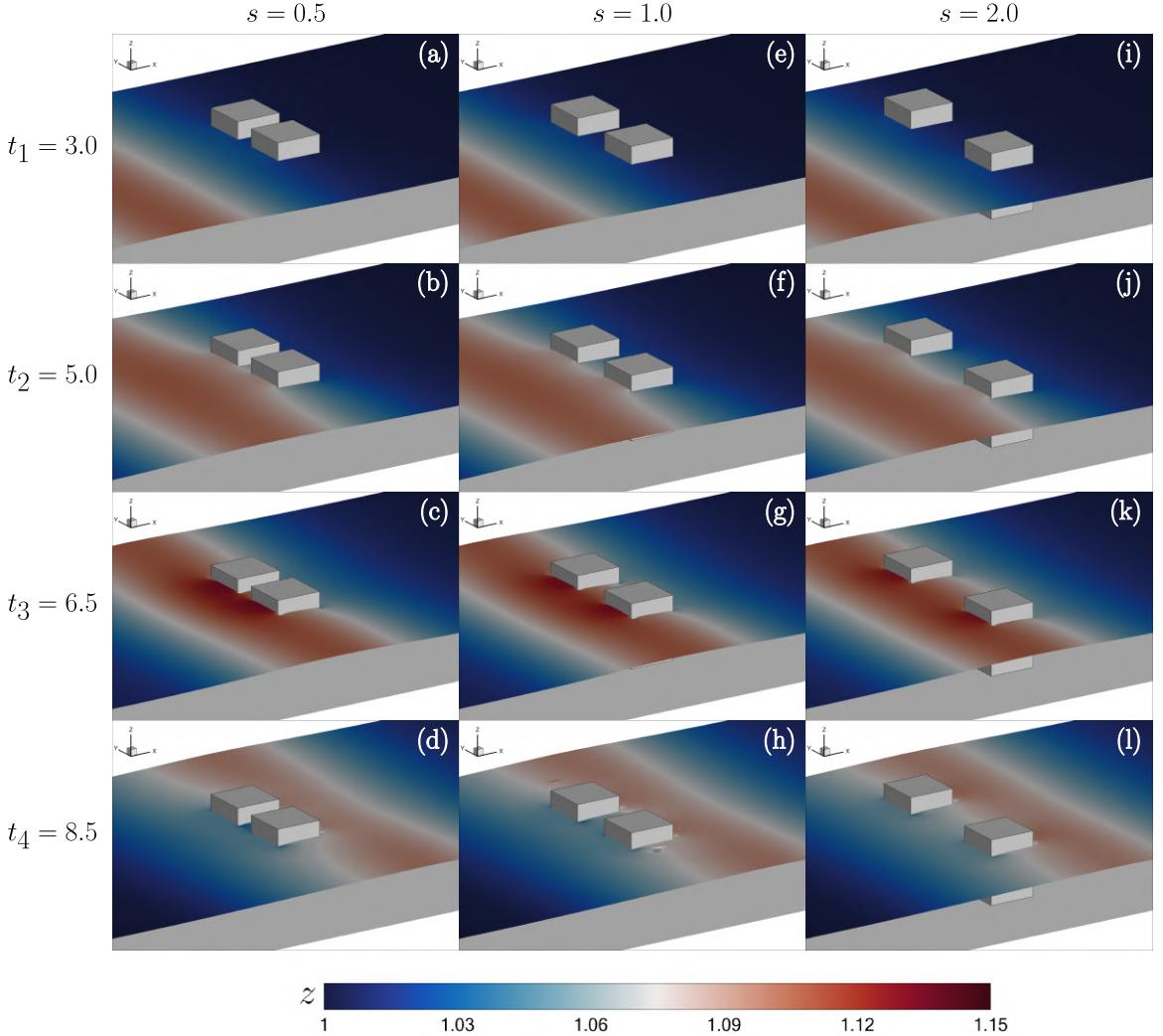


FIGURE 5.2. Snapshots of the wave elevation for the tested obstacle separations at four different times: $t_1 = 3.0$, $t_2 = 5.0$, $t_3 = 6.5$, and $t_4 = 8.5$. Panels (a)–(d) correspond to $s = 0.5$, (e)–(h) to $s = 1.0$, and (i)–(l) to $s = 2.0$.

As shown in the figure, the wave reaches its maximum height in the vicinity of the obstacles, accompanied by a slight deformation of the spanwise pattern around them. Nonetheless, no significant differences are observed between the tested separations. For

a more detailed analysis, the time series of the water depth at the centre of the front face is shown in Figure 5.3.

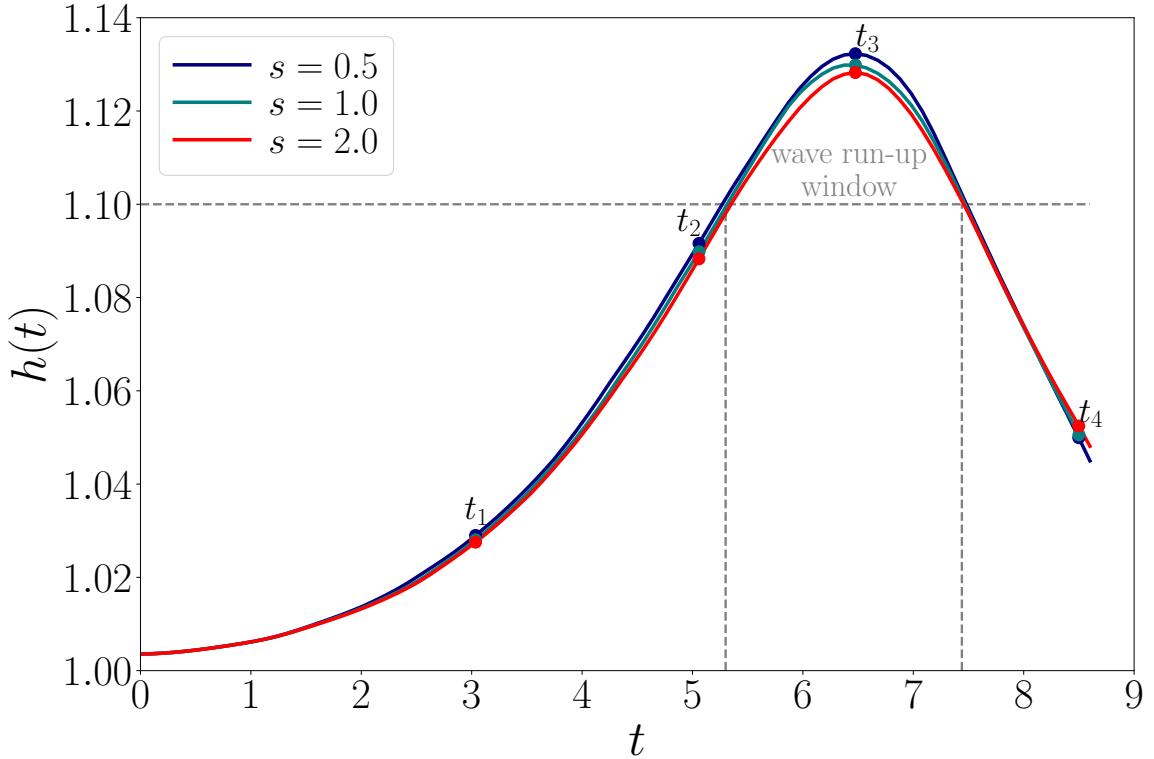


FIGURE 5.3. Time series of the wave run-up at the centre of the front face of the obstacles for the three tested separations. The dashed line at $h = 1.1$ corresponds to the undisturbed wave amplitude. The period in which the run-up exceeds this level defines the wave run-up window, used to characterize the peak interaction phase.

As can be seen, the three cases follow almost identical run-up evolutions, with slightly higher values for smaller separations. Nevertheless, the difference between the highest run-up for the smallest separation ($h_{\max}^{\text{run-up}} = 1.13$) and the lowest for the largest separation ($h_{\max}^{\text{run-up}} = 1.128$) is only 0.35%.

5.4. Wave hydrodynamic loading

To calculate the hydrodynamic horizontal loading on an obstacle, for a specific time, we compute (Christou et al., 2025)

$$F_x = F_p + F_\tau \quad (5.1)$$

Where F_p is the force induced by the pressure loading on the front and rear sides of the obstacle, and F_τ is the friction exerted by the flow on the lateral walls. They read

$$F_p = \int_S \mathbf{H}(\phi)(p \hat{n}) \cdot \hat{n} dS = \int_{S_{\text{front}}} \mathbf{H}(\phi)(p \hat{n}) \cdot \hat{n} dS + \int_{S_{\text{rear}}} \mathbf{H}(\phi)(p \hat{n}) \cdot \hat{n} dS \quad (5.2)$$

$$F_\tau = \int_S \mathbf{H}(\phi)(\boldsymbol{\tau} \cdot \hat{n}) \cdot \mathbf{e}_x dS = \int_{S_{\text{inner}}} \mathbf{H}(\phi)(\boldsymbol{\tau} \cdot \hat{n}) \cdot \mathbf{e}_x dS + \int_{S_{\text{outer}}} \mathbf{H}(\phi)(\boldsymbol{\tau} \cdot \hat{n}) \cdot \mathbf{e}_x dS \quad (5.3)$$

$\mathbf{H}(\phi)$ denotes the Heaviside step function, which is 1 if $\phi > 0$ and 0 otherwise. p is the total pressure, \hat{n} is the unitary vector pointing outwards from the obstacle surface, \mathbf{e}_x is the unitary vector in the x direction, and $\boldsymbol{\tau}$ is the shear stress tensor. The front, rear, inner and outer sides refer to the faces with normals $-\mathbf{e}_x$, \mathbf{e}_x , $-\mathbf{e}_y$, and \mathbf{e}_y , respectively.

Figure 5.4 shows the evolution of the frontal (solid lines) and rear (dashed lines) forces. The time series reveals minor differences between the tested separations. The highest frontal loading occurs for the case with $s = 0.5$, while the opposite trend is observed on the rear side of the obstacles. The duration of the peak force coincides with the wave run-up window defined in Figure 5.3, clearly linking wave height to obstacle loading.

The temporal evolution of the pressure-induced forces, particularly the slight variations between separations seen in Figure 5.4, can be further elucidated by considering the vorticity fields presented in Figures 5.9–5.11. As the solitary wave interacts with the obstacles, vortices form and grow near the front and rear corners. Within these coherent structures, the centrifugal effect induces a local pressure reduction (Saffman, 1992). Consequently, the development of vorticity near the front face temporally modulates the integrated pressure load, contributing to the force history's shape. The rear-face force, which becomes positive (downwave-directed) shortly after the peak frontal loading, is similarly influenced

by the shedding and advection of these corner vortices, which create low-pressure regions on the lee side of the obstacles. The minor differences in force magnitude between separations, such as the higher frontal force for $s = 0.5$, are consistent with the intensified and more merged vortical activity observed in the narrowed gap (e.g., Figure 5.9), which enhances the local pressure deficit on the inner portions of the front faces. Thus, the transient pressure forces result not only from the global wave kinematics but also from the dynamic evolution of vorticity generated at the obstacle edges.

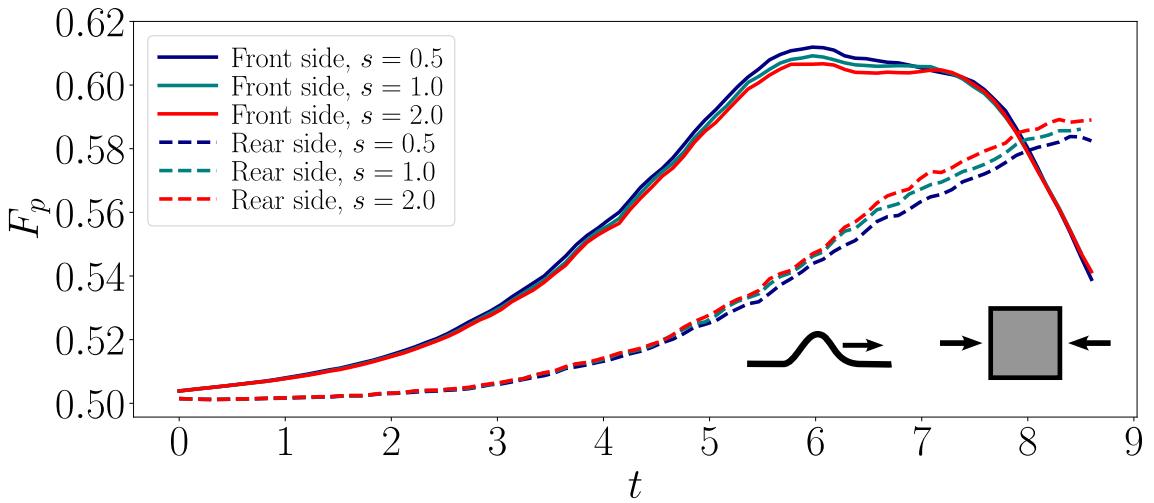


FIGURE 5.4. Pressure-induced force on the obstacles: front face (solid line) and rear face (dashed line).

Figure 5.5 shows the lateral force on the inner side (solid line) and outer side (dashed line) of the obstacle. The magnitude of these forces is three orders of magnitude smaller than the frontal and rear forces discussed previously. In addition, the force acting on the outer sides of the obstacle exhibits no significant variation across the tested separations. By contrast, a markedly different behaviour is observed on the inner side, where the force experienced by the obstacle in the smallest separation case exceeds that on the outer side by approximately 50%. This force imbalance generates a resultant torsional moment on the obstacle, which may lead to unexpected vibrations or even structural fatigue under prolonged or repeated loading conditions.

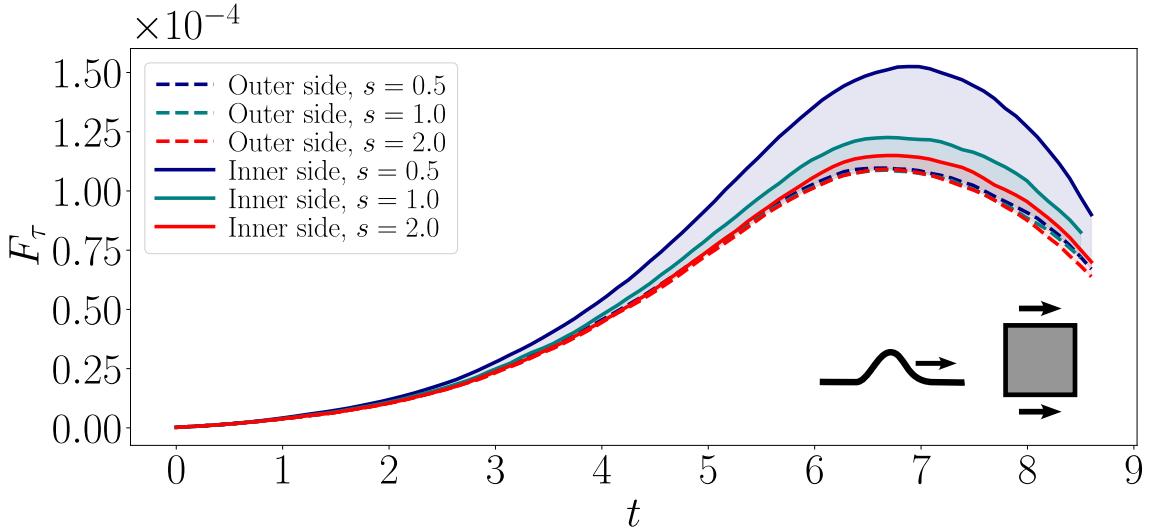


FIGURE 5.5. Friction-induced force on the obstacles: inner face (solid line) and outer face (dashed line).

5.5. Bed shear stresses

To better understand the potential effects of the force imbalance between the outer and inner sides on erosion and obstacle scour induced by the transient wave, an analysis of the bed shear stresses is presented. This analysis is based on the calculation of the non-dimensional shear velocity (Escauriaza & Sotiropoulos, 2011a).

$$u_\tau = \sqrt{\frac{1}{Re} \left. \frac{\partial |\mathbf{u}|}{\partial z} \right|_{z=0}} \quad (5.4)$$

Snapshots for t_1-t_4 for the tested separations, in the vicinity of the obstacles, are presented in Figure 5.6.

As shown, a propagating front of elevated u_τ values travels with the wave. Upon interacting with the obstacles, this front generates regions of both enhanced and reduced shear (see Figures 5.6c–d, g–h, k–l). In all three cases, increased shear is observed at the obstacle corners, where abrupt changes in flow direction lead to large velocity gradients. In contrast, the central region along the frontal face of the obstacles exhibits reduced shear stress. This is in agreement with the behaviour reported by Bagherizadeh et al. (2021),

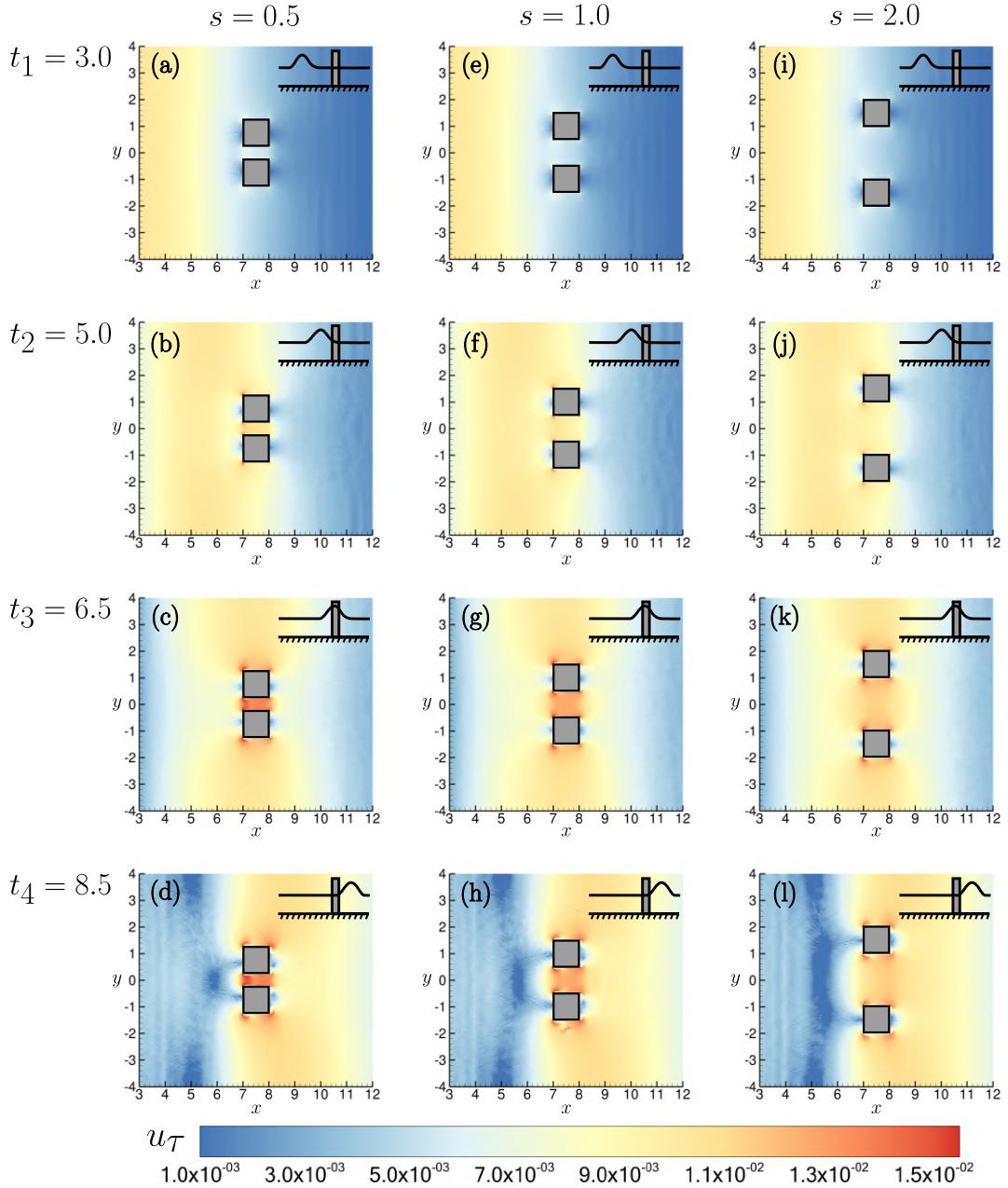


FIGURE 5.6. Snapshots of filled contour maps of the non-dimensional shear velocity (u_τ) for the tested obstacle separations at four different times: $t_1 = 3.0$, $t_2 = 5.0$, $t_3 = 6.5$, and $t_4 = 8.5$. Panels (a)–(d) correspond to $s = 0.5$, (e)–(h) to $s = 1.0$, and (i)–(l) to $s = 2.0$. A schematic sketch of the wave propagation stage is shown in the upper-right corner of each panel.

which observed that the dominant vortical structures due to the interaction between the wave and the obstacle are located around the corner.

This behaviour differs from that observed in shear currents around obstacles at high Reynolds numbers, where an adverse pressure gradient induces flow separation upstream of the obstacle, leading to the formation of a Horseshoe Vortex System (HSV). This persistent, self-sustained vortical structure enhances shear stresses and is responsible for the erosion typically observed in front of obstacles in rivers (Escauriaza & Sotiropoulos, 2011a, 2011b; Link et al., 2012). Figure 5.7 presents a snapshot of the velocity field for the case $s = 0.5$ at time $t_3 = 6.5$, showing planes located at the obstacle centreline ($y = -s/2 - 1/2$) and at $z = \Delta z$. The figure confirms the absence of an HSV in front of the obstacle, which, together with the velocity reduction in this region, explains the low shear stress observed in Figure 5.6.

In addition to the previously discussed features, Figure 5.6 also reveals that, in the case $s = 0.5$, the high shear stress regions at the obstacle corners merge to form a continuous high-shear channel between the obstacles (see Figures 5.6c–d). To further investigate this effect, time series of u_τ at points located 0.25 length units to the left and right of the centre of the obstacle are presented in Figure 5.8.

The figure depicts a similar behaviour to that observed in Figure 5.5. While the u_τ values on the outer side remain comparable across all cases, significant differences emerge on the inner side, where decreasing separation leads to a greater contrast between the inner and outer sides. This imbalance in shear velocity may potentially result in uneven erosion patterns, which could compromise the structural stability under wave action and lead to local scouring, foundation exposure, or eventual structural failure over time.

5.6. Vortical structures

As shown in the analyses presented in the preceding sections, the intrinsically unsteady nature of wave propagation and its interaction with the obstacles leads to different stages

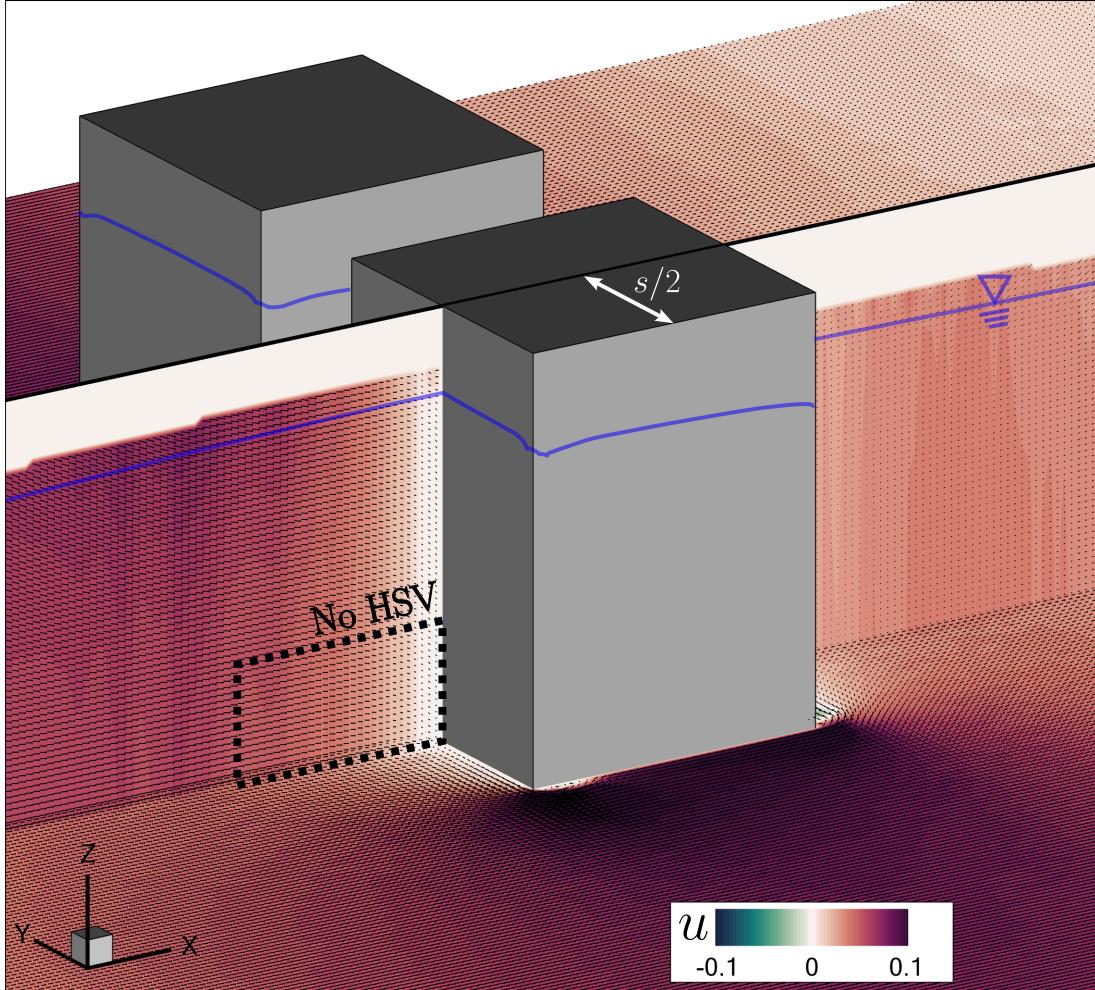


FIGURE 5.7. Snapshot of the u -velocity component around the obstacle at $t = 6.5$ for $s = 0.5$. The dashed-line rectangle in front of the obstacle highlights the absence of a Horseshoe Vortex (HSV) system.

in the flow dynamics. To further this analysis and examine the influence of obstacle separation on the coherent flow structures around the obstacles, plots of z -vorticity at $z = 0.5$ are presented in Figures 5.9, 5.10, and 5.11 for times $t_2 = 5$, $t_3 = 6.5$, and $t_4 = 8.5$, respectively.

The figures show that the vortical structures are primarily concentrated in the vicinity of the obstacles, with slightly broader spanwise regions of vorticity developing around the corners. As the wave passes over the obstacles, these vortices grow, leading to the formation of small recirculation cells of characteristic size $\mathcal{O}(10^{-1})$ at the rear corners.

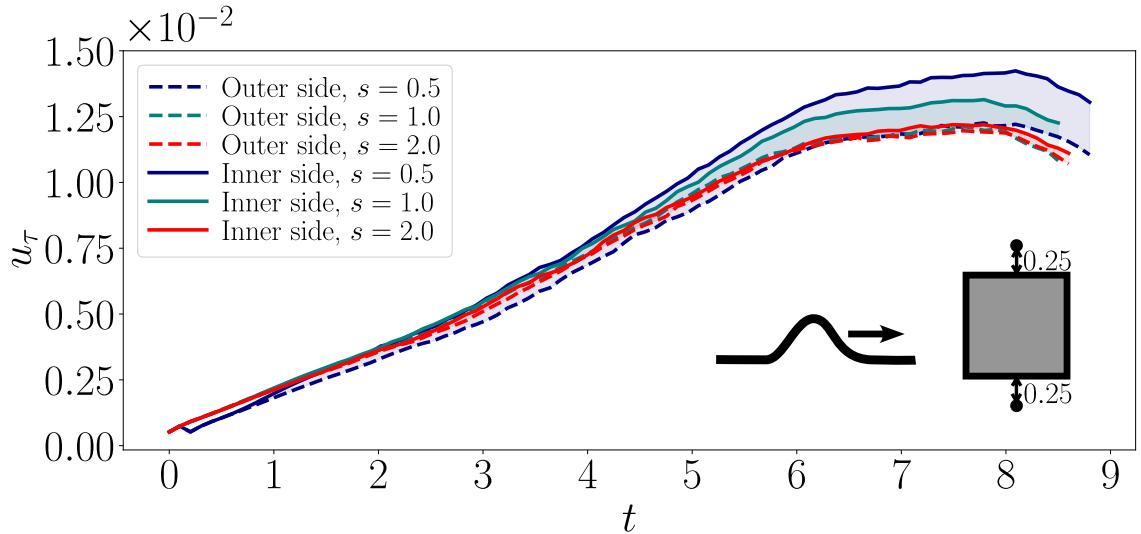


FIGURE 5.8. Time series of u_τ for a point located at a distance of 0.25 from the obstacle surface, measured outwards (dashed line), and inwards (solid line) for the three tested separations.

Despite the presence of these vortical structures, they do not evolve into a self-sustained shear layer. This can be attributed to the absence of a fully developed boundary layer along the obstacle surfaces that separates at the downstream edge. A boundary layer requires sufficient streamwise development to accumulate the shear necessary for instability mechanisms such as separation and shear-layer roll-up. On short surfaces, and under a decaying streamwise velocity transient (following the passage of the wave crest), this development is prematurely truncated, resulting in insufficient shear at the trailing edge. This interpretation aligns with findings from direct numerical simulations of wave boundary layers (Önder & Liu, 2020; Ozdemir et al., 2014; Sandoval & Eaves, 2025), where instabilities and transition mechanisms only emerge after substantial boundary-layer growth and streak development.

The vortices observed at the rear corners are likely caused by local flow separation due to pressure gradients and geometric effects, rather than by classical shear-layer instability. Experimental and numerical studies on solitary wave interaction with rectangular obstacles (Chang et al., 2001; Lin & Huang, 2010) confirm that sharp edges can induce

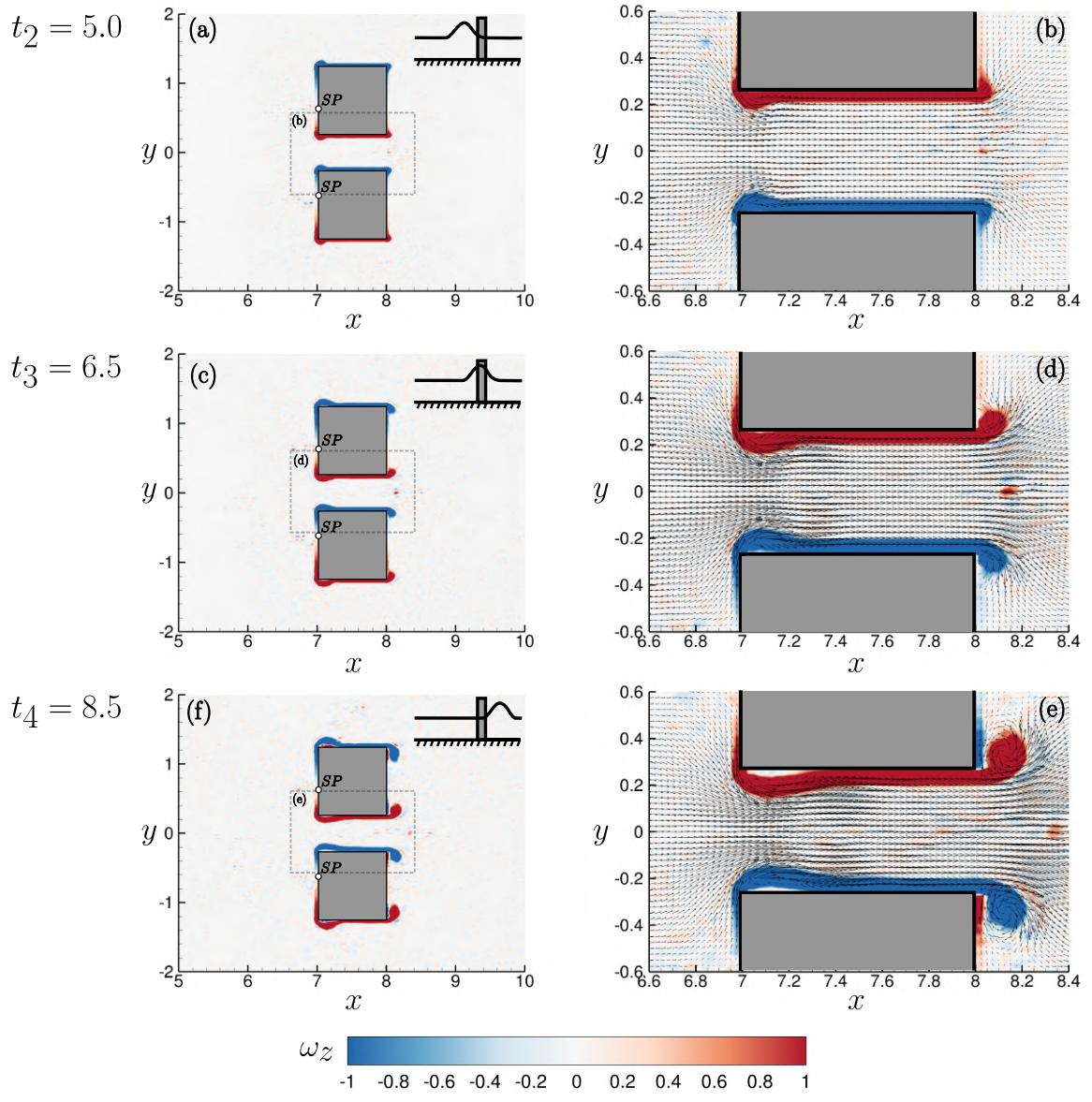


FIGURE 5.9. Computed z -vorticity at $z = 0.5$ for $s = 0.5$ at times $t_2 = 5.0$, $t_3 = 6.5$, and $t_4 = 8.5$. Panels (b), (d), and (e) are zoomed-in views of panels (a), (c), and (f), respectively. The stagnation point on the front face of the obstacle is indicated by SP .

vortical structures even when the flow remains largely inviscid or transitional. These vortices are expected to remain confined and not develop into large-scale mixing layers unless the obstacle length permits sufficient boundary-layer maturation.

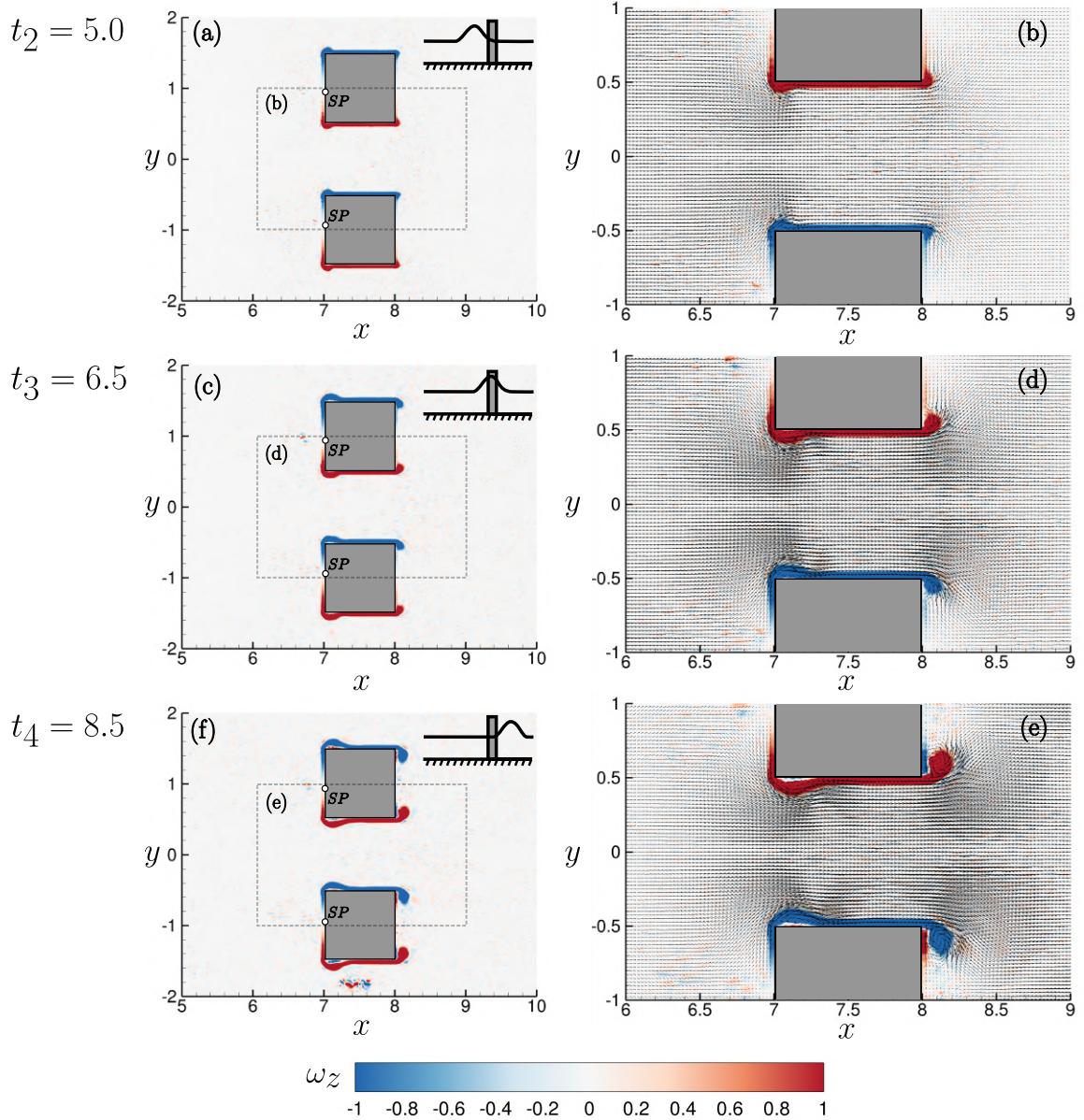


FIGURE 5.10. Computed z -vorticity at $z = 0.5$ for $s = 1.0$ at times $t_2 = 5.0$, $t_3 = 6.5$, and $t_4 = 8.5$. Panels (b), (d), and (e) are zoomed-in views of panels (a), (c), and (f), respectively. The stagnation point on the front face of the obstacle is indicated by SP .

A final observation concerns the influence of obstacle separation on the position of the stagnation point along the front face of the obstacles. The results indicate that, as the separation decreases, the stagnation point (defined here as the location where the y -component of the velocity changes sign in the vicinity of the obstacle) shifts progressively

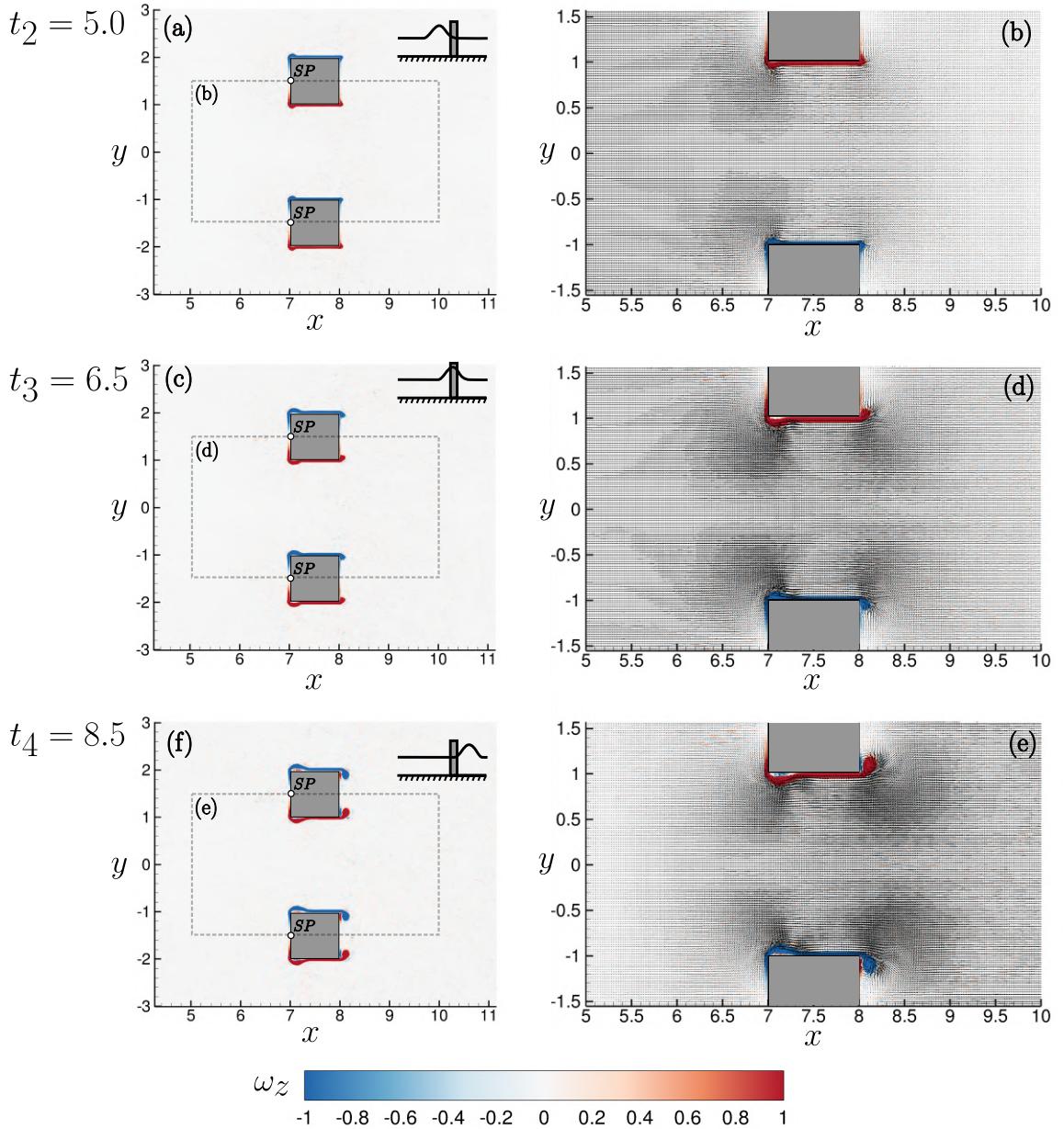


FIGURE 5.11. Computed z -vorticity at $z = 0.5$ for $s = 2.0$ at times $t_2 = 5.0$, $t_3 = 6.5$, and $t_4 = 8.5$. Panels (b), (d), and (e) are zoomed-in views of panels (a), (c), and (f), respectively. The stagnation point on the front face of the obstacle is indicated by SP .

towards the inner corner. This shift reflects the increasing asymmetry in the local flow field induced by the narrowing gap between obstacles. Table 5.2 presents the distance from the geometric centre of the front face to the stagnation point at time $t_3 = 6.5$. The rightmost

s	Distance from the Center	Deviation %
0.5	0.106	21.20
1.0	0.063	12.68
2.0	0.012	2.40

TABLE 5.2. Deviation of the stagnation point (SP) from the centre of the front face of the obstacle.

column reports the corresponding deviation as a percentage, where 0% indicates alignment with the face centre and 100% corresponds to the inner corner.

The inward shift of the stagnation point is consistent with the previously reported asymmetries in lateral loading and bottom shear stresses, highlighting the role of enhanced flow streaming through the inter-obstacle channel.

5.7. Discussion

The results presented in this chapter show that the interaction between a solitary wave and a pair of square obstacles is influenced by obstacle separation only in specific aspects of the flow. The evolution of the free-surface elevation, run-up, and frontal loading remains largely insensitive to the gap width within the range of separations tested. This behaviour indicates that, for solitary waves of moderate amplitude, the global hydrodynamic forcing is dominated by the kinematics of the incoming wave rather than by channelling effects between the obstacles. The differences in maximum run-up were found to be below 1%, confirming that the wave crest interacts with the obstacle front faces in a quasi-symmetric manner for all separations.

In contrast, the flow within the inter-obstacle region exhibits pronounced sensitivity to the gap width. The inner-side shear forces and bed shear stresses increase considerably as the separation decreases, with the case $s = 0.5$ showing both a persistent shear imbalance and the formation of a continuous high-shear region between the obstacles. This behaviour is consistent with the enhanced streaming induced by the restricted gap, which accelerates the flow and sharpens velocity gradients near the inner obstacle faces. The resulting force

asymmetry generates a net torsional moment that may be relevant in practical configurations where repeated loading or structural flexibility are present.

The vorticity fields reveal that the vortical structures produced during the interaction remain confined to the vicinity of the obstacle corners. These vortices do not evolve into a coherent shear layer or a large-scale recirculation system, regardless of obstacle spacing. This outcome derives from the transient nature of the boundary layer under solitary waves, which lacks the streamwise development required for separation-driven instability. The absence of a Horseshoe Vortex System (HSV) further confirms that the incoming flow does not maintain the adverse pressure gradient and steady upstream conditions typically associated with HSV formation in riverine currents. Instead, the generated vortices resemble small-scale separation bubbles whose growth is limited by the rapid decay in streamwise velocity following the passage of the wave crest.

A consistent trend observed across all analyses is the progressive shift of the stagnation point toward the inner obstacle corner as the separation decreases. This shift reflects a redistribution of the local flow field caused by the increased momentum flux through the inter-obstacle gap. The resulting asymmetry aligns with the trends observed in inner-side shear stresses and lateral force imbalances, demonstrating that the gap-induced streaming is the main mechanism driving the deviation from symmetric loading conditions.

Finally, although the simulation captures the essential flow physics during the wave-obstacle interaction, numerical instabilities arise shortly after the peak interaction phase. Their onset coincides with the stage at which the reflected wave, free-surface deformations, and vortical structures interact within a domain that is no longer dominated by a single incoming disturbance. These numerical issues are likely related to the combined effects of free-surface steepening, grid stretching near the interface, and the accumulation of numerical errors in the artificial-compressibility formulation under strongly unsteady conditions. Potential strategies to mitigate these instabilities include reconsidering the time-integration approach—particularly the use of a pointwise TVD Runge–Kutta scheme, which emphasises hyperbolic behaviour and may reduce elliptic smoothing. Alternative schemes with

enhanced dissipative or stabilising properties may offer improved robustness. These aspects will be examined in future developments of the numerical model.

CHAPTER 6

CONCLUSIONS AND FUTURE WORK

6. CONCLUSIONS AND FUTURE WORK

In this thesis, we presented a fully unsteady, single-phase three-dimensional free-surface viscous numerical model based on an Artificial Compressibility Navier–Stokes solver coupled with the Level-Set method for capturing the free-surface evolution. Through the development and application of this framework, the specific objectives established in the Introduction were addressed: implementing a robust and physically consistent treatment of free-surface boundary conditions, ensuring strong global volume preservation via a geometric reinitialisation strategy, and analysing the interaction between free-surface flows and in-stream obstacles with emphasis on the vortical dynamics generated by the wave–structure interaction.

First, a robust methodology was implemented to impose physically consistent normal and tangential boundary conditions at the free surface, achieved by integrating the Ghost-Fluid Method with a modified Weighted Least Squares procedure. Second, global volume preservation was ensured by incorporating a geometric-based reinitialisation strategy for the Level-Set function, adapted and extended to maintain strong mass conservation over long simulation times. Third, the solver was applied to analyse the interaction between free-surface flows and in-stream obstacles, with particular attention to the vortical dynamics arising from the wave–structure interaction. The emphasis placed on enforcing physically consistent free-surface boundary conditions and ensuring mass conservation proved critical for achieving accurate flow predictions, thereby confirming one of the central hypotheses of this work. The model was validated against benchmark cases of increasing complexity and subsequently applied to simulate the propagation of a solitary wave over two in-stream obstacles at varying separations, highlighting the rich hydrodynamic behaviour generated by the interaction. Together, these contributions provide a coherent and flexible numerical framework capable of accurately representing unsteady free-surface flows and advancing the modelling of wave–structure interactions relevant to coastal and marine energy applications.

The solution of the flow variables is carried out using the dual-time-stepping AC iteration scheme. The equations are transformed into generalised curvilinear coordinates and discretised using a second-order-accurate finite-volume method on a non-staggered computational grid. The system of equations is integrated using a pressure-based implicit preconditioner, enhanced with local-time-stepping, and advanced in time with a second-order accurate three-point-backwards Euler-type time stepping scheme. The free-surface evolution was incorporated explicitly into the momentum equations through a pressure-splitting procedure, treating the hydrostatic and dynamic pressure components separately. The hydrostatic pressure gradient is treated as a gravitational source term, and alongside the velocity field, acts as a predictor for the evolution and later convergence of the dynamic pressure field within pseudo-time. This method showed to be a robust approach to link the free-surface dynamics with the bulk flow field in a physically-consistent manner, and to the best of the authors knowledge, this is the first time in literature that an approach of these characteristics is coupled with an AC solver, thoroughly tested and validated, and used to simulate wave-obstacle interactions.

The single-phase approach adopted in this work, aimed for a physically-consistent solution procedure, where fluid material properties remained constant across the solution domain, and therefore, mass and momentum fluxes were not artificially smeared in the vicinity of the free surface due to the use of transition smoothing functions for density and viscosity. In order to capture the free-surface as a dynamic boundary condition, normal and tangential stress balances were incorporated into the equation through the ghost-fluid method. This formulation has the advantage of retaining the conservative properties of the solver, as well as reducing the amount of computational nodes outside the fluid domain, as the air phase is not needed to be resolved (just three nodes outside the fluid domain is needed for ghost-fluid extrapolation purposes). As a result, the model achieves enhanced computational performance and scalability for large-scale simulations in parallel environments.

The resulting system of equation is discretised using a pressure-based residual smoothing method, where the linear fluxes of the coupled system are treated implicitly. This approach, combined with the application of the Block-Diagonal algorithm with the Beam and Warming (1976) factorisation method and the Implicit Residual Smoothing of Sotiropoulos and Constantinescu (1997) lead to three tridiagonal matrices for ξ^1 , ξ^2 , and ξ^3 directions, respectively. These matrices are efficiently inverted in parallel using the state-of-the-art solver for massive tridiagonal systems PaScaL (Kim et al., 2021). Even though this method leads to a solution procedure that correctly captures the elliptic nature of the pressure gradient terms and continuity equation, difficulties were encountered when the free-surface was nearly flat, and only a small set of nodes were within the water phase in some of the directions. This produced that some regions of the domain led to tridiagonal matrices with just a few of non-zero entries, and therefore, ill-posed for inversion. Although numerical techniques, like slope limiters and implicit residual smoothing, were implemented to stabilise the solution, this issue remained as a dominant source of instability. Future work in this regard may be associated to the use of multi-stage pointwise method as Implicit Total-Variation-Diminishing Runge-Kutta (TVD-RK) algorithms. This methods advance the system at each grid point independently, and therefore variations of the flow variables travel across the domain as waves, enhancing the predominance of the hyperbolic terms of the equations and therefore lacking the elliptic coupling offered by the Block-diagonal algorithm, which could delay the convergence in pseudo-time over the whole domain. However, the stability, monotonicity and TVD properties of this approach could compensate these drawbacks.

The free surface evolution is captured with the Level-Set method. The non-linear hyperbolic equation that represents the free-surface advection is advanced in time using an explicit TVD-RK method. The convective term of the equation is discretised using a third-order-accurate Weighted Non-Oscillatory (WENO) method, which is able to produce stable solutions with relatively small amounts of numerical dissipation. Despite the use of high-order schemes, the Level-Set function's signed-distance property deteriorates after

advection, leading to cumulative mass loss—a well-known limitation of classical Level-Set formulations. To mitigate this issue, a geometric-based redistancing procedure adapted from Ausas et al. (2011) was implemented to enforce strong global mass conservation. The method reconstructs the interface as a piecewise linear surface using the Marching Tetrahedron Algorithm, enabling accurate computation of distances and local volumes from vectorial geometry formulas. Two additional steps enforcing local and global mass conservation yield a redistanced Level-Set field that maintains both the geometric consistency of the interface and the total fluid volume. The Zalesak’s disk benchmark case showed that applying the geometric reinitialisation at every time step preserves the global volume variation around 0.02 %. This represents a substantial improvement over the original geometric reinitialisation proposed by Ausas et al. (2011), which resulted in a volume error of approximately 5% for the same case. Furthermore, it greatly outperforms the widely used method of Sussman and Fatemi (1999), which produces a volume discrepancy of approximately 10%, even when the present reinitialisation method is not applied every time step, making our approach highly computationally efficient. This implementation effectively addresses two of the main limitations of standard Level-Set approaches: the progressive loss of volume during long simulations and the deterioration of interface geometry under strong deformations.

To test and demonstrate the numerical solver capabilities, a set of benchmark cases are evaluated for the different modules of the solver developed in this thesis. The first group of test cases showed the accuracy of the method in extrapolating the pressure and velocity fields onto the ghost fluid nodes by using the free-surface dynamic boundary conditions in different configurations. To accurately reconstruct the necessary velocity and pressure gradients to impose the dynamic boundary conditions in arbitrary spatial configurations of the free surface, a Weighted Least Square Method (WLSM) was implemented. The choice of the weights for the method was based on a combination of the local geometry of the free-surface and its relative position to the background computational grid, assigning larger influence to nodes located closer and alongside the normal direction towards the bulk flow. This approach, together with the adaptation of a Venkatakrishnan’s slope limiter

(Venkatakrishnan, 1993) used in unstructured grid, showed to be robust to produce accurate and stable reconstructed gradients in different configurations (purely geometric test cases as the rotating cylinder and sphere, or physically realistic ones, such as the Stokes wave velocity field extrapolation). The full model was tested for two wave problems: a low amplitude sloshing wave, and a propagating solitary wave. In both cases, the solver showed its capability to generate accurate and strongly mass preserving solutions.

In the final part of this thesis, the interaction of a 0.1 amplitude solitary wave with two identical obstacles, at three different separations, was investigated. Obstacle spacing changes showed to have limited influence on the run-up of the waves and the pressure loading on the front and rear sides of the obstacle. However, it showed to induce large discrepancies on the lateral friction forcing and bed shear stresses between inner and outer sides. It was observed that smaller separation values produced enhanced streaming through this passage and therefore induced larger values for lateral forcing and bed shear stresses. Conversely, the front side of the obstacles presented relatively smaller bed shear stresses, an outcome that differs from typical current-obstacle interactions. This may be attributed to the absence of a Horseshoe Vortex (HSV) system at the front of the obstacle in the solitary wave case. The explanation for this might be associated to the time scales of the wave propagation, as not enough centrifugal instability is developed by the adverse pressure gradient on the front of the obstacle to generate a self-sustained HSV system.

Similarly, wave propagation did not induce the formation of a self-sustained turbulent shear layer at the rear corners of the obstacles. An explanation for this observation may rely on similar arguments as the ones proposed in the HSV case: the combination of the obstacle edge length with the the time scale of the stream induced by the wave are not enough to drive a full transition to turbulence. On the one hand, the length is too small to generate a bypass transition through Tollmien–Schlichting waves, and the time scale is too short to allow the formation, growth and breakdown of streaks to roll-up into a self-sustained shear layer. Nevertheless, these observations are preliminary and highlight the need for further investigation with different wave amplitudes, depth, obstacle sizes and separations,

in order to draw a more general picture of the vortex dynamics governing wave–structure interactions.

Future work will be focused on the full integration of the turbine representation module developed in Sandoval et al. (2021), with the free-surface numerical framework developed herein. The aforementioned, in addition to the extension of the solver capabilities in order to recreate different wave and current conditions, will allow us to simulate the effects of complex sea states on marine energy devices. This is a key step towards assessing local-scale dynamics of hydrokinetic turbines operation, and a valuable resource for further design of reliable components, as well as optimised arrays layout that takes into account fully unsteady conditions.

Future work will focus on the full integration of the turbine representation module developed in Sandoval et al. (2021) with the free-surface numerical framework presented in this thesis. This integration will enable the simulation of hydrokinetic turbine–wave interactions under fully unsteady and three-dimensional flow conditions, providing a valuable tool to assess local-scale dynamics such as rotor loading, wake evolution, and energy extraction efficiency.

In parallel, we plan to extend the solver to reproduce more complex environmental conditions, including varying wave amplitudes, multi-directional wave spectra, and combined wave–current interactions. These developments will allow for parametric studies on the influence of obstacle geometry, turbine array layout, and flow conditions on hydrodynamic loads and vortex generation.

On the numerical side, future improvements could target enhanced stability and efficiency of the solver. This includes exploring multi-stage pointwise integration methods such as Implicit Total-Variation-Diminishing Runge-Kutta schemes for improved solution of the Navier-Stokes equations, and further optimization of the geometric-based level-set reinitialisation for large-scale simulations.

Finally, the framework can be extended to include sediment transport and scour prediction, allowing for comprehensive assessments of marine energy devices in realistic coastal

and riverine environments. These advancements will support the design of reliable turbine components, optimised array configurations, and effective mitigation strategies for environmental impacts under fully unsteady free-surface conditions.

REFERENCES

- Adcock, T. A., Draper, S., Willden, R. H., & Vogel, C. R. (2021). The fluid mechanics of tidal stream energy conversion. *Annual Review of Fluid Mechanics*, 53, 287–310.
- Ahn, H. T., & Kallinderis, Y. (2006). Strongly coupled flow/structure interactions with a geometrically conservative ale scheme on general hybrid meshes. *Journal of Computational Physics*, 219(2), 671–696.
- Albin, E., Knikker, R., Xin, S., Paschereit, C. O., & d'Angelo, Y. (2016). Computational assessment of curvatures and principal directions of implicit surfaces from 3d scalar data. *International Conference on Mathematical Methods for Curves and Surfaces*, 1–22.
- Aris, R. (2012). *Vectors, tensors and the basic equations of fluid mechanics*. Courier Corporation.
- Armenio, V., & Piomelli, U. (2000). A lagrangian mixed subgrid-scale model in generalized coordinates. *Flow, Turbulence and Combustion*, 65, 51–81.
- Ausas, R. F., Dari, E. A., & Buscaglia, G. C. (2011). A geometric mass-preserving redistancing scheme for the level set function. *International journal for numerical methods in fluids*, 65(8), 989–1010.
- Bagherizadeh, E., Zhang, Z., Farhadzadeh, A., Angelidis, D., Ghazian Arabi, M., Moghimi, S., & Khosronejad, A. (2021). Numerical modelling of solitary wave and structure interactions using level-set and immersed boundary methods by adopting adequate inlet boundary conditions. *Journal of Hydraulic Research*, 59(4), 559–585.
- Batchelor, G. K. (1967). *An introduction to fluid dynamics*. Cambridge university press.
- Beam, R. M., & Warming, R. F. (1976). An implicit finite-difference algorithm for hyperbolic systems in conservation-law form. *Journal of computational physics*, 22(1), 87–110.
- Blazek, J. (2015). *Computational fluid dynamics: Principles and applications*. Butterworth-Heinemann.

- Burton, T., Jenkins, N., Sharpe, D., & Bossanyi, E. (2011). *Wind energy handbook*. John Wiley & Sons.
- Carrica, P., Wilson, R., & Stern, F. (2007). An unsteady single-phase level set method for viscous free surface flows. *International Journal for Numerical Methods in Fluids*, 53(2), 229–256.
- Casulli, V. (1999). A semi-implicit finite difference method for non-hydrostatic, free-surface flows. *International journal for numerical methods in fluids*, 30(4), 425–440.
- Chamorro, L., Hill, C., Morton, S., Ellis, C., Arndt, R., & Sotiropoulos, F. (2013). On the interaction between a turbulent open channel flow and an axial-flow turbine. *Journal of Fluid Mechanics*, 716, 658–670.
- Chamorro, L., Hill, C., Neary, V., Gunawan, B., Arndt, R., & Sotiropoulos, F. (2015). Effects of energetic coherent motions on the power and wake of an axial-flow turbine. *Physics of Fluids*, 27(5), 055104.
- Chang, K.-A., Hsu, T.-J., & Liu, P. L.-F. (2001). Vortex generation and evolution in water waves propagating over a submerged rectangular obstacle: Part i. solitary waves. *Coastal Engineering*, 44(1), 13–36.
- Chen, L., Zang, J., Hillis, A. J., Morgan, G. C., & Plummer, A. R. (2014). Numerical investigation of wave–structure interaction using openfoam. *Ocean Engineering*, 88, 91–109.
- Chorin, A. J. (1967). A numerical method for solving incompressible viscous flow problems. *Journal of Computational Physics*, 2(1), 12–26. [https://doi.org/https://doi.org/10.1016/0021-9991\(67\)90037-X](https://doi.org/https://doi.org/10.1016/0021-9991(67)90037-X)
- Chorin, A. J. (1968). Numerical solution of the navier-stokes equations. *Mathematics of computation*, 22(104), 745–762.
- Chorin, A. J., Marsden, J. E., & Marsden, J. E. (1990). *A mathematical introduction to fluid mechanics* (Vol. 3). Springer.
- Christou, A., Stagonas, D., Buldakov, E., & Stoesser, T. (2025). Focused waves on shear currents interacting with a vertical cylinder. *Coastal Engineering*, 104698.

- Christou, A., Stoesser, T., & Xie, Z. (2021). A large-eddy-simulation-based numerical wave tank for three-dimensional wave-structure interaction. *Computers & Fluids*, 231, 105179.
- Creech, A. C., Borthwick, A. G., & Ingram, D. (2017). Effects of support structures in an les actuator line model of a tidal turbine with contra-rotating rotors. *Energies*, 10(5), 726.
- Dianat, M., Skarysz, M., & Garmory, A. (2017). A coupled level set and volume of fluid method for automotive exterior water management applications. *International Journal of Multiphase Flow*, 91, 19–38.
- Drikakis, D., & Rider, W. (2005). *High-resolution methods for incompressible and low-speed flows*. Springer Science & Business Media.
- Escauriaza, C., & Sotiropoulos, F. (2011a). Initial stages of erosion and bed form development in a turbulent flow around a cylindrical pier. *Journal of Geophysical Research: Earth Surface*, 116(F3).
- Escauriaza, C., & Sotiropoulos, F. (2011b). Reynolds number effects on the coherent dynamics of the turbulent horseshoe vortex system. *Flow, turbulence and combustion*, 86(2), 231–262.
- Fedkiw, R. P., Aslam, T., Merriman, B., & Osher, S. (1999). A non-oscillatory eulerian approach to interfaces in multimaterial flows (the ghost fluid method). *Journal of computational physics*, 152(2), 457–492.
- Ferziger, J. H., Perić, M., & Street, R. L. (2019). *Computational methods for fluid dynamics*. Springer.
- Fleisch, D. A. (2011). *A student's guide to vectors and tensors*. Cambridge University Press.
- Flora, K., & Khosronejad, A. (2021). On the impact of bed-bathymetry resolution and bank vegetation on the flood flow field of the american river, california: Insights gained using data-driven large-eddy simulation. *Journal of Irrigation and Drainage Engineering*, 147(9), 04021036.

- Foley, J. D. (1996). *Computer graphics: Principles and practice* (Vol. 12110). Addison-Wesley Professional.
- Fringer, O., Gerritsen, M., & Street, R. (2006). An unstructured-grid, finite-volume, non-hydrostatic, parallel coastal ocean simulator. *Ocean modelling*, 14(3-4), 139–173.
- Frohn, A., & Roth, N. (2000). *Dynamics of droplets*. Springer Science & Business Media.
- Fujii, K. (2010). A multidimensional analogue of the simpson's formula of integral. *International Journal of Geometric Methods in Modern Physics*, 7(03), 471–483.
- Gajardo, D., Escauriaza, C., & Ingram, D. M. (2019). Capturing the development and interactions of wakes in tidal turbine arrays using a coupled bem-des model. *Ocean Engineering*, 181, 71–88.
- Garcia-Villalba, M., Colonius, T., Desjardins, O., Lucas, D., Mani, A., Marchisio, D., Matar, O. K., Picano, F., & Zaleski, S. (2025). Numerical methods for multiphase flows. *International Journal of Multiphase Flow*, 191, 105285. <https://doi.org/https://doi.org/10.1016/j.ijmultiphaseflow.2025.105285>
- Ge, L. (2004). *Numerical simulation of three-dimensional, complex, turbulent flows with unsteady coherent structures: From hydraulics to cardiovascular fluid mechanics*. Georgia Institute of Technology.
- Ge, L., & Sotiropoulos, F. (2007). A numerical method for solving the 3d unsteady incompressible navier–stokes equations in curvilinear domains with complex immersed boundaries. *Journal of computational physics*, 225(2), 1782–1809.
- Germano, M., Piomelli, U., Moin, P., & Cabot, W. H. (1991). A dynamic subgrid-scale eddy viscosity model. *Physics of Fluids A: Fluid Dynamics*, 3(7), 1760–1765.
- Gibou, F., Fedkiw, R., & Osher, S. (2018). A review of level-set methods and some recent applications. *Journal of Computational Physics*, 353, 82–109.
- Gotelli, C., Musa, M., Guala, M., & Escauriaza, C. (2019). Experimental and numerical investigation of wake interactions of marine hydrokinetic turbines. *Energies*, 12(16), 3188.

- Gresho, P. M., & Sani, R. L. (1987). On pressure boundary conditions for the incompressible navier-stokes equations. *International Journal for Numerical Methods in Fluids*, 7(10), 1111–1145.
- Gropp, W., Lusk, E., & Skjellum, A. (1999). *Using mpi: Portable parallel programming with the message-passing interface* (Vol. 1). MIT press.
- Ha, T., Shim, J., Lin, P., & Cho, Y.-S. (2014). Three-dimensional numerical simulation of solitary wave run-up using the ib method. *Coastal Engineering*, 84, 38–55.
- Harlow, F. H., Welch, J. E. et al. (1965). Numerical calculation of time-dependent viscous incompressible flow of fluid with free surface. *Physics of fluids*, 8(12), 2182.
- Hergibo, P., Phillips, T. N., & Xie, Z. (2025). An adaptive dual grid moment-of-fluid method for multiphase flows. *Journal of Computational Physics*, 113908.
- Higuera, P., Liu, P.-F., Lin, C., Wong, W.-Y., & Kao, M.-J. (2018). Laboratory-scale swash flows generated by a non-breaking solitary wave on a steep slope. *Journal of Fluid Mechanics*, 847, 186–227.
- Hirt, C. W., & Nichols, B. D. (1981). Volume of fluid (vof) method for the dynamics of free boundaries. *Journal of computational physics*, 39(1), 201–225.
- Hirt, C. W., Amsden, A. A., & Cook, J. (1974). An arbitrary lagrangian-eulerian computing method for all flow speeds. *Journal of computational physics*, 14(3), 227–253.
- Howard, K. B., Singh, A., Sotiropoulos, F., & Guala, M. (2015). On the statistics of wind turbine wake meandering: An experimental investigation. *Physics of Fluids*, 27(7), 075103.
- Hu, Z. Z., Greaves, D., & Raby, A. (2016). Numerical wave tank study of extreme waves and wave-structure interaction using openfoam®. *Ocean Engineering*, 126, 329–342.
- Huang, L., Li, Y., Benites-Munoz, D., Windt, C. W., Feichtner, A., Tavakoli, S., Davidson, J., Paredes, R., Quintuna, T., Ransley, E., et al. (2022). A review on the modelling of wave-structure interactions based on openfoam. *OpenFOAM® Journal*, 2, 116–142.

- Hunt, J. C., Wray, A. A., & Moin, P. (1988). Eddies, streams, and convergence zones in turbulent flows. *Studying turbulence using numerical simulation databases, 2. Proceedings of the 1988 summer program*.
- Ingram, D. M., Causon, D. M., & Mingham, C. G. (2003). Developments in cartesian cut cell methods. *Mathematics and Computers in Simulation*, 61(3-6), 561–572.
- Ingram, D. M., Gao, F., Causon, D. M., Mingham, C. G., & Troch, P. (2009). Numerical investigations of wave overtopping at coastal structures. *Coastal Engineering*, 56(2), 190–202.
- Jiang, G.-S., & Shu, C.-W. (1996). Efficient implementation of weighted eno schemes. *Journal of computational physics*, 126(1), 202–228.
- Kajishima, T., & Taira, K. (2016). *Computational fluid dynamics: Incompressible turbulent flows*. Springer.
- Kang, S., & Sotiropoulos, F. (2012). Numerical modeling of 3d turbulent free surface flow in natural waterways. *Advances in water resources*, 40, 23–36.
- Kang, S., Yang, X., & Sotiropoulos, F. (2014). On the onset of wake meandering for an axial flow turbine in a turbulent open channel flow. *Journal of Fluid Mechanics*, 744, 376–403.
- Kara, S., Kara, M. C., Stoesser, T., & Sturm, T. W. (2015). Free-surface versus rigid-lid les computations for bridge-abutment flow. *Journal of Hydraulic Engineering*, 141(9), 04015019.
- Katsidoniotaki, E., Shahroozi, Z., Eskilsson, C., Palm, J., Engström, J., & Göteman, M. (2023). Validation of a cfd model for wave energy system dynamics in extreme waves. *Ocean Engineering*, 268, 113320.
- Khan, M., Bhuyan, G., Iqbal, M., & Quaicoe, J. (2009). Hydrokinetic energy conversion systems and assessment of horizontal and vertical axis turbines for river and tidal applications: A technology status review. *Applied energy*, 86(10), 1823–1835.
- Khosronejad, A., Ghazian Arabi, M., Angelidis, D., Bagherizadeh, E., Flora, K., & Farhadzadeh, A. (2019). Comparative hydrodynamic study of rigid-lid and level-set methods for les of open-channel flow. *Journal of Hydraulic Engineering*, 145(1), 04018077.

- Khosronejad, A., Arabi, M. G., Angelidis, D., Bagherizadeh, E., Flora, K., & Farhadzadeh, A. (2020a). A comparative study of rigid-lid and level-set methods for les of open-channel flows: Morphodynamics. *Environmental Fluid Mechanics*, 20(1), 145–164.
- Khosronejad, A., Flora, K., Zhang, Z., & Kang, S. (2020b). Large-eddy simulation of flash flood propagation and sediment transport in a dry-bed desert stream. *International Journal of Sediment Research*, 35(6), 576–586.
- Kim, J., & Moin, P. (1985). Application of a fractional-step method to incompressible navier-stokes equations. *Journal of computational physics*, 59(2), 308–323.
- Kim, K.-H., Kang, J.-H., Pan, X., & Choi, J.-I. (2021). Pascal_tdma: A library of parallel and scalable solvers for massive tridiagonal systems. *Computer Physics Communications*, 260, 107722.
- Kwak, D., & Kiris, C. C. (2010). *Computation of viscous incompressible flows*. Springer Science & Business Media.
- Lahey, T. M., & Ellis, T. (1994). *Fortran 90 programming*. Addison-Wesley Longman Publishing Co., Inc.
- Le Méhauté, B. (2013). *An introduction to hydrodynamics and water waves*. Springer Science & Business Media.
- Lee, J., Teubner, M., Nixon, J., & Gill, P. (2006). Applications of the artificial compressibility method for turbulent open channel flows. *International journal for numerical methods in fluids*, 51(6), 617–633.
- Leonard, B. P. (1979). A stable and accurate convective modelling procedure based on quadratic upstream interpolation. *Computer methods in applied mechanics and engineering*, 19(1), 59–98.
- LeVeque, R. J. (2002). *Finite volume methods for hyperbolic problems* (Vol. 31). Cambridge university press.
- Li, Z., Ghia, K., Li, Y., Fan, Z., & Shen, L. (2021). Unsteady reynolds-averaged navier–stokes investigation of free surface wave impact on tidal turbine wake. *Proceedings of the Royal Society A*, 477(2246), 20200703.

- Lilly, D. (1992). A proposed modification of the germano sugrid-scale closure method. *Phys Fluids A*, 4, 633–635.
- Lin, C.-L., Lee, H., Lee, T., & Weber, L. J. (2005). A level set characteristic galerkin finite element method for free surface flows. *International Journal for Numerical Methods in Fluids*, 49(5), 521–547.
- Lin, M.-Y., & Huang, L.-H. (2010). Vortex shedding from a submerged rectangular obstacle attacked by a solitary wave. *Journal of Fluid Mechanics*, 651, 503–518.
- Lin, P. (2008). *Numerical modeling of water waves*. CRC Press.
- Link, O., González, C., Maldonado, M., & Escauriaza, C. (2012). Coherent structure dynamics and sediment particle motion around a cylindrical pier in developing scour holes. *Acta Geophysica*, 60(6), 1689–1719.
- Liu, X.-D., Osher, S., & Chan, T. (1994). Weighted essentially non-oscillatory schemes. *Journal of computational physics*, 115(1), 200–212.
- Madsen, P. A., Fuhrman, D. R., & Schäffer, H. A. (2008). On the solitary wave paradigm for tsunamis. *Journal of Geophysical Research: Oceans*, 113(C12).
- Madsen, P. A., & Schäffer, H. (2006). A discussion of artificial compressibility. *Coastal engineering*, 53(1), 93–98.
- Marchandise, E., & Remacle, J.-F. (2006). A stabilized finite element method using a discontinuous level set approach for solving two phase incompressible flows. *Journal of computational physics*, 219(2), 780–800.
- McSherry, R. J., Chua, K. V., & Stoesser, T. (2017). Large eddy simulation of free-surface flows. *Journal of Hydrodynamics, Ser. B*, 29(1), 1–12.
- Merkle, C. (1987). Time-accurate unsteady incompressible flow algorithms based on artificial compressibility. *8th Computational Fluid Dynamics Conference*, 1137.
- Mut, F., Buscaglia, G. C., & Dari, E. A. (2006). New mass-conserving algorithm for level set redistancing on unstructured meshes. *Journal of Applied Mechanics*, 73(6), 1011–1016. <https://doi.org/10.1115/1.2198244>

- Nangia, N., Griffith, B. E., Patankar, N. A., & Bhalla, A. P. S. (2019). A robust incompressible navier-stokes solver for high density ratio multiphase flows. *Journal of Computational Physics*, 390, 548–594.
- Nishikawa, H. (2014). First-, second-, and third-order finite-volume schemes for diffusion. *Journal of Computational Physics*, 256, 791–805.
- Nishikawa, H. (2021). The quick scheme is a third-order finite-volume scheme with point-valued numerical solutions. *International Journal for Numerical Methods in Fluids*, 93(7), 2311–2338.
- Önder, A., & Liu, P. L.-F. (2020). Stability of the solitary wave boundary layer subject to finite-amplitude disturbances. *Journal of Fluid Mechanics*, 896, A20.
- Orszaghova, J., Taylor, P. H., Borthwick, A. G., & Raby, A. C. (2014). Importance of second-order wave generation for focused wave group run-up and overtopping. *Coastal Engineering*, 94, 63–79.
- Osher, S., & Fedkiw, R. P. (2003). *Level set methods and dynamic implicit surfaces* (Vol. 153). Springer.
- Osher, S., & Sethian, J. A. (1988). Fronts propagating with curvature-dependent speed: Algorithms based on hamilton-jacobi formulations. *Journal of computational physics*, 79(1), 12–49.
- Ouro, P., Harrold, M., Stoesser, T., & Bromley, P. (2017). Hydrodynamic loadings on a horizontal axis tidal turbine prototype. *Journal of Fluids and Structures*, 71, 78–95.
- Ouro, P., Mullings, H., Christou, A., Draycott, S., & Stallard, T. (2024). Wake characteristics behind a tidal turbine with surface waves in turbulent flow analyzed with large-eddy simulation. *Physical Review Fluids*, 9(3), 034608.
- Ouro, P., & Stoesser, T. (2019). Impact of environmental turbulence on the performance and loadings of a tidal stream turbine. *Flow, Turbulence and Combustion*, 102(3), 613–639.
- Ozdemir, C. E., Hsu, T.-J., & Balachandar, S. (2014). Direct numerical simulations of transition and turbulence in smooth-walled stokes boundary layer. *Physics of Fluids*, 26(4).

- Paik, J., Sotiropoulos, F., & Sale, M. J. (2005). Numerical simulation of swirling flow in complex hydroturbine draft tube using unsteady statistical turbulence models. *Journal of hydraulic engineering*, 131(6), 441–456.
- Peyret, R. (1976). Unsteady evolution of a horizontal jet in a stratified fluid. *Journal of Fluid Mechanics*, 78(1), 49–63.
- Popinet, S. (2018). Numerical models of surface tension. *Annual Review of Fluid Mechanics*, 50(1), 49–75.
- Press, W. H. (1996). *Numerical recipes in fortran 90: Volume 2, volume 2 of fortran numerical recipes: The art of parallel scientific computing* (Vol. 2). Cambridge university press.
- Ramaswamy, B. (1990). Numerical simulation of unsteady viscous free surface flow. *Journal of Computational Physics*, 90(2), 396–430.
- Robertson, I., Sherwin, S., & Graham, J. (2004). Comparison of wall boundary conditions for numerical viscous free surface flow simulation. *Journal of fluids and structures*, 19(4), 525–542.
- Rodi, W., Constantinescu, G., & Stoesser, T. (2013). Large-eddy simulation in hydraulics.
- Russo, G., & Smereka, P. (2000). A remark on computing distance functions. *Journal of computational physics*, 163(1), 51–67.
- Saffman, P. G. (1992). Vortex dynamics. *Theoretical approaches to turbulence* (pp. 263–277). Springer.
- Sagaut, P. (2005). *Large eddy simulation for incompressible flows: An introduction*. Springer Science & Business Media.
- Sandoval, J., & Eaves, T. S. (2025). Edge states and the periodic self-sustaining process in the Stokes boundary layer. *Journal of Fluid Mechanics*, 1022, A10.
- Sandoval, J., Soto-Rivas, K., Gotelli, C., & Escauriaza, C. (2021). Modeling the wake dynamics of a marine hydrokinetic turbine using different actuator representations. *Ocean Engineering*, 222, 108584.
- Saruwatari, A., Watanabe, Y., & Ingram, D. M. (2009). Scarifying and fingering surfaces of plunging jets. *Coastal Engineering*, 56(11-12), 1109–1122.

- Scardovelli, R., & Zaleski, S. (1999). Direct numerical simulation of free-surface and interfacial flow. *Annual review of fluid mechanics*, 31(1), 567–603.
- Shin, S., & Juric, D. (2002). Modeling three-dimensional multiphase flow using a level contour reconstruction method for front tracking without connectivity. *Journal of Computational Physics*, 180(2), 427–470.
- Shu, C.-W., & Osher, S. (1989). Efficient implementation of essentially non-oscillatory shock-capturing schemes, ii. *Journal of Computational Physics*, 83(1), 32–78. [https://doi.org/https://doi.org/10.1016/0021-9991\(89\)90222-2](https://doi.org/https://doi.org/10.1016/0021-9991(89)90222-2)
- Smagorinsky, J. (1963). General circulation experiments with the primitive equations: I. the basic experiment. *Monthly weather review*, 91(3), 99–164.
- Sørensen, J. N., & Shen, W. Z. (2002). Numerical modeling of wind turbine wakes. *Journal of Fluids Engineering*, 124(2), 393–399.
- Sotiropoulos, F., & Constantinescu, G. (1997). Pressure-based residual smoothing operators for multistage pseudocompressibility algorithms. *Journal of computational Physics*, 133(1), 129–145.
- Sotiropoulos, F., & Abdallah, S. (1992). A primitive variable method for the solution of three-dimensional incompressible viscous flows. *Journal of Computational Physics*, 103(2), 336–349.
- Soto-Rivas, K., Richter, D., & Escauriaza, C. (2019). A formulation of the thrust coefficient for representing finite-sized farms of tidal energy converters. *Energies*, 12(20), 3861.
- Stallard, T., Collings, R., Feng, T., & Whelan, J. (2013). Interactions between tidal turbine wakes: Experimental study of a group of three-bladed rotors. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 371(1985), 20120159.
- Stewart, J. (2012). *Calculus: Early transcendentals*. Cengage Learning.
- Sussman, M., & Fatemi, E. (1999). An efficient, interface-preserving level set redistancing algorithm and its application to interfacial incompressible fluid flow. *SIAM Journal on scientific computing*, 20(4), 1165–1191.

- Sussman, M., Fatemi, E., Smereka, P., & Osher, S. (1998). An improved level set method for incompressible two-phase flows. *Computers & Fluids*, 27(5-6), 663–680.
- Sussman, M., Smereka, P., & Osher, S. (1994). A level set approach for computing solutions to incompressible two-phase flow. *Journal of Computational physics*, 114(1), 146–159.
- Tadjbakhsh, I., & Keller, J. B. (1960). Standing surface waves of finite amplitude. *Journal of Fluid Mechanics*, 8(3), 442–451.
- Tang, H., & Sotiropoulos, F. (2007). Fractional step artificial compressibility schemes for the unsteady incompressible navier–stokes equations. *Computers & fluids*, 36(5), 974–986.
- Temam, R. (1969). Sur l’approximation de la solution des équations de navier-stokes par la méthode des pas fractionnaires (ii). *Archive for rational mechanics and analysis*, 33, 377–385.
- Tryggvason, G., Bunner, B., Esmaeeli, A., Juric, D., Al-Rawahi, N., Tauber, W., Han, J., Nas, S., & Jan, Y.-J. (2001). A front-tracking method for the computations of multiphase flow. *Journal of computational physics*, 169(2), 708–759.
- Tryggvason, G., Scardovelli, R., & Zaleski, S. (2011). *Direct numerical simulations of gas–liquid multiphase flows*. Cambridge university press.
- Unverdi, S. O., & Tryggvason, G. (1992). A front-tracking method for viscous, incompressible, multi-fluid flows. *Journal of computational physics*, 100(1), 25–37.
- Van der Pijl, S., Segal, A., Vuik, C., & Wesseling, P. (2005). A mass-conserving level-set method for modelling of multi-phase flows. *International journal for numerical methods in fluids*, 47(4), 339–361.
- Venkatakrishnan, V. (1993). On the accuracy of limiters and convergence to steady state solutions. *31st Aerospace sciences meeting*, 880.
- Venkatakrishnan, V. (1995). Convergence to steady state solutions of the euler equations on unstructured grids with limiters. *Journal of computational physics*, 118(1), 120–130.

- Wang, Y., & Liu, P. L.-F. (2022). On finite amplitude solitary waves—a review and new experimental data. *Physics of Fluids*, 34(10).
- Watanabe, Y. (2022). *Dynamics of water surface flows and waves*. CRC Press.
- Watanabe, Y., Saruwatari, A., & Ingram, D. M. (2008). Free-surface flows under impacting droplets. *Journal of Computational Physics*, 227(4), 2344–2365.
- White, F. M. (1994). Fluid mechanics, mcgraw-hill. New York.
- White, J. A., Nishikawa, H., & Baurle, R. A. (2019). Weighted least-squares cell-average gradient construction methods for the vulcan-cfd second-order accurate unstructured grid cell-centered finite-volume solver. *AIAA Scitech 2019 Forum*, 0127.
- Winckler, P., & Liu, P. L.-F. (2015). Long waves in a straight channel with non-uniform cross-section. *Journal of Fluid Mechanics*, 770, 156–188.
- Wu, B., & Fu, L. (2025). An implicit moving-least-squares immersed boundary method for high-fidelity fluid-structure interaction simulations. *Journal of Computational Physics*, 530, 113913.
- Wu, G., Eatock Taylor, R., & Greaves, D. (2001). The effect of viscosity on the transient free-surface waves in a two-dimensional tank. *Journal of Engineering Mathematics*, 40, 77–90.
- Yuan, H.-Z., Shu, C., Wang, Y., & Shu, S. (2018). A simple mass-conserved level set method for simulation of multiphase flows. *Physics of Fluids*, 30(4).
- Yue, W., Lin, C.-L., & Patel, V. C. (2005). Large eddy simulation of turbulent open-channel flow with free surface simulated by level set method. *Physics of fluids*, 17(2), 025108.
- Zalesak, S. T. (1979). Fully multidimensional flux-corrected transport algorithms for fluids. *Journal of computational physics*, 31(3), 335–362.

APPENDIX A. TRANSFORMATION OF NAVIER-STOKES EQUATIONS INTO GENERALISED CURVILINEAR COORDINATES

A.1. Non-dimensional Navier-Stokes equations for an incompressible fluid with a source term

A.1.1. Flow scales

Let's define the length and velocity scale for the flow, \mathcal{L} y \mathcal{U} , in such a way that velocities and spatial coordinates are defined as

$$\tilde{u}_i = u_i \mathcal{U} \quad (\text{A.1})$$

$$\tilde{x}_i = x_i \mathcal{L} \quad (\text{A.2})$$

Where $\widetilde{(\)}$ denotes that the variable is dimensional. By using these scales, the time scale of the flow is defined as

$$\tilde{t} = t \mathcal{T} = t \frac{\mathcal{L}}{\mathcal{U}} \quad (\text{A.3})$$

Where u_i , x_i y t are non-dimensional variables. On the other hand, the pressure is scaled as follows.

$$\tilde{P} = P \rho \mathcal{U}^2 \quad (\text{A.4})$$

A.1.2. Mass conservation

The mass conservation equation for an incompressible fluid, in dimensional form, expressed in Einstein notation, reads:

$$\frac{\partial \tilde{u}_i}{\partial \tilde{x}_i} = 0 \quad (\text{A.5})$$

Replacing the flow scales previously determined into the equation

$$\frac{\partial(u_i\mathcal{U})}{\partial(x_i\mathcal{L})} = 0 \quad (\text{A.6})$$

$$\frac{\mathcal{U}}{\mathcal{L}} \frac{\partial u_i}{\partial x_i} = 0 \quad (\text{A.7})$$

This leads to the non-dimensional mass conservation equation.

$$\frac{\partial u_i}{\partial x_i} = 0 \quad (\text{A.8})$$

A.1.3. Momentum conservation

The momentum conservation equation in Cartesian coordinates, in dimensional form, expressed in Einstein notation, reads as follows.

$$\rho \left(\frac{\partial \tilde{u}_i}{\partial \tilde{t}} + \frac{\partial(\tilde{u}_i \tilde{u}_j)}{\partial \tilde{x}_j} \right) = - \frac{\partial \tilde{P}}{\partial \tilde{x}_i} + \frac{\partial}{\partial \tilde{x}_j} (2 \mu \tilde{S}_{ij}) + \rho \tilde{f}_i \quad (\text{A.9})$$

Considering a constant specific mass within the fluid that is being solved and incorporating a modified expression for the pressure term as $\tilde{p} = \tilde{P}/\rho$, the equation can be expressed as follows:

$$\frac{\partial \tilde{u}_i}{\partial \tilde{t}} + \frac{\partial(\tilde{u}_i \tilde{u}_j)}{\partial \tilde{x}_j} = - \frac{\partial \tilde{p}}{\partial \tilde{x}_i} + \frac{\partial}{\partial \tilde{x}_j} (2 \nu \tilde{S}_{ij}) + \tilde{f}_i \quad (\text{A.10})$$

Expanding each using the flow scales previously defined:

- $\frac{\partial \tilde{u}_i}{\partial \tilde{t}} = \frac{\partial(u_i \mathcal{U})}{\partial(t \mathcal{T})} = \frac{\mathcal{U}}{\mathcal{T}} \frac{\partial u_i}{\partial t} = \frac{\mathcal{U}}{\mathcal{L}/\mathcal{U}} \frac{\partial u_i}{\partial t} = \frac{\mathcal{U}^2}{\mathcal{L}} \frac{\partial u_i}{\partial t}$

- $\frac{\partial(\tilde{u}_i \tilde{u}_j)}{\partial \tilde{x}_j} = \frac{\partial((u_i \mathcal{U}) \cdot (u_j \mathcal{U}))}{\partial(x_j \mathcal{L})} = \frac{\mathcal{U}^2}{\mathcal{L}} \frac{\partial(u_i u_j)}{\partial x_j}$

- $\frac{\partial \tilde{p}}{\partial \tilde{x}_i} = \frac{\partial(p \cdot \mathcal{U}^2)}{\partial(x_i \mathcal{L})} = \frac{\mathcal{U}^2}{\mathcal{L}} \frac{\partial p}{\partial x_i}$
- $$\begin{aligned} \frac{\partial}{\partial \tilde{x}_j} \left(2 \nu \widetilde{S}_{ij} \right) &= \nu \frac{\partial}{\partial \tilde{x}_j} \left(\frac{\partial \tilde{u}_i}{\partial \tilde{x}_j} + \frac{\partial \tilde{u}_j}{\partial \tilde{x}_i} \right) = \nu \frac{\partial}{\partial(x_j \cdot \mathcal{L})} \left(\frac{\partial(u_i \cdot \mathcal{U})}{\partial(x_j \cdot \mathcal{L})} + \frac{\partial(u_j \cdot \mathcal{U})}{\partial(x_i \cdot \mathcal{L})} \right) \\ &= \nu \frac{\mathcal{U}}{\mathcal{L}^2} \frac{\partial}{\partial x_j} \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right) \end{aligned}$$
- $\tilde{f}_i = g \hat{e}_{g_i}$

Where $\hat{e}_{g_i} = g_i/g$ is a unitary vector (non-dimensional) that indicates the gravity direction. The expansion of the viscous term also assumes the viscosity is constant so that it can be taken out from the derivative. Introducing this expansion into the momentum equation

$$\frac{\mathcal{U}^2}{\mathcal{L}} \frac{\partial u_i}{\partial t} + \frac{\mathcal{U}^2}{\mathcal{L}} \frac{\partial(u_i u_j)}{\partial x_j} = - \frac{\mathcal{U}^2}{\mathcal{L}} \frac{\partial p}{\partial x_i} + \nu \frac{\mathcal{U}}{\mathcal{L}^2} \frac{\partial}{\partial x_j} \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right) + g \hat{e}_{g_i} \quad (\text{A.11})$$

Multiplying both sides by $\mathcal{L}/\mathcal{U}^2$

$$\frac{\partial u_i}{\partial t} + \frac{\partial(u_i u_j)}{\partial x_j} = - \frac{\partial p}{\partial x_i} + \frac{\nu}{\mathcal{L} \mathcal{U}} \frac{\partial}{\partial x_j} \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right) + \frac{g \mathcal{L}}{\mathcal{U}^2} \hat{e}_{g_i}$$

Incorporating the Reynolds and Froude numbers into the equation

$$Re = \frac{\mathcal{L} \mathcal{U}}{\nu} \quad (\text{A.12})$$

$$Fr^2 = \frac{\mathcal{U}^2}{g \mathcal{L}} \quad (\text{A.13})$$

We finally get the non-dimensional momentum conservation equation.

$$\frac{\partial u_i}{\partial t} + \frac{\partial(u_i u_j)}{\partial x_j} = - \frac{\partial p}{\partial x_i} + \frac{1}{Re} \frac{\partial}{\partial x_j} \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right) + \frac{1}{Fr^2} \hat{e}_{g_i} \quad (\text{A.14})$$

A.2. Governing Equations in Strong Conservation Form

If we expand the indexes of the mass and momentum equations, we obtain a set of four partial differential equations:

$$\begin{aligned}
\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z} &= 0 \\
\frac{\partial u}{\partial t} + \frac{\partial(u^2)}{\partial x} + \frac{\partial(uv)}{\partial y} + \frac{\partial(uw)}{\partial z} &= -\frac{\partial p}{\partial x} + \frac{1}{Re} \left(\frac{\partial}{\partial x} \left(\frac{\partial u}{\partial x} + \frac{\partial u}{\partial x} \right) + \frac{\partial}{\partial y} \left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} \right) \right. \\
&\quad \left. + \frac{\partial}{\partial z} \left(\frac{\partial u}{\partial z} + \frac{\partial w}{\partial x} \right) \right) + \frac{e_{g_1}}{Fr^2} \\
\frac{\partial v}{\partial t} + \frac{\partial(uv)}{\partial x} + \frac{\partial(v^2)}{\partial y} + \frac{\partial(vw)}{\partial z} &= -\frac{\partial p}{\partial y} + \frac{1}{Re} \left(\frac{\partial}{\partial x} \left(\frac{\partial v}{\partial x} + \frac{\partial u}{\partial y} \right) + \frac{\partial}{\partial y} \left(\frac{\partial v}{\partial y} + \frac{\partial v}{\partial y} \right) \right. \\
&\quad \left. + \frac{\partial}{\partial z} \left(\frac{\partial v}{\partial z} + \frac{\partial w}{\partial y} \right) \right) + \frac{e_{g_2}}{Fr^2} \\
\frac{\partial w}{\partial t} + \frac{\partial(uw)}{\partial x} + \frac{\partial(vw)}{\partial y} + \frac{\partial(w^2)}{\partial z} &= -\frac{\partial p}{\partial z} + \frac{1}{Re} \left(\frac{\partial}{\partial x} \left(\frac{\partial w}{\partial x} + \frac{\partial u}{\partial z} \right) + \frac{\partial}{\partial y} \left(\frac{\partial w}{\partial y} + \frac{\partial v}{\partial z} \right) \right. \\
&\quad \left. + \frac{\partial}{\partial z} \left(\frac{\partial w}{\partial z} + \frac{\partial w}{\partial z} \right) \right) + \frac{e_{g_3}}{Fr^2}
\end{aligned}$$

Let's express this system in vectorial form

$$\begin{aligned}
\mathbf{I} \frac{\partial}{\partial t} \begin{bmatrix} p \\ u \\ v \\ w \end{bmatrix} + \frac{\partial}{\partial x} \begin{bmatrix} p + u^2 - \frac{1}{Re} \left(\frac{\partial u}{\partial x} + \frac{\partial u}{\partial x} \right) \\ uv - \frac{1}{Re} \left(\frac{\partial v}{\partial x} + \frac{\partial u}{\partial y} \right) \\ uw - \frac{1}{Re} \left(\frac{\partial w}{\partial x} + \frac{\partial u}{\partial z} \right) \end{bmatrix} + \frac{\partial}{\partial y} \begin{bmatrix} v \\ uv - \frac{1}{Re} \left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} \right) \\ p + v^2 - \frac{1}{Re} \left(\frac{\partial v}{\partial y} + \frac{\partial v}{\partial y} \right) \\ vw - \frac{1}{Re} \left(\frac{\partial w}{\partial y} + \frac{\partial v}{\partial z} \right) \end{bmatrix} \\
+ \frac{\partial}{\partial z} \begin{bmatrix} w \\ uw - \frac{1}{Re} \left(\frac{\partial u}{\partial z} + \frac{\partial w}{\partial x} \right) \\ vw - \frac{1}{Re} \left(\frac{\partial v}{\partial z} + \frac{\partial w}{\partial y} \right) \\ p + w^2 - \frac{1}{Re} \left(\frac{\partial w}{\partial z} + \frac{\partial w}{\partial z} \right) \end{bmatrix} = \begin{bmatrix} 0 \\ -e_{g_1}/Fr^2 \\ -e_{g_2}/Fr^2 \\ -e_{g_3}/Fr^2 \end{bmatrix}
\end{aligned}$$

Where

$$\Gamma = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (\text{A.15})$$

The vectorial equation can be expressed as follows:

$$\Gamma \frac{\partial \mathbf{Q}}{\partial t} + \frac{\partial \mathbf{F}}{\partial x} + \frac{\partial \mathbf{G}}{\partial y} + \frac{\partial \mathbf{H}}{\partial z} + \mathbf{S} = 0 \quad (\text{A.16})$$

Where F , G and H are the flux vectors in x , y , and z directions correspondingly, and S is the source term due to gravity forces. This equation can be expressed in strong conservation form (Drikakis & Rider, 2005; Kajishima & Taira, 2016) using the divergence form to group the spatial derivatives. The resulting equation, expressed in Einstein notation, reads:

$$\Gamma \frac{\partial \mathbf{Q}}{\partial t} + \frac{\partial \mathcal{F}_k}{\partial x_k} + \mathbf{S} = 0 \quad (\text{A.17})$$

Where \mathcal{F}_k is the flux tensor of the system defined as:

$$\mathcal{F}_k = \begin{bmatrix} u_k \\ p \delta_{1k} + u_1 u_k - \frac{1}{Re} \left(\frac{\partial u_1}{\partial x_k} + \frac{\partial u_k}{\partial x_1} \right) \\ p \delta_{2k} + u_2 u_k - \frac{1}{Re} \left(\frac{\partial u_2}{\partial x_k} + \frac{\partial u_k}{\partial x_2} \right) \\ p \delta_{3k} + u_3 u_k - \frac{1}{Re} \left(\frac{\partial u_3}{\partial x_k} + \frac{\partial u_k}{\partial x_3} \right) \end{bmatrix} \quad (\text{A.18})$$

Where x , y , and z are x_1 , x_2 , and x_3 , respectively. On the other hand, u , v and w are u_1 , u_2 and u_3 respectively. Finally, δ_{ij} is the Kronecker Delta tensor, which is equal to 1 when $i = j$, and 0 otherwise (Aris, 2012).

A.3. Transformation of the Governing Equations to Generalised Curvilinear Coordinates

The previous system of equations in Cartesian coordinates (x_k) is transformed into generalised non-orthogonal curvilinear coordinates (ξ_j). The transformation is defined as:

$$\xi_j = \xi_j(x_1, x_2, x_3) \quad (j = 1, 2, 3) \quad (\text{A.19})$$

And by using the chain rule, the Cartesian derivatives are transformed as follows:

$$\frac{\partial}{\partial x_k} = \frac{\partial \xi^j}{\partial x_k} \frac{\partial}{\partial \xi_j} = \frac{\partial \xi^1}{\partial x_k} \frac{\partial}{\partial \xi_1} + \frac{\partial \xi^2}{\partial x_k} \frac{\partial}{\partial \xi_2} + \frac{\partial \xi^3}{\partial x_k} \frac{\partial}{\partial \xi_3} \quad (\text{A.20})$$

Then, the system equation

$$\boldsymbol{\Gamma} \frac{\partial \boldsymbol{Q}}{\partial t} + \frac{\partial \mathcal{F}_k}{\partial x_k} + \boldsymbol{S} = 0$$

can be written as

$$\boldsymbol{\Gamma} \frac{\partial \boldsymbol{Q}}{\partial t} + \frac{\partial \xi^j}{\partial x_k} \frac{\partial \mathcal{F}_k}{\partial \xi^j} + S = 0 \quad (\text{A.21})$$

Let's introduce the Jacobian of the transformation, which is defined as

$$J = \det \left(\frac{\partial \xi^j}{\partial x_k} \right) \quad (\text{A.22})$$

If we manipulate the flux divergence of the equation A.21 by introducing the Jacobian

$$\boldsymbol{\Gamma} \frac{\partial \boldsymbol{Q}}{\partial t} + J \left(\frac{1}{J} \frac{\partial \xi^j}{\partial x_k} \frac{\partial \mathcal{F}_k}{\partial \xi^j} \right) + S = 0$$

The following identity can be applied to recall the previous expression (Kajishima & Taira, 2016):

$$\frac{1}{J} \frac{\partial \xi^j}{\partial x_k} \frac{\partial \mathcal{F}_k}{\partial \xi^j} = \frac{\partial}{\partial \xi^j} \left(\frac{\mathcal{F}_k}{J} \right) \frac{\partial \xi^j}{\partial x_k} \quad (\text{A.23})$$

Replacing this expression into A.23, we obtain the

$$\boldsymbol{\Gamma} \frac{\partial \boldsymbol{Q}}{\partial t} + J \frac{\partial}{\partial \xi^j} \left(\frac{\mathcal{F}_k}{J} \frac{\partial \xi^j}{\partial x_k} \right) + S = 0 \quad (\text{A.24})$$

This is a partial transformation of the equations, where the flow variables (p, u, v, w) are kept in Cartesian coordinates, and the independent variables (x_i) are transformed. With partial transformation, the Christoffel symbols, which are found in the full transformation, are avoided (Ge & Sotiropoulos, 2007). We can decompose the flux tensor into its linear, non-linear, and viscous components.

$$\mathcal{F}_k = \mathcal{F}_{L_k} + \mathcal{F}_{N_k} + \mathcal{F}_{v_k} \quad (\text{A.25})$$

Where

$$\mathcal{F}_{L_k} = \begin{bmatrix} u_k \\ p \delta_{1k} \\ p \delta_{2k} \\ p \delta_{3k} \end{bmatrix}; \quad \mathcal{F}_{N_k} = \begin{bmatrix} 0 \\ u_1 u_k \\ u_2 u_k \\ u_3 u_k \end{bmatrix}; \quad \mathcal{F}_{v_k} = \begin{bmatrix} 0 \\ -\frac{1}{Re} \left(\frac{\partial u_1}{\partial x_k} + \frac{\partial u_k}{\partial x_1} \right) \\ -\frac{1}{Re} \left(\frac{\partial u_2}{\partial x_k} + \frac{\partial u_k}{\partial x_2} \right) \\ -\frac{1}{Re} \left(\frac{\partial u_3}{\partial x_k} + \frac{\partial u_k}{\partial x_3} \right) \end{bmatrix} \quad (\text{A.26})$$

We operate the flux components with the metrics of the transformation ($\partial \xi^j / \partial x_k$) to obtain an expression in curvilinear coordinates for $\frac{\mathcal{F}_k}{J} \frac{\partial \xi^j}{\partial x_k} = \frac{\mathcal{F}_{L_k}}{J} \frac{\partial \xi^j}{\partial x_k} + \frac{\mathcal{F}_{N_k}}{J} \frac{\partial \xi^j}{\partial x_k} + \frac{\mathcal{F}_{v_k}}{J} \frac{\partial \xi^j}{\partial x_k}$.

A.3.1. Linear Fluxes

From the linear flux tensor, we can compute

$$\frac{\mathcal{F}_{L_k}}{J} \frac{\partial \xi^j}{\partial x_k} = \begin{bmatrix} u_k \frac{\partial \xi^j}{\partial x_k} \cdot \frac{1}{J} \\ p \delta_{1k} \frac{\partial \xi^j}{\partial x_k} \cdot \frac{1}{J} \\ p \delta_{2k} \frac{\partial \xi^j}{\partial x_k} \cdot \frac{1}{J} \\ p \delta_{3k} \frac{\partial \xi^j}{\partial x_k} \cdot \frac{1}{J} \end{bmatrix} \quad (\text{A.27})$$

We define the contravariant velocity as

$$U^j = u_1 \frac{\partial \xi^j}{\partial x_1} + u_2 \frac{\partial \xi^j}{\partial x_2} + u_3 \frac{\partial \xi^j}{\partial x_3} = u_k \frac{\partial \xi^j}{\partial x_k} \quad (\text{A.28})$$

On the other hand, for any tensor T_{ij} , the Kronecker Delta has the shifting property (Aris, 2012; Fleisch, 2011)

$$T_{ij}\delta_{jk} = T_{ik} \quad (\text{A.29})$$

Introducing (A.28) and (A.29) into (A.27), we finally obtain

$$\frac{\mathcal{F}_{L_k}}{J} \frac{\partial \xi^j}{\partial x_k} = \begin{bmatrix} \frac{U^j}{J} \\ \frac{1}{J} \cdot p \frac{\partial \xi^j}{\partial x_1} \\ \frac{1}{J} \cdot p \frac{\partial \xi^j}{\partial x_2} \\ \frac{1}{J} \cdot p \frac{\partial \xi^j}{\partial x_3} \end{bmatrix} \quad (\text{A.30})$$

A.3.2. Non-Linear Fluxes

From the non-linear flux tensor, we can compute

$$\frac{\mathcal{F}_{N_k}}{J} \frac{\partial \xi^j}{\partial x_k} = \begin{bmatrix} 0 \\ u_1 \cdot u_k \frac{\partial \xi^j}{\partial x_k} \cdot \frac{1}{J} \\ u_2 \cdot u_k \frac{\partial \xi^j}{\partial x_k} \cdot \frac{1}{J} \\ u_3 \cdot u_k \frac{\partial \xi^j}{\partial x_k} \cdot \frac{1}{J} \end{bmatrix} \quad (\text{A.31})$$

The term $u_k \frac{\partial \xi^j}{\partial x_k} \cdot \frac{1}{J} = \frac{U^j}{J}$. Introducing this into the previous expression, we finally get:

$$\frac{\mathcal{F}_{N_k}}{J} \frac{\partial \xi^j}{\partial x_k} = \begin{bmatrix} 0 \\ u_1 \frac{U^j}{J} \\ u_2 \frac{U^j}{J} \\ u_3 \frac{U^j}{J} \end{bmatrix} \quad (\text{A.32})$$

A.3.3. Viscous Fluxes

For the viscous flux, we have

$$\frac{\mathcal{F}_{v_k}}{J} \frac{\partial \xi^j}{\partial x_k} = \begin{bmatrix} 0 \\ -\frac{1}{Re} \left(\frac{1}{J} \frac{\partial u_1}{\partial x_k} \frac{\partial \xi^j}{\partial x_k} + \frac{1}{J} \frac{\partial u_k}{\partial x_1} \frac{\partial \xi^j}{\partial x_k} \right) \\ -\frac{1}{Re} \left(\frac{1}{J} \frac{\partial u_2}{\partial x_k} \frac{\partial \xi^j}{\partial x_k} + \frac{1}{J} \frac{\partial u_k}{\partial x_2} \frac{\partial \xi^j}{\partial x_k} \right) \\ -\frac{1}{Re} \left(\frac{1}{J} \frac{\partial u_3}{\partial x_k} \frac{\partial \xi^j}{\partial x_k} + \frac{1}{J} \frac{\partial u_k}{\partial x_3} \frac{\partial \xi^j}{\partial x_k} \right) \end{bmatrix} \quad (\text{A.33})$$

Let's operate the first component of the momentum equation, transforming the velocity gradients into curvilinear coordinates:

$$\frac{1}{J} \frac{\partial u_1}{\partial x_k} \frac{\partial \xi^j}{\partial x_k} + \frac{1}{J} \frac{\partial u_k}{\partial x_1} \frac{\partial \xi^j}{\partial x_k} = \frac{1}{J} \frac{\partial u_1}{\partial \xi^m} \frac{\partial \xi^m}{\partial x_k} \frac{\partial \xi^j}{\partial x_k} + \frac{\partial u_k}{\partial \xi^m} \frac{\partial \xi^m}{\partial x_1} \frac{\partial \xi^j}{\partial x_k}$$

The tensor $\frac{\partial \xi^m}{\partial x_k} \frac{\partial \xi^j}{\partial x_k}$ is the contravariant metric tensor g^{jm} . We also define the following tensor R_{ij} , which corresponds to the velocity gradient tensor

$$R_{ij} = \frac{\partial u_i}{\partial \xi^l} \frac{\partial \xi^l}{\partial x_j} \quad (\text{A.34})$$

Which implies

$$\frac{\partial u_k}{\partial \xi^m} \frac{\partial \xi^m}{\partial x_1} = R_{k1}$$

This leads to

$$\frac{1}{J} \frac{\partial u_1}{\partial x_k} \frac{\partial \xi^j}{\partial x_k} + \frac{1}{J} \frac{\partial u_k}{\partial x_1} \frac{\partial \xi^j}{\partial x_k} = \frac{1}{J} \frac{\partial u_1}{\partial \xi^m} g^{mj} + \frac{1}{J} R_{k1} \frac{\partial \xi^j}{\partial x_k}$$

Replacing the dummy summation index k by m in the second term, we obtain

$$\frac{1}{J} \frac{\partial u_1}{\partial x_k} \frac{\partial \xi^j}{\partial x_k} + \frac{1}{J} \frac{\partial u_k}{\partial x_1} \frac{\partial \xi^j}{\partial x_k} = \frac{1}{J} \frac{\partial u_1}{\partial \xi^m} g^{mj} + \frac{1}{J} R_{m1} \frac{\partial \xi^j}{\partial x_m}$$

The same procedure can be applied to the other components of the momentum equation. By applying this, we finally obtain

$$\frac{\mathcal{F}_{v_k}}{J} \frac{\partial \xi^j}{\partial x_k} = \begin{bmatrix} 0 \\ -\frac{1}{J} \frac{1}{Re} \left(\frac{\partial u_1}{\partial \xi^m} g^{mj} + R_{m1} \frac{\partial \xi^j}{\partial x_m} \right) \\ -\frac{1}{J} \frac{1}{Re} \left(\frac{\partial u_2}{\partial \xi^m} g^{mj} + R_{m2} \frac{\partial \xi^j}{\partial x_m} \right) \\ -\frac{1}{J} \frac{1}{Re} \left(\frac{\partial u_3}{\partial \xi^m} g^{mj} + R_{m3} \frac{\partial \xi^j}{\partial x_m} \right) \end{bmatrix} \quad (\text{A.35})$$

If we recall

$$\frac{\mathcal{F}_k}{J} \frac{\partial \xi^j}{\partial x_k} = \mathbf{F}^j \quad (\text{A.36})$$

We obtain the partially-transformed system equation in non-orthogonal generalised curvilinear coordinates

$$\Gamma \frac{\partial Q}{\partial t} + J \frac{\partial \mathbf{F}^j}{\partial \xi^j} + \mathbf{S} = 0 \quad (\text{A.37})$$

Which finally leads to

$$\Gamma \frac{\partial \mathbf{Q}}{\partial t} + J \frac{\partial}{\partial \xi^j} (\mathbf{F}_L^j + \mathbf{F}_N^j + \mathbf{F}_v^j) + \mathbf{S} = 0 \quad (\text{A.38})$$

Where

$$\frac{\mathcal{F}_{L_k}}{J} \frac{\partial \xi^j}{\partial x_k} = \mathbf{F}_L^j \quad ; \quad \frac{\mathcal{F}_{N_k}}{J} \frac{\partial \xi^j}{\partial x_k} = \mathbf{F}_N^j \quad ; \quad \frac{\mathcal{F}_{v_k}}{J} \frac{\partial \xi^j}{\partial x_k} = \mathbf{F}_v^j \quad (\text{A.39})$$

$$\mathbf{F}_L^j = \frac{1}{J} \begin{bmatrix} U^j \\ p \frac{\partial \xi^j}{\partial x_1} \\ p \frac{\partial \xi^j}{\partial x_2} \\ p \frac{\partial \xi^j}{\partial x_3} \end{bmatrix}; \quad \mathbf{F}_N^j = \frac{1}{J} \begin{bmatrix} 0 \\ u_1 U^j \\ u_2 U^j \\ u_3 U^j \end{bmatrix}; \quad \mathbf{F}_v^j = \frac{1}{J} \begin{bmatrix} 0 \\ -\frac{1}{Re} \left(\frac{\partial u_1}{\partial \xi^m} g^{mj} + R_{m1} \frac{\partial \xi^j}{\partial x_m} \right) \\ -\frac{1}{Re} \left(\frac{\partial u_2}{\partial \xi^m} g^{mj} + R_{m2} \frac{\partial \xi^j}{\partial x_m} \right) \\ -\frac{1}{Re} \left(\frac{\partial u_3}{\partial \xi^m} g^{mj} + R_{m3} \frac{\partial \xi^j}{\partial x_m} \right) \end{bmatrix} \quad (\text{A.40})$$

APPENDIX B. BLOCK-DIAGONAL ALGORITHM

The application of the second-difference implicit diagonal operator on the unknown variables, $\mathbf{T}_{\xi^1}(\overline{\Delta Q^*}_{i,j,k})$, $\mathbf{T}_{\xi^2}(\overline{\Delta Q^{**}}_{i,j,k})$, and $\mathbf{T}_{\xi^3}(\overline{\Delta Q^{\ell+1,n+1}}_{i,j,k})$, results in tridiagonal matrix systems in each direction on the left-hand side of the equation, which are solved using optimised Linear Algebra libraries. The notation of $R_{i,j,k}^{*\ell,n+1}$, $R_{i,j,k}^{**\ell,n+1}$ and $R_{i,j,k}^{***\ell,n+1}$ remarks that the right-hand side of the equation is updated every sub-step. It is important to keep this in mind because it will become relevant to set proper boundary conditions for the method at domain boundaries and nodes next to the free surface.

The diagonal differential operator in each direction reads (for better readability, the spectral radius in each direction at a point i, j, k is expressed as $\wp(\tilde{\mathbf{A}}_{L_{i,j,k}}^m) = \tilde{\wp}_{i,j,k}^m$)

$$\mathbf{T}_{\xi^1}()_{i,j,k} = \text{diag} \begin{bmatrix} 1 - \Delta\tau_{i,j,k} \varepsilon^1 \tilde{\wp}_{i,j,k}^1 \delta_{\xi^1 \xi^1}()_{i,j,k} \\ 1 - \Delta\tau_{i,j,k} \varepsilon^1 \tilde{\wp}_{i,j,k}^1 \delta_{\xi^1 \xi^1}()_{i,j,k} \\ 1 + \Delta\tau_{i,j,k} \theta_{i,j,k} \sqrt{g_{i,j,k}^{11}} \delta_{\xi^1}()_{i,j,k} - \Delta\tau_{i,j,k} \varepsilon^1 \tilde{\wp}_{i,j,k}^1 \delta_{\xi^1 \xi^1}()_{i,j,k} \\ 1 - \Delta\tau_{i,j,k} \theta_{i,j,k} \sqrt{g_{i,j,k}^{11}} \delta_{\xi^1}()_{i,j,k} - \Delta\tau_{i,j,k} \varepsilon^1 \tilde{\wp}_{i,j,k}^1 \delta_{\xi^1 \xi^1}()_{i,j,k} \end{bmatrix}$$

$$\mathbf{T}_{\xi^2}()_{i,j,k} = \text{diag} \begin{bmatrix} 1 - \Delta\tau_{i,j,k} \varepsilon^2 \tilde{\wp}_{i,j,k}^2 \delta_{\xi^2 \xi^2}()_{i,j,k} \\ 1 - \Delta\tau_{i,j,k} \varepsilon^2 \tilde{\wp}_{i,j,k}^2 \delta_{\xi^2 \xi^2}()_{i,j,k} \\ 1 + \Delta\tau_{i,j,k} \theta_{i,j,k} \sqrt{g_{i,j,k}^{22}} \delta_{\xi^2}()_{i,j,k} - \Delta\tau_{i,j,k} \varepsilon^2 \tilde{\wp}_{i,j,k}^2 \delta_{\xi^2 \xi^2}()_{i,j,k} \\ 1 - \Delta\tau_{i,j,k} \theta_{i,j,k} \sqrt{g_{i,j,k}^{22}} \delta_{\xi^2}()_{i,j,k} - \Delta\tau_{i,j,k} \varepsilon^2 \tilde{\wp}_{i,j,k}^2 \delta_{\xi^2 \xi^2}()_{i,j,k} \end{bmatrix}$$

$$\mathbf{T}_{\xi^3}(\)_{i,j,k} = \text{diag} \begin{bmatrix} 1 - \Delta\tau_{i,j,k} \varepsilon^3 \tilde{\wp}_{i,j,k}^3 \delta_{\xi^3 \xi^3}(\)_{i,j,k} \\ 1 - \Delta\tau_{i,j,k} \varepsilon^3 \tilde{\wp}_{i,j,k}^3 \delta_{\xi^3 \xi^3}(\)_{i,j,k} \\ 1 + \Delta\tau_{i,j,k} \theta_{i,j,k} \sqrt{g_{i,j,k}^{33}} \delta_{\xi^3}(\)_{i,j,k} - \Delta\tau_{i,j,k} \varepsilon^3 \tilde{\wp}_{i,j,k}^3 \delta_{\xi^3 \xi^3}(\)_{i,j,k} \\ 1 - \Delta\tau_{i,j,k} \theta_{i,j,k} \sqrt{g_{i,j,k}^{33}} \delta_{\xi^3}(\)_{i,j,k} - \Delta\tau_{i,j,k} \varepsilon^3 \tilde{\wp}_{i,j,k}^3 \delta_{\xi^3 \xi^3}(\)_{i,j,k} \end{bmatrix}$$

Which leads to

$$\mathbf{T}_{\xi^1}(\overline{\Delta Q_{i,j,k}^*}) = \begin{bmatrix} \overline{\Delta Q_{i,j,k}^{*1}} - \Delta \tau_{i,j,k} \varepsilon^1 \tilde{\phi}_{i,j,k}^1 & \frac{\overline{\Delta Q_{i+1,j,k}^{*1}} - 2\overline{\Delta Q_{i,j,k}^{*1}} + \overline{\Delta Q_{i-1,j,k}^{*1}}}{(\Delta \xi^1)^2} \\ \overline{\Delta Q_{i,j,k}^{*2}} - \Delta \tau_{i,j,k} \varepsilon^1 \tilde{\phi}_{i,j,k}^1 & \frac{\overline{\Delta Q_{i+1,j,k}^{*2}} - 2\overline{\Delta Q_{i,j,k}^{*2}} + \overline{\Delta Q_{i-1,j,k}^{*2}}}{(\Delta \xi^1)^2} \\ \overline{\Delta Q_{i,j,k}^{*3}} + \Delta \tau_{i,j,k} \theta_{i,j,k} \sqrt{g_{i,j,k}^{11}} \frac{\overline{\Delta Q_{i+1,j,k}^{*3}} - \overline{\Delta Q_{i-1,j,k}^{*3}}}{2\Delta \xi^1} - \Delta \tau_{i,j,k} \varepsilon^1 \tilde{\phi}_{i,j,k}^1 & \frac{\overline{\Delta Q_{i+1,j,k}^{*3}} - 2\overline{\Delta Q_{i,j,k}^{*3}} + \overline{\Delta Q_{i-1,j,k}^{*3}}}{(\Delta \xi^1)^2} \\ \overline{\Delta Q_{i,j,k}^{*4}} - \Delta \tau_{i,j,k} \theta_{i,j,k} \sqrt{g_{i,j,k}^{11}} \frac{\overline{\Delta Q_{i+1,j,k}^{*4}} - \overline{\Delta Q_{i-1,j,k}^{*4}}}{2\Delta \xi^1} - \Delta \tau_{i,j,k} \varepsilon^1 \tilde{\phi}_{i,j,k}^1 & \frac{\overline{\Delta Q_{i+1,j,k}^{*4}} - 2\overline{\Delta Q_{i,j,k}^{*4}} + \overline{\Delta Q_{i-1,j,k}^{*4}}}{(\Delta \xi^1)^2} \end{bmatrix} = \mathbf{M}_{1,i,j,k}^{-1} \begin{bmatrix} R_{i,j,k}^{*1} \\ R_{i,j,k}^{*2} \\ R_{i,j,k}^{*3} \\ R_{i,j,k}^{*4} \end{bmatrix}$$

$$\mathbf{T}_{\xi^2}(\overline{\Delta Q_{i,j,k}^{**}}) = \begin{bmatrix} \overline{\Delta Q_{i,j,k}^{**1}} - \Delta \tau_{i,j,k} \varepsilon^2 \tilde{\phi}_{i,j,k}^2 & \frac{\overline{\Delta Q_{i,j+1,k}^{**1}} - 2\overline{\Delta Q_{i,j,k}^{**1}} + \overline{\Delta Q_{i,j-1,k}^{**1}}}{(\Delta \xi^2)^2} \\ \overline{\Delta Q_{i,j,k}^{**2}} - \Delta \tau_{i,j,k} \varepsilon^2 \tilde{\phi}_{i,j,k}^2 & \frac{\overline{\Delta Q_{i,j+1,k}^{**2}} - 2\overline{\Delta Q_{i,j,k}^{**2}} + \overline{\Delta Q_{i,j-1,k}^{**2}}}{(\Delta \xi^2)^2} \\ \overline{\Delta Q_{i,j,k}^{**3}} + \Delta \tau_{i,j,k} \theta_{i,j,k} \sqrt{g_{i,j,k}^{22}} \frac{\overline{\Delta Q_{i,j+1,k}^{**3}} - \overline{\Delta Q_{i,j-1,k}^{**3}}}{2\Delta \xi^2} - \Delta \tau_{i,j,k} \varepsilon^2 \tilde{\phi}_{i,j,k}^2 & \frac{\overline{\Delta Q_{i,j+1,k}^{**3}} - 2\overline{\Delta Q_{i,j,k}^{**3}} + \overline{\Delta Q_{i,j-1,k}^{**3}}}{(\Delta \xi^2)^2} \\ \overline{\Delta Q_{i,j,k}^{**4}} - \Delta \tau_{i,j,k} \theta_{i,j,k} \sqrt{g_{i,j,k}^{22}} \frac{\overline{\Delta Q_{i,j+1,k}^{**4}} - \overline{\Delta Q_{i,j-1,k}^{**4}}}{2\Delta \xi^2} - \Delta \tau_{i,j,k} \varepsilon^2 \tilde{\phi}_{i,j,k}^2 & \frac{\overline{\Delta Q_{i,j+1,k}^{**4}} - 2\overline{\Delta Q_{i,j,k}^{**4}} + \overline{\Delta Q_{i,j-1,k}^{**4}}}{(\Delta \xi^2)^2} \end{bmatrix} = \mathbf{M}_{2,i,j,k}^{-1} \begin{bmatrix} R_{i,j,k}^{**1} \\ R_{i,j,k}^{**2} \\ R_{i,j,k}^{**3} \\ R_{i,j,k}^{**4} \end{bmatrix}$$

$$\mathbf{T}_{\xi^3}(\overline{\Delta Q_{i,j,k}^{***}}) = \begin{bmatrix} \overline{\Delta Q_{i,j,k}^{***1}} - \Delta \tau_{i,j,k} \varepsilon^3 \tilde{\wp}_{i,j,k}^3 & \frac{\overline{\Delta Q_{i,j,k+1}^{***1}} - 2\overline{\Delta Q_{i,j,k}^{***1}} + \overline{\Delta Q_{i,j,k-1}^{***1}}}{(\Delta \xi^3)^2} \\ \overline{\Delta Q_{i,j,k}^{***2}} - \Delta \tau_{i,j,k} \varepsilon^3 \tilde{\wp}_{i,j,k}^3 & \frac{\overline{\Delta Q_{i,j,k+1}^{***2}} - 2\overline{\Delta Q_{i,j,k}^{***2}} + \overline{\Delta Q_{i,j,k-1}^{***2}}}{(\Delta \xi^3)^2} \\ \overline{\Delta Q_{i,j,k}^{***3}} + \Delta \tau_{i,j,k} \theta_{i,j,k} \sqrt{g_{i,j,k}^{33}} \frac{\overline{\Delta Q_{i,j,k+1}^{***3}} - \overline{\Delta Q_{i,j,k-1}^{***3}}}{2\Delta \xi^3} - \Delta \tau_{i,j,k} \varepsilon^3 \tilde{\wp}_{i,j,k}^3 & \frac{\overline{\Delta Q_{i,j,k+1}^{***3}} - 2\overline{\Delta Q_{i,j,k}^{***3}} + \overline{\Delta Q_{i,j,k-1}^{***3}}}{(\Delta \xi^3)^2} \\ \overline{\Delta Q_{i,j,k}^{***4}} - \Delta \tau_{i,j,k} \theta_{i,j,k} \sqrt{g_{i,j,k}^{33}} \frac{\overline{\Delta Q_{i,j,k+1}^{***4}} - \overline{\Delta Q_{i,j,k-1}^{***4}}}{2\Delta \xi^3} - \Delta \tau_{i,j,k} \varepsilon^3 w_{i,j,k}^3 & \frac{\overline{\Delta Q_{i,j,k+1}^{***4}} - 2\overline{\Delta Q_{i,j,k}^{***4}} + \overline{\Delta Q_{i,j,k-1}^{***4}}}{(\Delta \xi^2)^2} \end{bmatrix} = \mathbf{M}_{3_{i,j,k}}^{-1} \begin{bmatrix} R_{i,j,k}^{***1} \\ R_{i,j,k}^{***2} \\ R_{i,j,k}^{***3} \\ R_{i,j,k}^{***4} \end{bmatrix}$$

Analysing the first equation by components and recasting the terms (it is analogous for the other directions):

$$\mathbf{T}_{\xi^1} (\overline{\Delta Q_{i,j,k}^*}) = \mathbf{M}_{1,i,j,k}^{-1} \cdot R_{i,j,k}^* = \overline{R_{i,j,k}^*}$$

$$\text{aw}_{\xi^1_{i,j,k}}^1 \overline{\Delta Q_{i-1,j,k}^{*1}} + \text{ap}_{\xi^1_{i,j,k}}^1 \overline{\Delta Q_{i,j,k}^{*1}} + \text{ae}_{\xi^1_{i,j,k}}^1 \overline{\Delta Q_{i+1,j,k}^{*1}} = \overline{R_{i,j,k}^{*1}} \quad (\text{B.1})$$

$$\text{aw}_{\xi^1_{i,j,k}}^2 \overline{\Delta Q_{i-1,j,k}^{*2}} + \text{ap}_{\xi^1_{i,j,k}}^2 \overline{\Delta Q_{i,j,k}^{*2}} + \text{ae}_{\xi^1_{i,j,k}}^2 \overline{\Delta Q_{i+1,j,k}^{*2}} = \overline{R_{i,j,k}^{*2}} \quad (\text{B.2})$$

$$\text{aw}_{\xi^1_{i,j,k}}^3 \overline{\Delta Q_{i-1,j,k}^{*3}} + \text{ap}_{\xi^1_{i,j,k}}^3 \overline{\Delta Q_{i,j,k}^{*3}} + \text{ae}_{\xi^1_{i,j,k}}^3 \overline{\Delta Q_{i+1,j,k}^{*3}} = \overline{R_{i,j,k}^{*3}} \quad (\text{B.3})$$

$$\text{aw}_{\xi^1_{i,j,k}}^4 \overline{\Delta Q_{i-1,j,k}^{*4}} + \text{ap}_{\xi^1_{i,j,k}}^4 \overline{\Delta Q_{i,j,k}^{*4}} + \text{ae}_{\xi^1_{i,j,k}}^4 \overline{\Delta Q_{i+1,j,k}^{*4}} = \overline{R_{i,j,k}^{*4}} \quad (\text{B.4})$$

Where:

- $\overline{\Delta Q_{i,j,k}^{*1}}$ (p - component)

$$\text{aw}_{\xi^1_{i,j,k}}^1 = -\frac{\Delta \tau_{i,j,k} \varepsilon^1 \tilde{\phi}_{i,j,k}^1}{(\Delta \xi^1)^2} ; \text{ ap}_{\xi^1_{i,j,k}}^1 = 1 + \frac{2 \Delta \tau_{i,j,k} \varepsilon^1 \tilde{\phi}_{i,j,k}^1}{(\Delta \xi^1)^2} ; \text{ ae}_{\xi^1_{i,j,k}}^1 = -\frac{\Delta \tau_{i,j,k} \varepsilon^1 \tilde{\phi}_{i,j,k}^1}{(\Delta \xi^1)^2}$$

- $\overline{\Delta Q_{i,j,k}^{*2}}$ (u - component)

$$\text{aw}_{\xi^1_{i,j,k}}^2 = -\frac{\Delta \tau_{i,j,k} \varepsilon^1 \tilde{\phi}_{i,j,k}^1}{(\Delta \xi^1)^2} ; \text{ ap}_{\xi^1_{i,j,k}}^2 = 1 + \frac{2 \Delta \tau_{i,j,k} \varepsilon^1 \tilde{\phi}_{i,j,k}^1}{(\Delta \xi^1)^2} ; \text{ ae}_{\xi^1_{i,j,k}}^2 = -\frac{\Delta \tau_{i,j,k} \varepsilon^1 \tilde{\phi}_{i,j,k}^1}{(\Delta \xi^1)^2}$$

- $\overline{\Delta Q_{i,j,k}^{*3}}$ (v - component)

$$\text{aw}_{\xi^1_{i,j,k}}^3 = -\frac{\Delta \tau_{i,j,k} \theta_{i,j,k} \sqrt{g_{i,j,k}^{11}}}{2\Delta \xi^1} - \frac{\Delta \tau_{i,j,k} \varepsilon^1 \tilde{\phi}_{i,j,k}^1}{(\Delta \xi^1)^2}; \quad \text{ap}_{\xi^1_{i,j,k}}^3 = 1 + \frac{2\Delta \tau_{i,j,k} \varepsilon^1 \tilde{\phi}_{i,j,k}^1}{(\Delta \xi^1)^2}; \quad \text{ae}_{\xi^1_{i,j,k}}^3 = \frac{\Delta \tau_{i,j,k} \theta_{i,j,k} \sqrt{g_{i,j,k}^{11}}}{2\Delta \xi^1} - \frac{\Delta \tau_{i,j,k} \varepsilon^1 \tilde{\phi}_{i,j,k}^1}{(\Delta \xi^1)^2}$$

- $\overline{\Delta Q_{i,j,k}^{*4}}$ (w - component)

$$\text{aw}_{\xi^1_{i,j,k}}^4 = \frac{\Delta \tau_{i,j,k} \theta_{i,j,k} \sqrt{g_{i,j,k}^{11}}}{2\Delta \xi^1} - \frac{\Delta \tau_{i,j,k} \varepsilon^1 \tilde{\phi}_{i,j,k}^1}{(\Delta \xi^1)^2}; \quad \text{ap}_{\xi^1_{i,j,k}}^4 = 1 + \frac{2\Delta \tau_{i,j,k} \varepsilon^1 \tilde{\phi}_{i,j,k}^1}{(\Delta \xi^1)^2}; \quad \text{ae}_{\xi^1_{i,j,k}}^4 = -\frac{\Delta \tau_{i,j,k} \theta_{i,j,k} \sqrt{g_{i,j,k}^{11}}}{2\Delta \xi^1} - \frac{\Delta \tau_{i,j,k} \varepsilon^1 \tilde{\phi}_{i,j,k}^1}{(\Delta \xi^1)^2}$$

B.1. Boundary conditions

The implicit operator leads to the following form of the equation at the nodes next to the boundary (we will use as example the pressure equation (variable 1) in the ξ^1 direction, next to the boundary at the end of the domain, $i = im$)

$$\text{aw}_{\xi^1_{im-1,j,k}} \overline{\Delta Q_{im-2,j,k}^{*1}} + \text{ap}_{\xi^1_{im-1,j,k}} \overline{\Delta Q_{im-1,j,k}^{*1}} + \text{ae}_{\xi^1_{im-1,j,k}} \overline{\Delta Q_{im,j,k}^{*1}} = \overline{R_{im-1,j,k}^{*1}} \quad (\text{B.5})$$

Sotiropoulos and Constantinescu (1997) remarks that any boundary condition can be formulated in a unified fashion as an extrapolation of the interior nodes values as

$$\Delta Q_{im} = \mathbf{S}_{im}^a \Delta Q_{im-1} + \mathbf{S}_{im}^b \Delta Q_{im-2} \quad (\text{B.6})$$

Where \mathbf{S}^a and \mathbf{S}^b are diagonal matrices with coefficients depending on the boundary condition type. Therefore, for the intermediate solution, it holds

$$\Delta Q_{im}^* = \mathbf{S}_{im}^a \Delta Q_{im-1}^* + \mathbf{S}_{im}^b \Delta Q_{im-2}^* \quad (\text{B.7})$$

Multiplying both sides by $\mathbf{M}_{1_{im}}^{-1}$

$$\mathbf{M}_{1_{im}}^{-1} \Delta Q_{im}^* = \mathbf{M}_{1_{im}}^{-1} (\mathbf{S}_{im}^a \Delta Q_{im-1}^* + \mathbf{S}_{im}^b \Delta Q_{im-2}^*) \quad (\text{B.8})$$

$$\overline{\Delta Q_{im}^*} = \mathbf{M}_{1_{im}}^{-1} (\mathbf{S}_{im}^a \Delta Q_{im-1}^* + \mathbf{S}_{im}^b \Delta Q_{im-2}^*) \quad (\text{B.9})$$

If we replace (B.9) into (B.10), then we get

$$\begin{aligned} & \text{aw}_{\xi^1_{im-1,j,k}} \overline{\Delta Q_{im-2,j,k}^{*1}} + \text{ap}_{\xi^1_{im-1,j,k}} \overline{\Delta Q_{im-1,j,k}^{*1}} \\ & + \text{ae}_{\xi^1_{im-1,j,k}} \mathbf{M}_{1_{im}}^{-1} (\mathbf{S}_{im}^a \Delta Q_{im-1,j,k}^* + \mathbf{S}_{im}^b \Delta Q_{im-2,j,k}^*) = \overline{R_{im-1,j,k}^{*1}} \end{aligned} \quad (\text{B.10})$$

This is an implicit boundary condition, as ΔQ_{im-1}^* and ΔQ_{im-2}^* are part of the solution of the system. In order to make the boundary conditions explicit, Sotiropoulos and Constantinescu (1997) propose to replace the unknown values by residuals at the corresponding nodes. This reads

$$\overline{\Delta Q_{im}^*} = \mathbf{M}_{1_{im}^{-1}} (\mathbf{S}_{im}^a R_{im-1}^* + \mathbf{S}_{im}^b R_{im-2}^*) \quad (\text{B.11})$$

Which finally leads to the following semi-implicit form of the equation system at the boundary

$$\begin{aligned} & \overline{aw_{\xi_{im-1,j,k}^1}^1 \Delta Q_{im-2,j,k}^{*1}} + \overline{ap_{\xi_{im-1,j,k}^1}^1 \Delta Q_{im-1,j,k}^{*1}} = \\ & \overline{R_{im-1,j,k}^{*1}} - ae_{\xi_{im-1,j,k}^1}^1 \left(\mathbf{M}_{1_{im,j,k}^{-1}} \left(\mathbf{S}_{im}^a R_{im-1,j,k}^{*1} + \mathbf{S}_{im}^b R_{im-2,j,k}^{*1} \right) \right) \end{aligned} \quad (\text{B.12})$$

An analogous procedure is implemented at the $i = 1$ boundary. The same procedure to set boundary conditions at the computational boundaries, applies to set boundary conditions over blanking regions. In that case, the whole system of equations is composed by more than one subset of equations that lead to tridiagonal submatrices.

B.2. Tridiagonal matrix system

Solving the system of equation resulting from the formulation developed above, leads to a tridiagonal matrix system to be solved in every direction. To illustrate this, we consider a very simple case, where we have a rectangular domain and we assemble the matrix system for an arbitrary j, k point in the computational domain to solve the resulting equation system along the ξ^1 direction (ξ^1 -sweep). The system structure is exactly the same for the four variables to be solved. For that reason, we dropped the super-index that indicates the variable to be solved for simplicity. The matrix coefficients, right-hand side and boundary conditions vary depending on the variable to be solved. To achieve the solution of the ξ^1 -sweep, we have to solve the tridiagonal system for every j, k nodes in the domain. The same procedure apply for the ξ^2 and ξ^3 -sweep.

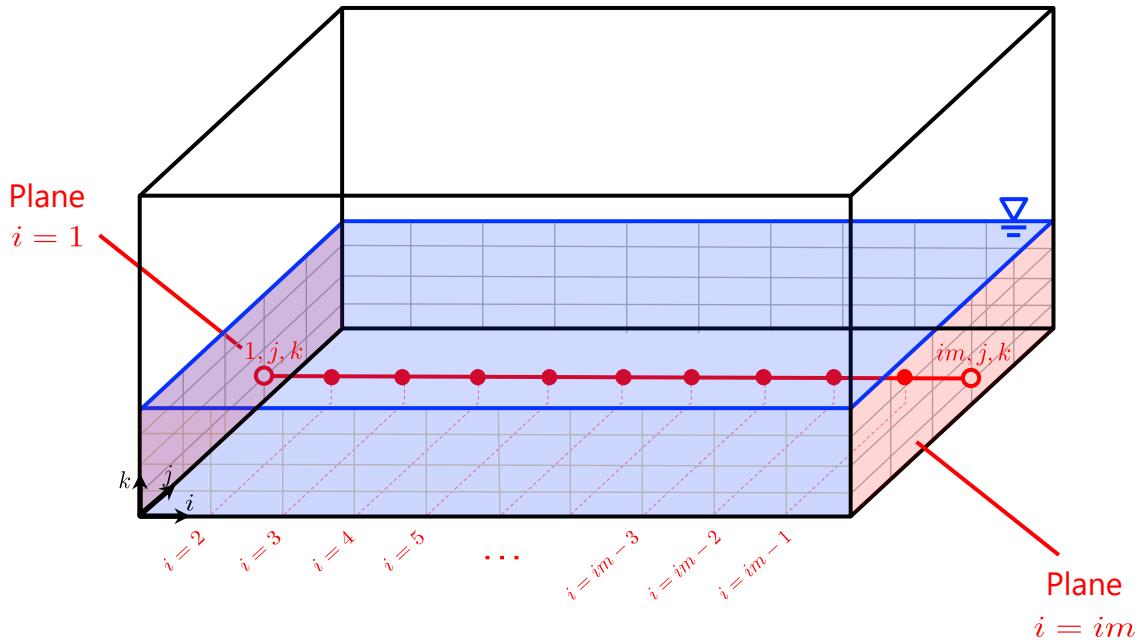


FIGURE B.1. ξ^1 -sweep for a j, k position in the computational domain

$$\begin{bmatrix}
\text{ap}_{\xi_{2,j,k}^1} & \text{ae}_{\xi_{2,j,k}^1} & 0 & 0 & \dots & 0 & 0 \\
\text{aw}_{\xi_{3,j,k}^1} & \text{ap}_{\xi_{3,j,k}^1} & \text{ae}_{\xi_{3,j,k}^1} & 0 & \dots & 0 & 0 \\
0 & \ddots & \ddots & \ddots & \ddots & 0 & 0 \\
\vdots & 0 & \text{aw}_{\xi_{i,j,k}^1} & \text{ap}_{\xi_{i,j,k}^1} & \text{ae}_{\xi_{i,j,k}^1} & 0 & \vdots \\
0 & 0 & \ddots & \ddots & \ddots & \ddots & 0 \\
0 & 0 & \dots & 0 & \text{aw}_{\xi_{im-2,j,k}^1} & \text{ap}_{\xi_{im-2,j,k}^1} & \text{ae}_{\xi_{im-2,j,k}^1} \\
0 & 0 & \dots & 0 & 0 & \text{aw}_{\xi_{im-1,j,k}^1} & \text{ap}_{\xi_{im-1,j,k}^1}
\end{bmatrix} \begin{bmatrix}
\overline{\Delta Q_{2,j,k}^*} \\
\overline{\Delta Q_{3,j,k}^*} \\
\vdots \\
\overline{\Delta Q_{i-1,j,k}^*} \\
\overline{\Delta Q_{i,j,k}^*} \\
\overline{\Delta Q_{i+1,j,k}^*} \\
\vdots \\
\overline{\Delta Q_{im-2,j,k}^*} \\
\overline{\Delta Q_{im-1,j,k}^*}
\end{bmatrix} = \begin{bmatrix}
\overline{R_{2,j,k}^*} - \overline{BC_{1,j,k}^*} \\
\overline{R_{3,j,k}^*} \\
\vdots \\
\overline{R_{i-1,j,k}^*} \\
\overline{R_{i,j,k}^*} \\
\overline{R_{i+1,j,k}^*} \\
\vdots \\
\overline{R_{im-2,j,k}^*} \\
\overline{R_{im-1,j,k}^*} - \overline{BC_{im,j,k}^*}
\end{bmatrix}$$

Where

$$\begin{aligned}
\overline{BC_{1,j,k}^*} &= \text{aw}_{\xi_{2,j,k}^1} \overline{\Delta Q_{1,j,k}^*} = \text{aw}_{\xi_{2,j,k}^1} \left(\mathbf{M}_{1_{1,j,k}^{-1}} (\mathbf{S}_1^a R_{2,j,k}^* + \mathbf{S}_1^b R_{3,j,k}^*) \right) \\
\overline{BC_{im,j,k}^*} &= \text{ae}_{\xi_{im-1,j,k}^1} \overline{\Delta Q_{im,j,k}^*} = \text{ae}_{\xi_{im-1,j,k}^1} \left(\mathbf{M}_{1_{im,j,k}^{-1}} (\mathbf{S}_{im}^a R_{im-1,j,k}^* + \mathbf{S}_{im}^b R_{im-2,j,k}^*) \right)
\end{aligned} \tag{B.13}$$

B.3. Free-surface boundary conditions

A flag variable H is set to 1 in water and 0 in air. It's is defined by a Heaviside function

$$H_{i,j,k} = \mathcal{H}(\phi_{i,j,k}) = \begin{cases} 0 & \phi_{i,j,k} \leq -\epsilon_{\text{prec}} \text{ (air)} \\ 1 & \phi_{i,j,k} > -\epsilon_{\text{prec}} \text{ (water)} \end{cases} \quad (\text{B.14})$$

Where ϵ_{prec} is the machine precision used for real numbers in the computational code. The general equation in the case with a dynamic free-surface reads (variable superscript were removed to improve readability)

$$\begin{aligned} H_{i-1,j,k} \cdot aw_{\xi^1_{i,j,k}} \overline{\Delta Q_{i-1,j,k}^*} + H_{i,j,k} \cdot ap_{\xi^1_{i,j,k}} \overline{\Delta Q_{i,j,k}^*} + H_{i+1,j,k} \cdot ae_{\xi^1_{i,j,k}} \overline{\Delta Q_{i+1,j,k}^*} = \\ \overline{R_{i,j,k}^*} - H_{i,j,k} \left((1 - H_{i-1,j,k}) \cdot aw_{\xi^1_{i,j,k}} \cdot \overline{R_{i-1,j,k}^{e*}} - (1 - H_{i+1,j,k}) \cdot ae_{\xi^1_{i,j,k}} \cdot \overline{R_{i+1,j,k}^{e*}} \right) \end{aligned} \quad (\text{B.15})$$

Where $\overline{R_{i-1,j,k}^{e*}}$ and $\overline{R_{i+1,j,k}^{e*}}$ are extrapolated from the water phase nodes using a Inverse Distance Weighting method.

An example of the application of this formulation is presented in Figure B.2.

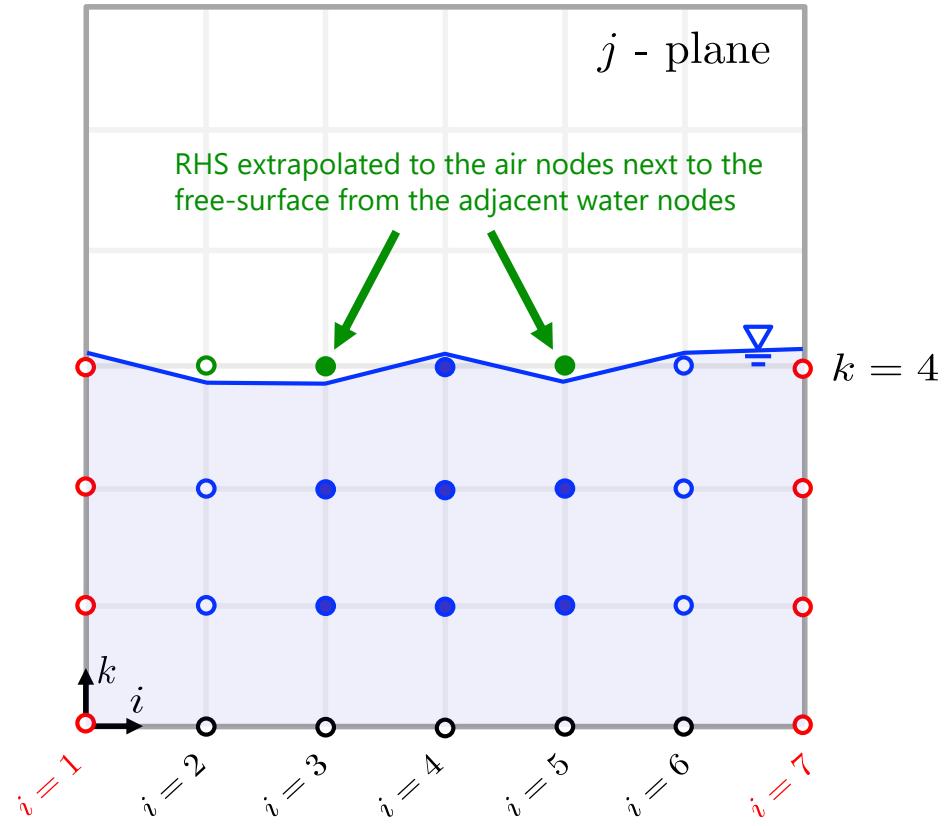


FIGURE B.2. ξ^1 -sweep for a j, k position in the computational domain

Then, the tridiagonal matrix for any variable for the example presented in Figure B.2 reads (indexes j, k in the subscript were dropped intentionally for readability)

$$\begin{bmatrix} H_2 \cdot ap_{\xi_2^1} & H_3 \cdot ae_{\xi_2^1} & 0 & 0 & 0 \\ H_2 \cdot aw_{\xi_3^1} & H_3 \cdot ap_{\xi_3^1} & H_4 \cdot ae_{\xi_3^1} & 0 & 0 \\ 0 & H_3 \cdot aw_{\xi_4^1} & H_4 \cdot ap_{\xi_4^1} & H_5 \cdot ae_{\xi_4^1} & 0 \\ 0 & 0 & H_4 \cdot aw_{\xi_5^1} & H_5 \cdot ap_{\xi_5^1} & H_6 \cdot ae_{\xi_5^1} \\ 0 & 0 & 0 & H_5 \cdot aw_{\xi_6^1} & H_6 \cdot ap_{\xi_6^1} \end{bmatrix} \begin{bmatrix} \overline{\Delta Q}_2^* \\ \overline{\Delta Q}_3^* \\ \overline{\Delta Q}_4^* \\ \overline{\Delta Q}_5^* \\ \overline{\Delta Q}_6^* \end{bmatrix} = \begin{bmatrix} \overline{R}_2^* - \overline{BC}_1^* - H_2 \left((1 - H_1) \cdot aw_{\xi_2^1} \cdot \overline{R}_1^{e^*} - (1 - H_3) \cdot ae_{\xi_2^1} \cdot \overline{R}_3^{e^*} \right) \\ \overline{R}_3^* - H_3 \left((1 - H_2) \cdot aw_{\xi_3^1} \cdot \overline{R}_2^{e^*} - (1 - H_4) \cdot ae_{\xi_3^1} \cdot \overline{R}_4^{e^*} \right) \\ \overline{R}_4^* - H_4 \left((1 - H_3) \cdot aw_{\xi_4^1} \cdot \overline{R}_3^{e^*} - (1 - H_5) \cdot ae_{\xi_4^1} \cdot \overline{R}_5^{e^*} \right) \\ \overline{R}_5^* - H_5 \left((1 - H_4) \cdot aw_{\xi_5^1} \cdot \overline{R}_4^{e^*} - (1 - H_6) \cdot ae_{\xi_5^1} \cdot \overline{R}_6^{e^*} \right) \\ \overline{R}_6^* - \overline{BC}_7^* - H_6 \left((1 - H_5) \cdot aw_{\xi_6^1} \cdot \overline{R}_5^{e^*} - (1 - H_7) \cdot ae_{\xi_6^1} \cdot \overline{R}_7^{e^*} \right) \end{bmatrix}$$

Which, given the configuration of the figure, results in

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & ap_{\xi_4^1} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & ap_{\xi_6^1} \end{bmatrix} \begin{bmatrix} \overline{\Delta Q}_2^* \\ \overline{\Delta Q}_3^* \\ \overline{\Delta Q}_4^* \\ \overline{\Delta Q}_5^* \\ \overline{\Delta Q}_6^* \end{bmatrix} = \begin{bmatrix} \overline{R}_2^* - \overline{BC}_1^* \\ \overline{R}_3^* \\ \overline{R}_4^* - aw_{\xi_4^1} \cdot \overline{R}_3^{e^*} - ae_{\xi_4^1} \cdot \overline{R}_5^{e^*} \\ \overline{R}_5^* \\ \overline{R}_6^* - \overline{BC}_7^* - aw_{\xi_6^1} \cdot \overline{R}_5^{e^*} \end{bmatrix}$$

APPENDIX C. DYNAMIC BOUNDARY CONDITION AT THE FREE SURFACE

C.1. Tangential Boundary Condition

The stress tensor at any point of the incompressible fluid domain is defined as

$$\sigma_{ij} = -P\delta_{ij} + 2\mu S_{ij} = -P\delta_{ij} + \tau_{ij} \quad (\text{C.1})$$

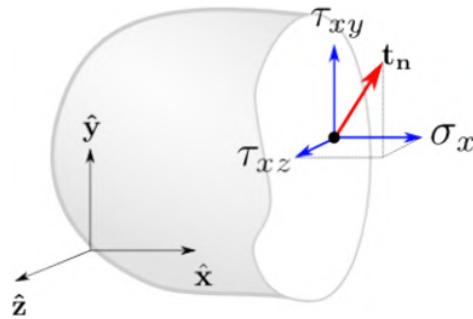


FIGURE C.1. Traction vector at an arbitrary point for a plane defined by normal direction \underline{n} .

To extract the traction vector at a surface with normal direction \underline{n} , we operate the stress tensor with the normal vector (see Figure C.1)

$$\underline{t}_n = \underline{\underline{\sigma}} \cdot \underline{n} \iff t_{n_i} = \sigma_{ij} n_j \quad (\text{C.2})$$

The traction vector t_n is not necessarily aligned with the normal direction of the surface defined by \underline{n} . Recasting the previous expression

$$t_{n_i} = \sigma_{ij} \cdot n_j \quad (\text{C.3})$$

$$t_{n_i} = (-P\delta_{ij} + \tau_{ij}) \cdot n_j \quad (\text{C.4})$$

$$t_{n_i} = -P\delta_{ij} n_j + \tau_{ij} n_j \quad (\text{C.5})$$

$$t_{n_i} = -Pn_i + \tau_{ij} n_j \quad (\text{C.6})$$

To obtain the tangential component of the traction vector, we project it onto the \underline{s} direction

$$t_s = t_{n_i} s_i = P n_i s_i + \tau_{ij} n_j s_i \quad (\text{C.7})$$

\underline{n} and \underline{s} are mutually orthogonal, then $n_i s_i = 0$

$$t_s = \tau_{ij} n_j s_i = 2\mu \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right) n_j s_i \quad (\text{C.8})$$

Finally, the zero-shear condition at the free surface implies

$$t_s = 0 \Rightarrow \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right) n_j s_i = 0 \quad (\text{C.9})$$

C.2. Normal Boundary Condition

The dynamic boundary condition for an interface between two fluids in equilibrium is given by (Batchelor, 1967):

$$[\widetilde{\sigma_{ij}} n_j] = \widetilde{\sigma_{ij}}^{f_1} n_j - \widetilde{\sigma_{ij}}^{f_2} n_j = \gamma \widetilde{\kappa} n_i \quad (\text{C.10})$$

Where $(\widetilde{\cdot})$ denotes that the variable is dimensional and the operator $[\cdot]$ indicates the difference between the value of a variable on both sides of an interface (jump condition). f_1 and f_2 indicate the two fluids at the boundary, where f_1 is the heavier fluid, γ is the surface tension coefficient, $\widetilde{\kappa}$ is the dimensional curvature and n_i is the normal direction.

If we use the velocity and length scales \mathcal{U} and \mathcal{L} , we can express the previous relations as follows:

$$\rho_{f_1} \mathcal{U}^2 \left(\sigma_{ij}^{f_1} n_j - \frac{\rho_{f_2}}{\rho_{f_1}} \sigma_{ij}^{f_1} n_j \right) = \gamma \frac{\kappa}{\mathcal{L}} n_i$$

Rearranging terms, we obtain:

$$\sigma_{ij}^{f_1} n_j - \frac{\rho_{f_2}}{\rho_{f_1}} \sigma_{ij}^{f_1} n_j = \frac{\gamma}{\rho_{f_1} \mathcal{L} \mathcal{U}^2} \kappa n_i$$

$$\sigma_{ij}^{f_1} n_j - \frac{\rho_{f_2}}{\rho_{f_1}} \sigma_{ij}^{f_1} n_j = \frac{\kappa}{We} n_i$$

Where $We = \rho_{f_1} \mathcal{L} \mathcal{U}^2 / \gamma$ is the Weber number, which is the ratio between inertia and surface tension forces (Frohn & Roth, 2000). If we project the previous expression onto the normal direction to obtain the normal dynamic boundary condition, we obtain the following:

$$\sigma_{ij}^{f_1} n_i n_j - \frac{\rho_{f_2}}{\rho_{f_1}} \sigma_{ij}^{f_1} n_i n_j = \frac{\kappa}{We}$$

If we consider $\rho_{f_1} \gg \rho_{f_2}$, which is the case in free-surface channel flows, and we neglect the effect of the curvature ($\kappa \ll 1$) and surface tension ($We \gg 1$) at the free surface (fs), we end up with the normal dynamic boundary condition

$$(\sigma_{ij}^{f_1} n_i n_j)_{fs} = 0$$

If we expand $\sigma_{ij} n_i n_j$ (we dropped the fluid superscript for better readability)

$$(\sigma_{ij} n_i n_j)_{fs} = 0$$

APPENDIX D. SIMPSON RULE IN GENERALISED CURVILINEAR COORDINATES

To obtain a filtered variable over a domain Ω_x , we need to average it over this region

$$\hat{f}(x, y, z) = \frac{\iiint_{\Omega_x} f(x, y, z) \, dx \, dy \, dz}{\iiint_{\Omega_x} dx \, dy \, dz} \quad (\text{D.1})$$

If the computational domain is defined over a curvilinear space, then Ω_x is not necessarily a parallelogram, and as a consequence, it might be very difficult to evaluate this integral numerically. Using the change of variable theorem (Stewart, 2012), we can express the previous integral as

$$\iiint_{\Omega_x} f(x, y, z) \, dx \, dy \, dz = \iiint_{\Omega_\xi} f(x(\xi, \eta, \zeta), y(\xi, \eta, \zeta), z(\xi, \eta, \zeta)) \frac{1}{J} \, d\xi \, d\eta \, d\zeta \quad (\text{D.2})$$

Where J is the determinant of the Jacobian matrix $\partial \xi^i / \partial x_j$. Defining

$$\mathbf{f}^\xi(\xi, \eta, \zeta) \equiv \frac{f(x(\xi, \eta, \zeta), y(\xi, \eta, \zeta), z(\xi, \eta, \zeta))}{J} \quad (\text{D.3})$$

We can re-express the integral D.2 as

$$\iiint_{\Omega_x} f(x, y, z) \, dx \, dy \, dz = \iiint_{\Omega_\xi} \mathbf{f}^\xi(x(\xi, \eta, \zeta), y(\xi, \eta, \zeta), z(\xi, \eta, \zeta)) \, d\xi \, d\eta \, d\zeta \quad (\text{D.4})$$

Where Ω_ξ is now a parallelogram and an equally spaced domain, so the integral limits can be defined explicitly

$$\iiint_{\Omega_\xi} \mathbf{f}^\xi(x(\xi, \eta, \zeta), y(\xi, \eta, \zeta), z(\xi, \eta, \zeta)) = \int_{\zeta_{i,j,k}-\Delta_\zeta}^{\zeta_{i,j,k}+\Delta_\zeta} \int_{\eta_{i,j,k}-\Delta_\eta}^{\eta_{i,j,k}+\Delta_\eta} \int_{\xi_{i,j,k}-\Delta_\xi}^{\xi_{i,j,k}+\Delta_\xi} \mathbf{f}^\xi(\xi, \eta, \zeta) d\xi d\eta d\zeta d\xi d\eta d\zeta \quad (\text{D.5})$$

Similarly

$$\iiint_{\Omega_x} dx dy dz = \int_{\zeta_{i,j,k}-\Delta_\zeta}^{\zeta_{i,j,k}+\Delta_\zeta} \int_{\eta_{i,j,k}-\Delta_\eta}^{\eta_{i,j,k}+\Delta_\eta} \int_{\xi_{i,j,k}-\Delta_\xi}^{\xi_{i,j,k}+\Delta_\xi} \frac{1}{J} d\xi d\eta d\zeta \quad (\text{D.6})$$

The Simpson formula, which is obtained by approximating the function by a quadratic polynomial over Ω , allows us to approximate the integral using nodal values as follows (Fujii, 2010)

$$\begin{aligned} & \int_{\zeta_{i,j,k}-\Delta_\zeta}^{\zeta_{i,j,k}+\Delta_\zeta} \int_{\eta_{i,j,k}-\Delta_\eta}^{\eta_{i,j,k}+\Delta_\eta} \int_{\xi_{i,j,k}-\Delta_\xi}^{\xi_{i,j,k}+\Delta_\xi} \mathbf{f}^\xi(\xi, \eta, \zeta) d\xi d\eta d\zeta \\ & \approx \frac{\Delta_\xi \Delta_\eta \Delta_\zeta}{27} \cdot \left[64 \mathbf{f}_{i,j,k}^\xi \right. \\ & \quad + 16 \left(\mathbf{f}_{i+1,j,k}^\xi + \mathbf{f}_{i-1,j,k}^\xi + \mathbf{f}_{i,j+1,k}^\xi + \mathbf{f}_{i,j-1,k}^\xi + \mathbf{f}_{i,j,k+1}^\xi + \mathbf{f}_{i,j,k-1}^\xi \right) \\ & \quad + 4 \left(\mathbf{f}_{i+1,j+1,k}^\xi + \mathbf{f}_{i+1,j,k+1}^\xi + \mathbf{f}_{i,j+1,k+1}^\xi + \mathbf{f}_{i-1,j-1,k}^\xi + \mathbf{f}_{i-1,j,k-1}^\xi + \mathbf{f}_{i,j-1,k-1}^\xi \right. \\ & \quad \left. + \mathbf{f}_{i+1,j-1,k}^\xi + \mathbf{f}_{i-1,j+1,k}^\xi + \mathbf{f}_{i+1,j,k-1}^\xi + \mathbf{f}_{i-1,j,k+1}^\xi + \mathbf{f}_{i,j-1,k+1}^\xi + \mathbf{f}_{i,j+1,k-1}^\xi \right) \\ & \quad + \mathbf{f}_{i+1,j+1,k+1}^\xi + \mathbf{f}_{i-1,j-1,k-1}^\xi + \mathbf{f}_{i+1,j+1,k-1}^\xi + \mathbf{f}_{i+1,j-1,k-1}^\xi \\ & \quad \left. + \mathbf{f}_{i-1,j+1,k+1}^\xi + \mathbf{f}_{i-1,j-1,k+1}^\xi + \mathbf{f}_{i-1,j+1,k-1}^\xi + \mathbf{f}_{i+1,j-1,k+1}^\xi \right] \quad (\text{D.7}) \end{aligned}$$

So, finally, the filtered variable can be approximated as

$$\hat{f}(x_{i,j,k}, y_{i,j,k}, z_{i,j,k}) \equiv \hat{f}_{i,j,k} \approx \frac{\mathcal{N}_{i,j,k}}{\mathcal{D}_{i,j,k}} \quad (\text{D.8})$$

Where

$$\begin{aligned}
\mathcal{N}_{i,j,k} = & 64 \mathbf{f}_{i,j,k}^\xi \\
& + 16 \left(\mathbf{f}_{i+1,j,k}^\xi + \mathbf{f}_{i-1,j,k}^\xi + \mathbf{f}_{i,j+1,k}^\xi + \mathbf{f}_{i,j-1,k}^\xi + \mathbf{f}_{i,j,k+1}^\xi + \mathbf{f}_{i,j,k-1}^\xi \right) \\
& + 4 \left(\mathbf{f}_{i+1,j+1,k}^\xi + \mathbf{f}_{i+1,j,k+1}^\xi + \mathbf{f}_{i,j+1,k+1}^\xi + \mathbf{f}_{i-1,j-1,k}^\xi + \mathbf{f}_{i-1,j,k-1}^\xi + \mathbf{f}_{i,j-1,k-1}^\xi \right. \\
& \quad \left. + \mathbf{f}_{i+1,j-1,k}^\xi + \mathbf{f}_{i-1,j+1,k}^\xi + \mathbf{f}_{i+1,j,k-1}^\xi + \mathbf{f}_{i-1,j,k+1}^\xi + \mathbf{f}_{i,j-1,k+1}^\xi + \mathbf{f}_{i,j+1,k-1}^\xi \right) \\
& + \mathbf{f}_{i+1,j+1,k+1}^\xi + \mathbf{f}_{i-1,j-1,k-1}^\xi + \mathbf{f}_{i+1,j+1,k-1}^\xi + \mathbf{f}_{i+1,j-1,k-1}^\xi \\
& \quad + \mathbf{f}_{i-1,j+1,k+1}^\xi + \mathbf{f}_{i-1,j-1,k+1}^\xi + \mathbf{f}_{i-1,j+1,k-1}^\xi + \mathbf{f}_{i+1,j-1,k+1}^\xi \tag{D.9}
\end{aligned}$$

And

$$\begin{aligned}
\mathcal{D}_{i,j,k} = & 64 \frac{1}{J_{i,j,k}} \\
& + 16 \left(\frac{1}{J_{i+1,j,k}} + \frac{1}{J_{i-1,j,k}} + \frac{1}{J_{i,j+1,k}} + \frac{1}{J_{i,j-1,k}} + \frac{1}{J_{i,j,k+1}} + \frac{1}{J_{i,j,k-1}} \right) \\
& + 4 \left(\frac{1}{J_{i+1,j+1,k}} + \frac{1}{J_{i+1,j,k+1}} + \frac{1}{J_{i,j+1,k+1}} + \frac{1}{J_{i-1,j-1,k}} + \frac{1}{J_{i-1,j,k-1}} + \frac{1}{J_{i,j-1,k-1}} \right. \\
& \quad \left. + \frac{1}{J_{i+1,j-1,k}} + \frac{1}{J_{i-1,j+1,k}} + \frac{1}{J_{i+1,j,k-1}} + \frac{1}{J_{i-1,j,k+1}} + \frac{1}{J_{i,j-1,k+1}} + \frac{1}{J_{i,j+1,k-1}} \right) \\
& + \frac{1}{J_{i+1,j+1,k+1}} + \frac{1}{J_{i-1,j-1,k-1}} + \frac{1}{J_{i+1,j+1,k-1}} + \frac{1}{J_{i+1,j-1,k-1}} \\
& \quad + \frac{1}{J_{i-1,j+1,k+1}} + \frac{1}{J_{i-1,j-1,k+1}} + \frac{1}{J_{i-1,j+1,k-1}} + \frac{1}{J_{i+1,j-1,k+1}} \tag{D.10}
\end{aligned}$$