



MOVING BOUNDARY METHODS FOR THE SIMULATION OF TWO-FLUID AND SEDIMENT-FLUID INTERACTIONS

Malú Grave

Tese de Doutorado apresentada ao Programa de Pós-graduação em Engenharia Civil, COPPE, da Universidade Federal do Rio de Janeiro, como parte dos requisitos necessários à obtenção do título de Doutor em Engenharia Civil.

Orientador: Alvaro Luiz Gayoso de Azeredo
Coutinho

Rio de Janeiro
Março de 2020

MOVING BOUNDARY METHODS FOR THE SIMULATION OF TWO-FLUID AND
SEDIMENT-FLUID INTERACTIONS

Malú Grave

TESE SUBMETIDA AO CORPO DOCENTE DO INSTITUTO ALBERTO LUIZ
COIMBRA DE PÓS-GRADUAÇÃO E PESQUISA DE ENGENHARIA DA
UNIVERSIDADE FEDERAL DO RIO DE JANEIRO COMO PARTE DOS
REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE DOUTOR
EM CIÊNCIAS EM ENGENHARIA CIVIL.

Orientador: Alvaro Luiz Gayoso de Azeredo Coutinho

Aprovada por: Prof. Alvaro Luiz Gayoso de Azeredo Coutinho

Prof. Abimael Fernando Dourado Loula

Prof. Adriano Maurício de Almeida Côrtes

Prof. Fernando Pereira Duda

Prof. José Jerônimo Camata

Prof. Luiz Landau

RIO DE JANEIRO, RJ – BRASIL

MARÇO DE 2020

Grave, Malú

Moving boundary methods for the simulation of two-fluid and sediment-fluid interactions/Malú Grave. – Rio de Janeiro: UFRJ/COPPE, 2020.

XVI, 125 p.: il.; 29, 7cm.

Orientador: Alvaro Luiz Gayoso de Azeredo Coutinho

Tese (doutorado) – UFRJ/COPPE/Programa de Engenharia Civil, 2020.

Referências Bibliográficas: p. 105 – 118.

1. Finite Element. 2. Level-set. 3. Two-phase flows. 4. Mesh movement. 5. Sediment transport. I. Coutinho, Alvaro Luiz Gayoso de Azeredo. II. Universidade Federal do Rio de Janeiro, COPPE, Programa de Engenharia Civil. III. Título.

*Aos meus pais, que nunca mediram
esforços para que eu continuasse
estudando.*

Agradecimentos

Primeiramente, eu gostaria de agradecer meu orientador, Professor Alvaro Coutinho. Sou muito grata por ter tido a oportunidade de trabalhar com ele. Ele é aquele orientador que todos gostariam de ter. Sempre disponível quando eu precisava, me dando liberdade para tomar minhas próprias decisões durante o desenvolvimento da tese, sem deixar de me direcionar quando achou necessário.

Um agradecimento especial vai para o Professor Jose Camata, que foi quase um co-orientador durante essa pesquisa. Serei eternamente grata pela sua ajuda que foi essencial para a construção desta tese. Obrigada também pela paciência e ensinamentos ao longo desses anos.

Gostaria de agradecer os amigos (alunos, funcionários e professores) do Núcleo de Atendimento em Computação de Alto Desempenho (NACAD). Obrigada pela troca de conhecimentos, cooperação e, claro, amizade. Um agradecimento especial à minha amiga Linda Gesenhues, com quem pude compartilhar ótimos momentos dentro e fora do NACAD; ao Professor Adriano Côrtes, que com uma aula ajudou a minha pesquisa deslanchar; e à Mara Prata, por toda ajuda burocrática ao longo desses anos.

Obrigada também à CAPES (Coordenação de Aperfeiçoamento de Pessoal de Nível Superior) e à PETROBRAS pelo suporte financeiro tão importante para a realização deste trabalho.

Por fim, gostaria de expressar minha profunda gratidão aos meus pais e ao meu namorado por me fornecerem incentivo e apoio irrestrito ao longo dos meus anos de estudo e no processo de desenvolvimento e redação desta tese. Também sou grata a todos os meus outros familiares e amigos que me apoiaram ao longo do caminho. Vocês foram muito importantes, principalmente nos meses que antecederam a defesa desta tese, pois foi quando o meu pai veio a falecer. Ele não pôde comemorar esta conquista com a gente, mas sei que deve estar muito orgulhoso.

Essa conquista não teria sido possível sem todos vocês.

Obrigada.

Resumo da Tese apresentada à COPPE/UFRJ como parte dos requisitos necessários para a obtenção do grau de Doutor em Ciências (D.Sc.)

MÉTODOS DE FRONTEIRAS MÓVEIS PARA A SIMULAÇÃO DE
ESCOAMENTOS COM INTERAÇÃO ENTRE DOIS FLUIDOS E
FLUIDO-SEDIMENTO

Malú Grave

Março/2020

Orientador: Alvaro Luiz Gayoso de Azeredo Coutinho

Programa: Engenharia Civil

Uma categoria desafiadora no cálculo de escoamentos de fluidos é a modelagem de fronteiras e superfícies móveis. Existem duas abordagens para modelar esses problemas: métodos de rastreamento de interface e de captura de interface. Neste trabalho, dois métodos são estudados, sendo um de cada abordagem. Essa escolha se dá considerando os diferentes tipos de problemas tratados neste trabalho, que são de interações entre dois fluidos e fluido-sedimento. O primeiro está relacionado à interação entre dois fluidos com propriedades diferentes, como água e ar, e o segundo é sobre a interação entre um fluido e um fundo de sedimentos que se deforma a medida que o escoamento evolui. Para modelar problemas com dois fluidos, o modelo matemático resulta das equações de Navier-Stokes tratadas com a Formulação Variacional Multiescala baseada no resíduo, juntamente com o método Curva de Nível Convectada para rastrear a interface entre os diferentes fluidos. Por outro lado, para modelar a interação fluido-sedimento, uma abordagem completa dentro da descrição Lagrangeana-Euleriana arbitrária com adaptação de malha é apresentada. As equações de Navier-Stokes são combinadas com uma equação de advecção-difusão para sedimentos em suspensão. As alterações morfológicas são calculadas por modelos empíricos usados para representar a taxa de erosão, bem como o carregamento de sedimentos no leito. Os deslocamentos e, conseqüentemente, o movimento da interface, são aplicados como movimentação da malha. Novas técnicas são introduzidas para melhorar os resultados de ambos os métodos. Os algoritmos são validados com problemas em que dados experimentais e numéricos estão disponíveis. Por fim, os resultados são discutidos e analisados.

Abstract of Thesis presented to COPPE/UFRJ as a partial fulfillment of the requirements for the degree of Doctor of Science (D.Sc.)

MOVING BOUNDARY METHODS FOR THE SIMULATION OF TWO-FLUID AND SEDIMENT-FLUID INTERACTIONS

Malú Grave

March/2020

Advisor: Alvaro Luiz Gayoso de Azeredo Coutinho

Department: Civil Engineering

One challenging category of problems in flow computation is the modeling of moving boundaries and interfaces. There are different techniques to model such problems: interface tracking and interface capturing methods. In this work, we study two methods of modeling moving boundaries and interfaces in CFD problems. This choice is made considering the type of problem under consideration: two-fluid and sediment-fluid interactions. The first one is related to the interaction between two fluids with different properties, like water and air, and the second one is about the interaction between a fluid and a sediment bottom that deforms as the flow evolves. Therefore, we implement the convected level-set method to model two-fluid problems. The mathematical model results from the Navier-Stokes equations treated with the Residual Based Variational Multiscale formulation, together with the convected level-set method, used to track the interfaces between the different fluids. Here, we introduce a new truncated signed distance function to get a smooth truncation far from the interface. On the other hand, to model sediment-fluid interaction, we present a full approach within the Arbitrary Lagrangian-Eulerian framework with mesh adaptivity. The Navier-Stokes equations are combined with an advection-diffusion equation for suspended sediments. The morphological changes are calculated by empirical models used to represent the entrainment rate as well as the bed-load transport. The displacements, and consequently, the interface motion, are applied as mesh movement. We introduce a new smoothing technique called upwind sand-slide, in which we adjust the slope angle considering the flow direction. We validate the algorithm with test cases where experimental and numerical data are available. Finally, the results are discussed and analyzed.

Contents

List of Figures	xi
List of Tables	xvi
1 Introduction	1
1.1 Objectives and methodology	3
1.2 Background	4
1.2.1 Moving boundaries	4
1.2.2 Flow model	8
1.2.3 Sediment transport	8
1.3 Literature review	9
1.4 Structure of the thesis	10
2 Interface tracking and interface capturing methods	12
2.1 Convected Level-Set formulation	12
2.1.1 Governing equations	13
2.1.2 Time integration	16
2.1.3 Finite element formulation	16
2.1.4 Global mass conservation	18
2.2 Moving mesh method	19
3 Navier-Stokes equations	21
3.1 Governing equations	21
3.2 The N-S equations within an ALE framework	23
3.3 Residual-Based Variational Multiscale formulation	24
3.4 Time integration	27
3.5 Linearization of the Navier-Stokes Equations	28
3.6 Nondimensional Equations and Numbers	29
3.7 Boundary conditions	31
4 Sediment transport model	35
4.1 Governing equations	35

4.1.1	Suspended Load	35
4.1.2	Bed-load transport	36
4.1.3	Entrainment and Deposition Rate	37
4.2	Bed morphology model with ALE	38
4.3	Upwind sand-slide	38
5	Computational aspects	42
5.1	The <code>libmesh</code> library	42
5.2	Adaptive Mesh Refinement (AMR)	43
5.3	Two-fluid algorithm	43
5.4	Sediment-fluid algorithm	44
6	Results and Discussion	46
6.1	Verification of the fluid flow model	46
6.1.1	Lid-driven cavity test case	46
6.1.2	Flow over a backward facing-step test case	47
6.2	Verification of the convected level-set	51
6.2.1	Pure convection of a circle	51
6.2.2	Advection of a sphere	55
6.2.3	Zalesak’s problem	56
6.2.4	Single vortex problem	57
6.3	Verification of the surface tension model	59
6.3.1	Film-like interface	60
6.3.2	Pressure inside a bubble due the surface tension	61
6.4	Verification of the moving mesh method	62
6.4.1	Mesh movement	62
6.4.2	Upwind sand-slide	63
6.5	Two-fluid simulations	68
6.5.1	2D rising bubble	70
6.5.2	3D rising bubble	71
6.5.3	Grace’s bubble diagram	75
6.5.4	Coalescence of two rising gas bubbles	80
6.6	Suspended load of sediments simulations	83
6.6.1	Net entrainment test case	84
6.6.2	Net deposition test case	87
6.7	Sediment-fluid interaction	91
6.7.1	Wall Jet Scour test case	91
6.7.2	Scour Over a Pipeline test case	94
7	Conclusions and future work	102

References	105
A RBVMS formulation applied to the Navier-Stokes equations	119
B RBVMS formulation applied to the sediment transport equations	125

List of Figures

1.1	Comparison of Lagrangian and Eulerian methods for tracking moving boundaries [1].	5
1.2	Types of sediment transport	9
2.1	Distinguishing two phases separated by a circular interface with the standard level-set method	13
2.2	Comparison between the standard and modified SDFs with $E = 0.1$	15
3.1	Comparison between the regularized and scaled Heaviside and Dirac functions.	24
4.1	Upwind sand-slide model: (a) Initial configuration in which the slope angle is bigger than the sediment angle of repose; (b) Traditional sand-slide models in which we rotate the element boundary around its gravity center until the slope becomes lower or equal to the sediment angle of repose; (c) Upwind sand-slide model in which we consider the direction of the flow. If the flow is coming in the up-slope direction, we fix the first node, and we move the two subsequent nodes in the upwind flow direction, one to adjust the slope, and the other to keep the mass conservation. When the flow is coming in the down-slope direction, we can use the traditional method. The green area is the sediment mass added, and the red area is the sediment mass removed.	40
4.2	Two iterations of the sand-slide algorithm; (a) initial bed form, (b) first iteration, (c) second iteration [2]	41
5.1	Flow chart of computation of two-fluid problems	44
5.2	Flow chart of computation of sediment-fluid problems	45
6.1	Lid-driven cavity: Geometry	47
6.2	Lid-driven cavity: Velocity profile using different meshes with $Re = 5000$	48
6.3	Lid-driven cavity: Velocity profile with $Re = 1000$	48
6.4	Lid-driven cavity: Velocity profile with $Re = 5000$	49

6.5	Lid-driven cavity: Velocity profile with $Re = 10000$	49
6.6	Lid-driven cavity: Streamlines for different Reynolds.	50
6.7	Backward facing-step: Geometry	50
6.8	Backward facing-step: velocity profile using different meshes with $Re = 800$	50
6.9	Backward facing-step: Streamlines for $Re = 800$ in $0 < x < 10$	50
6.10	Backward facing-step: Dynamic pressure at $x=20$	51
6.11	Pure convection of a circle: Circle deformation on different meshes, with different time-steps and without the global mass conservation procedure. In the simple green line: the initial zero iso-value of the distance function. In the blue dashed line: the same iso-value at the final time-step (after one turn) with the BDF2 time discretization. In the red dotted line: the same iso-value at the final time-step (after one turn) with the BE time discretization. (a)-(h) correspond to values given in Table 6.1	53
6.12	Pure convection of a circle: Area loss per time of all simulations [(a)-(h)] given in Table 6.1	53
6.13	Pure convection of a circle: Circle deformation on different meshes, with different time-steps and with the global mass conservation procedure. In the simple green line: the initial zero iso-value of the distance function. In the blue dashed line: the same iso-value at the final time-step (after one turn) with the BDF2 time discretization. In the red dotted line: the same iso-value at the final time-step (after one turn) with the BE time discretization. (a)-(h) correspond to values given in Table 6.1	54
6.14	Pure convection of a circle: Position of the circle and mesh refinement at different time-steps during one turn.	55
6.15	Pure convection of a circle: iso-values when using the reinitialization or not.	55
6.16	Pure convection of a sphere: Initial mesh refinement of the advection of a sphere problem.	56
6.17	Pure convection of a sphere: Position of the sphere during one turn.	57
6.18	Zalesak's problem: Configuration.	57
6.19	Zalesak's problem: Position of the disk and mesh refinement at different time-steps during one turn.	58
6.20	Zalesak's problem: Initial disk configuration (blue line) and final configuration after one turn (green line with markers)	58
6.21	Single vortex problem: Vortex deformation and mesh refinement at different time-steps for $T = 4$	59
6.22	Single vortex problem: Vortex deformation and mesh refinement at different time-steps for $T = 8$	59

6.23	Single vortex problem: Initial configuration (blue line) and final configuration after one period (dashed green line).	60
6.24	Film-like interface: Geometry of the closed contour.	60
6.25	Pressure inside a bubble due the surface tension: Results.	62
6.26	Mesh movement: displacements when the elements have the same area.	64
6.27	Mesh movement: displacements when the elements have different areas.	65
6.28	Mesh movement: comparison between an anisotropic diffusion (left) and a isotropic diffusion in the gravity acceleration direction (right).	66
6.29	Upwind sand-slide: Geometry and mesh	67
6.30	Upwind sand-slide: Mesh displacements of case A	67
6.31	Upwind sand-slide: Mesh displacements of case B	68
6.32	Upwind sand-slide: Mesh displacements of case C	68
6.33	Upwind sand-slide: Initial (dashed lines) and final geometry of case A	68
6.34	Upwind sand-slide: Initial (dashed lines) and final geometry of case B	68
6.35	Upwind sand-slide: Initial (dashed lines) and final geometry of case C	69
6.36	Upwind sand-slide: Mesh displacements with smoothing of case A	69
6.37	Upwind sand-slide: Mesh displacements with smoothing of case B	69
6.38	Upwind sand-slide: Mesh displacements with smoothing of case C	69
6.39	2D rising bubble: Initial configuration and boundary conditions for the test cases.	70
6.40	2D rising bubble: Test A results.	72
6.41	2D rising bubble: Test B results.	73
6.42	3D rising bubble: Initial configuration and boundary conditions for the test cases.	74
6.43	3D rising bubble: Bubble shapes at different times for the test A.	75
6.44	3D rising bubble: Test A results.	76
6.45	3D rising bubble: Bubble shapes at different times for the test B.	77
6.46	3D rising bubble: Test B results.	78
6.47	3D rising bubble: Comparison of the mesh refinement.	79
6.48	3D rising bubble: Test A comparison of the results with different mesh refinements. The smallest element size for $h_{max(LS)} = 2$ is 0.025 m, for $h_{max(LS)} = 3$ is 0.0125 m and $h_{max(LS)} = 4$ is 0.00625.	80
6.49	3D rising bubble: Test B comparison of the results with different mesh refinements. The smallest element size for $h_{max(LS)} = 2$ is 0.025 m, for $h_{max(LS)} = 3$ is 0.0125 m and $h_{max(LS)} = 4$ is 0.00625.	81
6.50	Grace's bubble diagram: Test A comparison of the results with different mesh refinements.	83
6.51	Grace's bubble diagram: Test B comparison of the results with different mesh refinements.	84

6.52	Grace's bubble diagram: Test C comparison of the results with different mesh refinements.	85
6.53	Grace's bubble diagram: Test D comparison of the results with different mesh refinements.	86
6.54	Grace's bubble diagram: Test E comparison of the results with different mesh refinements.	87
6.55	Coalescence of two rising gas bubbles: Snapshots of the co-axial coalescence together with the experimental data of [3].	88
6.56	Coalescence of two rising gas bubbles: Snapshots of the oblique coalescence together with the experimental data of [3].	89
6.57	Net entrainment test case: Schematic definition of the problem.	89
6.58	Net entrainment test case: Geometry, boundary conditions, time-step, and mesh.	89
6.59	Net entrainment test case: Comparison of simulations (solid lines) and measured (squares) suspended sediment concentrations.	90
6.60	Net deposition test case: Schematic definition of the problem.	90
6.61	Net deposition test case: Geometry, boundary conditions, time-step, and mesh.	90
6.62	Net deposition test case: Comparison of simulations (solid lines) and measured (squares) suspended sediment concentrations.	91
6.63	Wall Jet Scour test case: Schematic definition of the problem.	92
6.64	Wall Jet Scour test case: Geometry, boundary conditions, time-step, and number of elements in the unstructured mesh.	92
6.65	Wall Jet Scour test case: Comparison of simulations (solid lines) and measured (circles) scour profile at $t = 60s$	93
6.66	Wall Jet Scour test case: Scour profile evolution in different time-steps.	94
6.67	Wall Jet Scour test case: Comparison between the solutions for the fixed and the adapted meshes at $t = 60s$	95
6.68	Wall Jet Scour test case: Detail of the initial and final unstructured quadrilateral fixed mesh with 34085 elements and hanging nodes at different times.	95
6.69	Wall Jet Scour test case: Detail of the unstructured quadrilateral adapted meshes at $t=0$, $t=20$, $t=40$ and $t=60$ seconds with respectively 17889, 23853, 24294 and 24300 elements and hanging nodes at different times.	96
6.70	Scour over a pipeline test case: Geometry, boundary conditions, time-step, and number of elements in the unstructured mesh.	96
6.71	Scour over a pipeline test case: Comparison of the simulation (solid lines), with the measured (circles) scour profile at $t = 600s$	97
6.72	Scour over a pipeline test case: Scour profile evolution at different time-steps.	98

6.73	Scour over a pipeline test case: Comparison between the solutions for the fixed and the adapted meshes $t = 600$ s.	99
6.74	Scour over a pipeline test case: Detail of the initial and final unstructured quadrilateral fixed mesh with 85796 elements and hanging nodes at different times.	100
6.75	Scour over a pipeline test case: Detail of the unstructured quadrilateral adapted meshes at $t=0$, $t=200$, $t=400$ and $t=600$ seconds with respectively 18820, 26617, 25471 and 26641 elements and hanging nodes.	101

List of Tables

6.1	Pure convection of a circle: Parameters of the different simulations performed.	52
6.2	Film-like interface: Integration errors for the surface tension force on the studied closed contour with successively refined uniform grids.	61
6.3	Pressure inside a bubble due the surface tension: Data.	62
6.4	2D rising bubble: Data.	70
6.5	2D rising bubble: Comparison of the code features and schemes.	71
6.6	3D rising bubble: Comparison of the code features and schemes [4].	74
6.7	Grace's bubble diagram: Morton (Mo), Eötvös (Eo) and Reynolds (Re) numbers according to the experiment of [5].	77
6.8	Grace's bubble diagram: Data.	82
6.9	Grace's bubble diagram: The terminal bubble shapes compared with the results of SPH [6], FTM [7] and the experimental data [5].	82
6.10	Coalescence of two rising gas bubbles: Data.	88

Chapter 1

Introduction

Computational fluid dynamics (CFD) is a branch of fluid mechanics that uses numerical analysis and data structures to analyze and solve problems that involve fluid flows. One challenging category of problems in flow computation is the modeling of moving boundaries and interfaces. These include fluid-particle, sediment-fluid, fluid-object and fluid-structure interactions, free-surface and two-fluid flows, and flows with moving mechanical components. The interaction of fluid flows with moving boundaries leads to a highly coupled, nonlinear system. Across these interfaces, phases, material properties, and flow features can vary abruptly. The main difficulty in the solution of these problems is that the moving boundary itself must be determined as part of the solution of the system of equations that govern the behavior of the fluid flow. The interfaces move under the influence of the flow field and, in turn, affect its behavior. Therefore, not only the transport of momentum and mass play significant roles in defining the behavior of the system, but also the formation, evolution, and dynamics of the interface [1].

Some examples of application of moving boundaries in engineering and science problems are the simulation of materials processing, as ironmaking and steelmaking, evaporation of metals, electromagnetic levitation, and high-pressure gas atomization of liquid metals [8]; fluid-structure interaction, as stability analysis of airplane wings, turbomachinery design, design of bridges, risers, ships, and the flow of blood through arteries [9]; oil recovery [10], and so on.

Several techniques exist for tracking the moving interfaces, each with its strengths and weaknesses. These techniques may be classified under two main categories: *interface tracking* or predominantly Lagrangian methods, and *interface capturing* or Eulerian methods [11]. In the Lagrangian methods, the grid is configured to conform to the shape of the interface, and thus it adapts continually to it. The Lagrangian methods are preferred for problems that have only small fluid motion because they can lead to severely entangled meshes in flow problems with large amplitude motions, resulting in the failure of the algorithms or gross inaccuracies in the results. On the other hand, the Eulerian methods usually employ a fixed grid, and the interface is not explicitly tracked but is reconstructed

from the properties of appropriate field variables. However, if the discontinuity across the interface is to be maintained with fidelity, and if interfacial behavior is the focus, Lagrangian methods hold an advantage.

Therefore, in an attempt to combine the advantages of the Eulerian and Lagrangian descriptions while minimizing their respective drawbacks as much as possible, the Arbitrary Lagrangian Eulerian (ALE) framework was introduced [12, 13]. This approach is based on an arbitrary motion of the reference frame, which is continuously rezoned to allow a precise representation of the moving interfaces. Another alternative that not only retains the advantage of structured Cartesian grids but also provides the ability to handle complex geometries is the Immersed-Boundary method [14, 15], where complex geometries within a Cartesian grid can be replaced by generating an external force field that acts on the fluid in the same manner as a solid boundary would.

In this work, we are interested in two categories of problems involving moving interfaces: two-fluid flows and sediment-fluid interaction. The first one is important to reproduce free-surface problems and the interaction between two different fluids, as water and air, oil and gases, etc. The second is essential to understand the transport of sediment in fluid flows, where the sediment bottom deforms as the flow evolves, i.e., the particles that make up the soil are dragged because of the flow and the morphology of the sediment bottom changes. Moreover, both situations may occur in one single problem, where there is a free-surface flow transporting sediment.

Since we are working with distinct category of problems, we use different solutions to model the interfaces. We use an interface-capturing method called convected level-set to represent the interface between two-fluid problems, and we use an ALE framework to represent moving sediment bottoms.

Among the several engineering problems involving two-fluid flows, the study of bubbles motion is a good option to validate the numerical implementation of a moving boundary, since there is a lot of experimental and numerical data for comparison. These problems are of fundamental importance in many physical, chemical and biological processes, such as boiling [16], cloud cavitation in hydraulic systems [17], bubble columns and centrifuges in the petrochemical industry [18], bubble nucleation and cooperativity in DNA melting [19]. The study of bubbles is also important to understand natural phenomena as the sound propagation in the ocean [20], the exchange of gases and heat between the oceans and the atmosphere [20], and explosive volcanic eruptions [21].

On the other hand, the natural transport of sediments is one of the most common processes in human environments. Sedimentation occurs in rivers, lakes, seas, and also in the atmosphere in the form of dust, smoke, or chemical pollutants. Sediment transport can change the course of a river due to erosion and deposition, can generate scour around piles [22], siltation in dams [23], and contribute to the formation of geological features [24]. All these phenomena cause changes at the sediment bottom.

Although physical modeling of sediment transport in fluid flows in the field or the laboratory may produce reliable results, scaling problems and the costs to conduct tests can be very high. Thus, numerical simulations offer an interesting alternative, even though the prediction of this phenomenon is still challenging due to its complexity. First of all, the flow field itself will be a turbulent flow, often three-dimensional, with a free-surface between water and air, not simply to simulate. Secondly, the transport of sediments is a very complex phenomenon that depends on empirical formulations, as the processes of sediment erosion, that includes the entrainment rate of sediments into suspension, and the bed-load transport where the particles are moved near the bottom. Also, the suspended load and deposition depend on a range of parameters. Besides, we have the coupling of flow with sediment transport. The geometry of the flow field will change with the sediment transport processes, and the suspended sediment may also influence its properties.

Reproducing two-fluid flows is also complex. The surface tension has an important role in such problems and has to be modeled as well. Also, the simulations that involve surface tension need to be three-dimensional because this force acts in all directions. Therefore, reproducing such problems is not trivial, and the moving surface techniques are essential in these cases.

1.1 Objectives and methodology

The main objective of this work is to simulate moving interfaces in fluid flows: by using the convected-level set method to represent two-fluid flows and a moving mesh technique to represent the morphological changes between sediment and fluid. Therefore, we work with two different problems. We apply the convected level-set to simulate gas bubbles motion in viscous fluids, and the ALE formulation is used to simulate sediment transport in fluid flows problems.

The numerical models for simulating these problems are implemented in `libMesh`, an open-source library that provides a platform for parallel, adaptive, multiphysics finite element simulations, considering adaptive mesh refinement [25]. The Navier-Stokes equations that govern the fluid are treated with the Residual-Based Variational Multiscale method (RBVMS). The convected level-set is solved by using a stabilized method with a discontinuity capturing operator and with a new truncated signed distance function, introduced in this work, to get a smooth truncation far from the interface. We use the Continuum Surface Force model to represent the surface tension between fluids, and we apply a global mass conservation procedure to enforce the mass balance since the convected level-set method does not guarantee mass conservation between phases. In the sediment transport problems, an advection-diffusion equation representing particle concentration is solved using the RBVMS formulation as well. Empirical models are used to represent the entrainment rate of sediments into suspension as well as the bed-load transport. Mor-

phological changes are applied as mesh movement, with the resulting scheme described within the ALE framework. Moreover, we consider adaptive mesh refinement based on the flux jump of the level-set and velocity errors to save computational effort, especially in 3D problems.

1.2 Background

We begin by reviewing the concepts used in two-fluid and sediment-fluid problems. First, we describe the techniques for modeling the moving boundaries. Then, we present the mathematical framework of the flow model. Finally, we present aspects of the modeling and simulation of sediment transport problems.

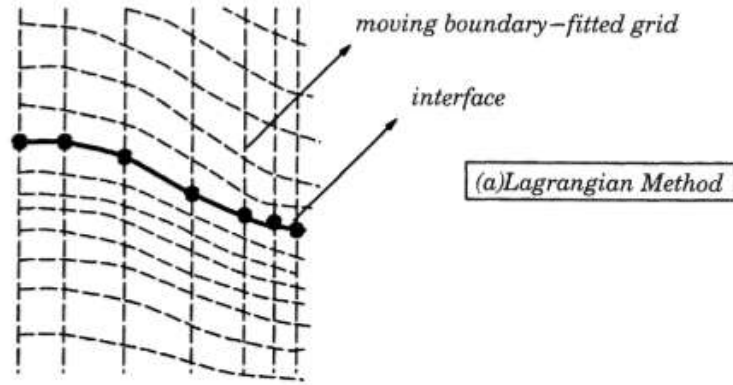
1.2.1 Moving boundaries

The techniques for tracking the moving interfaces may be classified under two main categories, interface tracking and interface capturing methods. The main difference can be located in the viewpoint, which can be either Eulerian or Lagrangian. The main features of the two categories are shown in Fig. 1.1.

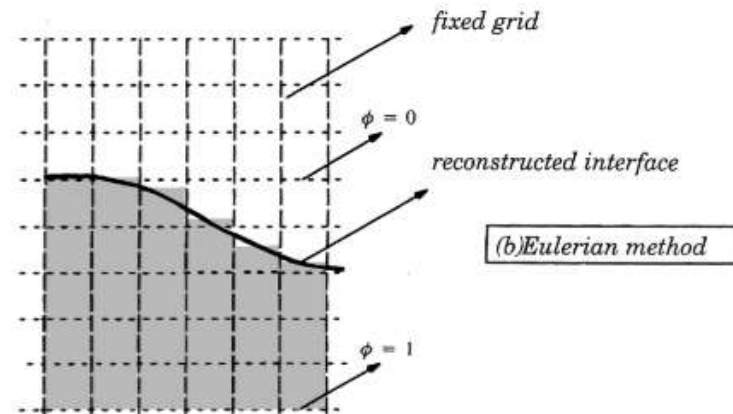
Interface capturing approaches are based on an Eulerian description and widely used in deformable domain problems. They define the interface implicitly on a fixed mesh, in which an appropriate scalar field is used to identify the two phases as well as the interface between them. Depending on the different methods, this scalar field may be described, for example, by a discontinuous Heaviside function or a signed distance function. In order to account for the interface motion, a standard convection equation is solved considering the velocity field. A significant advantage of the interface capturing approaches is that they are inherently able to account for topological changes of the interface. This allows for a much more flexible interface description than in interface tracking approaches. However, the difficulties are in the aspects of the discontinuity treatment across the interface, mass conservation, and the application of boundary conditions along the interface.

In interface tracking approaches, the interface is described explicitly on an interface conforming mesh. The idea is to track the position of the mesh nodes in a Lagrangian fashion. We can distinguish between fully Lagrangian approaches, where all mesh nodes are treated in a Lagrangian fashion, and ALE approaches, where the Lagrangian treatment is only applied to a portion of the mesh nodes, usually those along the interface. Interface tracking approaches offer an accurate and computationally efficient approximation of the interface. Furthermore, the imposition of boundary conditions at the interface is simple, with mesh nodes lying on the interface itself. However, the mesh quality will usually degrade significantly in the course of large deformations.

Various techniques exist to perform the simulation of moving boundaries, and an ex-



(a) Purely Lagrangian method with a moving boundary conforming grid.



(b) Fixed grid Eulerian method with a phase fraction (ϕ) definition of the interface.

Figure 1.1: Comparison of Lagrangian and Eulerian methods for tracking moving boundaries [1].

cellent summary of those methods can be found in a review of ELGETI e SAUERLAND [26]. The most common methods are described in the following.

Volume of Fluid (VOF) method

The Volume of Fluid method, introduced by HIRT e NICHOLS [27], makes use of a volume fraction function defined as the volume ratio of one phase to the total volume of the cell, on an Eulerian fixed grid. One phase is assigned with the fraction variable equal to 1, and the other one equal to 0. The interface is located in the cells with values between 0 and 1. The volume fraction is then advected with the local flow velocity.

One advantage of the VOF method is that as volumetric data is used to store interface location, conservation of volume is guaranteed (assuming incompressible fluids). However, the disadvantages include inaccuracy to evaluate the normals and curvature from the volume fraction function and abrupt change of the discontinuous physical quantities near the interface.

Immersed-boundary method (IBM)

The immersed-boundary method was introduced by [14] to study flow patterns around heart valves and has evolved into a generally useful method for problems of fluid-structure interaction. The mathematical formulation employs a mixture of two coordinate systems: a Eulerian fixed Cartesian mesh for the fluid, and a Lagrangian mesh for the boundaries, generally non-stationary and unstructured. Both coordinate systems are linked by interaction equations that involve a smoothed approximation to the Dirac delta function.

This method solves the fluid equations with an additional term, which represents the effects of the immersed boundary acting on the fluid motion. The imposition of the boundary conditions is not straightforward, and the ramifications of the boundary treatment on the accuracy and conservation properties of the numerical scheme are not obvious. One of the advantages of using IBM is that mesh does not need to be updated.

Phase-Field methods

The phase-field methods are based on models of energy by expressing the free energy of the system as a Cahn-Hilliard (or Allen-Cahn) functional. They use an auxiliary field that takes the role of an order parameter. The order parameter, or the phase-field, takes two distinct values (for instance, +1 and -1) in each phase, with a smooth change between both values in the zone around the interface, which is represented by thin but nonzero thickness regions. A discrete location of the interface may be defined as the collection of all points where the phase-field takes a particular value (e.g., 0). However, the phase-field equations never require the knowledge of the exact interface location. A phase-field model is usually constructed in such a way that in the limit of an infinitesimal interface width (the so-called sharp interface limit), the correct interfacial dynamics are recovered. This approach permits to solve the problem by integrating a set of partial differential equations for the whole system, thus avoiding the explicit treatment of the boundary conditions at the interface.

An advantage of phase-field methods is the ability to compute geometric quantities easily. However, since the phase function changes quickly near the interface, a large number of grid points is necessary to solve this region well. An adaptative mesh refinement method would be appropriate.

Interface tracking approach

In interface tracking methods, the boundaries are explicitly resolved by the computational mesh that follows the interface's motion. The necessity of accurately following the deforming interfaces requires some degree of Lagrangian description in the formulation. However, the fully Lagrangian approach may result in significant mesh distortion

or invalid meshes. Then, methods capable of combining the Lagrangian and Eulerian approaches in the same domain, as the ALE framework, were developed.

The basic idea of ALE is to work with a deforming mesh, as in the Lagrangian method, but to decouple the mesh deformation from the displacement of the fluid particles, which means that the mesh velocity is no longer equal to the fluid velocity, and can be chosen arbitrarily. The usual choice in ALE is to track boundary and interface nodes in a Lagrangian manner, while all other mesh nodes are adjusted solely to preserve mesh quality as best as possible. The ALE approach requires that the mesh velocity enters the momentum equation explicitly, thus modifying the original flow equations, precisely the convection term.

To track the sediment bottom motion in sediment transport problems, we use an interface-tracking method, as in [2, 28–39]. Our mesh represents only the fluid with the suspended sediments, while a boundary of our domain represents the sediment bottom. In this boundary, we represent the morphological changes as mesh movement by using an ALE framework, as presented in [40] and [37]. By applying this method in sediment transport problems, we may face unrealistic bed shapes that might cause numerical instabilities. Different smoothing techniques are used to avoid these problems, and we introduce a new smoothing procedure called *upwind sand-slide*.

Level-set method

The level-set method is an interface capturing strategy that mitigates difficulties associated with a discontinuous volume-fraction function, as in the VOF method, by using a smooth signed distance function instead. The original level-set formulation is divided into two steps: first, a convection equation is solved to calculate the evolution of the interface, and after, a reinitialization step is solved to keep the level-set function a signed distance function from the interface. More details will be addressed along this work.

To track the interface motion of two-fluid problems, we implement a modified level-set method called *convected level-set* [41, 42]. The convected level-set associates both reinitialization and convection steps by including the reinitialization in the convection equation. The original level-set formulation also makes use of a signed distance function defined as the Euclidian distance to the interface. In this work, we introduce a new truncated signed distance function based on [43]. The modified signed distance function concentrates the iso-surfaces near the interface and provides a smooth truncation far from there. We also apply a global mass conservation procedure to enforce the mass balance, since the principal drawback of the level-set method is the lack of conservation of mass.

1.2.2 Flow model

The governing equations of the flow model are the well-known Navier-Stokes equations. The Navier-Stokes equations, along with proper boundary and initial conditions, contain all information necessary to model fluid flows, including turbulent effects. It is possible to approximate them numerically using a computational grid fine enough to represent all relevant scales of the studied problem. However, the computational effort tends to limit the application of this method only to low Reynolds numbers. Thus, direct numerical simulation (DNS) is still a challenge, particularly at high Reynolds numbers.

An alternative approach to DNS consists of Large Eddy simulations (LES). In this method, only large flow structures are solved, while the smaller ones, called subgrid components, are modeled [44]. Inspired on LES models, the variational multiscale methods (VMS) were introduced by Hughes *et al.* in [45] as a general technique to model the subgrid scales in the numerical solution of partial differential equations. Here, we use the Residual-Based Variational Multiscale method (RBVMS), in which the subgrid components are modeled in terms of the residual of the large scales, and they are introduced in the resolved scale equations. This approach was introduced in Calo [46], and it has been applied successfully to many problems, with several discretization methods.

The RBVMS method does not include an eddy viscosity, which appears as the numerical diffusion inherent to the model. Thus, it does not need any modeling of the subgrid scales by statistical theories of turbulence. From this method, it is also possible to derive stabilized methods as the streamline upwind/Petrov-Galerkin (SUPG), the least-squares on the incompressibility constraint (LSIC) method and the pressure stabilizing/Petrov-Galerkin (PSPG) method [11, 47, 48]. It is known that the Galerkin formulation of the finite element method performs poorly when the advection or reaction terms of a partial differential equation dominate the problem [13]. Thus, these stabilization terms are important, and they come naturally from the RBVMS framework. Detailed reviews of the RBVMS formulation can be found in [47, 49–51].

1.2.3 Sediment transport

Sediment transport can be divided into two phenomena: suspended load and bed-load transport. The suspended load involves the finer sediment maintained in suspension by turbulence, whereas, bed-load transport consists of coarse particles transported along the bed. The majority of the numerical studies divide the sediment transport in fluid flows into bed-load and suspended load components [2, 28–39, 52–59].

When the shear rate at the bottom is strong enough, sediments may entrain into suspension as well. Thus, sediments are lifted from the bed-load zone and, once out of the bottom layer, start being governed by the suspended load. Furthermore, suspended grains can also be deposited due to their settling velocity and gravitational acceleration. All

these phenomena are schematically represented in Figure 1.2.

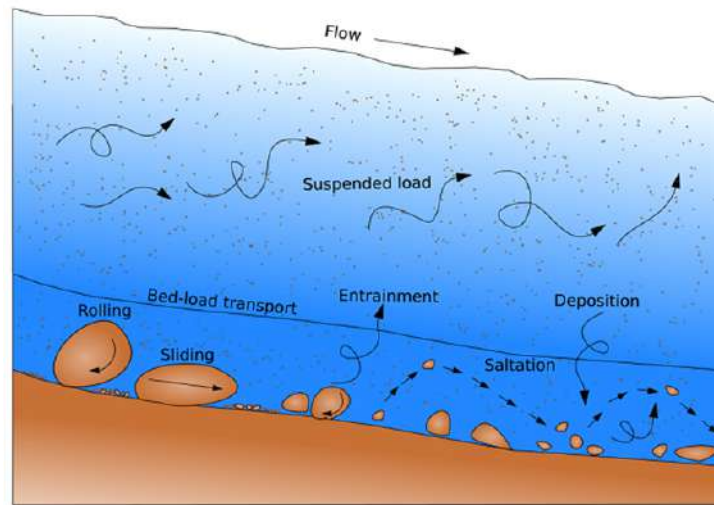


Figure 1.2: Types of sediment transport

The coupling between sediment transport and fluid also has some considerations. A sediment flow mixture can be viewed as a very dilute suspension when the volumetric concentration (or simply concentration) is less than 0.001, such that particle interactions can be neglected [60]. Most sediment transport models are based on this simple assumption. However, this “passive scalar hypothesis” is valid only under very restrictive assumptions: the particle diameter must be small in comparison with the smallest dynamic length-scale of the turbulence (the Kolgomorov length-scale), and the time response of the particle must be smaller than the dissipation timescale [61]. By exceeding a certain particle concentration in the fluid, the flow behavior turns non-Newtonian. This characteristic may be considered in the fluid simulation solver by a constitutive model [62], and it has a great influence on fluid behavior [63]. An intermediate assumption between the “passive scalar hypothesis” and the non-Newtonian fluid is the consideration of the concentration of the sediment particles in the Navier-Stokes equations, by applying it as a forcing term [64]. The flow behavior is affected, but not enough to be considered as a non-Newtonian fluid. In this work, we consider only small particle concentrations, and we use the “passive scalar hypothesis” in which the sediment concentration does not affect the flow behavior.

1.3 Literature review

One reason for studying bubbles motion to validate the two-fluid flow model is the fact that there are lots of experimental and numerical studies about this subject. The observations of BHAGA e WEBER [5] and LIU *et al.* [65] showed the characteristics of a single rising bubble, in terms of the bubble shape and terminal velocity. GRACE [66] studied a large number of experimental data on the shapes and velocities of bubbles rising

in quiescent viscous liquids and condensed the data into one diagram. The interaction between multiple bubbles rising in different liquids was investigated by BRERETON e KOROTNEY [3] and MANGA e STONE [67].

In numerical studies, different approaches have been used to model bubble problems, such as the VOF method [68, 69], the Front Tracking Method (FTM) [7, 70, 71], the Lattice Boltzmann Method (LBM) [72–75], the Smoothed Particle Hydrodynamics (SPH) [6, 76, 77], the Phase-Field method [78] and the Level-Set method [43, 79, 80].

Different cases have been investigated about sediment transport in fluid flows as well, as scour around offshore pipelines [28–34, 52, 53], scour around vertical piles [35, 36, 54, 55, 81], scour due a submerged wall jet [37, 56], sediment transport in curved channels with mobile beds [38, 39, 82], sediment transport and morphology of ripples [2, 57–59].

To reproduce such problems numerically, some authors have used the VOF method to model the interface between water-air, while used a Lagrangian method to model the interface between sediment-water [32, 37, 39]. AHMAD *et al.* [54], KRAFT *et al.* [57], QUEZADA *et al.* [53] and AFZAL *et al.* [55] have used the level-set method for the sediment-water tracking. On the other hand, KHOSRONEJAD *et al.* [82], LEFTHERIOTIS e DIMAS [58] and KHOSRONEJAD e SOTIROPOULOS [59] employed the immersed boundary method, and ZHAO e FERNANDO [52] and HOSSEIN KAZEMINEZHAD *et al.* [81] employed an Eulerian two-phase model for this same purpose.

The suspended load in sediment transport problems is often solved by an advection-diffusion equation and the entrainment and bed-load transport need empirical models to be solved. Two entrainment models commonly used can be found in [83] and [84]. The bed-load transport rate models are usually related to the bed shear stress and some of them can be found in [85], [86], [87].

The majority of works about sediment transport in fluids use the Reynolds-Averaged Navier-Stokes method (RANS) to model the fluid flow, using different approaches for the turbulence closure model [2, 28–39, 52–59, 81, 82]. In this work, we treat both, the Navier-Stokes equations and the suspended load transport equation, with the RBVMS formulation.

1.4 Structure of the thesis

The remainder of this work is organized as follows. In chapter 2, we present the interface tracking methods used in this work, with a description of the convected level-set and moving mesh methods. Then, in chapter 3, we present the mathematical setting for the numerical simulation of the flow, the corresponding RBVMS finite element formulation, and some relevant aspects about flow boundary conditions. After that, we show the sediment transport governing equations, its corresponding finite element formulation, the sediment boundary conditions, and the upwind sand-slide, used to smooth the mesh

movement. In chapter 5, we present some aspects of the implementation and the libraries used in this work. Then, we show numerical examples. The thesis ends with a summary of our main findings and the perspectives for the next steps of this work.

Chapter 2

Interface tracking and interface capturing methods

In this chapter, we present both methods used in this work to represent moving surfaces. First, we introduce the governing equations of the convected level-set formulation for two-fluid problems. Then, we present the ALE framework to solve sediment-fluid problems.

2.1 Convected Level-Set formulation

The level-set method was first introduced by OSHER e SETHIAN [88] in the late 1980s as a technique for capturing evolving interfaces and tracking the propagation of fronts. The method consists in to separate two phases with signed distance functions (SDFs), in which the interface between phases is defined as the zero level-set.

The original level-set function determines the smallest distance of a point to the interface, and the signal is used to represent the different phases (positive values represent one phase, while negative values the other one). Equation (2.1) defines the standard level-set function α as,

$$\alpha(\mathbf{x}) = \begin{cases} d(\mathbf{x}, \Gamma) & \text{for } \mathbf{x} \in \Omega^+ \\ 0 & \text{for } \mathbf{x} \in \Gamma \\ -d(\mathbf{x}, \Gamma) & \text{for } \mathbf{x} \in \Omega^- \end{cases} \quad (2.1)$$

where $d(\cdot, \cdot)$ is the Euclidian distance to Γ , Ω^+ and Ω^- are the subdomains corresponding to each phase, and Γ is the interface between the subdomains. Figure 2.1 exemplifies how two phases separated by a circular interface are modeled by using the standard level-set formulation.

The idea of the level-set method is to move the interface by solving a pure convection equation and, after that, to enforce the SDF properties by applying a reinitialization step.

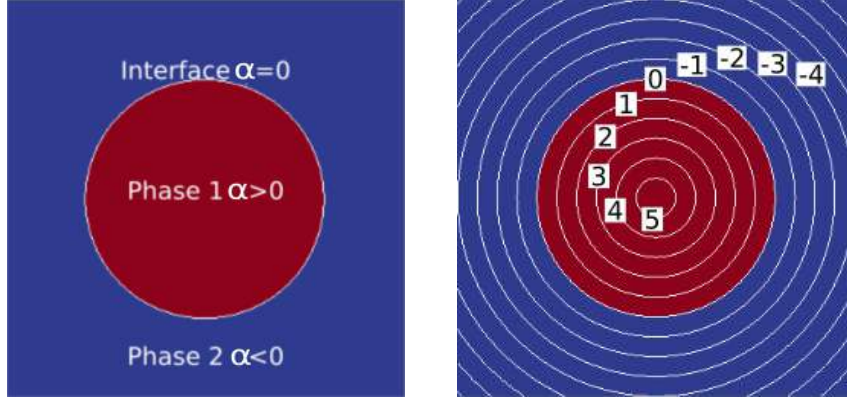


Figure 2.1: Distinguishing two phases separated by a circular interface with the standard level-set method

As the level-set function is transported, it may lose its signed distance properties, and the reinitialization recovers it by fixing the position of the iso-surfaces. In this work, we implement a modified level-set method that associates both reinitialization and convection steps. The method is called convected level-set [41, 42]. The inclusion of the reinitialization in the convection equation avoids the extra step that appears in the original level-set formulation.

In the following, we assume incompressible flows, that are flows in which the material density is constant within a fluid parcel, i.e., an infinitesimal volume that moves with the flow velocity. Incompressible flows also consider the divergence of the flow velocity equals to zero.

2.1.1 Governing equations

To follow the evolution of the level-set function, a pure convection problem is solved. Assuming that a divergence-free velocity field, \mathbf{u} , is known everywhere in the domain Ω , the interface motion is given by the following pure convection equation,

$$\begin{aligned} \frac{\partial \alpha}{\partial t} + \mathbf{u} \cdot \nabla \alpha &= 0 \\ \alpha(t = 0, \mathbf{x}) &= \alpha_0(\mathbf{x}). \end{aligned} \quad (2.2)$$

The gradient of the standard level-set function, as a SDF, remains constant along the domain, i.e. it satisfies the Eikonal equation $\|\nabla \alpha\| = 1$. However, when the interface is convected by a given velocity, it loses its distance property and needs to be reinitialized. Classically, the reinitialization is obtained by solving a Hamilton-Jacobi equation that reconstructs the level-set with the exact zero iso-value of $\alpha(\mathbf{x}, t)$. Introducing a virtual time τ , we want to keep $\|\nabla \alpha\|$ close to a given function S , solving the following equation,

$$\begin{aligned}\frac{\partial \alpha}{\partial \tau} + \text{sgn}(\alpha)(\|\nabla \alpha\| - S) &= 0 \\ \alpha(\tau = 0, x) &= \alpha(x, t)\end{aligned}\tag{2.3}$$

with $S = 1$ and $\text{sgn}(\alpha)$ being the sign function defined as,

$$\text{sgn}(\alpha) = \begin{cases} 1 & \text{for } \alpha > 0 \\ 0 & \text{for } \alpha = 0 \\ -1 & \text{for } \alpha < 0. \end{cases}\tag{2.4}$$

Note that without the reinitialization, the level-set equation (2.2) is almost the same of the VOF method [89].

The level-set method can be generalized to any SDF having the same independent measure properties. The only useful information is the exact zero iso-value position. Thus, the level-set function does not need to be computed in the whole domain. We may cut off the level-set function at a thickness of E , filtering the level-set function close to the interface. Here, we introduce a new modified SDF (ϕ), inspired in [43], given by,

$$\phi = \frac{1}{1 + e^{\frac{-\alpha}{E}}} - 0.5.\tag{2.5}$$

Figure 2.2 shows the difference between the standard SDF and the new modified one presented in Eq. (2.5). While in the standard level-set method $\alpha \in [-\infty, \infty]$, in our modified SDF $\phi \in [-0.5, 0.5]$. The new modified SDF, ϕ , gives a form to define a smooth truncation for the standard level-set function and can be generalized to any number of spatial dimensions.

We compute the derivative of ϕ with respect to the distance function,

$$\nabla \phi = \nabla \alpha \frac{e^{\frac{-\alpha}{E}}}{E(1 + e^{\frac{-\alpha}{E}})^2}.\tag{2.6}$$

The truncated level-set function now verifies the following property,

$$S = \|\nabla \phi\| = \frac{1}{4E} - \frac{\phi^2}{E}.\tag{2.7}$$

Rewriting equations (2.2) and (2.3), the pure convection and reinitialization equations for the modified SDF ϕ are,

$$\begin{aligned}\frac{\partial \phi}{\partial t} + \mathbf{u} \cdot \nabla \phi &= 0 \\ \phi(t = 0, \mathbf{x}) &= \phi_0(\mathbf{x})\end{aligned}\tag{2.8}$$

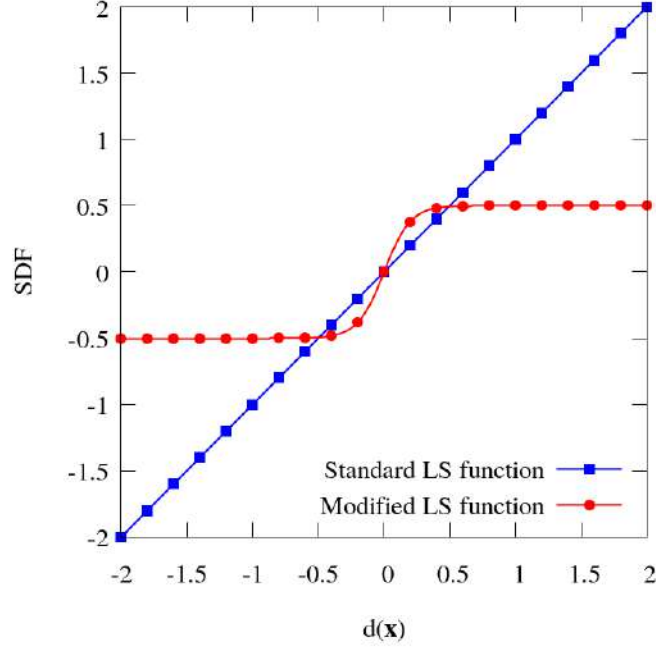


Figure 2.2: Comparison between the standard and modified SDFs with $E = 0.1$.

$$\frac{\partial \phi}{\partial \tau} + \text{sgn}(\phi) \left(\|\nabla \phi\| - \left(\frac{1}{4E} - \frac{\phi^2}{E} \right) \right) = 0 \quad (2.9)$$

$$\phi(\tau = 0, x) = \phi(x, t).$$

Here, we smooth the sign function using a equation similar to Eq. (2.5),

$$\text{sgn}(\phi) = \frac{2}{1 + e^{-\frac{\phi}{E}}} - 1. \quad (2.10)$$

Note that $\text{sgn}(\phi) = 0$ where $\phi = 0$, thus this reinitialization step does not influence the interface position.

It is possible to avoid the reinitialization step using the called convected reinitialization [41, 42, 90]. Once the Hamilton-Jacobi reinitialization is combined with the convection equation, it can be solved using,

$$\frac{\partial \phi}{\partial t} + \mathbf{u} \cdot \nabla \phi + \lambda \text{sgn}(\phi) (\|\nabla \phi\| - S) = 0 \quad (2.11)$$

$$\phi(t = 0, \mathbf{x}) = \phi_0(\mathbf{x})$$

where λ is a penalty constant. The penalty coefficient λ defines the contribution of the reinitialization equation in the convection equation. A small λ may not be enough to correct the iso-surfaces and to recover the signed distance properties adequately, while a

large one may change the interface shape.

Now, introducing the convection velocity coming from the Hamilton-Jacobi equation, we can rewrite Eq. (2.11) as,

$$\begin{aligned}\frac{\partial \phi}{\partial t} + (\mathbf{u} + \lambda \mathbf{U}) \cdot \nabla \phi - \lambda \operatorname{sgn}(\phi) S &= 0 \\ \phi(t = 0, \mathbf{x}) &= \phi_0(\mathbf{x})\end{aligned}\tag{2.12}$$

where $\mathbf{U} = \operatorname{sgn}(\phi) \frac{\nabla \phi}{\|\nabla \phi\|}$.

The final model is given by,

$$\begin{aligned}\frac{\partial \phi}{\partial t} + (\mathbf{u} + \lambda \mathbf{U}) \cdot \nabla \phi - \lambda \operatorname{sgn}(\phi) \left(\frac{1}{4E} - \frac{\phi^2}{E} \right) &= 0 \\ \phi(t = 0, \mathbf{x}) &= \phi_0(\mathbf{x}).\end{aligned}\tag{2.13}$$

2.1.2 Time integration

Equation (2.13) can be integrated in time with many different time integration methods [91]. The Backward Euler method is widely applied for two-fluid models because of its unconditional numerical stability characteristics. However, it has the disadvantage of being only first-order accurate, which introduces a significant amount of numerical diffusion. Then, we use the second-order Backward Differentiation Formula (BDF2), that, compared to the prevailing Backward Euler method, has a significantly better accuracy (second order) while retaining the important property of unconditional linear stability. The one-equation model becomes,

$$\begin{aligned}\frac{1.5\phi_{n+1} - 2\phi_n + 0.5\phi_{n-1}}{\Delta t} + (\mathbf{u}_{n+1} + \lambda \mathbf{U}_n) \cdot \nabla \phi_{n+1} \\ - \lambda \operatorname{sgn}(\phi_n) \left(\frac{1}{4E} - \frac{\phi_n^2}{E} \right) &= 0.\end{aligned}\tag{2.14}$$

The subscript $n + 1$ is associated to $t = t_{n+1}$ and n , and $n - 1$ to the previous time-steps. It is important to remember that Eq. (2.13) is non-linear, because \mathbf{U} depends on ϕ . Thus we linearize it by evaluating \mathbf{U} with the value of ϕ at the previous time-step (or time-lagging). The slope S is also evaluated at the previous time-step.

2.1.3 Finite element formulation

We introduce a stabilized finite element variational formulation to discretize in space the present model. We assume the weight space S_ϕ^h for ϕ^h along with its trial space

counterpart V_ϕ^h and the weight function ω^h , where the superscript h reminds us of the underlying finite element mesh on which the basis functions are defined, such that,

$$S_\phi^h = \{\phi^h(\cdot, t) \in H^1(\Omega) \mid \phi^h = \phi_D \text{ on } \Gamma_D\} \quad (2.15)$$

$$V_\phi^h = \{\omega^h(\cdot, t) \in H^1(\Omega) \mid \omega^h = 0 \text{ on } \Gamma_D\} \quad (2.16)$$

in which ϕ_D is a prescribed SDF value on the Dirichlet boundary Γ_D .

The weak formulation is then: find $\phi^h \in S_\phi^h$ such that $\forall \omega^h \in V_\phi^h$,

$$\begin{aligned} & \left(\frac{1.5\phi_{n+1}^h - 2\phi_n^h + 0.5\phi_{n-1}^h}{\Delta t}, \omega^h \right)_\Omega + ((\mathbf{u}_{n+1}^h + \lambda \mathbf{U}_n^h) \cdot \nabla \phi_{n+1}^h, \omega^h)_\Omega \\ & - \left(\lambda \operatorname{sgn}(\phi_n^h) \left(\frac{1}{4E} - \frac{\phi_n^{h2}}{E} \right), \omega^h \right)_\Omega + \sum_e (\mathcal{R}_\phi, \tau_\phi (\mathbf{u}_{n+1}^h + \lambda \mathbf{U}_n^h) \cdot \nabla \omega^h)_{\Omega_e} = 0. \end{aligned} \quad (2.17)$$

Here we define the operation $(\cdot, \cdot)_\Omega$ as the standard scalar product in $L^2(\Omega)$. In Eq. (2.1.3), \mathcal{R}_ϕ is the residual of the convection equation (2.13), and τ_ϕ is the stabilization parameter of the SUPG formulation [92], that are,

$$\mathcal{R}_\phi = \frac{1.5\phi_{n+1}^h - 2\phi_n^h + 0.5\phi_{n-1}^h}{\Delta t} + (\mathbf{u}_{n+1}^h + \lambda \mathbf{U}_n^h) \cdot \nabla \phi_{n+1}^h - \lambda \operatorname{sgn}(\phi_n^h) \left(\frac{1}{4E} - \frac{\phi_n^{h2}}{E} \right) \quad (2.18)$$

$$\tau_\phi = \frac{1}{n_{en} \|(\mathbf{u}_{n+1}^h + \lambda \mathbf{U}_n^h) \cdot \nabla \omega^h\|} \approx \frac{h_e}{n_{en} \|(\mathbf{u}_{n+1}^h + \lambda \mathbf{U}_n^h)\|} \quad (2.19)$$

with n_{en} being the number of nodes per element and h_e the mesh size.

In addition, a small stabilizing diffusion term κ_d , based on the $YZ\beta$ discontinuity-capturing operator [93], is added to improve stability. It is defined as,

$$\kappa_d = |Y^{-1}Z| \left(\sum_{i=1}^{n_{dim}} \left| Y^{-1} \frac{\partial \phi^h}{\partial x_i} \right| \right)^{\beta^s / 2 - 1} h_e^{\beta^s} \quad (2.20)$$

where

$$Y = \bar{\phi} \quad (2.21)$$

$$Z = \mathcal{R}_\phi. \quad (2.22)$$

The parameter β^s in Eq. (2.20) influences the smoothness of the layer and the parameter $\bar{\phi}$ is a reference value ($\bar{\phi} = 1$). For smooth layers $\beta^s = 1$, and this value is adopted

here. Note that if $\beta^s = 1$ and the reference value $\bar{\phi} = 1$, the discontinuity-capturing term is equivalent to the Consistent Approximated Upwind method [94].

The final stabilized finite element variational formulation of the convected level-set method is,

$$\begin{aligned}
& \left(\frac{1.5\phi_{n+1}^h - 2\phi_n^h + 0.5\phi_{n-1}^h}{\Delta t}, \omega^h \right)_{\Omega} + ((\mathbf{u}_{n+1}^h + \lambda \mathbf{U}_n^h) \cdot \nabla \phi_{n+1}^h, \omega^h)_{\Omega} \\
& - \left(\lambda \operatorname{sgn}(\phi_n^h) \left(\frac{1}{4E} - \frac{\phi_n^{h2}}{E} \right), \omega^h \right)_{\Omega} + (\nabla \omega^h, \kappa_d \nabla \phi_{n+1}^h) \\
& + \sum_e (\mathcal{R}_{\phi}, \tau_{\phi}(\mathbf{u}_{n+1}^h + \lambda \mathbf{U}_n^h) \cdot \nabla \omega^h)_{\Omega_e} = 0.
\end{aligned} \tag{2.23}$$

2.1.4 Global mass conservation

Due to the incompressibility assumption, the area of the region occupied by each fluid must be conserved during the whole computational process. However, the convected level-set method cannot guarantee mass conservation. Numerical errors may decrease/increase the area/volume of one of the fluids about several percent after many time-steps. That is why some authors combine the level-set method with some enforcement of mass balance (see [95] and [96]).

Mass conservation means that the current area/volume of one of the phases S^+ is equal to its initial area/volume S_0^+ . The initial conditions define S_0^+ . If S^+ is different from S_0^+ , we need to introduce a correction. It is necessary only to correct one of the domains since their union is constant. However, we do not want to change the shape of the interface between the two phases. Therefore, we change the zero level-set, accepting as a new zero level-set some near isoline, since it has almost the same shape. To do that, we move the level-set function upward or downward, by adding to ϕ a global constant perturbation c_{ϕ} , introduced for 2D simulations in [97], and generalized here for any number of spatial dimensions as,

$$c_{\phi} = \frac{S_0^+ - S^+}{L_t(\Gamma)} \tag{2.24}$$

where $L_t(\Gamma)$ is the length/area of the interface Γ at time t .

We can approximate S^+ and $L_t(\Gamma)$ by,

$$S^+ = \int_{\Omega} \mathcal{H}_{\epsilon}(\alpha) d\Omega \tag{2.25}$$

$$L_t = \int_{\Omega} \delta_{\epsilon}(\alpha) d\Omega \tag{2.26}$$

being \mathcal{H}_ϵ the regularized Heaviside function and δ_ϵ the regularized Dirac function. Here, ϵ is also defined as $\epsilon = h_e$. The functions \mathcal{H}_ϵ and δ_ϵ are,

$$\mathcal{H}_\epsilon(\alpha) = \begin{cases} 0 & \text{for } \alpha < -\epsilon \\ \frac{1}{2} \left(1 + \frac{\alpha}{\epsilon} + \frac{1}{\pi} \sin \left(\frac{\pi\alpha}{\epsilon} \right) \right) & \text{for } |\alpha| \leq \epsilon \\ 1 & \text{for } \alpha > \epsilon \end{cases} \quad (2.27)$$

$$\delta_\epsilon(\alpha) = \begin{cases} 0 & \text{for } \alpha > \epsilon \\ \frac{1}{2\epsilon} \left(1 + \cos \left(\frac{\pi\alpha}{\epsilon} \right) \right) & \text{for } |\alpha| \leq \epsilon \\ 0 & \text{for } \alpha < -\epsilon. \end{cases} \quad (2.28)$$

2.2 Moving mesh method

The ALE reference frame is responsible for tracking the sediment bottom displacements in fluid-sediment problems. An efficient mesh deformation scheme is necessary to allow large rotations and motion of the sediment bottom. Among different options available for addressing node repositioning, we apply an adaptive mesh rezoning technique based on [98].

The formal statement of the boundary-value problem is: given $\boldsymbol{\eta}$, the prescribed displacements vector containing the η values, the displacements per node, find the mesh displacement field $\mathbf{d} : \Omega \subset R^{nsd}$ such that,

$$\nabla \cdot ([1 + \tau_{mesh}](\mathbf{e} \otimes \mathbf{e})\nabla)\mathbf{d} = \mathbf{0} \text{ in } \Omega \quad (2.29)$$

$$\mathbf{d} = \boldsymbol{\eta} \text{ on } \Gamma_m \quad (2.30)$$

$$\mathbf{d} = \mathbf{0} \text{ on } \Gamma_f. \quad (2.31)$$

Equation (2.29) is the governing equation while Equations (2.30) and (2.31) are the boundary conditions at the moving and fixed boundaries, Γ_m and Γ_f , respectively. τ_{mesh} is a bounded, non-dimensional function that imposes spatially varying stiffness effect in the computational domain. One simple definition of τ_{mesh} , proposed in MASUD e HUGHES [98] is,

$$\tau_{mesh} = \frac{1 - V_{min}/V_{max}}{V/V_{max}} \quad (2.32)$$

where V , V_{max} and V_{min} represent the volumes of the current, the largest and the smallest elements in a given mesh, respectively. Because of the spatially varying stiffness, the

region with small elements moves almost like a rigid body while the biggest elements absorb the deformation. The tensor $\mathbf{e} \otimes \mathbf{e}$ induces movement in the preferred direction of motion. The original formulation considers an anisotropic diffusivity (that is, $\mathbf{e} \otimes \mathbf{e} = \mathbf{I}$) [98].

Since the displacements resulting from the morphological changes are only in the gravity acceleration direction, we change the Equation (2.29), allowing the displacements of all nodes only in the gravity acceleration direction as well ($\mathbf{e} = \mathbf{e}^g$). This modification avoids bad displacements when adjacent boundary nodes have displacements in opposite directions.

A Galerkin variational formulation is used to solve the mesh boundary value problem, yielding a symmetric set of algebraic equations of the form:

$$\mathbf{K}(\tau_{mesh})\mathbf{d} = \mathbf{r} \quad (2.33)$$

where \mathbf{K} and \mathbf{r} are respectively the global stiffness matrix and the forcing term, which contains the boundary conditions. We solve Equation (2.33) and once we have the displacements, the mesh velocity is computed by,

$$\mathbf{u}_{mesh} = \frac{\mathbf{d}}{\Delta t}. \quad (2.34)$$

Chapter 3

Navier-Stokes equations

In this chapter, the partial differential equations used for the simulation of turbulent fluid flows are presented. First, the well-known Navier-Stokes (N-S) equations are introduced. Then, we present the RBVMS framework, the temporal discretization, and an iterative solution procedure to solve the nonlinear problem. We also present nondimensional equations and numbers often used within the N-S equations. Moreover, at the end of the chapter, we introduce the flow boundary conditions.

3.1 Governing equations

In this work, we are working with two different problems. One is a two-fluid problem, and the other is a single-fluid problem that involves sediment-fluid interaction. The Navier-Stokes equations are different for single and two-fluid flows. Single-fluid flows are simpler to model, and the N-S equations can be applied straightforward. On the other hand, in two-fluid flows, there are other considerations. First, it is necessary to recover information from the convected level-set since it is the SDF that defines where the interface between both fluids is, and it is its signal that defines the properties of each phase. Moreover, there is a force at the interface between fluids called surface tension that have to be added. Therefore, in the following, we present the Navier-Stokes equations for two-fluid flows, and, for single-fluid problems, we simplify it by keeping constant the properties of the fluid in all domain.

The two-fluid flow is governed by the well known Navier-Stokes equations, which leads to the following nonlinear mathematical problem to be solved: let us consider a space-time domain in which the flow takes place along the interval $[0, t_f]$ given by $\Omega \subset R^{n_{sd}}$, where n_{sd} is the number of space dimensions. Let $\bar{\Gamma}$ denote the boundary of Ω . Find the pressure p and the velocity \mathbf{u} satisfying the following equations,

$$\rho \frac{\partial \mathbf{u}}{\partial t} + \rho \mathbf{u} \cdot \nabla \mathbf{u} - \nabla \cdot \boldsymbol{\sigma} - \rho \mathbf{g} - \mathbf{F}_{st} = 0 \text{ in } \Omega \times [0, t_f] \quad (3.1)$$

$$\nabla \cdot \mathbf{u} = 0 \text{ in } \Omega \times [0, t_f] \quad (3.2)$$

where ρ is the density, $\mathbf{g} = g\mathbf{e}^g$ is the gravitational acceleration vector (g is the absolute value and \mathbf{e}^g is the direction vector) and \mathbf{F}_{st} is the surface tension force between phases.

Considering only Newtonian behavior, the Cauchy stress tensor $\boldsymbol{\sigma}$, is given by,

$$\boldsymbol{\sigma}(p, \mathbf{u}) = -p\mathbf{I} + 2\mu\boldsymbol{\epsilon}(\mathbf{u}) \quad (3.3)$$

where \mathbf{I} is the identity tensor, μ is the dynamic viscosity and $\boldsymbol{\epsilon}(\mathbf{u})$ is the strain-rate tensor defined as

$$\boldsymbol{\epsilon}(\mathbf{u}) = \frac{1}{2}(\nabla\mathbf{u} + (\nabla\mathbf{u})^T). \quad (3.4)$$

For a complete problem description, proper boundary conditions on the boundary $\bar{\Gamma} = \bar{\Gamma}_D \cup \bar{\Gamma}_N$ must be prescribed. The boundary conditions can be written as,

$$\mathbf{u} = \mathbf{u}_D \text{ on } \bar{\Gamma}_D \quad (3.5)$$

$$\boldsymbol{\sigma} \cdot \mathbf{n} = \mathbf{t} \text{ on } \bar{\Gamma}_N \quad (3.6)$$

in which \mathbf{u}_D is a prescribed velocity value on $\bar{\Gamma}_D$ and \mathbf{t} is the interaction traction force on $\bar{\Gamma}_N$. Besides, a divergence-free initial condition for the velocity field must be specified.

To represent the heterogeneous flow material properties, such as the density ρ and the dynamic viscosity μ , we introduce to the following mixing laws,

$$\rho = \rho^+ \mathcal{H}_{\text{scaled}}(\alpha) + \rho^-(1 - \mathcal{H}_{\text{scaled}}(\alpha)) \quad (3.7)$$

$$\mu = \mu^+ \mathcal{H}_{\text{scaled}}(\alpha) + \mu^-(1 - \mathcal{H}_{\text{scaled}}(\alpha)) \quad (3.8)$$

where ρ^+ and ρ^- are the densities, and μ^+ and μ^- are the dynamic viscosities of the two different fluids marked with the SDF (positive values represent one phase, while negative values the other one). $\mathcal{H}_{\text{scaled}}$ is a non-symmetrical, smoothed Heaviside function [99], also related to the SDF, and given by,

$$\mathcal{H}_{\text{scaled}}(\alpha) = \begin{cases} 0 & \text{for } \alpha < -\epsilon \\ \frac{1}{2} \left(\frac{1}{2} + \frac{\alpha}{\epsilon} + \frac{\alpha^2}{2\epsilon^2} - \frac{1}{4\pi^2} \left(\cos\left(\frac{2\pi\alpha}{\epsilon}\right) - 1 \right) + \frac{\epsilon+\alpha}{\epsilon\pi} \sin\left(\frac{\pi\alpha}{\epsilon}\right) \right) & \text{for } |\alpha| \leq \epsilon \\ 1 & \text{for } \alpha > \epsilon \end{cases} \quad (3.9)$$

in which ϵ is a thickness related to the mesh size. We define $\epsilon = h_e$ in this work.

The surface tension force, which appears on the interface between two fluids, is applied using the Continuum Surface Force model (CSF) [100], that is,

$$\mathbf{F}_{st} = \sigma_{st} \kappa \mathbf{n} \delta(\phi) \quad (3.10)$$

where σ_{st} is the surface tension coefficient, κ is the curvature, \mathbf{n} is the normal vector to the interface and $\delta(\phi)$ is a thickness, calculated by a Dirac function, where the surface tension is distributed.

An advantage of an SDF is that the normal vector to the interface is readily obtained by,

$$\mathbf{n} = \nabla \phi \quad (3.11)$$

and the curvature ends up being,

$$\kappa(\phi) = \nabla \cdot \frac{\nabla \phi}{\|\nabla \phi\|}. \quad (3.12)$$

The Dirac function $\delta(\phi)$ satisfies $\delta(\phi) = \frac{\partial \mathcal{H}}{\partial \phi}$. Thus, we have,

$$\delta_{scaled}(\alpha) = \begin{cases} 0 & \text{for } \alpha > \epsilon \\ \frac{1}{2} \left(\frac{1}{\epsilon} + \frac{\alpha}{\epsilon^2} + \frac{1}{2\pi\epsilon} \sin\left(\frac{2\pi\alpha}{\epsilon}\right) + \frac{1}{\epsilon\pi} \sin\left(\frac{\pi\alpha}{\epsilon}\right) + \frac{\epsilon+\alpha}{\epsilon^2} \cos\left(\frac{\pi\alpha}{\epsilon}\right) \right) & \text{for } |\alpha| \leq \epsilon \\ 0 & \text{for } \alpha < -\epsilon. \end{cases} \quad (3.13)$$

The scaled Heaviside and Dirac functions are functions of the standard level-set SDF. Thus, before calculating these parameters, we must recall the standard SDF,

$$\alpha = -E \ln \left(\frac{1}{\phi + 0.5} - 1 \right). \quad (3.14)$$

The difference between the scaled (\mathcal{H}_{scaled} and δ_{scaled}) and the regularized (\mathcal{H}_ϵ and δ_ϵ) Heaviside and Dirac functions is that the scaled functions shift the distribution of the regularized Dirac function to the higher density region in the transition of the two phases, as shown in Fig. 3.1, and improves the stability of the CSF model [99–101]. However, to evaluate the perimeter, area, and volume, we must use the regularized functions, since they are signed-symmetric to the interface between the two phases, while the scaled functions are not.

3.2 The N-S equations within an ALE framework

The ALE approach introduces a new frame of reference, related to a moving space-time domain, in which the N-S equations are solved. To take into account the effect of the

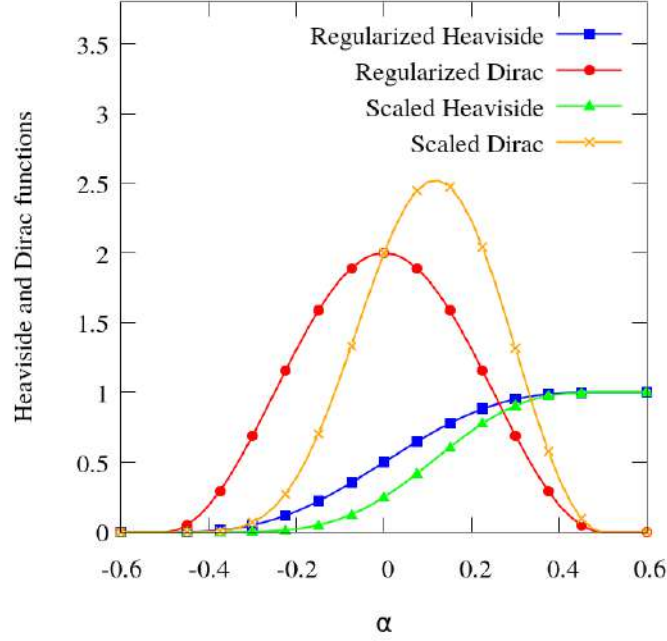


Figure 3.1: Comparison between the regularized and scaled Heaviside and Dirac functions.

mesh movement, the mesh velocity \mathbf{u}_{mesh} is introduced in Eq. 3.1 as follows

$$\rho \frac{\partial \mathbf{u}}{\partial t} + \rho(\mathbf{u} - \mathbf{u}_{\text{mesh}}) \cdot \nabla \mathbf{u} - \nabla \cdot \boldsymbol{\sigma} - \rho \mathbf{g} - \mathbf{F}_{\text{st}} = 0 \text{ in } \Omega \times [0, t_f] \quad (3.15)$$

For the sake of simplicity, we are going to introduce the term \mathbf{u}_M in which

$$\mathbf{u}_M = \mathbf{u} - \mathbf{u}_{\text{mesh}} \quad (3.16)$$

For fixed meshes, \mathbf{u}_{mesh} is equal to zero.

3.3 Residual-Based Variational Multiscale formulation

To approximate the Navier-Stokes equations we use a finite element scheme derived from the Residual-Based Variational Multiscale Formulation (RBVMS). The basic idea is to consider that the velocity and the pressure can be split into two components, a coarse one and a fine one. The fine scale components are modeled in terms of the residual of the large scales and then we replace arising their effect into the large-scale equation. The well-known stabilization formulations SUPG, PSPG, and LSIC, appear naturally within this framework [47], [11], [48], and other terms that arise from this formulation might be interpreted as numerical representations of the cross and Reynolds stresses arising in Large Eddy Simulation (LES) formulations, and therefore are used here as turbulence modeling [102–104]. Detailed reviews of the RBVMS formulation can be found at [47,

49–51].

For the present problem, we assume that the primitive variables are decomposed in two fields:

$$\mathbf{u} = \mathbf{u}^h + \mathbf{u}' \quad (3.17)$$

$$p = p^h + p' \quad (3.18)$$

where the subscript h denotes the coarse-scale component of the solution, while the superscript $'$ refers to the fine one. We insert the previous scale splitting in a standard variational Galerkin formulation built upon Eqs. (3.1) and (3.2). We assume the following weight spaces \mathbf{S}_u^h and S_p^h for velocity and pressure respectively, along with their trial spaces counterparts \mathbf{V}_u^h and V_p^h , and the weight functions \mathbf{w} and q such that,

$$\mathbf{S}_u^h = \{\mathbf{u}^h(\cdot, t) \in H^1(\Omega) \mid \mathbf{u}^h = \mathbf{u}_D \text{ on } \bar{\Gamma}_D\} \quad (3.19)$$

$$\mathbf{V}_u^h = \{\mathbf{w}^h(\cdot, t) \in H^1(\Omega) \mid \omega^h = 0 \text{ on } \bar{\Gamma}_D\} \quad (3.20)$$

$$S_p^h = V_p = \{q^h(\cdot, t) \mid q^h(\cdot, t) \in L^2(\Omega)\}. \quad (3.21)$$

The weak formulation is then: find $\mathbf{u} \in \mathbf{S}_u^h$ and $p \in S_p^h$ such that $\forall \mathbf{w} \in \mathbf{V}_u^h$ and $\forall q \in V_p^h$,

$$\left(\mathbf{w}, \rho \frac{\partial \mathbf{u}}{\partial t} \right)_\Omega + (\mathbf{w}, \rho \mathbf{u}_M \cdot \nabla \mathbf{u})_\Omega + (\nabla \mathbf{w}, \mu \nabla \mathbf{u})_\Omega - (\nabla \cdot \mathbf{w}, p)_\Omega \quad (3.22)$$

$$- (\mathbf{w}, \rho \mathbf{g})_\Omega - (\mathbf{w}, \mathbf{F}_{st})_\Omega - (\mathbf{w}, \mu \nabla \mathbf{u} \cdot \mathbf{n})_{\bar{\Gamma}} + (\mathbf{w}, p \mathbf{n})_{\bar{\Gamma}} + (q, \nabla \cdot \mathbf{u})_\Omega = 0 \quad (3.23)$$

To come up with a feasible numerical scheme, some simplifications are introduced. We admit the static hypothesis, such that the fine scales are algebraically related to the residuals of the governing equations. The time derivatives of the fine scales are supposed to vanish (rendering the denomination static fine scales), and the spatial derivatives in the fine scales equations are approximated through algebraic operators (much inspired on stabilized FEM) leading to the following set of fine scales equations:

$$\mathbf{u}' = -\tau_m \mathcal{R}_m \quad (3.24)$$

$$p' = -\tau_c \mathcal{R}_c \quad (3.25)$$

where τ_m and τ_c are stabilization parameters given by the standard expressions of stabi-

lized methods,

$$\tau_m = \left(\frac{4}{\Delta t^2} + \mathbf{u}^h \cdot \mathbf{G} \mathbf{u}^h + \nu^2 \mathbf{G} : \mathbf{G} \right)^{-\frac{1}{2}} \quad (3.26)$$

$$\tau_c = (\mathbf{g}^* \cdot \tau_m \mathbf{g}^*)^{-1} \quad (3.27)$$

with \mathbf{G} a second rank metric tensor,

$$\mathbf{G} = \frac{\partial \boldsymbol{\xi}^T}{\partial \mathbf{x}} \frac{\partial \boldsymbol{\xi}}{\partial \mathbf{x}} \quad (3.28)$$

and \mathbf{g}^* a vector obtained from the column sums of $\frac{\partial \boldsymbol{\xi}}{\partial \mathbf{x}}$, the inverse Jacobian of the element mapping between the parent and the physical domain,

$$\mathbf{g}^* = \{g_i^*\} \quad (3.29)$$

$$g_i^* = \sum_{j=1}^d \left(\frac{\partial \boldsymbol{\xi}}{\partial \mathbf{x}} \right)_{ji} . \quad (3.30)$$

The residuals for the continuity and momentum equations are given in function of the coarse-scales, that is,

$$\mathcal{R}_m = \rho \frac{\partial \mathbf{u}^h}{\partial t} + \rho \mathbf{u}_M^h \cdot \nabla \mathbf{u}^h + \nabla p^h - \mu \Delta \mathbf{u}^h - \rho \mathbf{g} - \mathbf{F}_{st} \quad (3.31)$$

$$\mathcal{R}_c = \nabla \cdot \mathbf{u}^h . \quad (3.32)$$

It is important to remember that the weighting functions also are decomposed into coarse and fine scales, resulting in two different equations. However, since we are considering an approximation of the fine scales based on the residuals, we only have to solve the coarse-scale equations.

After substituting and rearranging terms, we can obtain the RBVMS formulation for the Navier-Stokes equations as follows: find $\mathbf{u}^h \in \mathbf{S}_u^h$ and $p^h \in S_p^h$ such that $\forall \mathbf{w}^h \in \mathbf{V}_u^h$ and $\forall q^h \in V_p^h$,

$$\begin{aligned}
& \left(\mathbf{w}^h, \rho \frac{\partial \mathbf{u}^h}{\partial t} \right)_{\Omega} + \left(\mathbf{w}^h, \rho \mathbf{u}_M^h \cdot \nabla \mathbf{u}^h \right)_{\Omega} + \left(\nabla \mathbf{w}^h, \mu \nabla \mathbf{u}^h \right)_{\Omega} \\
& - \left(\nabla \cdot \mathbf{w}^h, p^h \right)_{\Omega} - \left(\mathbf{w}^h, \rho \mathbf{g} \right)_{\Omega} - \left(\mathbf{w}^h, \mathbf{F}_{st} \right)_{\Omega} - \left(\mathbf{w}^h, \mu \nabla \mathbf{u}^h \cdot \mathbf{n} \right)_{\bar{\Gamma}} \\
& + \left(\mathbf{w}^h, p^h \mathbf{n} \right)_{\bar{\Gamma}} + \left(q^h, \nabla \cdot \mathbf{u}^h \right)_{\Omega} + \sum_e \left(\nabla \mathbf{w}^h, \mathbf{u}^h \otimes \tau_m \mathcal{R}_{\mathbf{m}} \right)_{\Omega_e} \\
& + \sum_e \left(\nabla \mathbf{w}^h, \tau_m \mathcal{R}_{\mathbf{m}} \otimes \mathbf{u}_M^h \right)_{\Omega_e} - \sum_e \frac{1}{\rho} \left(\nabla \mathbf{w}^h, \tau_m \mathcal{R}_{\mathbf{m}} \otimes \tau_m \mathcal{R}_{\mathbf{m}} \right)_{\Omega_e} \\
& + \sum_e \left(\nabla \cdot \mathbf{w}^h, \rho \tau_c \mathcal{R}_c \right)_{\Omega_e} + \sum_e \frac{1}{\rho} \left(\nabla q^h, \tau_m \mathcal{R}_{\mathbf{m}} \right)_{\Omega_e} = 0
\end{aligned} \tag{3.33}$$

3.4 Time integration

For the time integration, we use the theta method, in which for a function $\frac{dy}{dt} = f(y)$ it has the general form of,

$$\frac{dy}{dt} \approx \frac{y_{n+1} - y_n}{\Delta t} = \theta f(y_{n+1}) + (1 - \theta) f(y_n) \tag{3.34}$$

where θ determines the method used, explicit or implicit.

Implicit methods find a solution by solving an equation involving the current state of the system, together or not with the solution of the previous time-step. It usually allows larger time-steps than explicit methods, that use only the previous time-step solution to find the current solution. The use of $\theta = 0.5$ yields the Crank-Nicolson method, which has second-order accuracy. However, this method does not handle problems with discontinuous initial data well and can have oscillatory results.

Therefore, in this work, we use the Backward Euler (BE) method, i.e., $\theta = 1$. This implicit method is only first order accurate on time, and we chose to sacrifice accuracy in time in favour of stability. Our temporal discretization becomes,

$$\frac{dy}{dt} \approx \frac{y_{n+1} - y_n}{\Delta t} = f(y_{n+1}). \tag{3.35}$$

Applying this method to Eq. (3.33) yields,

$$\begin{aligned}
& \left(\mathbf{w}^h, \rho \mathbf{u}_{n+1}^h \right)_\Omega - \left(\mathbf{w}^h, \rho \mathbf{u}_n^h \right)_\Omega + \Delta t \left[\left(\mathbf{w}^h, \rho \mathbf{u}_{M_{n+1}}^h \cdot \nabla \mathbf{u}_{n+1}^h \right)_\Omega + \left(\nabla \mathbf{w}^h, \mu \nabla \mathbf{u}_{n+1}^h \right)_\Omega \right. \\
& - \left(\nabla \cdot \mathbf{w}^h, p_{n+1}^h \right)_\Omega - \left(\mathbf{w}^h, \rho \mathbf{g} \right)_\Omega - \left(\mathbf{w}^h, \mathbf{F}_{\text{st}} \right)_\Omega - \left(\mathbf{w}^h, \mu \nabla \mathbf{u}_{n+1}^h \cdot \mathbf{n} \right)_{\bar{\Gamma}} \\
& + \left(\mathbf{w}^h, p_{n+1}^h \mathbf{n} \right)_{\bar{\Gamma}} + \left(q^h, \nabla \cdot \mathbf{u}_{n+1}^h \right)_\Omega + \sum_e \left(\nabla \mathbf{w}^h, \mathbf{u}_{n+1}^h \otimes \tau_m \mathcal{R}_m \right)_{\Omega_e} \\
& + \sum_e \left(\nabla \mathbf{w}^h, \tau_m \mathcal{R}_m \otimes \mathbf{u}_{M_{n+1}}^h \right)_{\Omega_e} - \sum_e \frac{1}{\rho} \left(\nabla \mathbf{w}^h, \tau_m \mathcal{R}_m \otimes \tau_m \mathcal{R}_m \right)_{\Omega_e} \\
& \left. + \sum_e \left(\nabla \cdot \mathbf{w}^h, \rho \tau_c \mathcal{R}_c \right)_{\Omega_e} + \sum_e \frac{1}{\rho} \left(\nabla q^h, \tau_m \mathcal{R}_m \right)_{\Omega_e} \right] = 0.
\end{aligned} \tag{3.36}$$

3.5 Linearization of the Navier-Stokes Equations

The N-S equations are non-linear because of the existence of the convective term. The usual approach to deal with this situation is to solve a linearized version of the equations at each time-step. Newton's method is one of the most popular numerical methods to solve this problem. In this method, each iteration solves a linear system of equations until the solution converges.

It is needed to provide an initial guess (\mathbf{u}_0, p_0) , which if it is close enough to the solution, will result in a quadratic convergence of the method. A bad initial guess may lead to the system not converging at all. Sometimes the solution converges even with an initial guess being null.

Newton's method reads: Given an initial guess $\mathbf{U}_0 \in V_N$ we seek to find $\delta \mathbf{U}^h \in V_N$ for $k = 1, 2, \dots$ until convergence such that

$$\mathbf{J}(\mathbf{U})_k^h \cdot \delta \mathbf{U}^h = -\mathbf{R}(\mathbf{U})_k^h \tag{3.37}$$

where $\delta \mathbf{U}^h = \mathbf{U}_{k+1}^h - \mathbf{U}_k^h$, the unknowns vector $\mathbf{U}^h = [\mathbf{u}^h, p^h]$, k is the Newton iteration number, $\mathbf{J}(\mathbf{U})_k^h$ is the Jacobian matrix with respect to $(\mathbf{U})_k^h$, and $\mathbf{R}(\mathbf{U})_k^h$ is the residual of Equation (3.33) at Newton iteration k .

To implement this method, we separate terms containing an unknown (i.e., a variable at iteration $k + 1$), keeping it on the left-hand-side, whereas all other terms are taken to the right-hand-side vector. The system of equations that is solved at each iteration is thus,

$$\mathbf{J}(\mathbf{U})_k^h \cdot \mathbf{U}_{k+1}^h = -\mathbf{R}(\mathbf{U})_k^h + \mathbf{J}(\mathbf{U})_k^h \cdot \mathbf{U}_k^h \tag{3.38}$$

All terms used for the RBVMS implementation are given in Appendix A.

3.6 Nondimensional Equations and Numbers

The nondimensional analysis is a common approach when working with fluids. It is possible to obtain information about the flow with characteristic parameters such as a dynamic viscosity μ_c , a flow speed u_c , a length scale l_c , a density ρ_c and the surface tension coefficient. Suitably combining these parameters, it yields dimensionless quantities that allow making statements for flows on any scale.

The Reynolds number, Re , is a dimensionless number defined as the ratio of inertial forces to viscous forces and consequently quantifies the relative importance of these two forces. At low Reynolds numbers, the flow is dominated by the viscous forces, while at high Reynolds numbers, the viscous forces can be neglected, and the inertial forces are dominating. Thus, the Reynolds number is,

$$Re = \frac{\rho_c u_c l_c}{\mu_c}. \quad (3.39)$$

The Froude number, Fr , is defined as the ratio of the flow inertia to the external field, that is,

$$Fr = \frac{u_c}{\sqrt{\|\mathbf{g}_c\| l_c}}. \quad (3.40)$$

The Weber number, We , is often useful where there is an interface between two different fluids, particularly for multi-phase flows with strongly curved surfaces. It can be thought of as a measure of the relative importance of the fluid's inertia compared to its surface tension. The quantity is useful in analyzing thin-film flows and the formation of droplets and bubbles, and is given by,

$$We = \frac{\rho_c u_c^2 l_c}{\sigma_{st}}. \quad (3.41)$$

The Eötvös number, EO , and the Morton number, MO , measure the importance of gravitational forces compared to surface tension forces and are used to characterize the shape of bubbles or drops moving in a surrounding fluid. They are defined as,

$$EO = \frac{g \Delta \rho D^2}{\sigma_{st}}. \quad (3.42)$$

$$MO = \frac{g \mu_L^4 \Delta \rho}{\rho_L^2 \sigma_{st}^3}. \quad (3.43)$$

Here, D is the bubble diameter, $\Delta \rho = \rho_L - \rho_G$, ρ_L is the density of the liquid medium, and ρ_G is the density of the gaseous one. In bubbles problems, we use $u_c = u_\infty$ and $L_c = D$ to calculate the Reynolds number, where u_∞ is the bubble terminal rise velocity.

Some bubble benchmarks are calculated with the dimensionless equations. Based on this concept, we introduce the following dimensionless variables,

$$\mathbf{x}^* = \frac{\mathbf{x}}{l_c} \quad (3.44)$$

$$t^* = \frac{u_c t}{l_c} \quad (3.45)$$

$$\mathbf{u}^* = \frac{\mathbf{u}}{u_c} \quad (3.46)$$

$$p^* = \frac{p - p_c}{\rho_c u_c^2} \quad (3.47)$$

$$\rho^* = \frac{\rho}{\rho_c} \quad (3.48)$$

$$\mu^* = \frac{\mu}{\mu_c} \quad (3.49)$$

$$\mathbf{g}^* = \frac{\mathbf{g}}{\|\mathbf{g}_c\|} \quad (3.50)$$

$$\mathbf{F}_{st}^* = \frac{\mathbf{F}_{st}}{\sigma_{st}}. \quad (3.51)$$

Recasting Eqs. (3.1) and (3.2) in the variables of Eqs. (3.44) to (3.51), we have the dimensionless form of N-S equations,

$$\rho^* \frac{\partial \mathbf{u}^*}{\partial t} + \rho^* \mathbf{u}_M^* \cdot \nabla \mathbf{u}^* + \nabla p^* - \frac{1}{Re} \nabla \cdot (\mu^* \nabla \mathbf{u}^*) - \frac{\rho^* \mathbf{g}^*}{Fr^2} - \frac{\mathbf{F}_{st}^*}{We} = 0 \text{ in } \Omega \times [0, t_f] \quad (3.52)$$

$$\nabla \cdot \mathbf{u}^* = 0 \text{ in } \Omega \times [0, t_f]. \quad (3.53)$$

From now on, we drop the asterisk for simplicity. The finite element formulation also applies to Eqs. (3.52) and (3.53). In addition, we introduce the following quantities of interest, which will be used to assist in describing the temporal evolution of the bubbles quantitatively.

Center of mass. To track the translation of bubbles, it is common to use the center of mass,

$$\mathbf{X}_c = \frac{\int_{\Omega^-} \mathbf{x} dx}{\int_{\Omega^-} 1 dx} \quad (3.54)$$

where Ω^- denotes the region that the bubble occupies.

Circularity/Sphericity. The “degree of circularity”, ϕ , introduced by WADELL [105], can be defined as,

$$\phi = \frac{P_a}{P_b} = \frac{\text{perimeter of a area-equivalent circle}}{\text{perimeter of the bubble}} = \frac{2\pi R}{P_b}. \quad (3.55)$$

Here, P_a denotes the perimeter or circumference of a circle with radii R which has

an area equal to that of a bubble with perimeter P_b . Similarly, for spheres, the “degree of sphericity”, ϕ , is given by,

$$\phi = \frac{A_a}{A_b} = \frac{\text{surface area of a volume-equivalent sphere}}{\text{surface area of the bubble}} = \frac{4\pi R^2}{A_b}. \quad (3.56)$$

Rise Velocity. The mean velocity, \mathbf{U}_{mean} , with which a bubble is rising or moving, is a particularly interesting quantity since it does not only measure how the interface tracking algorithm behaves but also the quality of the overall solution. We define the mean bubble velocity as,

$$\mathbf{U}_{\text{mean}} = \frac{\int_{\Omega^-} \mathbf{u} dx}{\int_{\Omega^-} 1 dx}. \quad (3.57)$$

Diameter. Denotes the bubble’s maximum extension in each coordinate direction.

3.7 Boundary conditions

The boundary conditions (BCs) are equations that give the constraints of the problem under consideration. There are three types of boundary conditions commonly encountered in the solution of partial differential equations: Dirichlet BC, specify the value of the function on a surface; Neumann BC, specify the normal derivative of the function on a surface; and Robin BC, specify a linear combination of the values of a function and the values of its derivative on the boundary of the domain.

The most common Dirichlet BCs used in CFD are the no-slip and the free-slip BCs. The no-slip BC means that the velocities at the boundary are zero ($\mathbf{u}^h = \mathbf{0}$), while the free-slip represents no friction between the fluid and boundary, i.e., the normal velocity is set to zero ($\mathbf{u}^h \cdot \mathbf{n} = 0$).

The Dirichlet BC usually is applied using the penalty method, called the strong form. The penalty method was put in a finite element context by BABUŠKA [106]. The use of this method replaces the Dirichlet boundary condition $\mathbf{u}^h = \mathbf{u}_D^h$ on $\bar{\Gamma}_D$ by a penalized Robin-type boundary condition, as follows,

$$\mathbf{t}' + \beta \mathbf{u}^h = \beta \mathbf{u}_D^h \text{ on } \bar{\Gamma}_D \quad (3.58)$$

where β acts as a penalty constant.

The boundary term that arises from the integration by parts of the N-S equations (Eq. 3.22) is now split into a Neumann portion $\bar{\Gamma}_N$ and a penalized Robin portion,

$$(\mathbf{w}^h, \mu \nabla \mathbf{u}^h \cdot \mathbf{n})_{\bar{\Gamma}} - (\mathbf{w}, p^h \mathbf{n})_{\bar{\Gamma}} = (\mathbf{w}^h, \mathbf{t}')_{\bar{\Gamma}_N} + (\mathbf{w}^h, \beta \mathbf{u}_D^h - \beta \mathbf{u}^h)_{\bar{\Gamma}_D}. \quad (3.59)$$

Therefore the momentum equation (Eq. 3.22) changes as follows using the penalty method,

$$\begin{aligned} & \left(\mathbf{w}^h, \rho \frac{\partial \mathbf{u}^h}{\partial t} \right)_{\Omega} + (\mathbf{w}^h, \rho \mathbf{u}_M^h \cdot \nabla \mathbf{u}^h)_{\Omega} + (\nabla \mathbf{w}^h, \mu \nabla \mathbf{u}^h)_{\Omega} - (\nabla \cdot \mathbf{w}^h, p^h)_{\Omega} \\ & - (\mathbf{w}^h, \rho \mathbf{g})_{\Omega} - (\mathbf{w}^h, \mathbf{F}_{st})_{\Omega} - (\mathbf{w}^h, \mathbf{t}')_{\bar{\Gamma}_N} - (\mathbf{w}^h, \beta \mathbf{u}_D^h - \beta \mathbf{u}^h)_{\bar{\Gamma}_D} + (q^h, \nabla \cdot \mathbf{u}^h)_{\Omega} = 0. \end{aligned} \quad (3.60)$$

In order to simplify the understanding, let us consider a matrix-vector system $\mathbf{K}_J \mathbf{u}_{nk} = \mathbf{f}_{RHS}$ with \mathbf{K}_J the Jacobian matrix, \mathbf{u}_{nk} the vector of unknowns and \mathbf{f}_{RHS} the right-hand-side vector. The penalty method for this system yields

$$\mathbf{K}_J \mathbf{u}_{nk} + \beta \mathbf{K}_{LD} \mathbf{u}_{nk} = \mathbf{f}_{RHS} + \beta \mathbf{K}_{JD} \mathbf{u}_{nk} \quad (3.61)$$

where \mathbf{K}_{JD} is a diagonal constraint matrix which has unit entries only for the Dirichlet boundary nodes.

Since β is very large, the penalty terms dominate this equation. At the same time, large values of β may cause ill-conditioning of the matrix, although this problem is alleviated by the use of preconditioners and iterative linear system equations solvers. The value of β is taken as 10^{10} .

The Dirichlet BC also may be applied weakly, by adding terms to the variational formulation. Considering the normal component of the velocity enforced strongly ($\mathbf{u}^h \cdot \mathbf{n} = 0$), we have to add the following terms to the momentum equation in order to enforce the tangential component of the velocity weakly,

$$-\sum_e (\mathbf{w}^h, \mu \nabla \mathbf{u}^h \cdot \mathbf{n} - p^h \mathbf{n})_{\bar{\Gamma}_D} - \sum_e (\mu \nabla \mathbf{w}^h \cdot \mathbf{n} - q^h \mathbf{n}, \mathbf{u}^h - \mathbf{b}^h)_{\bar{\Gamma}_D} + \sum_e (\mathbf{w}^h, \tau_B (\mathbf{u}^h - \mathbf{b}^h))_{\bar{\Gamma}_D} \quad (3.62)$$

with $\mathbf{b}^h = \mathbf{0}$ for the no-slip BC.

The first term corresponds to the resolved viscous shear stress at the boundary. However, the computed velocity is poorly resolved in the near-wall region. Thus this stress is not expected to match the correct wall shear stress. Therefore, a penalty stress is added corresponding to the third term of the equation. The second term is the called adjoint consistent term.

BAZILEVS *et al.* [107] have concluded that the best penalty stress to be considered is the law of the wall. They have implemented the one given by Spalding [108] being

$$\tau_b = \rho \frac{u^{h*2}}{\|\mathbf{u}_p\|} \quad (3.63)$$

with u^* the friction velocity and \mathbf{u}_p the velocity parallel to the wall. The friction velocity is calculated by

$$y^+ = f(u^+) = u^+ + e^{-\chi B} \left(e^{\chi u^+} - 1 - \chi u^+ - \frac{(\chi u^+)^2}{2} - \frac{(\chi u^+)^3}{6} \right) \quad (3.64)$$

where y^+ and u^+ denote the distance from the wall and the mean fluid velocity, respectively, expressed in non-dimensional wall units as

$$y^+ = \frac{y u^*}{\nu} \quad (3.65)$$

$$u^+ = \frac{\|\mathbf{u}_p\|}{u^*} \quad (3.66)$$

Also, χ is 0.4 and B is 5.5. The Spalding law is valid for all ranges of y^+ . This is important because the flow behavior changes on layers near the wall, and this law can account these effects.

It is known that weak and strong boundary conditions give the same results on stretched meshes. However, on coarse meshes, the weakly imposed boundary conditions deliver a significantly more accurate mean flow velocity than their strong counterpart. The combination of the RBVMS with the weakly imposed boundary condition acts like a RANS-type model applied to meshes too coarse for conventional LES models [107].

Sometimes, one may face a huge domain even if the interest is only in the initial part of it. Thus, the domain is cut in a way to reduce the computation effort by creating an Outflow BC. However, this is not trivial when the flux is turbulent and how to deal with outflows for incompressible flow simulations is still a challenge. In this work, we are not interested in reducing the domain significantly. Therefore, in the absence of a body force field (e.g., a gravitational field), we may prescribe the normal stress equal to zero on the Outflow BC, defined as,

$$(\mathbf{w}^h, (-p^h \mathbf{n} + \mu \nabla \mathbf{u}^h \cdot \mathbf{n}))_{\Gamma_O} = 0. \quad (3.67)$$

However, when we do consider a body force field, this equation changes a little. The pressure field can be decomposed in two components,

$$p^h = p_0 + p_d \quad (3.68)$$

where p^h is the total pressure, p_0 is the hydrostatic pressure and p_d is the dynamic pressure.

At the outlet, only the dynamic pressure is specified as zero normal gradient and not the total pressure. Thus, the hydrostatic pressure has to be subtracted from the total pressure in this boundary. The resulting equation is,

$$\begin{aligned}
(\mathbf{w}^h, -(p_d \mathbf{n} + \mu \nabla \mathbf{u}^h \cdot \mathbf{n}))_{\bar{\Gamma}_O} &= 0 \\
(\mathbf{w}^h, -((p^h - p_0) \mathbf{n} + \mu \nabla \mathbf{u}^h \cdot \mathbf{n}))_{\bar{\Gamma}_O} &= 0 \\
(\mathbf{w}^h, -(p^h \mathbf{n} + \mu \nabla \mathbf{u}^h \cdot \mathbf{n}))_{\bar{\Gamma}_O} + (\mathbf{w}^h, (p_0 \mathbf{n}))_{\bar{\Gamma}_O} &= 0 \\
(\mathbf{w}^h, -(p^h \mathbf{n} + \mu \nabla \mathbf{u}^h \cdot \mathbf{n}))_{\bar{\Gamma}_O} &= -(\mathbf{w}^h, (p_0 \mathbf{n}))_{\bar{\Gamma}_O}.
\end{aligned} \tag{3.69}$$

Note that in this case, the tangential velocity is set to be zero [109, 110].

The Outflow BC is important in two-fluid problems where the hydrostatic pressure must be considered to track the air/water interface.

Chapter 4

Sediment transport model

In this section, we present the governing equations of the sediment transport model, including the suspended and bed-load. We also present the bed morphology model and the upwind sand-slide algorithm, developed to smooth and minimize the unrealistic bed shapes that may occur when the predicted slope of the elements exceeds the sediment angle of repose.

4.1 Governing equations

The governing equations of the sediment transport model are composed of an advection-diffusion equation that represents the suspended load, together with empirical formulations that represent the bed-load and entrainment rate of sediments. Here we present all the components necessary to model these phenomena.

4.1.1 Suspended Load

In the suspended load, the turbulence of the flow carries the sediment particles, and an advection-diffusion equation governs it. By assuming a dilute suspension of small particles, we can neglect particle inertia and any particle-particle interactions. Consequently, the particles are driven by fluid motion, with a settling velocity acting in the direction of gravity, u_s , and the following equation governs it,

$$\frac{\partial c}{\partial t} + (\mathbf{u}_M + u_s \mathbf{e}^g) \cdot \nabla c = \nabla \cdot (\epsilon_s \nabla c) \text{ in } \Omega \times [0, t_f] \quad (4.1)$$

in which c is the sediment concentration and ϵ_s is the sediment diffusivity considered to be very small. The terms which are coming from the Eq. (4.1), resulting from applying the RBVMS formulation and the Backward Euler time integration, are given in Appendix B.

The settling velocity can be estimated by experiments or analytically. Stokes' solution for the drag resistance of the flow past a sphere can be expressed by the simplified Navier-

Stokes equation together with the continuity equation in polar coordinates. By using his solution, the following expression for the settling velocity of spherical particles can be derived [111],

$$u_s = \frac{d_{50}^2(\rho_s - \rho)g}{18\mu} \quad (4.2)$$

where d_{50} is the average particle size, ρ is the fluid density, ρ_s is the particle density, g is the gravitational acceleration and μ is the fluid viscosity.

4.1.2 Bed-load transport

The bed-load transport is the sediment movement near the bottom, where the particles are moved by sliding, rolling or by saltation. In our work, it is represented as mesh movement, since the boundary of the domain represents the sediment bottom.

There will be bed-load transport if the boundary shear stress exceeds the sediment critical bed shear stress. The critical bed shear stress for a particular sediment is a function of the flow intensity and sediment properties, and it is calculated from the Shields diagram [112]. It is always given for a flat bed of identically-sized grains, and it can be determined from the Soulsby-Whitehouse [86] equation, that is,

$$\tau_{\theta c} = \frac{0.3}{1 + 1.2d_*} + 0.055(1 - e^{-0.02d_*}) \quad (4.3)$$

where d_* is the dimensionless grain size given by,

$$d_* = d_{50} \left(\frac{gR}{\nu^2} \right)^{\frac{1}{3}} \quad (4.4)$$

in which R is the sediment submerged specific gravity $\left(R = \frac{\rho - \rho_s}{\rho} \right)$.

The dimensionless shear stress for motion initiation, the Shields number, is defined as,

$$\tau_\theta = \frac{\tau_b}{\rho g R d_{50}} \quad (4.5)$$

where τ_b is the bed shear stress calculated with the fluid flow model.

In this way, if the Shields number calculated in Equation (4.5) is higher than the critical shear stress, it is necessary to calculate the bed-load transport. There are many bed-load transport rate formulas in the literature [87, 113, 114]. For the current model, the formula of Engelund and Fredsøe [87] is chosen, and it is defined as,

$$q^* = \begin{cases} 18.74(\tau_\theta - \tau_{\theta c})(\tau_\theta^{0.5} - 0.7\tau_{\theta c}^{0.5}), & \text{if } \tau_\theta > \tau_{\theta c} \\ 0, & \text{otherwise} \end{cases} \quad (4.6)$$

where q^* is the dimensionless bed-load transport rate, known as the Einstein number, which is given by,

$$q^* = \frac{q_0}{\sqrt{Rgd_{50}d_{50}}}. \quad (4.7)$$

Here q_0 is the bed-load transport rate for flat bed, and it is used to find the bed-load transport rates in different directions considering the slope effect according to the Equation (4.8), that is,

$$q_i = q_0 \frac{\tau_{bi}}{|\boldsymbol{\tau}_b|} - C|q_0| \frac{\partial \eta}{\partial x_i} \quad (4.8)$$

where q_i is the bed-load rate on the i direction, τ_{bi} is the bed shear stress on the i direction, $\boldsymbol{\tau}_b$ is the vector containing τ_{bi} , η is the bed elevation and C is a constant in the range of 1.5-2.3 which is used to represent the slope effect on the sediment flux [29].

4.1.3 Entrainment and Deposition Rate

Depending on the shear stress at the bottom, sediments can entrain into suspension, going out of the bed-load zone and starting to be governed by the suspended load equations. It is also possible that suspended particles are settled on the bottom. We represent these exchanges between bed-load and suspended load zones as boundary conditions together with mesh movement.

The flux of sediments \mathbf{q}_s including convective and diffusive fluxes is defined as,

$$\mathbf{q}_s = (\mathbf{u}_M + u_s \mathbf{e}^g)c - \epsilon_s \nabla c. \quad (4.9)$$

The sediment flux at the atmosphere boundary and the rigid walls is zero ($\mathbf{q}_s \cdot \mathbf{n} = 0$), and at the sediment bed, the sediment flux is equal to $E - D$, where E is the entrainment rate and D is the deposition rate ($\mathbf{q}_s \cdot \mathbf{n} = E - D$). Besides the flux boundary conditions, we may have a boundary where there is a prescribed flux of sediments. In this case we use a Dirichlet boundary condition with $c = c_{in}$ on Γ_{in}^s .

The entrainment rate also needs an empirical model and, in this work, we adopt the van Rijn model [83]. The entrainment rate is then given by,

$$E = C_{ae} u_s \quad (4.10)$$

where

$$C_{ae} = 0.015 \frac{d_{50} T^{1.5}}{a d_*^{0.3}} \quad (4.11)$$

$$T = \frac{(u_*' - u_{*c})}{u_{*c}} \quad (4.12)$$

$$u_{*c} = \sqrt{\frac{\tau_{bc}}{\rho}} \quad (4.13)$$

$$\tau_{bc} = \tau_{\theta c} \rho g R d_{50} \quad (4.14)$$

$$u'_{*} = \frac{g^{0.5}}{C'} \bar{u} \quad (4.15)$$

$$C' = 18 \log \left(\frac{12R_b}{3d_{90}} \right). \quad (4.16)$$

In equations (4.10) to (4.16), C_{ae} is the near-bed sediment concentration in volume under capacity (equilibrium) status, a is the reference level above the bed, T is the transport parameter, u'_{*} is the effective bed shear velocity related to grain friction, u_{*c} is the critical bed-shear velocity for initial motion of sediment, $\tau_{\theta c}$ is the critical Shields number given by Equation (4.3), \bar{u} is the mean flow velocity, R_b is the hydraulic radius according to the method of Vanoni-Brooks [115] and d_{90} is the grain size corresponding to 90% finer.

On the other hand, the deposition rate at the bed is straightforward,

$$D = u_s c_b \quad (4.17)$$

where c_b is the sediment concentration very near the bed.

4.2 Bed morphology model with ALE

Bed elevation changes are based on the continuity of sediment. The equation which describes the sediment continuity is the Exner equation [116], given by:

$$\frac{\partial \eta}{\partial t} = \frac{1}{1-n} (-\nabla \cdot \mathbf{q}_b + D - E) \quad (4.18)$$

where η is the bed elevation, n is the porosity of the bed, \mathbf{q}_b is the bed-load transport rate vector whose components are given by Equation (4.8), E is the entrainment rate given by Equation (4.10) and D is the deposition rate given by Equation (4.17).

The Exner equation relates all possible ways of sediment moves in a giving time-step. We apply the bed elevation resulting from Equation (4.18) as mesh movement, with the resulting scheme described within the ALE framework, as explained in chapter 2.2.

4.3 Upwind sand-slide

In sediment transport problems, unrealistic bed shapes as sharp corners sometimes occur and undermine numerical stability. It often occurs when the predicted slope of the

elements exceeds the sediment angle of repose. Therefore, different smoothing techniques were developed to minimize these unrealistic bed shapes [2, 30, 59, 117].

In this work, we developed a new smoothing procedure called upwind sand-slide. The traditional smoothing techniques rotate the element boundary around its gravity center until the slope becomes lower or equal to the sediment angle of repose. Thus, sediment mass is removed and reallocated in the domain to avoid big slopes and its consequences in the numerical simulation.

Our smoothing technique also changes the position of the sediments, but we change it by considering the direction of the flow. If the flow is coming in the up-slope direction, it does not make sense that sediments move in the opposite direction as occurs on traditional sand-slide models. Instead of rotating the element boundary around its gravity center, we fix the first node, and we move the two subsequent nodes in the upwind flow direction, one to adjust the slope, and the other to keep the mass conservation. When the flow is coming in the down-slope direction, we can use the traditional method, since the sediment movement follows already the flow direction. We can see how it works in Figure 4.1.

We apply the upwind sand-slide if the computed bed slope between two adjacent bed nodes is found to be larger than the sediment angle of repose. The distance d_y used to adjust the slope is given by,

$$d_y = \frac{x_i - x_{i-1}}{2} (\tan \alpha^s - \tan \phi_c^s) \quad (4.19)$$

where α^s is the angle between the bed and the horizontal axis, x_i and x_{i-1} are the horizontal coordinates of the nodes and ϕ_c^s is the critical slope. This method requires a slope change on the adjacent elements every time it is applied. Thus, after that, we have to check the slope of all elements again, and the algorithm is iterated until convergence is achieved. Figure 4.2 shows the iteration procedure of this method. The sediment angle of repose, ϕ^s , has been fixed and the angle of stability limit is chosen to be $\phi_c^s = \phi^s + 1^\circ$ according to [118].

It is important to remember that these methods are geometrical, and the horizontal distance between nodes have to be the same to conserve the mass of sediments. It is also important to verify if the integral of the vertical coordinates at the bottom is constant along the time interval of interest.

Besides the upwind sand-slide model, we also apply a Laplacian smoothing in the boundary nodes to avoid small oscillations that may harm numerical stability. This smoothing is made after a certain number of time-steps, considering the quantity of sediments in movement. This smoothing tends to lead the sediment bottom to a horizontal shape. Thus we have to be careful about the frequency that it is applied. Considering we have only bilinear elements, the Laplacian smoothing on the boundary can be written as,

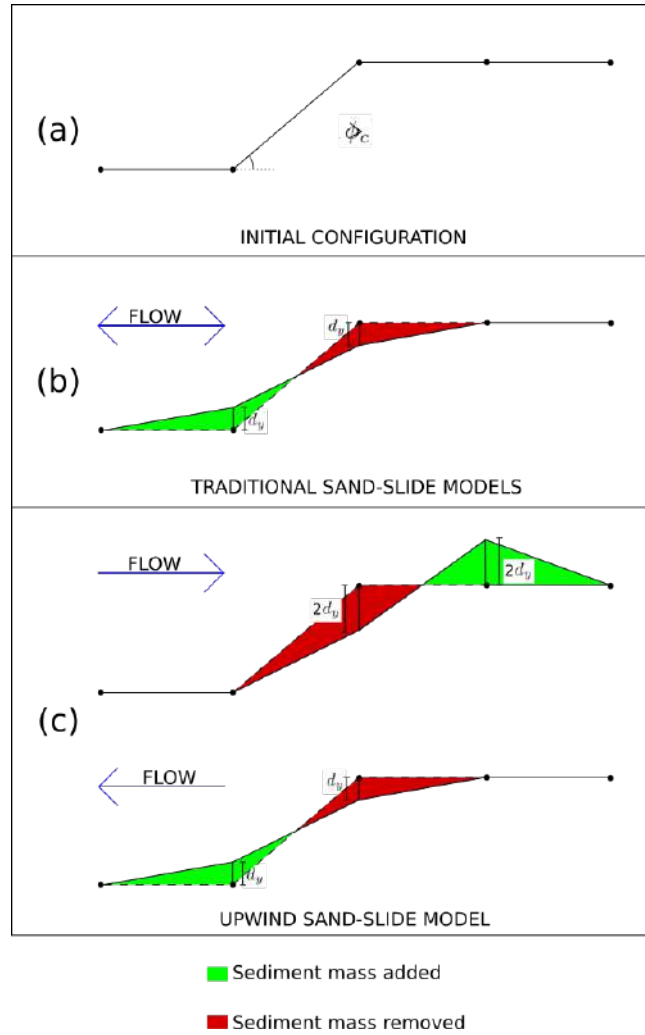


Figure 4.1: Upwind sand-slide model: (a) Initial configuration in which the slope angle is bigger than the sediment angle of repose; (b) Traditional sand-slide models in which we rotate the element boundary around its gravity center until the slope becomes lower or equal to the sediment angle of repose; (c) Upwind sand-slide model in which we consider the direction of the flow. If the flow is coming in the up-slope direction, we fix the first node, and we move the two subsequent nodes in the upwind flow direction, one to adjust the slope, and the other to keep the mass conservation. When the flow is coming in the down-slope direction, we can use the traditional method. The green area is the sediment mass added, and the red area is the sediment mass removed.

$$y_i = 0.5y_i + 0.25(y_{i-1} + y_{i+1})$$

where y_i is the vertical coordinate of the node, and y_{i-1} and y_{i+1} are the vertical coordinates of its neighbor nodes.

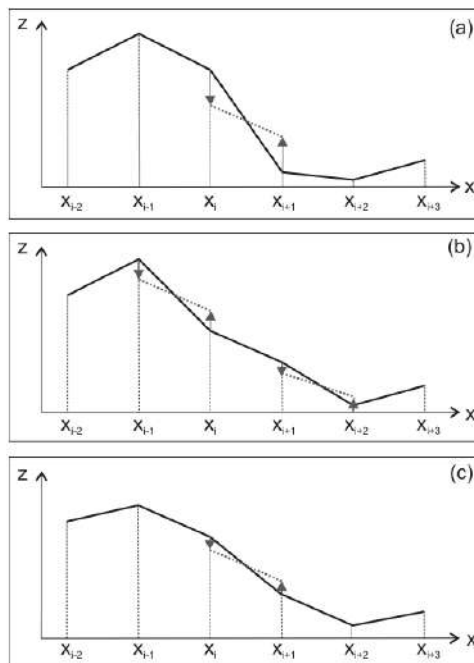


Figure 4.2: Two iterations of the sand-slide algorithm; (a) initial bed form, (b) first iteration, (c) second iteration [2]

Chapter 5

Computational aspects

This chapter discusses the computation aspects of the models implemented on `libmesh` [25]. We present the structure of the algorithm and how the model is solved using the available libraries.

5.1 The `libmesh` library

`libmesh` provides a finite element framework that can be used for numerical simulation of partial differential equations on serial and parallel platforms. This library is an excellent tool for programming the finite element method and can be used for one-, two-, and three-dimensional steady and transient simulations. The `libmesh` library also has adaptive mesh refinement and coarsening strategies, also used in this work.

The main advantage of this library is the possibility to focus on the implementation of the specific features of the modeling without worrying about adaptivity and code parallelization. Consequently, the effort to build a high performance computing code tends to be minimized.

The `libmesh` library is written in C++ [119], being compatible with several software and hardware. There are many external libraries used by `libmesh` as the `LASPack` [120], which has a package with serial routines of linear algebra and iterative solution methods and the Portable Extensible Toolkit for Scientific Computation (PETSc) library [121], for the solution of linear and non-linear systems, both in serial and parallel. We also can cite `METIS` [122] and `ParMETIS` [123] used for the domain partitioning, among other available libraries.

The PETSc library gives the support for the numerical solution of the partial differential equations in parallel. It has efficient data storage and manipulation of vectors and matrices structures. The PETSc library also has several methods for the solution of sparse linear systems as the Generalized Minimal Residual Method (GMRES) and the bi-conjugated gradient, and preconditioners as the ILU(k) and block-Jacobi. It is the user that defines which kind of linear solver and preconditioner will be used within the algo-

rithm. For more information about this library, linear solvers and preconditioners, see [124] and [125].

5.2 Adaptive Mesh Refinement (AMR)

The AMR procedure uses a local error estimator considering the error of an element relative to its neighbor elements in the mesh. This error may come from any variable of the system. As it is standard in `libMesh`, Kelly's error indicator is employed, which uses the H^1 -seminorm to estimate the error ([126]). Apart from the element interior residual, the flux jumps across the inter-element faces influence the element error. The flux jump of each face is computed and added to the error contribution of the cell. For both the residual and flux jump, the value of the desired variables at each node is necessary. Being so, the error $\|e\|^2$ can be stated as,

$$\|e\|^2 = \sum_{i=1}^n \|e\|_i^2 \quad (5.1)$$

where $\|e\|_i^2$ is the error of each variable. In this study we use only the velocity as variable for the error estimator.

After computing the error values the elements are "flagged" for refining and coarsening regarding their relative error. This is done by a statistical element flagging strategy. It is assumed that the element error $\|e\|$ is distributed approximately in a normal probability function. Here, the statistical mean μ_s and standard deviation σ_s of all errors are calculated. Whether an element is flagged is depending on a refining (r_f) and a coarsening (c_f) fraction. For all errors $\|e\| < \mu_s - \sigma_s c_f$ the elements are flagged for coarsening and for all $\|e\| > \mu_s + \sigma_s r_f$ the elements are marked for refinement. The refinement is performed by local isotropic subdivision (h-refinement) with hanging nodes. Here, the refinement level is limited by a maximum h -level (h_{max}) and the coarsening is done by h-restitution of subelements ([127], [128], [129]).

5.3 Two-fluid algorithm

To solve two-fluid problems, we start the algorithm by initializing the level-set variables, building the iso-surfaces according to the modified SDF. Then, it calls a routine to calculate the normals and curvature. Afterward, it calculates the Navier-Stokes equation using the previously defined fluid properties. The PETSc library solves the linear system of equations coming from the linearization of the Navier-Stokes equations invoked by `libMesh`, applying the GMRES with Block-Jacobi preconditioner together with ILU(0) within each block. Then, the interface movement is calculated with the convected level-

set method using the velocities coming from the fluid flow model. The resulting linear system of the pure convection equation is solved in the same way as the fluid flow model. The global mass conservation is then enforced. With the flow and level-set updated, the adaptive mesh refinement algorithm is applied, refining, or coarsening the mesh depending on the flux jump of the level-set and velocity errors. Finally, before going for the next time-step, the flow and level-set equations are recalculated with the updated mesh. Figure 5.1 illustrate this sequence.

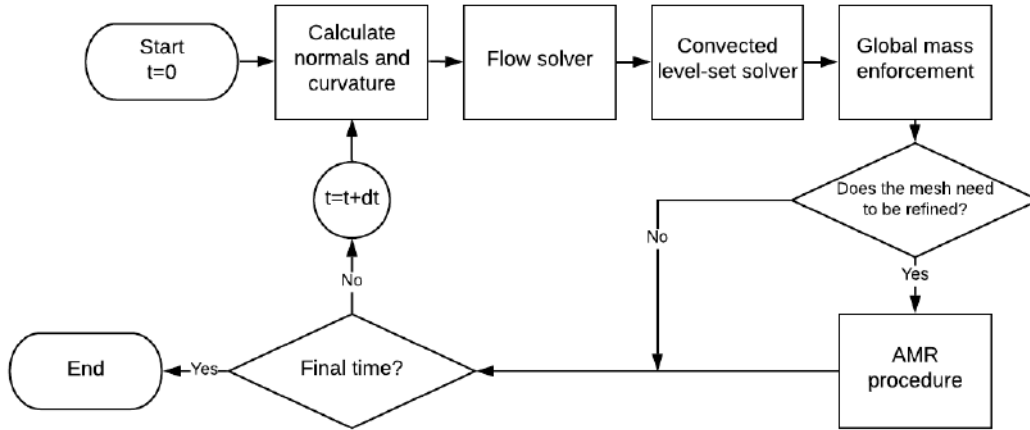


Figure 5.1: Flow chart of computation of two-fluid problems

The use of the algorithm is straightforward. To give the information about the problem under consideration, we need to provide two input files. One contains mesh data, including node coordinates, element connections, and physical groups of each boundary, and another contains all information and parameters to close the problem, as time-step, final time, the maximum number of non-linear and linear iterations, etc. Here, we also define what happens with each physical group by defining the boundary conditions. The `libmesh` library also allows the mesh creation inside the source code. However, depending on the geometry, it is easier to use an external mesh generator as GMSH, for example [130].

5.4 Sediment-fluid algorithm

In sediment transport problems, we only consider the fluid and the sediment bottom. We are still not considering free-surface problems together with morphological changes. However, we intend to couple both methods in future works. Therefore, we only have one phase, representing the fluid, while the sediment bottom is represented as mesh movement within the ALE framework. It is possible to avoid some steps of calculation used in the previous algorithm by defining a constant density in the Navier-Stokes equation.

Therefore, the algorithm starts calculating the fluid flow model using AMR based on the flux jump of the velocities error. After that, the suspended sediment transport is calculated using the velocities coming from the fluid flow model. If there is no sediment suspended and there is no entrainment rate, the algorithm jumps to the next part. Otherwise, the resulting linear system is solved in the same way (using the preconditioned GMRES).

Finally, we calculate morphological contributions. It comes from the bed-load transport and entrainment and deposition rate, according to the Exner equation (4.18). Here, we calculate the bottom displacements, and we check the angle of repose before going to the mesh movement solver. We adjust the displacements using the upwind sand-slide algorithm, and once this process is over, it is possible to move the mesh.

The mesh movement system of equations is symmetric, and it is solved using the preconditioned Conjugate Gradients method, together with IC(0). The mesh is updated, and the velocities coming from these changes are included in the flow calculation in the next time-step. Figure 5.2 illustrate this sequence.

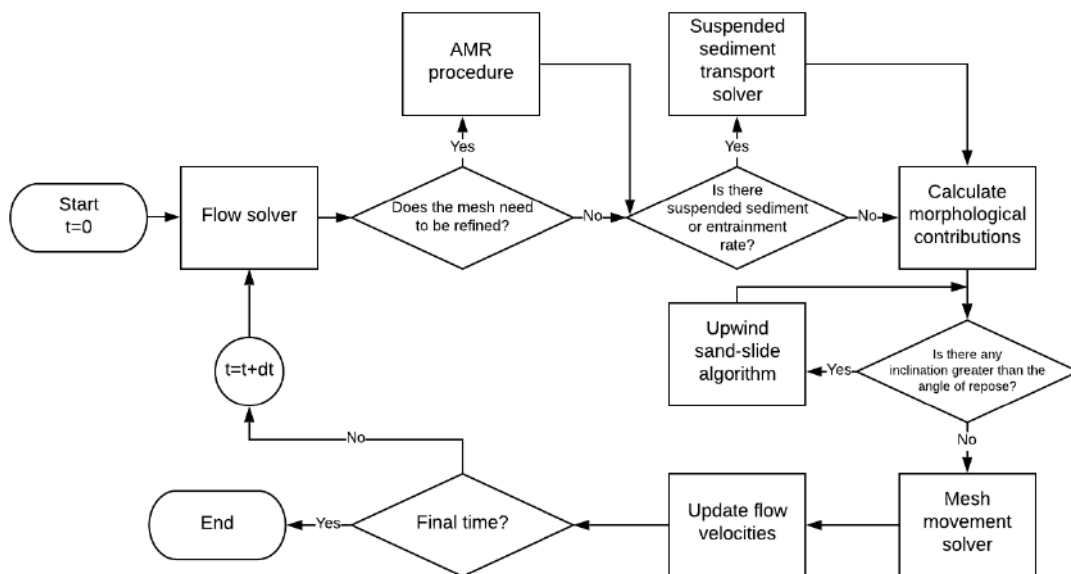


Figure 5.2: Flow chart of computation of sediment-fluid problems

Note that we are not considering that the suspended sediment affects the fluid behavior. The sediments are considered passive scalars to the flow. They are only carried by the flow without changing its course. However, it is possible to consider the sediment influence and to couple both modules. We only have to add a forcing term correspondent to the suspended sediment concentration in the Navier-Stokes equations [64]. The test cases used in this work do not require this coupling because the influence of the particles on the fluid is minimal to them. This is the reason we are considering the “passive scalar hypothesis”, in which the sediment concentration is too low to be considered in the fluid model.

Chapter 6

Results and Discussion

In this chapter, we verify our algorithm by solving several problems. The verification goes through several stages until reaching the complete model. These steps are essential to understand each phase of the developed models and also to isolate errors when they appear.

We start verifying the fluid flow model by solving single-phase problems. Then we verify the implementation of the convected level-set method with pure convection problems. After, we check the surface tension model by comparing the results with analytical responses. Finally, it is possible to solve two-fluid problems, and we solve several bubbles motion problems.

To solve sediment-fluid problems, we start verifying the mesh movement procedure by applying it to a simple configuration where it is possible to check all considerations done for these problems. Then, we show the influence of the upwind sand-slide smoothing technique. After, the suspended load of sediments is verified against experimental data. Finally, we present the results of the complete model of sediment-fluid interaction.

6.1 Verification of the fluid flow model

The first challenge is to model the flow field correctly. Therefore, we begin with two classical single-phase problems: the lid-driven cavity, and the flow over a backward facing-step. These problems are useful benchmarks to verify the implemented model.

6.1.1 Lid-driven cavity test case

We begin solving the classical lid-driven flow problem. This test has been widely used as a benchmark for numerical methods and has been analyzed by many authors [131–133]. The problem consists of a square container with dimensions 1×1 filled with a fluid, and the lid of the container moves at a given velocity, thereby setting the fluid in motion. The problem description and boundary conditions are shown in Figure 6.1, where no-slip

conditions are imposed on all segments of the boundary except for the upper boundary, along which the velocity u is not set to zero, but equals to 1.

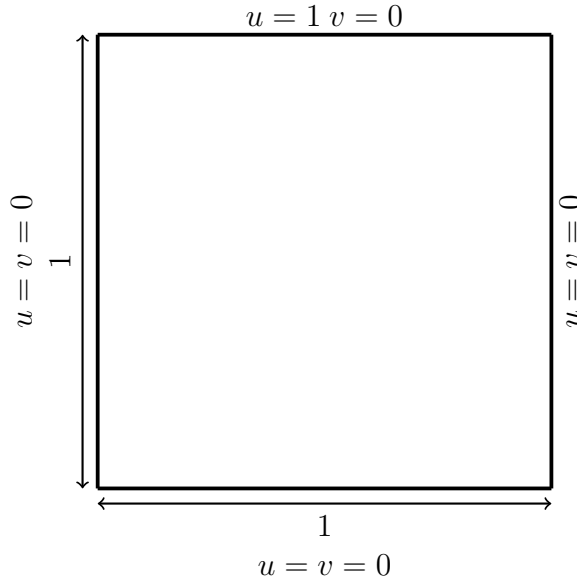


Figure 6.1: Lid-driven cavity: Geometry

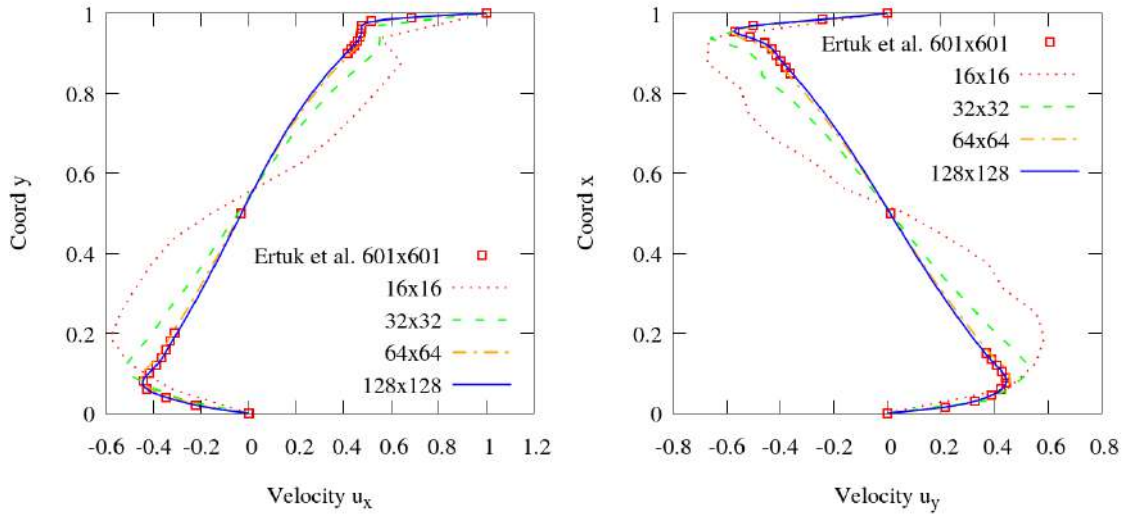
First, we perform a mesh sensitivity study to validate the capability of the method. Therefore, four different structured grids are used for these comparisons with a mesh of 16×16 , 32×32 , 64×64 , and 128×128 bilinear quadrilateral elements respectively. The Reynolds number is chosen to be equal to 5000, and we use a time-step equal to 0.1 until reach the steady-state. All numerical experiments are compared to the very known reference of ERTURK *et al.* [132]. In [132] the 2D steady incompressible Navier–Stokes equations were solved using a very efficient finite difference numerical method (fourth-order compact formulation) on a very fine grid of 601×601 .

The velocity profiles for u_x and u_y along $x = 0.5$ and $y = 0.5$ respectively employing different grid resolutions together with the reference solution are shown in Fig. 6.2. It is possible to see that the present method converges rather rapidly to the given benchmark solution.

Now we present the velocity profiles (Fig. 6.3 to Fig. 6.5) and streamlines (Fig. 6.6) of three situations, in which the Reynolds number is defined as 1000, 5000 and 10000. We use a fixed mesh of 160×160 bilinear quadrilateral elements and a time-step of 0.1. All simulations have good agreement with the numerical reference of ERTURK *et al.* [132].

6.1.2 Flow over a backward facing-step test case

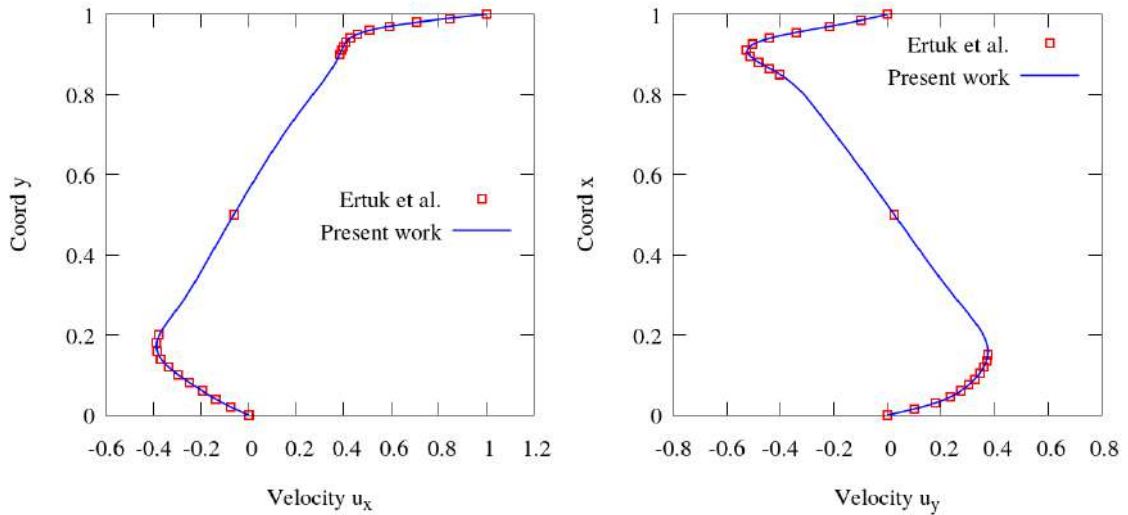
As a second test case, we consider a popular test problem for developing flow simulation codes, which is the flow over a backward facing-step. It consists of a fluid flowing in a straight channel, which abruptly widens on one side. The geometry and the boundary



(a) u_x along $x = 0.5$.

(b) u_y along $y = 0.5$.

Figure 6.2: Lid-driven cavity: Velocity profile using different meshes with $Re = 5000$.



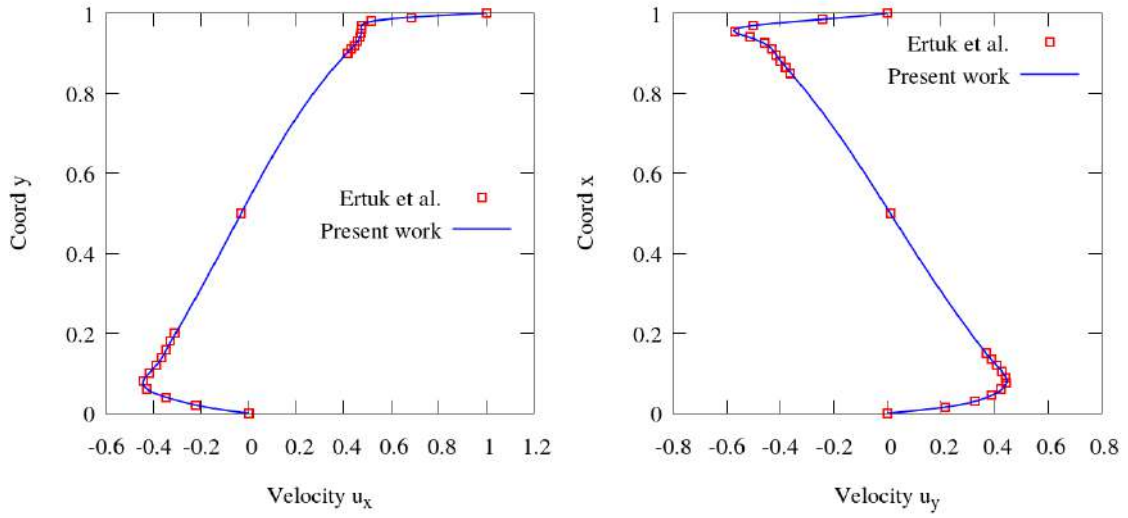
(a) u_x along $x = 0.5$.

(b) u_y along $y = 0.5$.

Figure 6.3: Lid-driven cavity: Velocity profile with $Re = 1000$.

conditions of the problem are shown in Figure 6.7. We apply the no-slip boundary condition in all boundaries, except for the inlet and outlet. At the inlet boundary, we apply an inflow velocity that has parabolic shape given by $u_{in} = 24y(0.5 - y)$, and at the outlet boundary, we apply nothing, representing the Outflow BC when we do not consider the hydrostatic pressure as a body force.

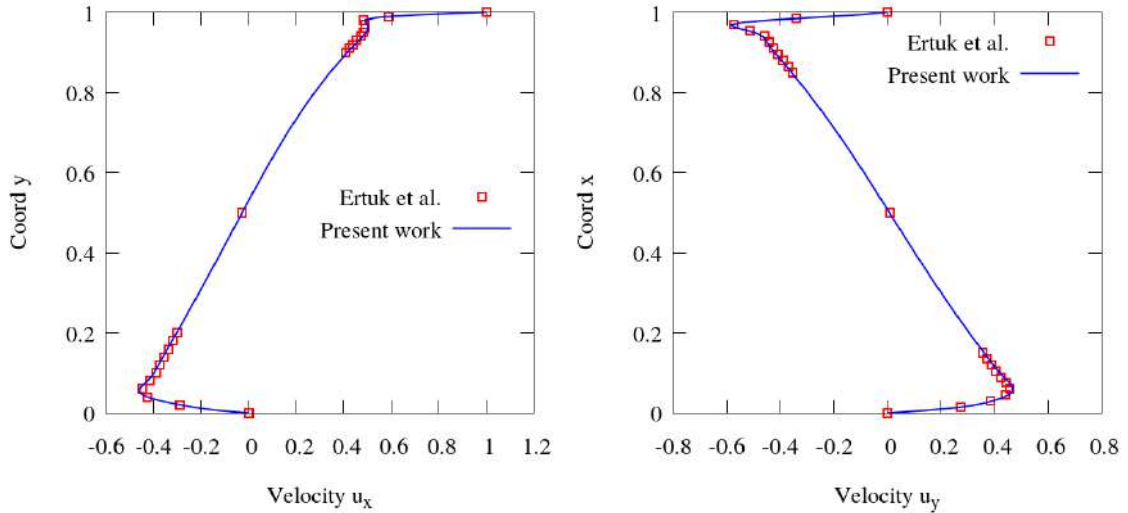
Once again, we perform a mesh sensitivity study. Three different structured grids are used for these comparisons with a mesh of 10×200 , 20×400 and 40×800 bilinear quadrilateral elements. The Reynolds number is chosen to be equal to 800, and



(a) u_x along $x = 0.5$.

(b) u_y along $y = 0.5$.

Figure 6.4: Lid-driven cavity: Velocity profile with $Re = 5000$.



(a) u_x along $x = 0.5$.

(b) u_y along $y = 0.5$.

Figure 6.5: Lid-driven cavity: Velocity profile with $Re = 10000$.

we use a time-step equal to 0.1 until reach the steady-state. All numerical experiments are compared to the very known reference of GARTLING [134], where the 2D steady incompressible Navier–Stokes equations were solved on a grid of 40×800 .

The velocity profiles for u_x along $x = 7$ and $x = 15$ employing different grid resolutions together with the reference solution are shown in Fig. 6.8. In Fig. 6.9 we also present the streamlines of the solution between $x = 0$ and $x = 10$.

We also test the Outflow BC in this problem, by considering the hydrostatic pressure as a body force term. The velocity profiles should be the same by considering the hydrostatic

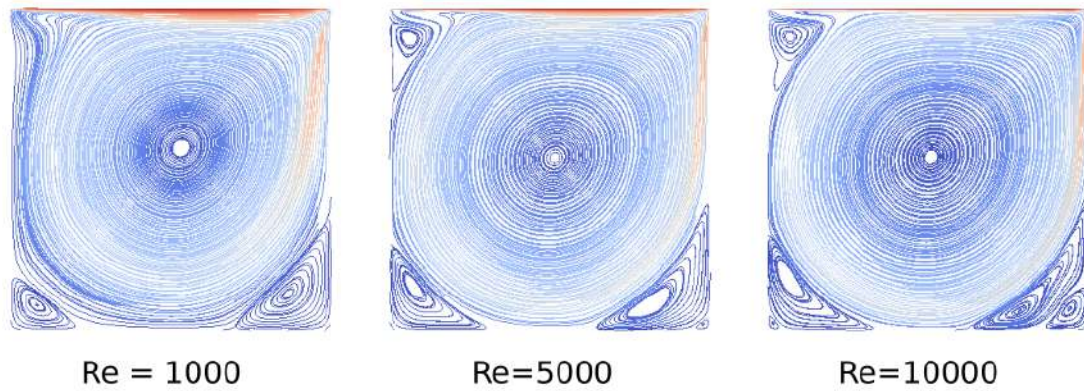


Figure 6.6: Lid-driven cavity: Streamlines for different Reynolds.

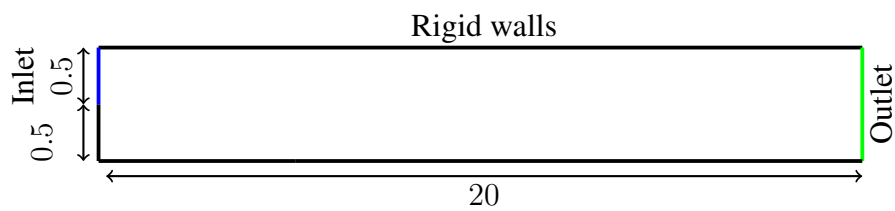


Figure 6.7: Backward facing-step: Geometry

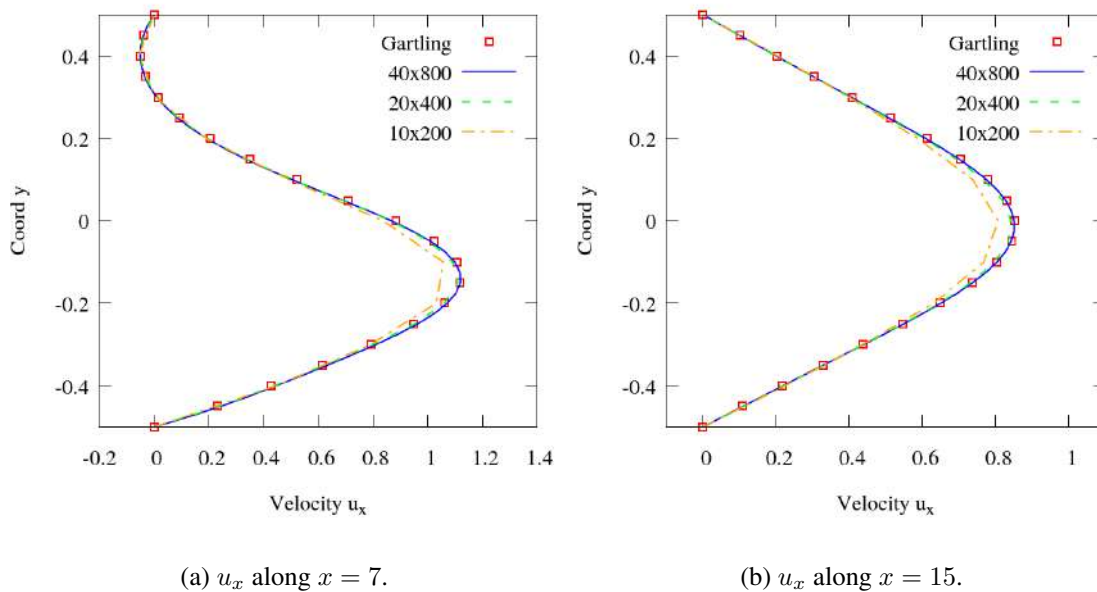


Figure 6.8: Backward facing-step: velocity profile using different meshes with $Re = 800$.

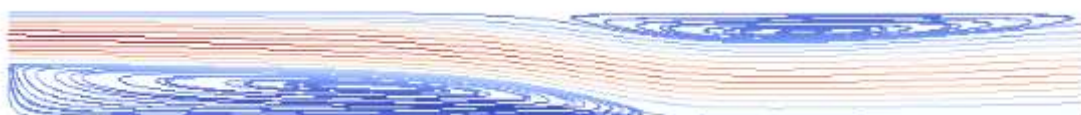


Figure 6.9: Backward facing-step: Streamlines for $Re = 800$ in $0 < x < 10$.

pressure or not. Therefore, we have to add the Outflow BC term at the outlet boundary to consider its influence, as explained in section 3.7. This condition works pretty well, although it generates a small perturbation of the pressure at the corners of the boundary, as shown in Fig. 6.10. The Outflow BC is important when working with two-fluid problems.

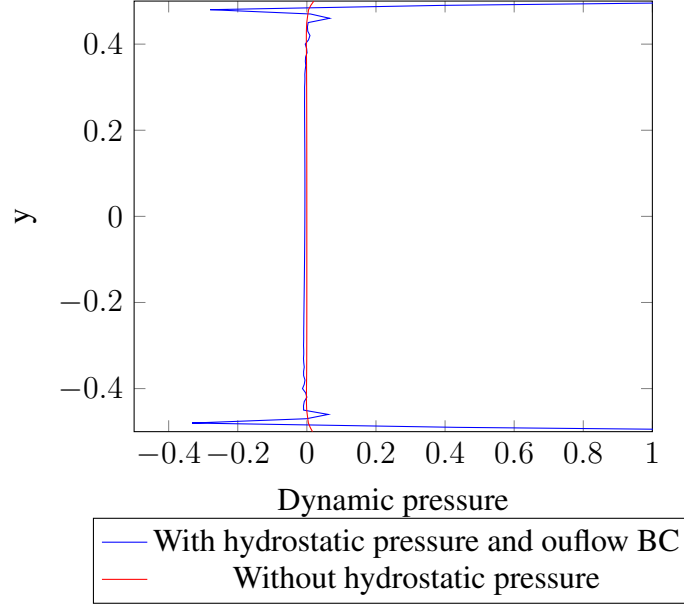


Figure 6.10: Backward facing-step: Dynamic pressure at $x=20$

6.2 Verification of the convected level-set

Before simulating two-fluid problems, it is important to verify if the interface motion is working correctly. In this section, we verify the implementation of the convected level-set method by solving simple convection problems. In all simulations, we assign the value of the penalty constant λ to 1, and we define a time-step low enough to properly reinitialize the SDF, by considering the value of λ and mesh size.

6.2.1 Pure convection of a circle

In this two-dimensional benchmark, we consider the rotation of a circle placed in a square computational domain of side equal to 1 and initially centered at $(0.25, 0.5)$ with a radius of 0.15. The imposed divergence-free velocity field \mathbf{u} is,

$$\begin{aligned} u &= -2\pi(y - y_{(Square\ Center)}) \\ v &= 2\pi(x - x_{(Square\ Center)}). \end{aligned} \tag{6.1}$$

This test case is used to verify different assumptions adopted in this work, as the temporal integration method and the importance of the global mass enforcement procedure.

To do that, a series of runs of this case with fixed mesh is performed. We also outline the importance of the mesh size and the corresponding time-step. In section 2.1.2, we have defined the time integration method for the convected level-set as the BDF2. Here, we compare the BDF2 with the Backward Euler (BE) method, which is given by Eq. (3.35). Even though we know the BDF2 method should provide better results than the BE because it has second-order accuracy while the BE is only first-order, we would like to outline that there is no extra computational cost by applying it. We also would like to show what is the difference between both methods with different mesh refinement and time-step. Applying the BE method to the convected level-set equation (Eq. 2.13) yields,

$$\frac{\phi_{n+1} - \phi_n}{\Delta t} + (\mathbf{u}_{n+1} + \lambda \mathbf{U}_n) \cdot \nabla \phi_{n+1} - \lambda \operatorname{sgn}(\phi_n) \left(\frac{1}{4E} - \frac{\phi_n^2}{E} \right) = 0. \quad (6.2)$$

Table 6.1 provides the data used for the simulations. We use the SDF presented in Eq. (2.2) with a thickness $E = 0.05$. The computational times were obtained in an Intel[®] Core[™] i5-8400 CPU @ 2.80GHz \times 6. In Figure 6.11, the shape's deformation between the initial and the final time are shown for all these simulations.

Table 6.1: Pure convection of a circle: Parameters of the different simulations performed.

Case	Mesh Grid	time-step (s)	CPU time (s)	
			(BE)	(BDF2)
(a)	50 \times 50	0.01	10	10
(b)	50 \times 50	0.005	20	17
(c)	50 \times 50	0.0025	30	31
(d)	100 \times 100	0.005	66	66
(e)	100 \times 100	0.0025	117	118
(f)	100 \times 100	0.001	265	266
(g)	200 \times 200	0.001	961	990
(h)	200 \times 200	0.0005	1096	1041

The most challenging feature for a good interface-capturing method resides in its ability to preserve the mass of the species involved. In these first simulations, we do not apply the global mass conservation routine, presented in section 2.1.4, aiming to observe what happens by using this method without a mass conservation enforcement. The mass conservation is given by the relative error between the final area of the deformed circle and the initial area. Figure 6.12 shows the percentage of area loss per time of the simulations given by Table 6.1. It is possible to see that the BDF2 method provides more accurate results in all simulations, and there is no extra computational cost by using it. The mass conservation is also dependent on the mesh size. Refined meshes lose less mass than the coarse ones. The area loss of the BE method (Fig. 6.12a) is significantly larger than the area loss of the BDF2 method (Fig. 6.12b). That is why we chose the BDF2 method for

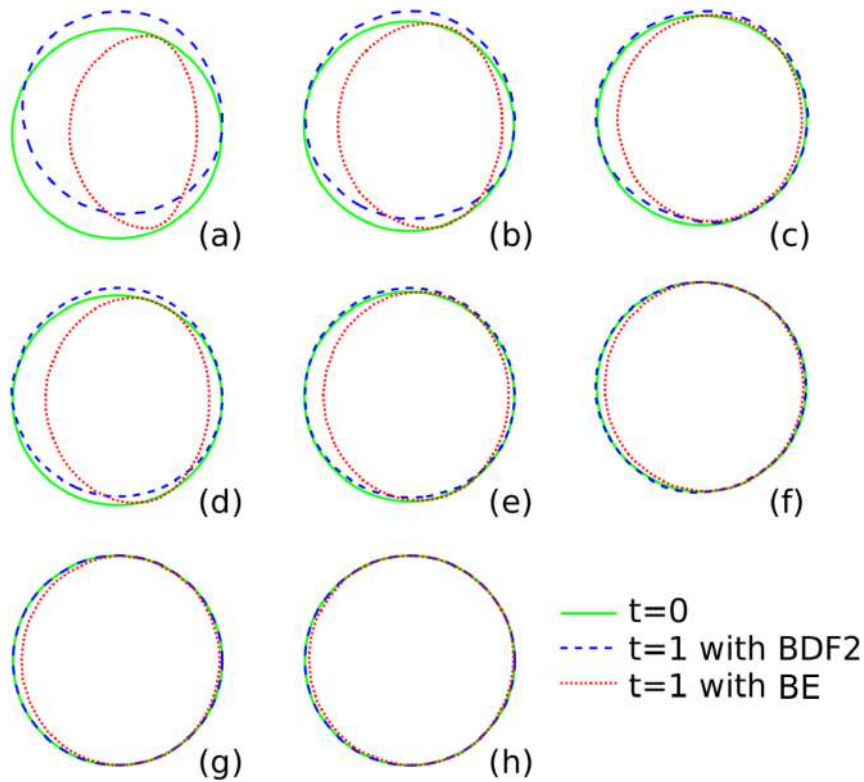


Figure 6.11: Pure convection of a circle: Circle deformation on different meshes, with different time-steps and without the global mass conservation procedure. In the simple green line: the initial zero iso-value of the distance function. In the blue dashed line: the same iso-value at the final time-step (after one turn) with the BDF2 time discretization. In the red dotted line: the same iso-value at the final time-step (after one turn) with the BE time discretization. (a)-(h) correspond to values given in Table 6.1

the time integration of the convected level-set method.

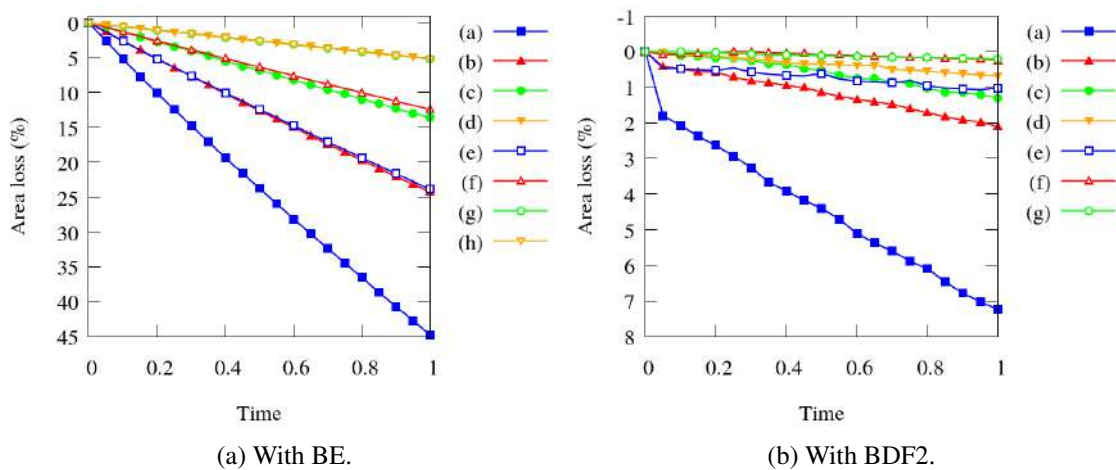


Figure 6.12: Pure convection of a circle: Area loss per time of all simulations [(a)-(h)] given in Table 6.1

In Fig. 6.13, we present the shape's deformation between the initial and the final time

when using the global mass conservation procedure presented in section 2.1.4. There is no longer mass loss between phases. However, these runs indicate the dependency of the accuracy of the convected level-set method on the mesh size. The coarser the mesh, the more distorted the interface becomes. Moreover, refining the domain implies a smaller time-step for a better precision.

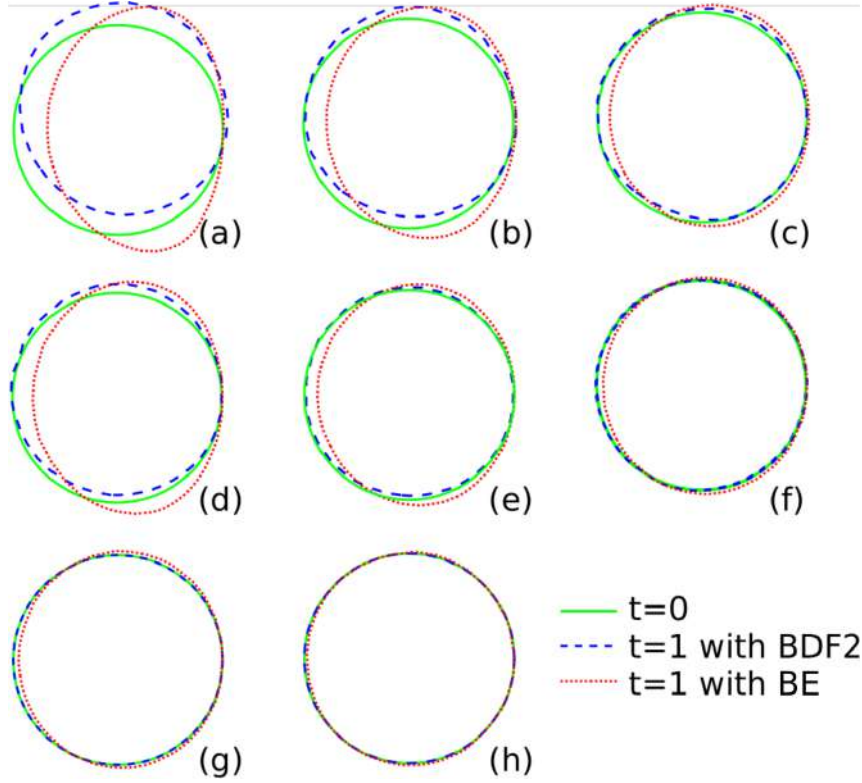


Figure 6.13: Pure convection of a circle: Circle deformation on different meshes, with different time-steps and with the global mass conservation procedure. In the simple green line: the initial zero iso-value of the distance function. In the blue dashed line: the same iso-value at the final time-step (after one turn) with the BDF2 time discretization. In the red dotted line: the same iso-value at the final time-step (after one turn) with the BE time discretization. (a)-(h) correspond to values given in Table 6.1

Finally, we present this simulation using AMR, together with the BDF2 method and with the global mass conservation procedure. We use an adaptive mesh refinement based on the flux jump of the level-set function error, in which $h_{max(LS)} = 2$, the coarsening fraction $c_f = 0.01$ and the refining fraction $r_f = 0.99$. We apply the adaptive mesh refinement every 50 time-steps. The initial mesh has 50×50 bilinear quadrilateral elements, and after the refinement, the smallest element has size 0.005. We initially refine in two levels the region where the circle is located. The time-step is 0.0005. Figure 6.14 shows the simulation results at different steps during one revolution of the circle. The zero level-set is highlighted, and it is possible to see the mesh adaptivity. Note that after one turn, the circle keeps its shape. The computational time for the result with AMR is 711 seconds. Therefore, by using AMR we save computational effort, and we obtain the

same accuracy of the results of the run (h) that took 1096 seconds.

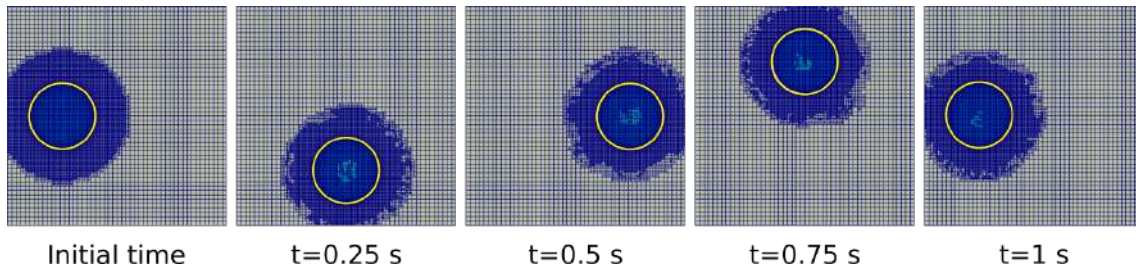


Figure 6.14: Pure convection of a circle: Position of the circle and mesh refinement at different time-steps during one turn.

In Figure 6.15, we show what happens when we consider only the pure convection parcel of the convected level-set equation to capture the interface motion, i.e., $\lambda = 0$, aiming to understand the importance of the reinitialization. After one turn, even that the zero iso-value keeps its shape close to a circle, the other iso-values has distortions and become ellipses when using the BE method (Fig. 6.15b). On the other hand, the BDF2 method preserves better the iso-values shape, even though they still are not perfect circles (Fig. 6.15c). Figure 6.15d shows the iso-values using the complete convected level-set. It is possible to see how the reinitialization adjusts the level-set field to keep it an SDF. We also conclude that the BDF2 preserves the iso-values shape better than the BE, and for this reason, we only use the BDF2 for the time integration of the convected level-set method in the next simulations.

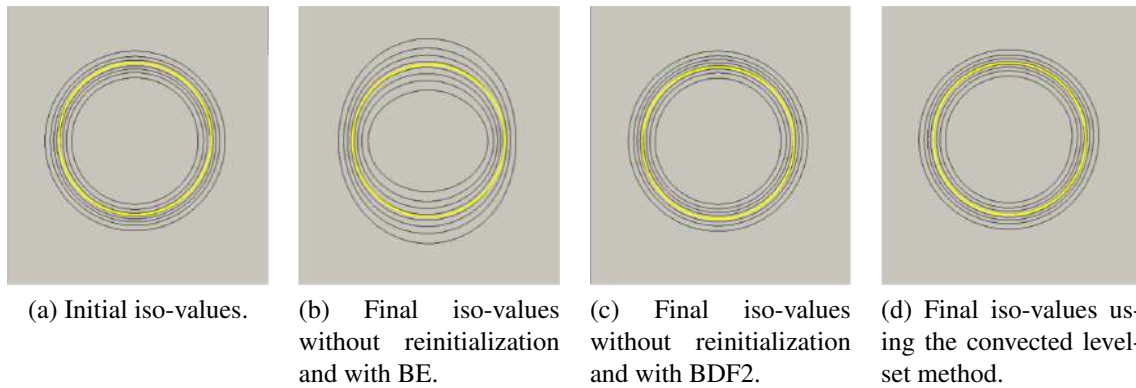


Figure 6.15: Pure convection of a circle: iso-values when using the reinitialization or not.

6.2.2 Advection of a sphere

Now, we consider a three-dimensional problem that is the rotation of a sphere. The sphere is initially centered at $(0.25, 0.5, 0.5)$, the side of the computational domain is set to

1, with a radius of 0.15. The imposed velocity field is,

$$\begin{aligned} u &= -2\pi(y - y_{(Square\ Center)}) \\ v &= 2\pi(x - x_{(Square\ Center)}) \\ w &= 0 \end{aligned} \tag{6.3}$$

We use an adaptive mesh refinement with $h_{max} = 2$, $c_f = 0.01$ and $r_f = 0.99$. We apply the AMR every 25 time-steps. The initial mesh has $25 \times 25 \times 25$ trilinear hexahedral elements, and after the refinement, the smallest element has size 0.01. We initially refine in two levels the sphere region, as we can see in Figure 6.16. The time-step is 0.001 seconds, and we choose a thickness $E = 0.025$. Figure 6.17 shows the simulation at different steps during one rotation. The result shows that the convected level-set method preserves a good interface shape for three-dimensional problems as well.

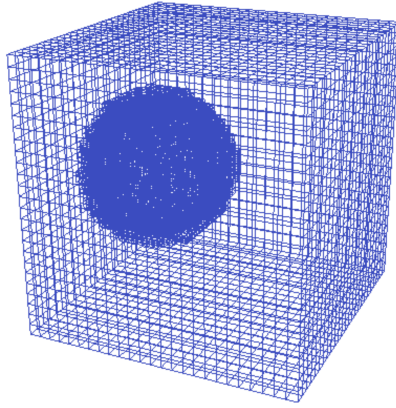


Figure 6.16: Pure convection of a sphere: Initial mesh refinement of the advection of a sphere problem.

6.2.3 Zalesak's problem

The Zalesak's problem consists of the rotation of a slotted disk under a given velocity field [135]. The side of the squared computational domain is 1.2, and the disk is initially centered in $(0.3, 0.6)$ with a radius of 0.2 and a slot length of 0.35 (Fig. 6.18). The velocity field is also given by Eq. (6.1).

We choose a thickness $E = 0.0125$ and we use an adaptive mesh refinement with $h_{max(LS)} = 3$, $c_f = 0.01$ and $r_f = 0.99$. We apply the adaptive mesh refinement every 50 time-steps. The initial mesh has 60×60 bilinear quadrilateral elements, and after the refinement, the smallest element has size 0.0025. We initially refine in three levels the circle region. This example needs more refinement than the previous ones because of the sharp corners of the disk. The time-step is 0.0001. Figure 6.19 shows the simulation results at different steps during one revolution. The zero level-set is highlighted, and it

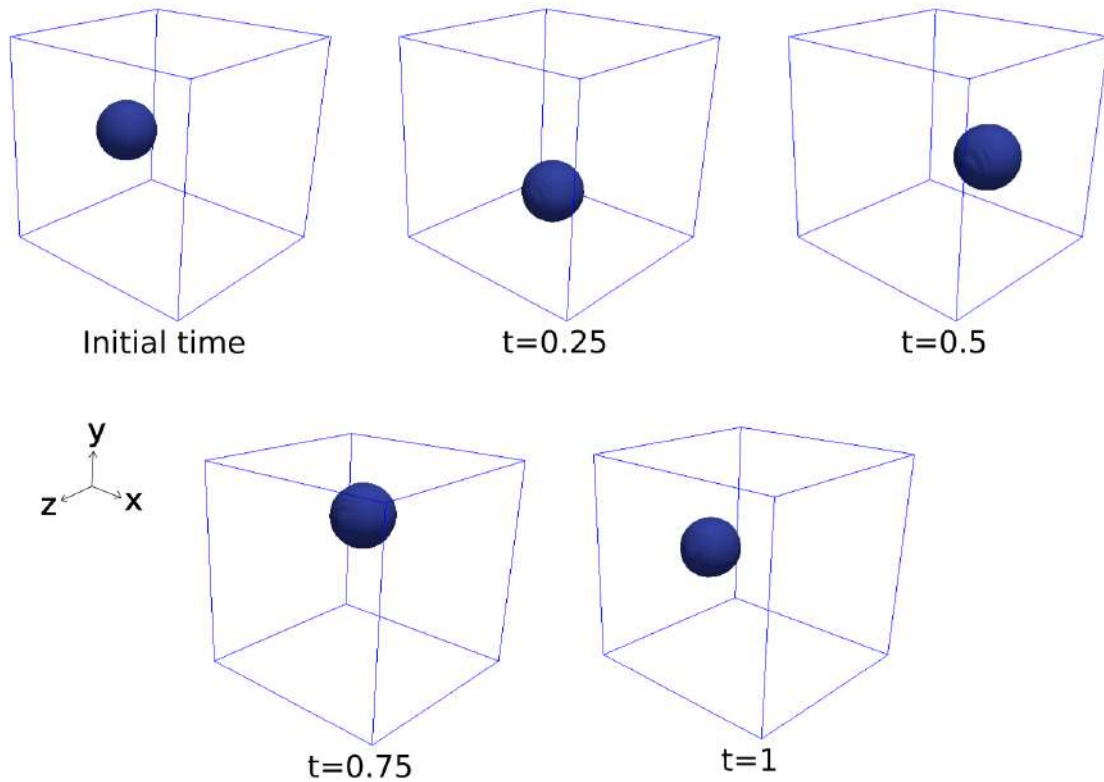


Figure 6.17: Pure convection of a sphere: Position of the sphere during one turn.

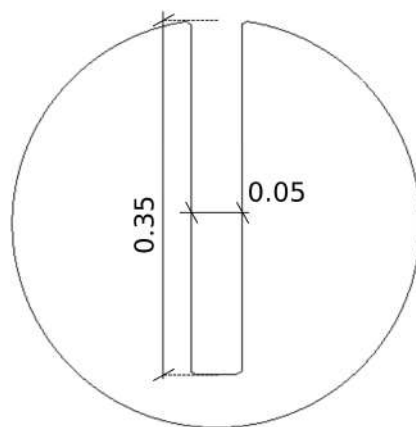


Figure 6.18: Zalesak's problem: Configuration.

is possible to see the mesh adaptivity. In Fig. 6.20, we compare the initial configuration with the final one after one turn, and the interface shape is preserved.

6.2.4 Single vortex problem

The single vortex problem was firstly introduced by BELL *et al.* [136] as a more severe test case than the Zalesak's problem for testing the ability of the numerical scheme to resolve thin filaments. The problem consists of a circle initially centered at $(0.5, 0.75)$

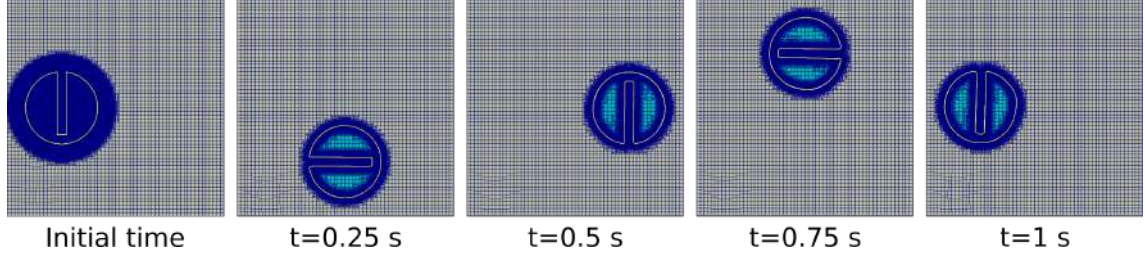


Figure 6.19: Zalesak's problem: Position of the disk and mesh refinement at different time-steps during one turn.

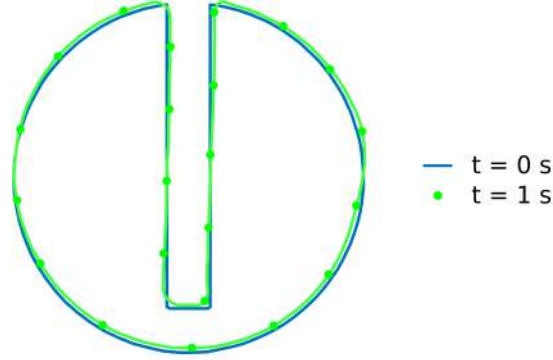


Figure 6.20: Zalesak's problem: Initial disk configuration (blue line) and final configuration after one turn (green line with markers)

in a unit square domain, and it is advected by a velocity field given by the following stream function,

$$\Psi = \frac{1}{\pi} \sin^2(\pi x) \sin^2(\pi y) \quad (6.4)$$

which leads to the velocity field,

$$u = \frac{\partial \Psi}{\partial y} = \sin(2\pi y) \sin^2(\pi x) \quad (6.5)$$

$$v = \frac{\partial \Psi}{\partial x} = \sin(2\pi x) \sin^2(\pi y). \quad (6.6)$$

For the purpose of error analysis, LEVEQUE [137] suggested to multiply the velocity field by

$$g(t) = \cos(\pi t/T) \quad (6.7)$$

where T is the period at which the flow returns to its initial state. Following other authors [89, 138], we adopt a period $T = 8$, but we also show the results using $T = 4$.

We choose a thickness $E = 0.0078125$ and we use an adaptive mesh refinement with $h_{max(LS)} = 4$, $c_f = 0.01$ and $r_f = 0.99$. We apply the adaptive mesh refinement every 50 time-steps. The initial mesh has 80×80 bilinear quadrilateral elements, and after the refinement, the smallest element has size 0.0015625. We refine the initial region in

four levels. This level of refinement is important to reproduce the thin “tail” that arises after a few simulation instants. The time-step is 0.0025. Figures 6.21 and 6.22 show the simulation results at different steps during one revolution of the single vortex problem for $T = 4$ and $T = 8$, respectively. The zero level-set is highlighted, and it is possible to see the mesh adaptivity. In Fig. 6.23, we compare the initial configuration with the final one after one period. In the simulation with $T = 8$, the “tail” becomes thinner than for the simulation with $T = 4$, and the mesh used is not refined enough to capture this thickness, which distorts the final interface shape. For better results, we should use an even more refined mesh in that region. On the other hand, the level of refinement used is enough to provide good results with $T = 4$, and after one turn, the interface shape is preserved.

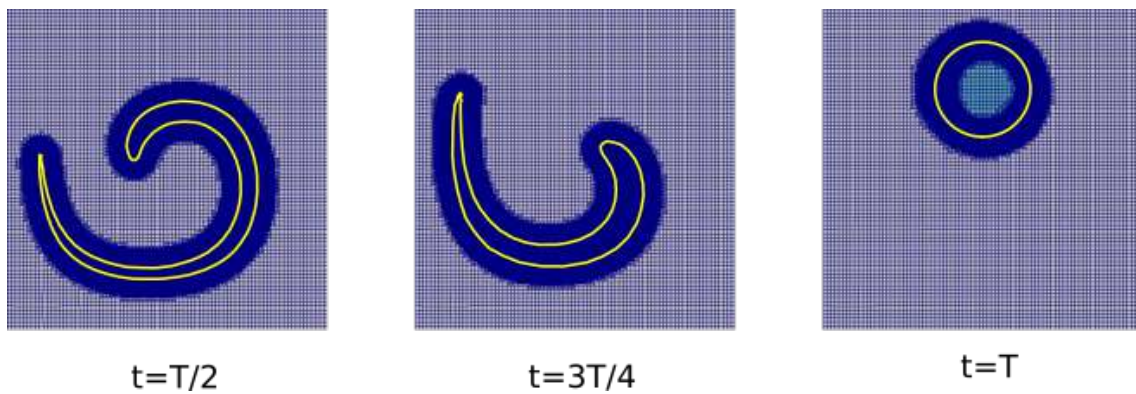


Figure 6.21: Single vortex problem: Vortex deformation and mesh refinement at different time-steps for $T = 4$.

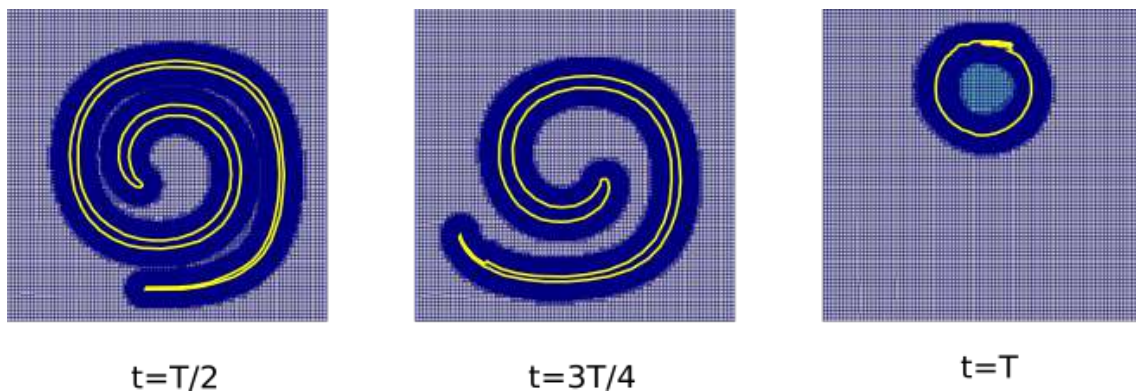


Figure 6.22: Single vortex problem: Vortex deformation and mesh refinement at different time-steps for $T = 8$.

6.3 Verification of the surface tension model

In this section, we verify the implementation of the surface tension model. First, we check the calculation of the surface tension force in a given contour, in which we have the

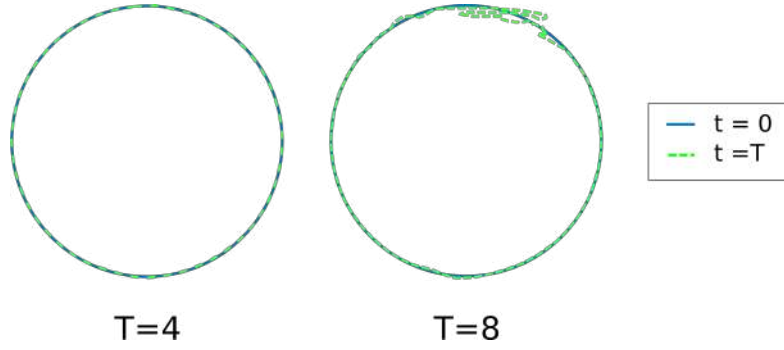


Figure 6.23: Single vortex problem: Initial configuration (blue line) and final configuration after one period (dashed green line).

analytical response based on the geometry of the problem. Then, we check a two-fluid problem in the absence of gravity, where we can verify the pressure resulting from the surface tension.

6.3.1 Film-like interface

In this test case, we want to check the calculation of the surface tension force in a given contour. We reproduce the contour used by SHEPEL e SMITH [139], shown in Fig. 6.24. It consists of four 180° arcs, two large arcs of radii R_o and R_i and two small arcs of radius $(R_o - R_i)/2$. The surface tension coefficient is set to $\sigma_{st} = 0.1$ and the radii are set to $R_o = 1.0625$ and $R_i = 0.9375$.

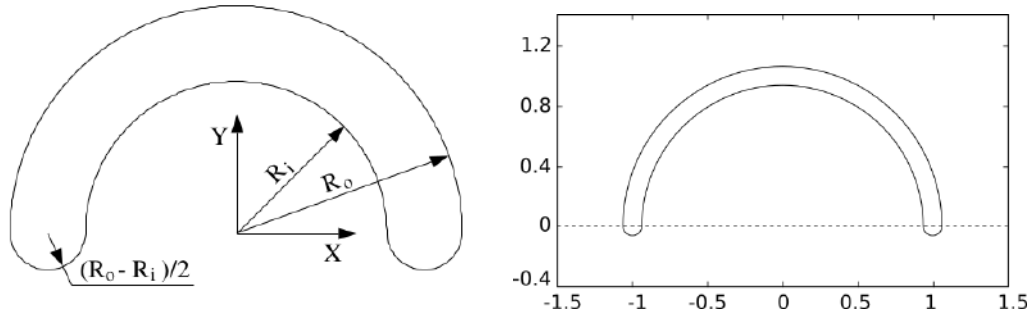


Figure 6.24: Film-like interface: Geometry of the closed contour.

The computational domain is divided into two parts by a horizontal line passing through the origin of the coordinate system and the points where the arcs are connected. Therefore, we denote the surface tension force in the upper part of the domain, i.e., for $y \geq 0$, as \mathbf{F}_{st}^+ and in the lower part of the domain, $y \leq 0$, as \mathbf{F}_{st}^- .

As the contour is symmetric with respect to the y -axis, the horizontal component of \mathbf{F}_{st} is not considered in the analysis, whereas the vertical component is calculated as follows: $F_{st}^y = h^2 \sum_{i,j} \mathbf{J} \cdot \mathbf{F}_{st}(\mathbf{x}_i, j)$, where $\mathbf{J} = (0, 1)$ is the unit vector along the y -axis.

The exact solution to this problem obtained by contour integration is given by $F_{st}^{y+} = -4$ and $F_{st}^{y-} = +4$.

The contour is modeled using a thickness $E = 0.1$. We define the half-thickness of the band where the surface tension force is distributed equally to $\epsilon = 5h_e$, being h_e the element size. Table 6.2 shows the results of a convergence study for successively refined uniform grids. The errors are calculated as $E^+ = (4 + F_{st}^{y+})/4$ and $E^- = (4 - F_{st}^{y-})/4$. It is possible to see the importance of mesh refinement to achieve good results.

Table 6.2: Film-like interface: Integration errors for the surface tension force on the studied closed contour with successively refined uniform grids.

	h_e			
Error	0.05	0.025	0.0125	0.00625
$ E^+ $	8.64E-02	3.78E-02	1.91E-04	5.10E-05
$ E^- $	1.23E-01	1.84E-02	2.78E-03	6.15E-04

6.3.2 Pressure inside a bubble due the surface tension

The surface tension induces an excess of pressure inside bubbles which, for a spherical shape, can be calculated from the Youngs-Laplace equation,

$$\Delta p = \frac{2\sigma_{st}}{R}. \quad (6.8)$$

Here, we present a simulation of a drop of water with a radius $R = 2$ cm in stagnant air in the absence of gravity. Under these conditions, the drop should acquire a spherical shape and remain in equilibrium. The data given in Table 6.3 is used for this simulation. The free-slip boundary condition is used at all boundaries. We use two different meshes (case A and case B) to compare the grid influence for this test case. Although we obtain the pressure solution with only one time-step, we run this simulation during a larger period to confirm that the surface tension keeps the bubble in a spherical shape.

According to Laplace's formula, the exact solution to this problem is a discontinuous pressure distribution equals to 0 outside the drop and 7.2 Pa inside the drop, with a jump at the drop surface. In Fig. 6.25, we plot our results and compare them with the exact solution. It can be seen that the excess pressure inside the bubble is predicted with reasonable accuracy in both cases (deviation from the analytical solution is 3.83% for case A and 2.45% for case B), which indicates that the surface tension model has been implemented correctly. Moreover, the spherical bubble shape is maintained. Comparing the results of cases A and B, the more refined mesh provides better results.

Table 6.3: Pressure inside a bubble due the surface tension: Data.

Computational domain	$0.08 \times 0.08 \times 0.08$	m
Grid sizes	(A) 0.004 / (B) 0.002	m
Number of time-steps	1000	(-)
Time-step	0.001	s
Bubble radius	0.02	m
Initial bubble position	$(x, y, z) = (0.04, 0.04, 0.04)$	m
Liquid density	1000	kg/m ³
Liquid viscosity	10^{-3}	kg/(ms)
Gas density	1.163	kg/m ³
Gas viscosity	1.85^{-5}	kg/(ms)
Surface tension	0.072	N/m
Gravity	0.0	m/s ²

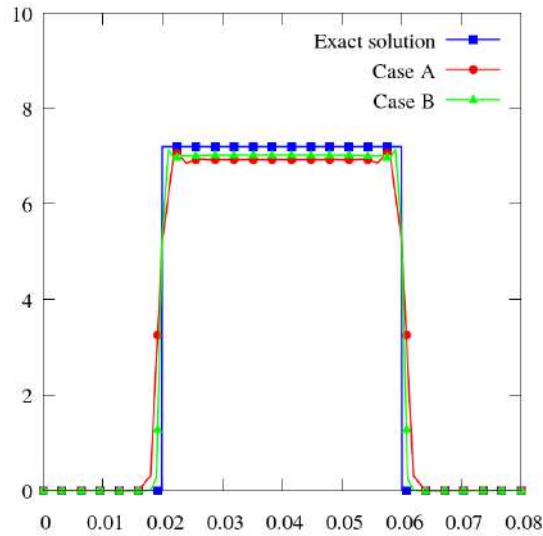


Figure 6.25: Pressure inside a bubble due the surface tension: Results.

6.4 Verification of the moving mesh method

In this section, we verify the implementation of the moving mesh method. As mentioned before, the ALE framework is used in sediment-fluid problems, and the morphological changes are represented as mesh movement. First, we verify the algorithm of mesh movement. Then, we verify the upwind sand-slide mechanism, developed to limit the slope angle to the sediment angle of repose.

6.4.1 Mesh movement

In sediment transport problems, the movement of the sediment boundary is essential, playing the main role in the resolution of these problems. The desired result is the final

profile of the sediment bed, and this is represented by mesh movement in this work.

To verify the mesh movement algorithm, we use a rectangular domain with 10×50 bilinear quadrilateral elements, and we apply mesh movement boundary conditions as follows: we fix the superior part of the rectangular domain, $\mathbf{d} = \mathbf{0}$ on Γ_f and we apply a sinusoidal displacement that depends on time at the bottom boundary, $\mathbf{d} = -\mathbf{C}_M \sin(t)$ on Γ_m , where \mathbf{C}_M is a vector with all entries equal to a positive constant C_M . The laterals are free to move. In Figure 6.26, we can see the domain at different times. It is possible to note that all elements were deformed in the same way. The diffusion applied was dependent on the area of the elements, and since all elements have the same area, the displacements were the same.

Now, we take the same domain and mesh movement boundary conditions, but we refine the middle south part of the rectangular elements. In this case, the displacements of the biggest elements are bigger than the small ones (Figure 6.27). Usually, the boundaries are refined. Therefore this method of calculating the diffusion is good to keep the shape of the small elements (at boundaries) while the biggest ones are more deformed.

As a last mesh movement test, we would like to show the importance of the choice of a gravity acceleration direction diffusion, especially when there are displacements in opposite directions. For this test, the displacements at the bottom are,

$$\begin{aligned} \text{if } 2 < \text{coord}(x) \leq 2.5 \quad d &= -C_M \sin(t) \\ \text{if } 2.5 < \text{coord}(x) \leq 3 \quad d &= C_M \sin(t) \\ \text{otherwise } d &= 0 \end{aligned} \tag{6.9}$$

where $\text{coord}(x)$ is the coordinate in x direction. Figure 6.28 compares the mesh when using a motion diffusion only in the gravity acceleration direction, with an anisotropic motion diffusion.

When there are displacements in opposite directions, an anisotropic diffusion tends to cause large deformations in the mesh, as it is possible to see in Figure 6.28. These large deformations may make the system unstable, and the simulation might diverge. On the other hand, by using a diffusion applied only in the gravity acceleration direction, only a few elements are distorted, and more elements absorb the displacements. In sediment transport simulations, displacements in opposite directions are expected, and that is why we choose a diffusion applied only in the gravity acceleration direction.

6.4.2 Upwind sand-slide

In order to test the upwind sand-slide mechanism, a heap of sand with initial slopes larger than a pre-determined critical bed slope is used as a test case. This test case is de-

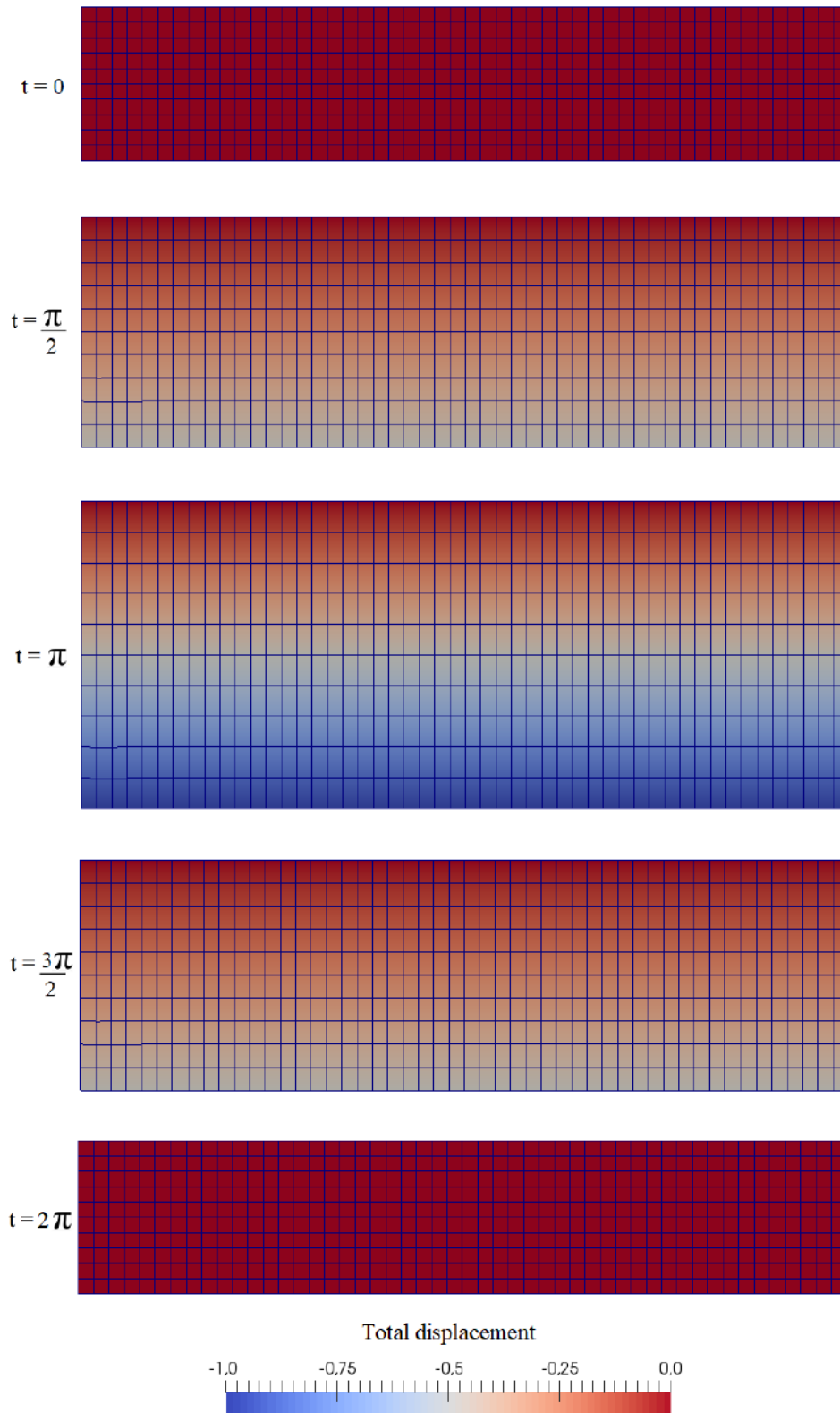


Figure 6.26: Mesh movement: displacements when the elements have the same area.

signed to test only the upwind sand-slide mechanism. Thus no sediment transport models are included in the simulation.

A sketch of the case can be seen in Fig. 6.29. The left slope angle is 40° , and the right

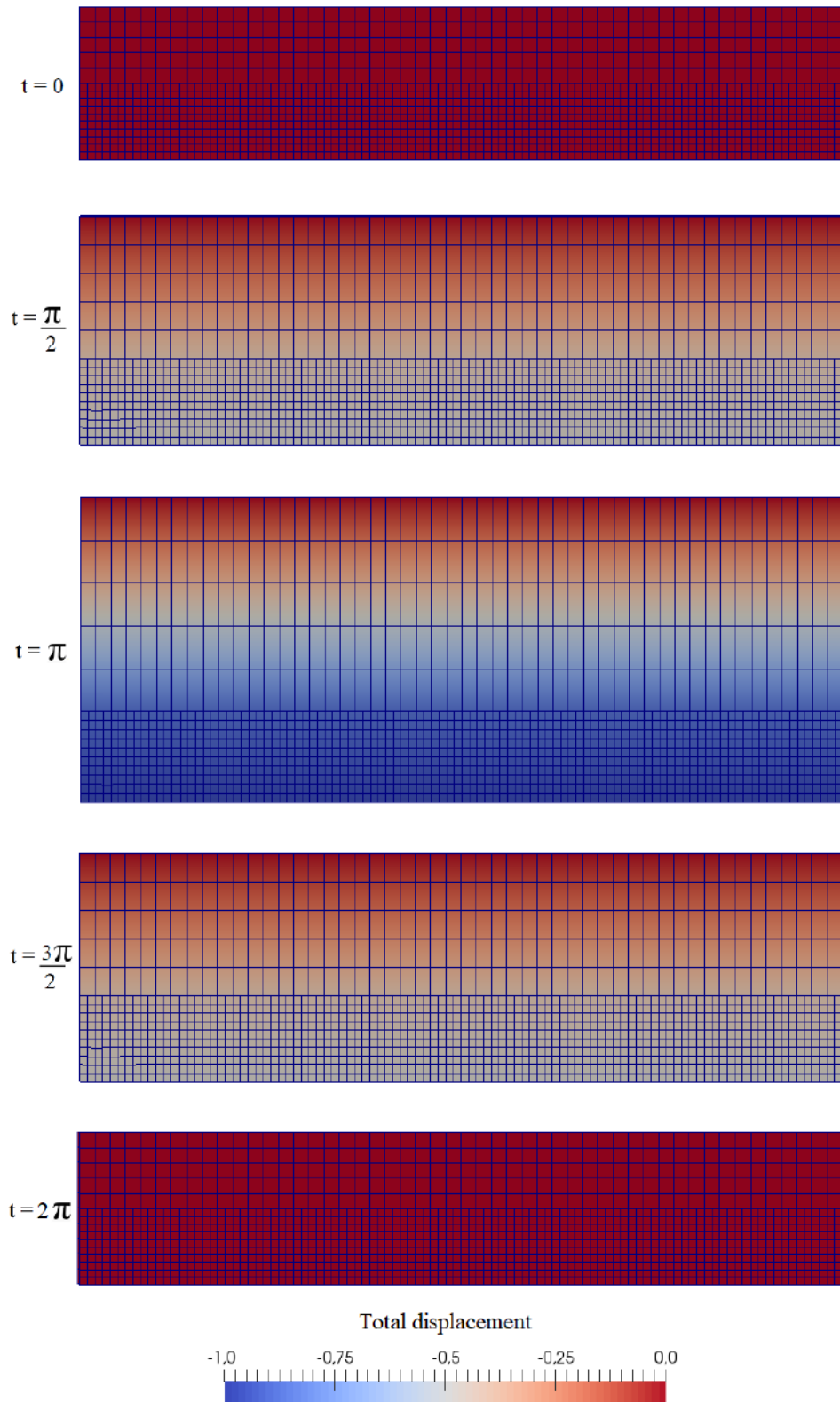


Figure 6.27: Mesh movement: displacements when the elements have different areas.

slope angle is 35° . The critical bed slope is set to be 30° . Therefore both slopes need to be corrected although the correction is different on each side.

The computational mesh for this test case is two-dimensional with 750 bilinear quadri-

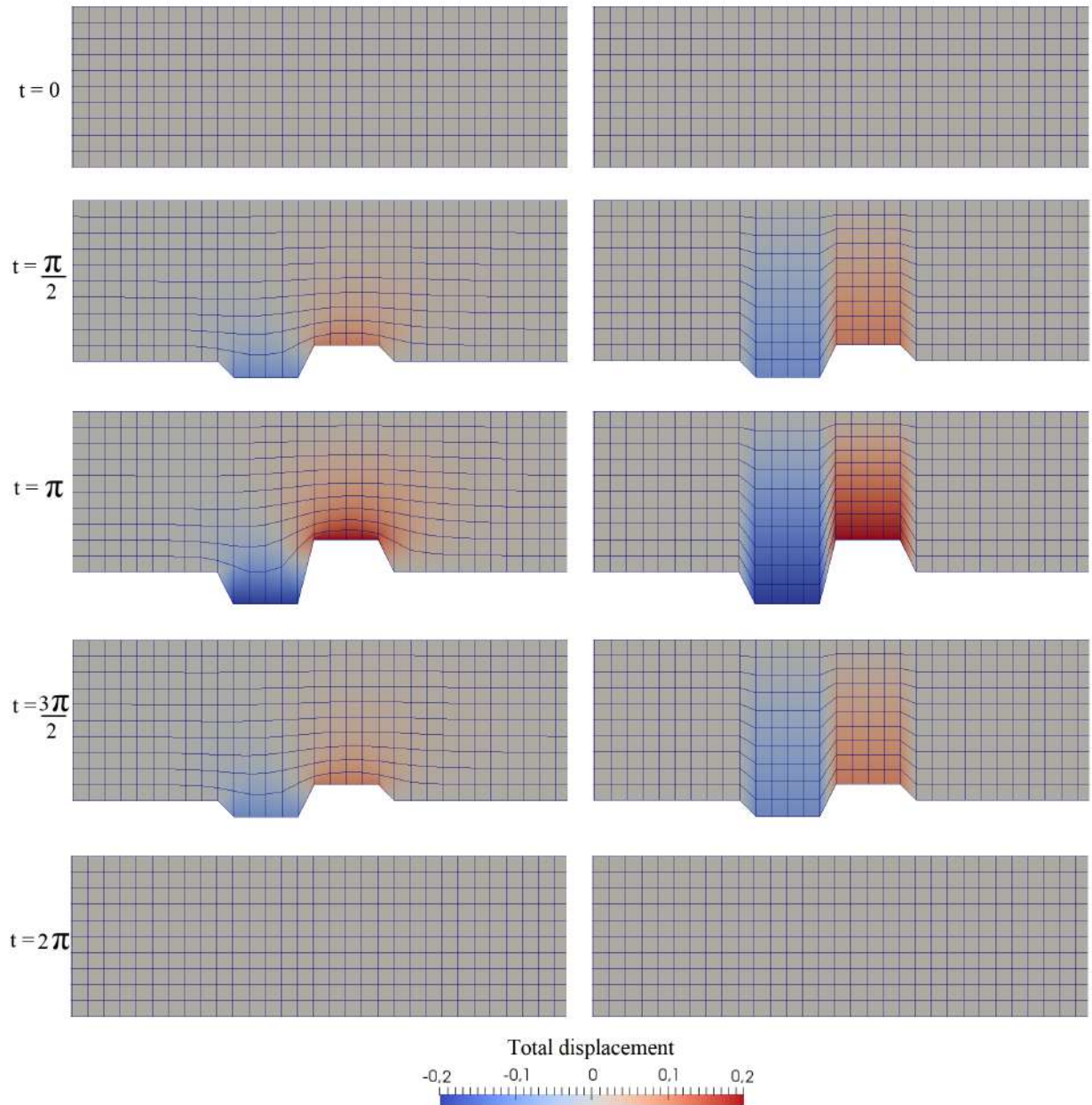


Figure 6.28: Mesh movement: comparison between an anisotropic diffusion (left) and a isotropic diffusion in the gravity acceleration direction (right).

laterals elements. The boundary conditions for the mesh displacements are fixed to zero for all the boundaries except for the bottom boundary where the values are given by the upwind sand-slide mechanism. We present three situations: (A) without a flow field. (B) the flow comes from the left to the right side of the domain. (C) the flow comes from the right to the left side of the domain. Here, we are still not applying the Laplacian smoothing.

The displacements of the mesh after the procedure are shown in Figures 6.30 to 6.32. The initial and final geometry of each case are shown in Figures 6.33 to 6.35. It is possible to see the influence of the flow field in the final geometry. If we are not considering any

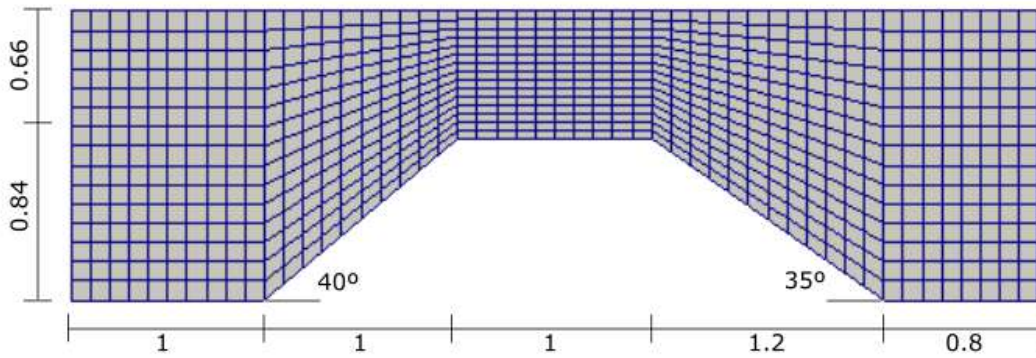


Figure 6.29: Upwind sand-slide: Geometry and mesh

flow field as the test case A, the correction can be done using the classical smoothing techniques, which rotate the element boundary around its gravity center until the slope becomes lower or equal to the sediment angle. On the other hand, in the test cases B and C, we apply the upwind sand-slide procedure, considering the direction of the flow. Note how the geometry changes and follows the direction of the flow.

It is important to check if the area of the domain is the same in all geometries because we do not want to lose or to gain sediments with this process. In this test case, the initial and final domains have an area of 5.736, which means that the mass is conserved. This test demonstrates that the upwind sand-slide mechanism works correctly to restrict the bed slope to the critical bed slope of the sediment, considering the direction of the flow.

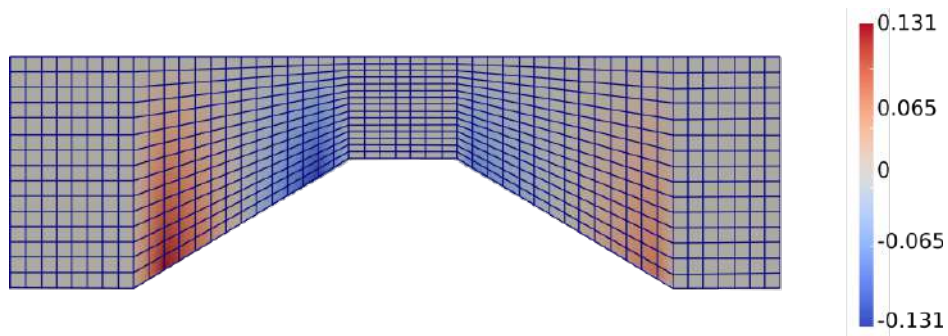


Figure 6.30: Upwind sand-slide: Mesh displacements of case A

Now, we show the influence of the Laplacian smoothing used to avoid sharp corners and small oscillations. We repeat the three test cases considering the smoothing technique only when a certain quantity of sediment moves. Therefore, it is possible to find the equilibrium geometry with the Laplacian smoothing. The new geometries are shown in Figures 6.36 to 6.38. The smoothing has rounded the corners and only affects the regions where the boundary angle changes.

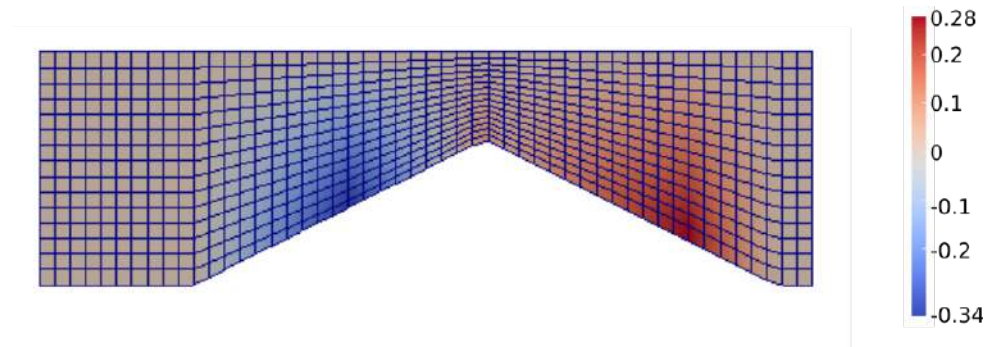


Figure 6.31: Upwind sand-slide: Mesh displacements of case B

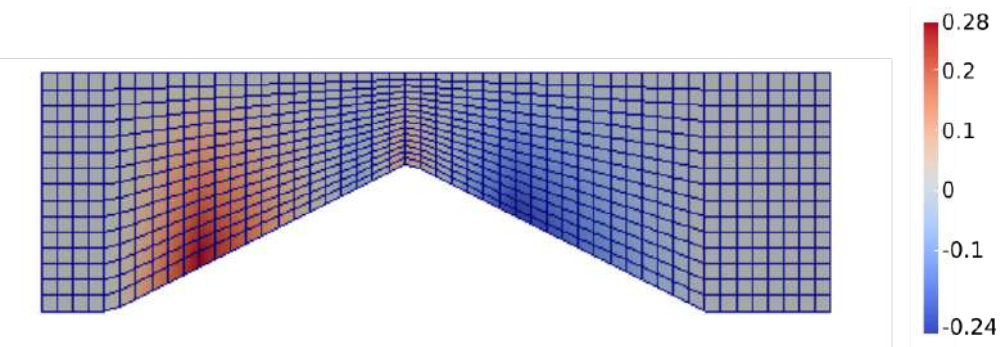


Figure 6.32: Upwind sand-slide: Mesh displacements of case C

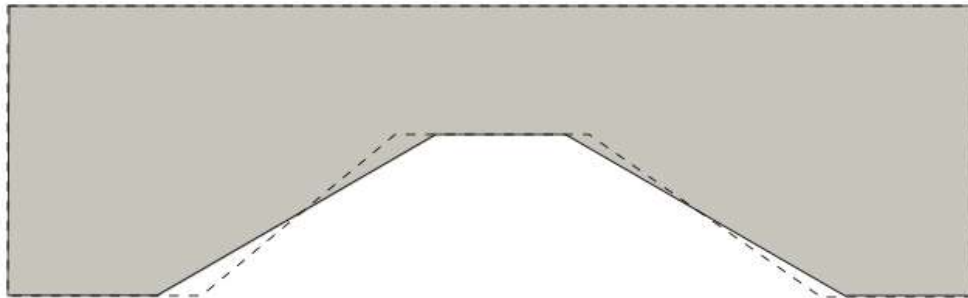


Figure 6.33: Upwind sand-slide: Initial (dashed lines) and final geometry of case A

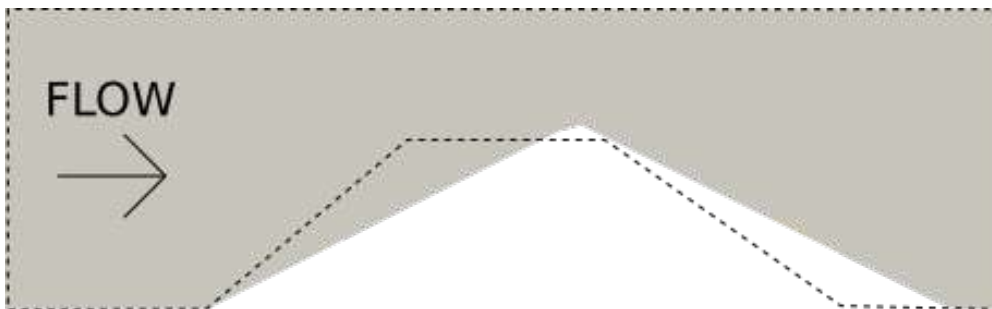


Figure 6.34: Upwind sand-slide: Initial (dashed lines) and final geometry of case B

6.5 Two-fluid simulations

For the next simulations, we include the effects of the gravity acceleration by simulating the rise of gas bubbles in quiescent viscous liquids. These are the complete simu-

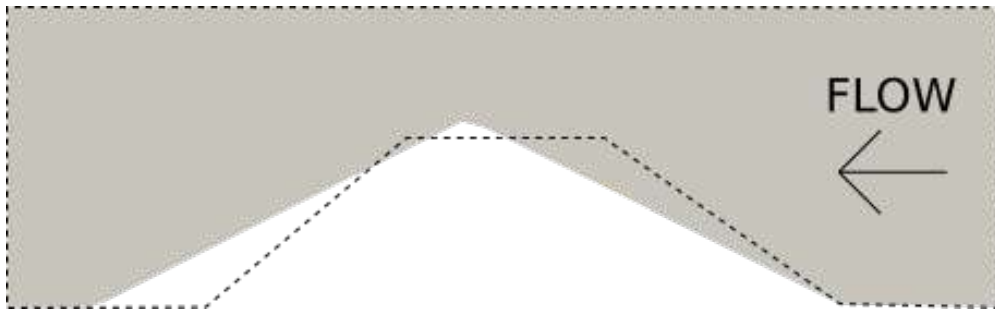


Figure 6.35: Upwind sand-slide: Initial (dashed lines) and final geometry of case C

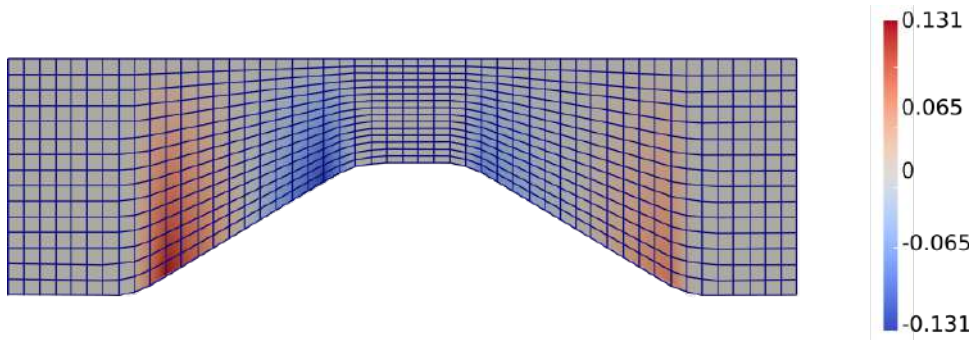


Figure 6.36: Upwind sand-slide: Mesh displacements with smoothing of case A

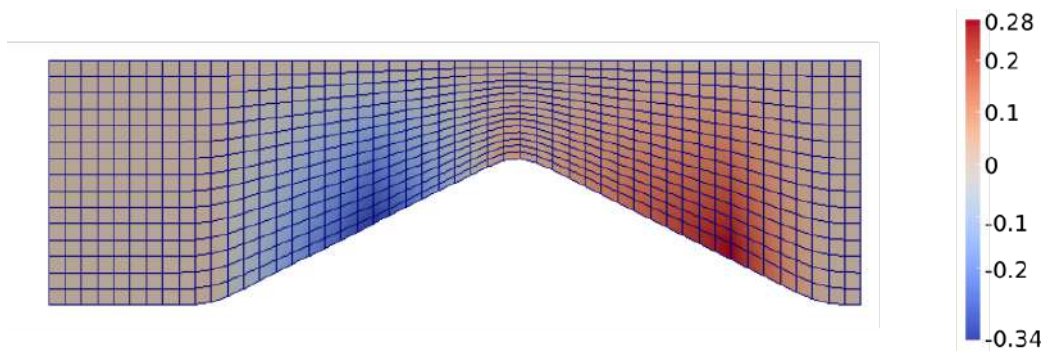


Figure 6.37: Upwind sand-slide: Mesh displacements with smoothing of case B

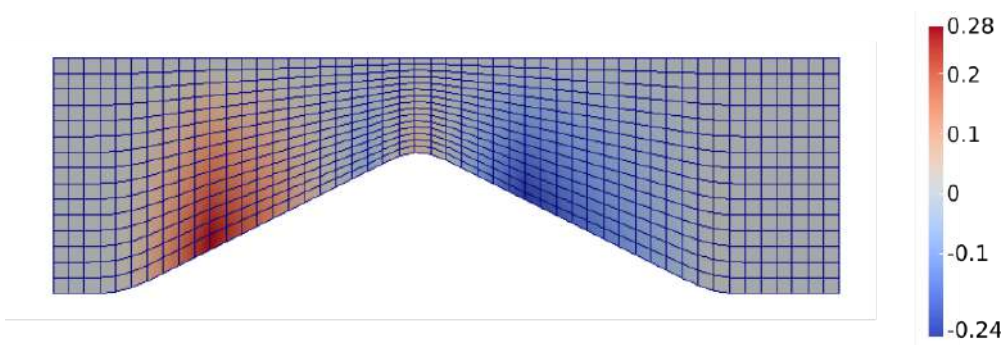


Figure 6.38: Upwind sand-slide: Mesh displacements with smoothing of case C

lations of two-fluid problems.

6.5.1 2D rising bubble

First, we simulate two benchmarks proposed by HYSING *et al.* [140] (case A and case B). The task of the proposed benchmarks is to track the evolution of a two-dimensional bubble rising in a liquid column, with the initial configuration described in Fig. 6.39.

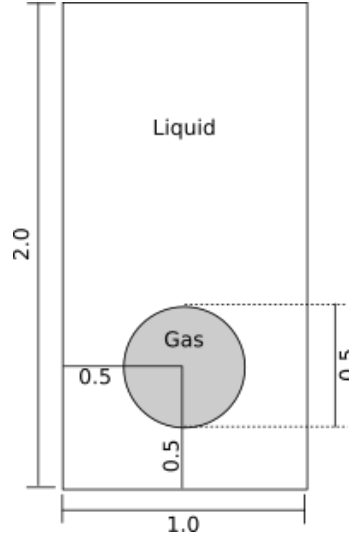


Figure 6.39: 2D rising bubble: Initial configuration and boundary conditions for the test cases.

The initial configuration is identical for both test cases and consists of a circular bubble of radius $R = 0.25$ m centered at $[0.5, 0.5]$ m in a $[1 \times 2]$ m rectangular domain. The no-slip boundary condition is used at the top and bottom boundaries, whereas the free-slip condition is imposed on the vertical walls. Table 6.4 lists the parameters used for this simulation.

Table 6.4: 2D rising bubble: Data.

Computational domain	1×2	(m)
Grid sizes	0.05 to 0.003125	(m)
Number of time-steps	960	(-)
Time-step	0.003125	s
Bubble radius	0.25	m
Initial bubble position	$(x, y) = (0.5, 0.5)$	m
Liquid density	(A) 1000 (B) 1000	kg/m ³
Liquid viscosity	(A) 10 (B) 10	kg/(ms)
Gas density	(A) 100 (B) 1	kg/m ³
Gas viscosity	(A) 1 (B) 0.1	kg/(ms)
Surface tension	(A) 24.5 (B) 1.96	N/m
Gravity	(A) 0.98 (B) 0.98	m/s ²

We use an adapted mesh, initially with 20×40 bilinear quadrilateral elements, and

after the refinement, the smallest element has size 0.003125 m. We initially refine the region where the bubble is located in four levels. The adaptive mesh refinement is based on the flux jump of the level-set function error, in which $h_{max(LS)} = 4$, and in the flux jump of the velocity error, with $h_{max(FLow)} = 2$. We apply the adaptive mesh refinement every eight time-steps. The interface is modeled with $E = 0.015625$.

In Fig. 6.40a, we present the bubble shape at the final time ($t = 3$) for the test case A. We compare our results with the ones obtained by three different groups [140]: TP2D [141], FreeLIFE [142], and MooNMD [143], which are available in [140]. We list the main features of each code in Table 6.5. Our prediction is in good agreement with the others (Fig. 6.40a). However, the final shape is not sufficient to validate the code, and it is now that the previously defined quantities of interest become particularly useful. Thus, in Figures 6.40b, 6.40c, 6.40d we compare the circularity, center of mass position, and rise velocity for test case A. All groups have a good agreement for the quantities of interest.

Table 6.5: 2D rising bubble: Comparison of the code features and schemes.

	TP2D	FreeLIFE	MooNMD	Present work
Space discretization	FEM	FEM	FEM	FEM
Interface capturing	Level-set	Level-set	ALE	Convected level-set
Time discretization	Fractional-step θ method	BDF2	Fractional-step θ method	Backward Euler / BDF2

In Fig. 6.41a, we present the bubble shape at the final time ($t = 3$) for the test case B. Again, we compare our results with [141], [142], and [143]. It is evident that although all codes predict a similar shape for the main bulk of the bubble, there is no agreement concerning the thin filamentary regions. There are discrepancies when we compare all quantities of interest. The circularity for all groups (Fig. 6.41b) agrees very well until about $t = 1.75$ seconds, and for later times significant differences start to appear, that is, when the thin filaments are present. The center of mass, shown in Fig. 6.41c, is predicted similarly for all groups despite the shape differences, and the mean rise velocity also presents a quite good agreement between the different codes (Fig. 6.41d).

6.5.2 3D rising bubble

The bubble benchmark presented above is a good test to validate the code because several results are available. However, modeling bubbles in 2D is not the best alternative, since the surface tension acts in all directions, and this force has significant influence in the bubble shape and the rise velocity. Therefore, from now on, we present only 3D simulations.

We simulate a 3D benchmark proposed by ADELSBERGER *et al.* [4] in which the results of several flow solvers are available for comparison. The only difference introduced with respect to the 2D benchmark (aside from the additional spatial dimension) is

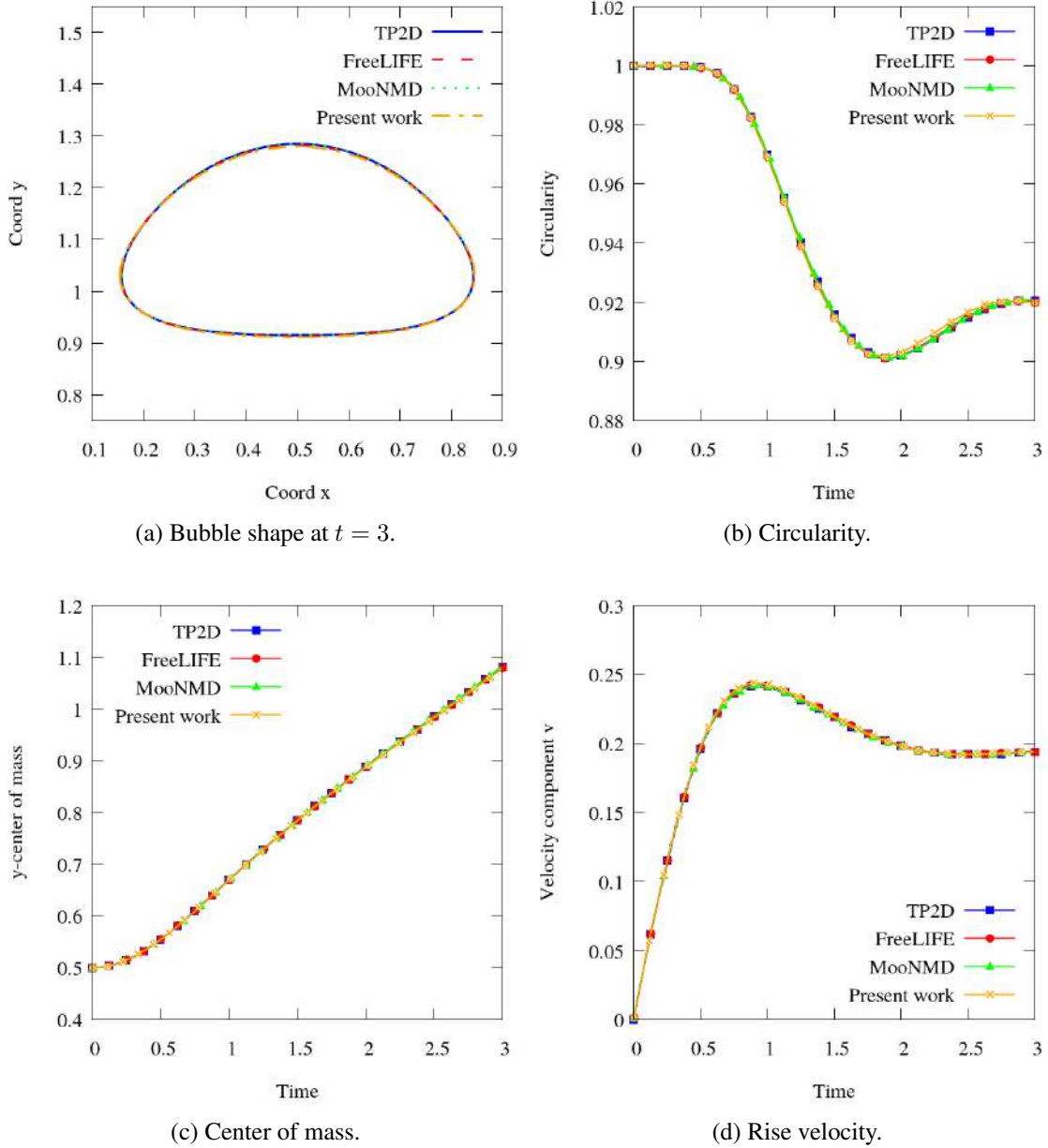


Figure 6.40: 2D rising bubble: Test A results.

the type of boundary condition applied at the vertical walls, which in 2D is a free-slip boundary condition, but in the 3D benchmark, it is a no-slip boundary condition. The geometrical setup for the 3D configuration is depicted in Fig. 6.42, and we use the same physical parameters presented in Table 6.4, except for the grid size and time-step.

We use an adapted mesh, initially with $10 \times 20 \times 10$ trilinear hexahedral elements, and after the refinement, the smallest element has size 0.00625 m. We refine the initial region where the bubble is located into four levels. The adaptive mesh refinement is based on the flux jump of the level-set function error, in which $h_{max(LS)} = 4$, and in the flux jump of the velocity error, with $h_{max(FLow)} = 3$. We apply the adaptive mesh refinement every 20 time-steps. The interface is modeled with $E = 0.0125$, and the time-step is defined as

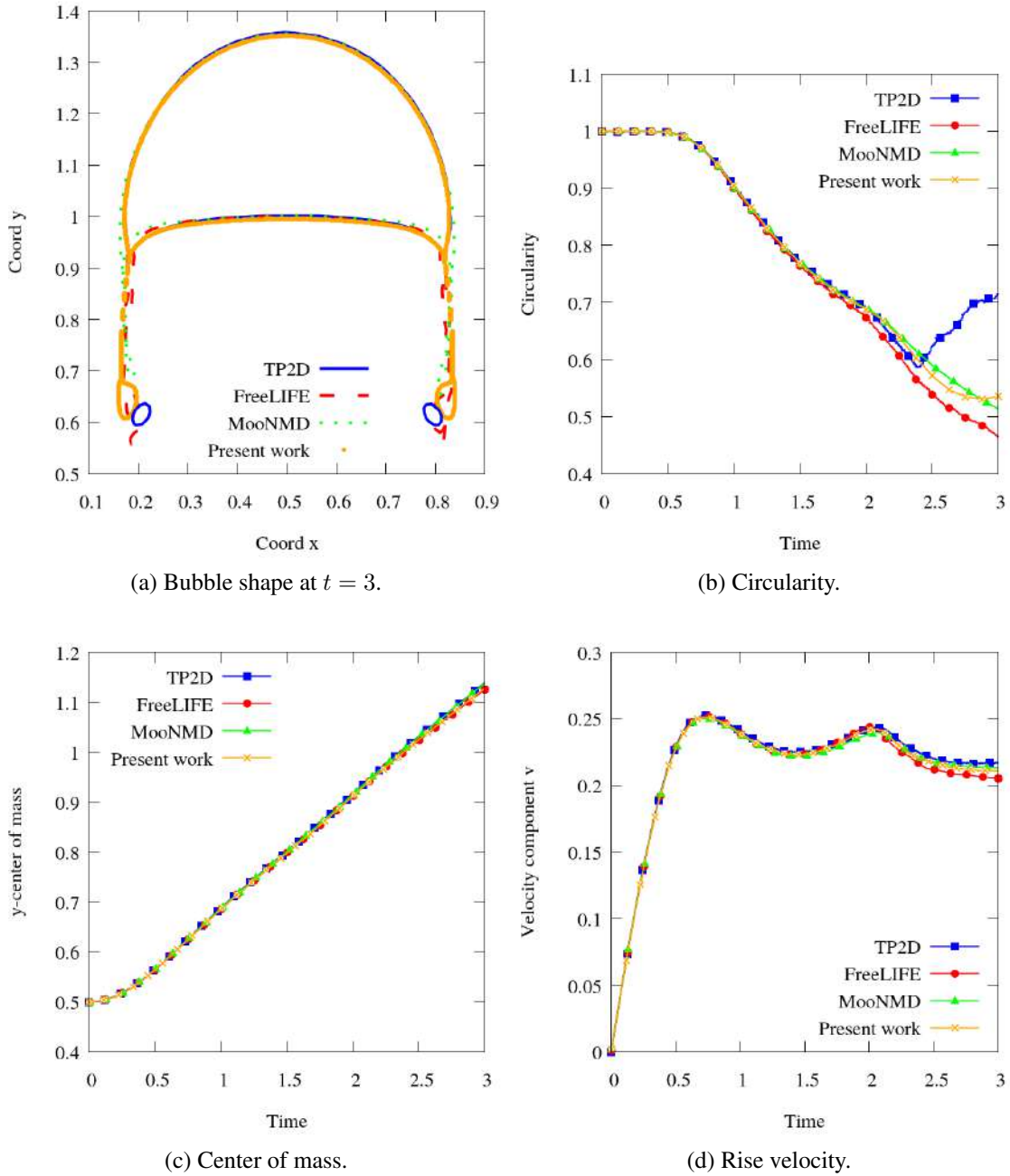


Figure 6.41: 2D rising bubble: Test B results.

0.0025 seconds.

We compare our results with the ones obtained by the flow solvers DROPS [144], NaSt3DGPF [79] and OpenFOAM [145], which are available in [4]. All codes adopt different numerical techniques. We list the main features of each code in Table 6.6. The simulations of DROPS, NaSt3DGPF, and OpenFOAM were performed on the HPC cluster *Siebengebirge* featuring 160 Intel Xeon X7560 2.226 GHz CPU cores and total memory of 2560 GB. Our simulations run on the HPC cluster *Lobo Carneiro*, featuring 504 CPUs Intel Xeon E5-2670v3 (Haswell) 2.3 GHz: 6048 cores and total memory of 16 TB. In Table 6.6, it is possible to see how our computational time was considerably lower than

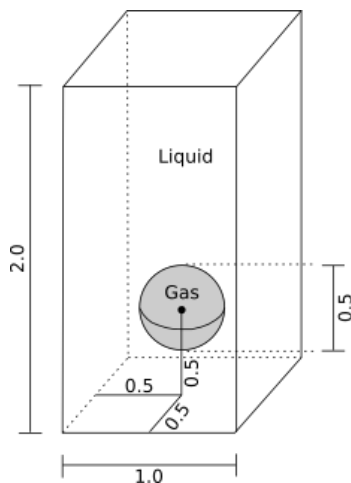


Figure 6.42: 3D rising bubble: Initial configuration and boundary conditions for the test cases.

the other solvers. It happens not only because of the bigger computational power of *Lobo Carneiro* but also because we use larger time-steps and a lower number of elements. The AMR procedure, together with the implicit solution of the Navier-Stokes and convected level-set equations, allows saving much computational effort.

Table 6.6: 3D rising bubble: Comparison of the code features and schemes [4].

	DROPS	NaSt3D	OpenFOAM	Present work
Space discretization	XFEM	Finite differences	Finite volumes	FEM
Interface capturing	Level-set	Level-set	VOF	Convected level-set
Time discretization	Crank-Nicholson	Adams-Bashforth 2nd	Backward Euler	Backward Euler / BDF2
Δt	2.5×10^{-4}	10^{-4}	10^{-4}	2.5×10^{-3}
Mesh	AMR	Fixed	Fixed	AMR
$\min h_e$	1/32	1/121	1/128	1/160
Number of elements	$\leq 65,536$	3,543,122	4,194,304	max of 1,400,000
Computational time	2 weeks / 1 proc	1 week / 32 procs	2.5 days / 32 procs	24 hours / 16 procs

In Fig. 6.43, we present the bubble shapes at different times for the test case A. In Figures 6.44a, 6.44b, 6.44c, 6.44d we show the diameter, sphericity, center of mass position and rise velocity for test case A, and we add the results of the other solvers presented in ADELBERGER *et al.* [4]. All groups have a good agreement for the quantities of interest, except OpenFOAM, which presents some discrepancies.

In Fig. 6.45, we present the bubble shape at different time-steps for the test case B. As in the 2D benchmark, after 1.75 seconds of simulation, the results for the bubble diameter (Fig. 6.46a) and sphericity (Fig. 6.46b) differ. The center of mass, shown in Fig. 6.46c, is predicted very similarly, and the mean rise velocity also presents a quite good agreement between the different solvers (Fig. 6.46d).

To show the influence of the mesh refinement in our results, we run three different meshes considering the same time-step of 0.0025 seconds, being the initial mesh equal to

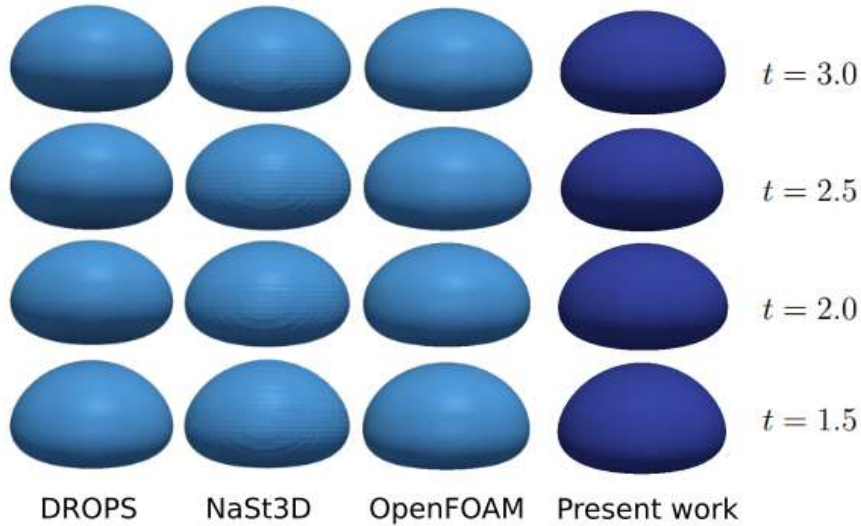


Figure 6.43: 3D rising bubble: Bubble shapes at different times for the test A.

$10 \times 20 \times 10$ trilinear hexahedral elements, with a refined region near the initial bubble position. The difference between the meshes is the maximum level of refinement allowed. Mesh 1 has $h_{max(LS)} = 2$, Mesh 2 has $h_{max(LS)} = 3$ and Mesh 3 has $h_{max(LS)} = 4$. We show the initial meshes and the number of elements with time in Figures 6.47a and 6.47b. If we would not use the AMR procedure and refine all domain with elements sizes equal to the smallest element size of each mesh, we would have 128,000 elements for Mesh 1, 1,024,000 elements for Mesh 2 and 8,192,000 elements for Mesh 3. Using the AMR procedure, we save much computational effort, working with meshes about one-eighth of the mesh without AMR. In Figures 6.48a to 6.49d, it is possible to see how the mesh refinement influences the quantities of interest. It happens because of the surface tension model adopted. We distribute the surface tension in an area near the interface, and the thickness where it is applied depends on the mesh size. Therefore, smaller element sizes will yield more accurate results. The difference between the results of $h_{max(LS)} = 3$ and $h_{max(LS)} = 4$ is minimal, and we chose to limit the simulations until $h_{max(LS)} = 4$, not only because of the computational cost but also because we already have good agreement with the other solvers. However, the results certainly would be a little more accurate with more refinement, as the simulations of the 2D benchmark, that we have used a smaller element size than for the 3D one.

6.5.3 Grace's bubble diagram

Depending on the characteristics of the gas and liquid phases, the bubble reaches different shapes and terminal velocities. GRACE [66] has analyzed a large body of experimental data on shapes and rise velocities of bubbles in quiescent viscous liquids and has

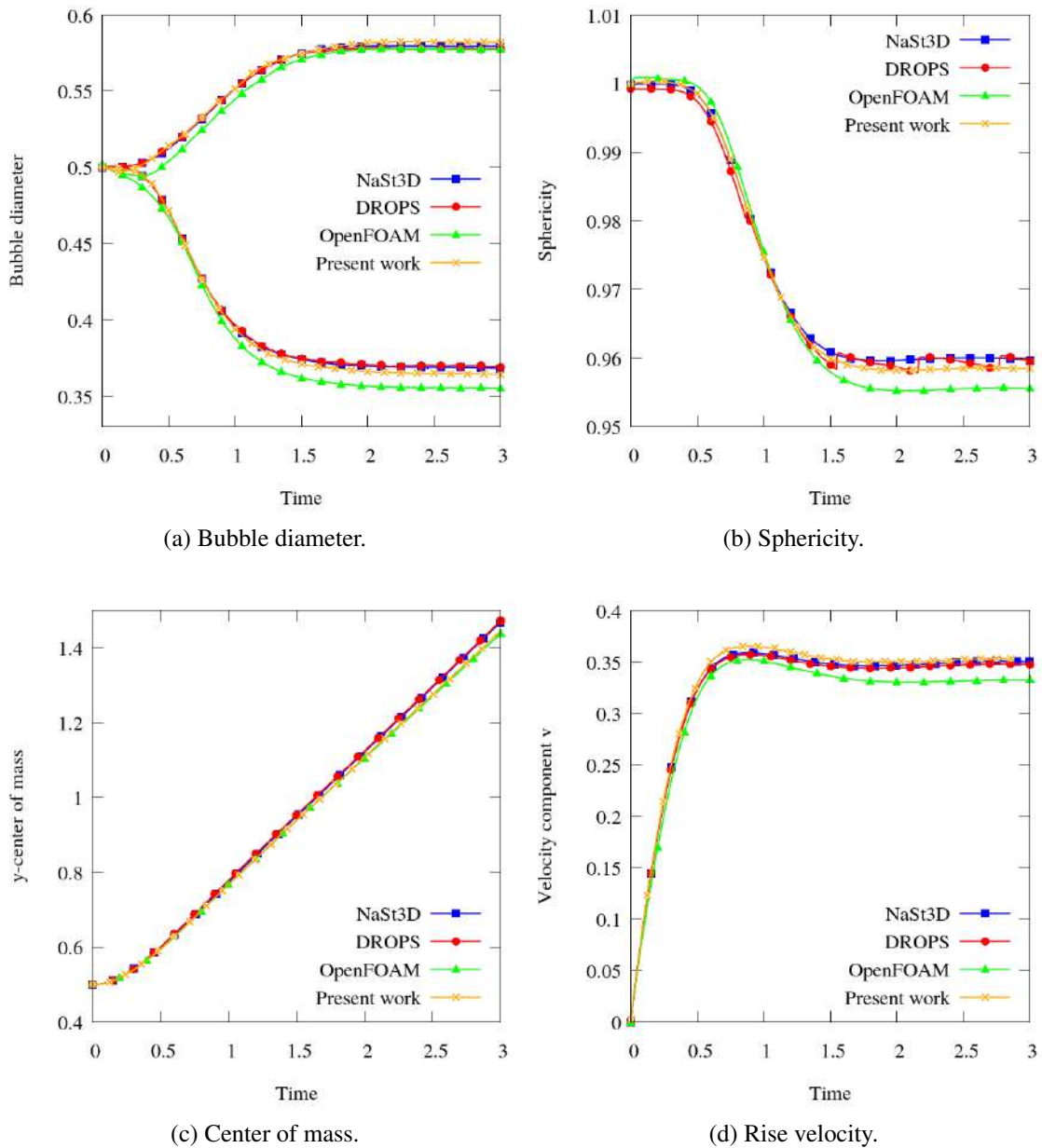


Figure 6.44: 3D rising bubble: Test A results.

shown that this data can be condensed into one diagram. The Grace's diagram relates the Morton, Eötvös and Reynolds numbers, and depending on how they are combined, the shape of the bubble may become spherical, ellipsoidal, skirted, dimple ellipsoidal cap, spherical cap or wobbling.

We simulate five regimes given in Grace's diagram, using the same values of Morton and Eötvös numbers used in the simulations in [71] and in the experiments in [5], which are presented in Table 6.7, where we also present the Reynolds calculated in this work. The surface tension, density, and viscosity are adapted in each simulation to result in the desired Morton and Eötvös numbers. We simulate two different meshes for each case, Mesh 1 and Mesh 2. The data used in these simulations are presented in Table 6.8.

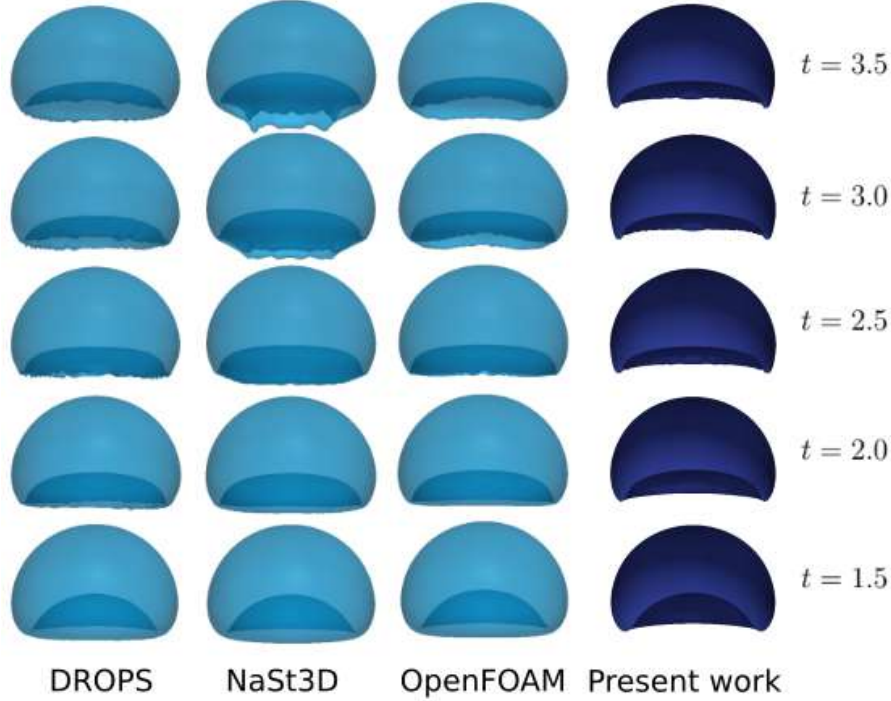


Figure 6.45: 3D rising bubble: Bubble shapes at different times for the test B.

Table 6.7: Grace’s bubble diagram: Morton (Mo), Eötvös (EO) and Reynolds (Re) numbers according to the experiment of [5].

Bubble regime	Mo	EO	Re_{exp} [5]	$Re_{present\ work}$	Case
Spherical	711	17.7	0.232	0.172	A
Ellipsoidal	8.2×10^{-4}	32.2	55.3	51.86	B
Skirted	43.1	339	18.3	18.25	C
Dimpled ellipsoidal cap	266	243	7.77	7.86	D
Spherical cap	4.63×10^{-3}	115	94.0	87.6	E

To reduce the computational cost, we simulate only one-quarter of the domain, applying symmetry (free-slip) boundary conditions at all boundaries. Then, we use an adapted mesh, initially with $10 \times 10 \times 40$ trilinear hexahedral elements, and after the refinement, the smallest element has size 0.00025 m in Mesh 1 and 0.000125 m in Mesh 2. We initially refine the region where the bubble is located into three/four levels. The adaptive mesh refinement is based on the flux jump of the level-set function error, in which $h_{max(LS)} = 3$ in Mesh 1 and $h_{max(LS)} = 4$ in Mesh 2, and on the flux jump of the velocity, in which $h_{max(FLow)} = h_{max(LS)} - 1$. The coarsening fraction is $c_f = 0.01$ and the refining fraction is $r_f = 0.99$. We apply the adaptive mesh refinement every 5 time-steps. The interface is modeled with $E = 0.0005$ m in Mesh 1 and $E = 0.00025$ m in Mesh 2.

For future comparisons, in Figures 6.50 to 6.54, we present the bubble quantities obtained with our solver for these cases, considering the two different meshes. The mesh

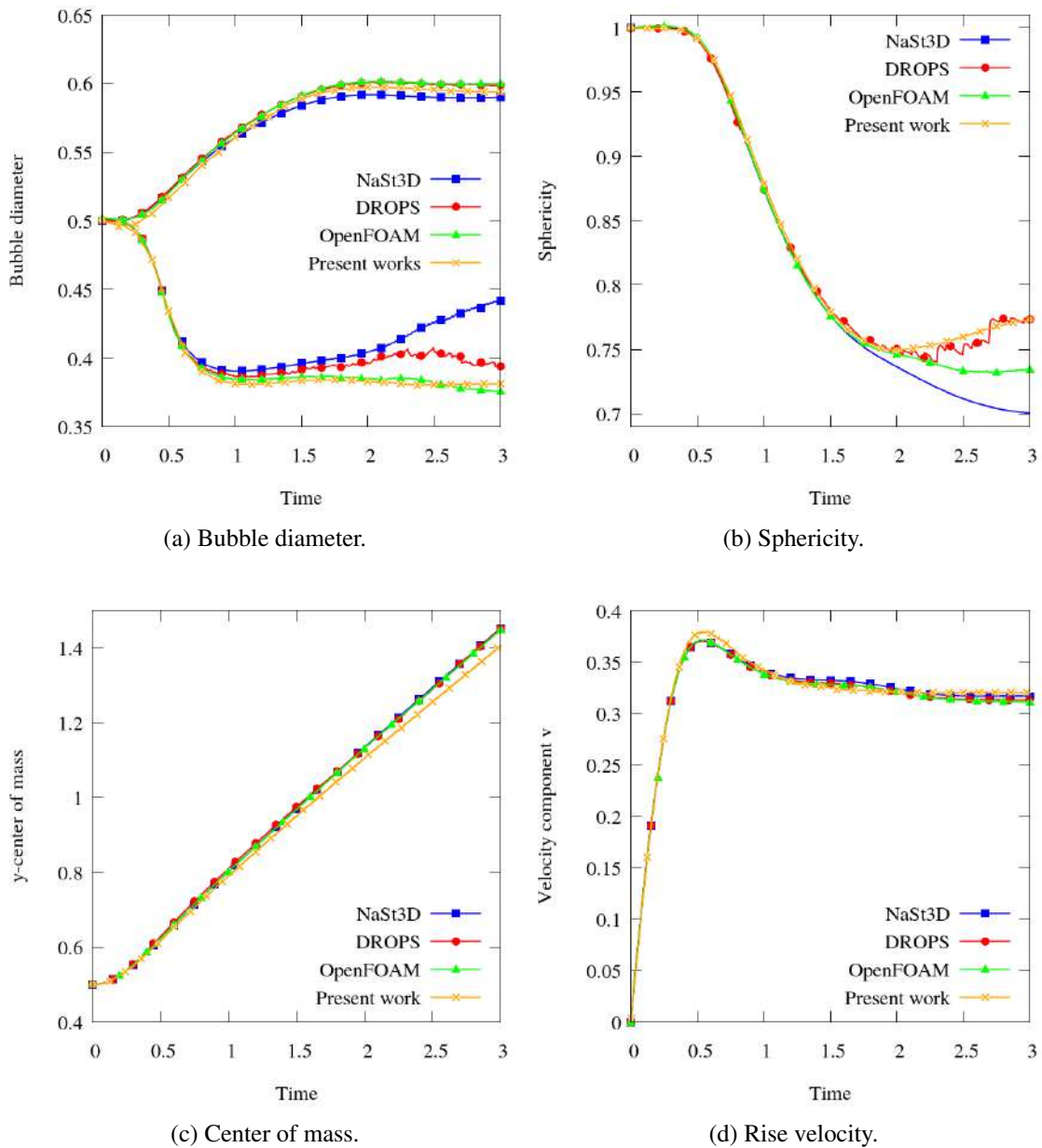
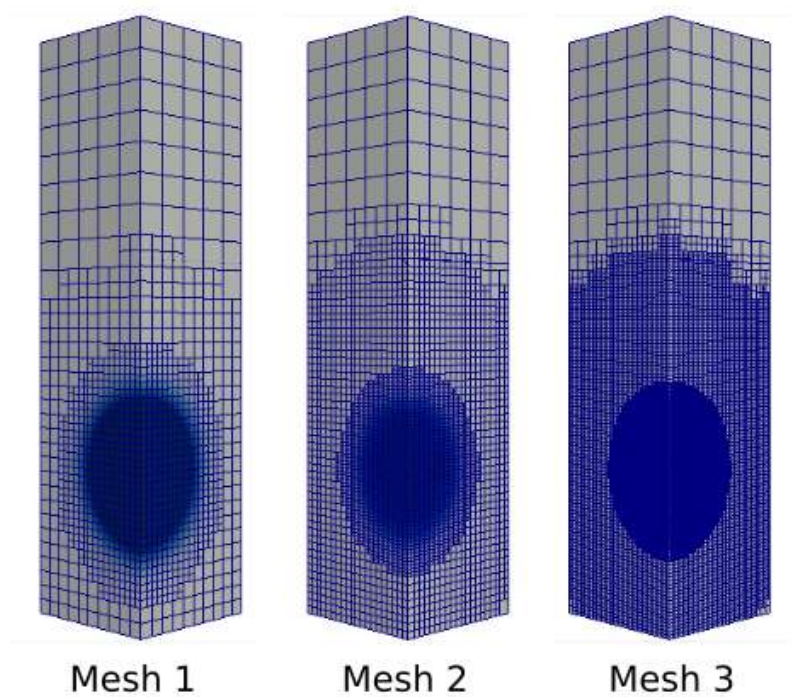


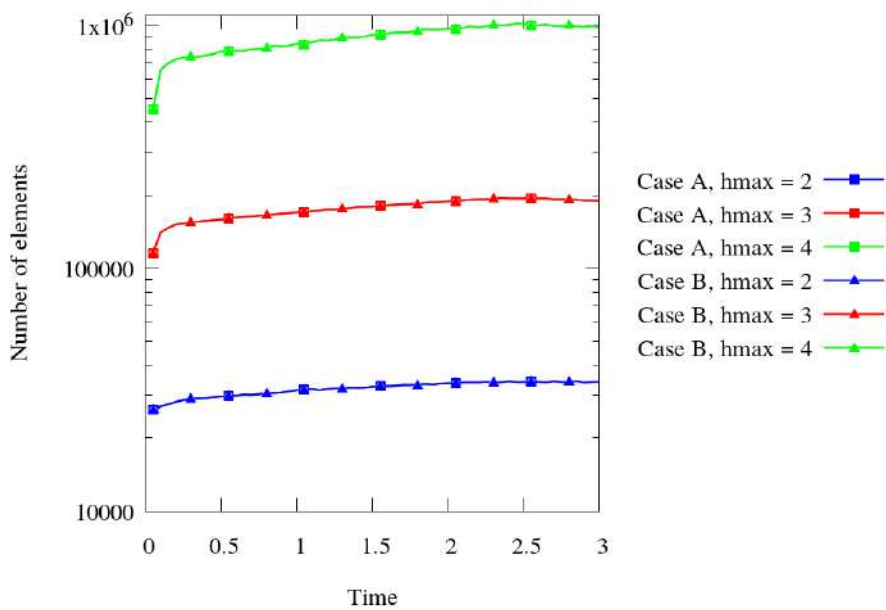
Figure 6.46: 3D rising bubble: Test B results.

refinement has minimal influence in all quantities of tests A and B (Fig. 6.50 and Fig. 6.51). On the other hand, in tests C and D, the bubble shapes between the different meshes differ after some simulation time, which also influences the rise velocity results (Fig. 6.52 and Fig. 6.53). Curiously, in test E (Fig. 6.54), the bubble shapes of both meshes agree, although the rise velocity changes a little. However, the small differences in the results make us conclude that the mesh refinement used is enough to represent well the studied cases.

In Table 6.9, snapshots are given of the shapes of the computed bubbles in the different regimes using Mesh 2, and they are compared with other numerical and experimental results. The computed Reynolds numbers and bubble shapes compare well with the results



(a) The three used meshes.



(b) Number of elements with time.

Figure 6.47: 3D rising bubble: Comparison of the mesh refinement.

obtained by ZHANG *et al.* [6] and HUA e LOU [7]. Although the Reynolds numbers are a little different for the different methods, they all have the same order, and the shapes qualitatively agree.

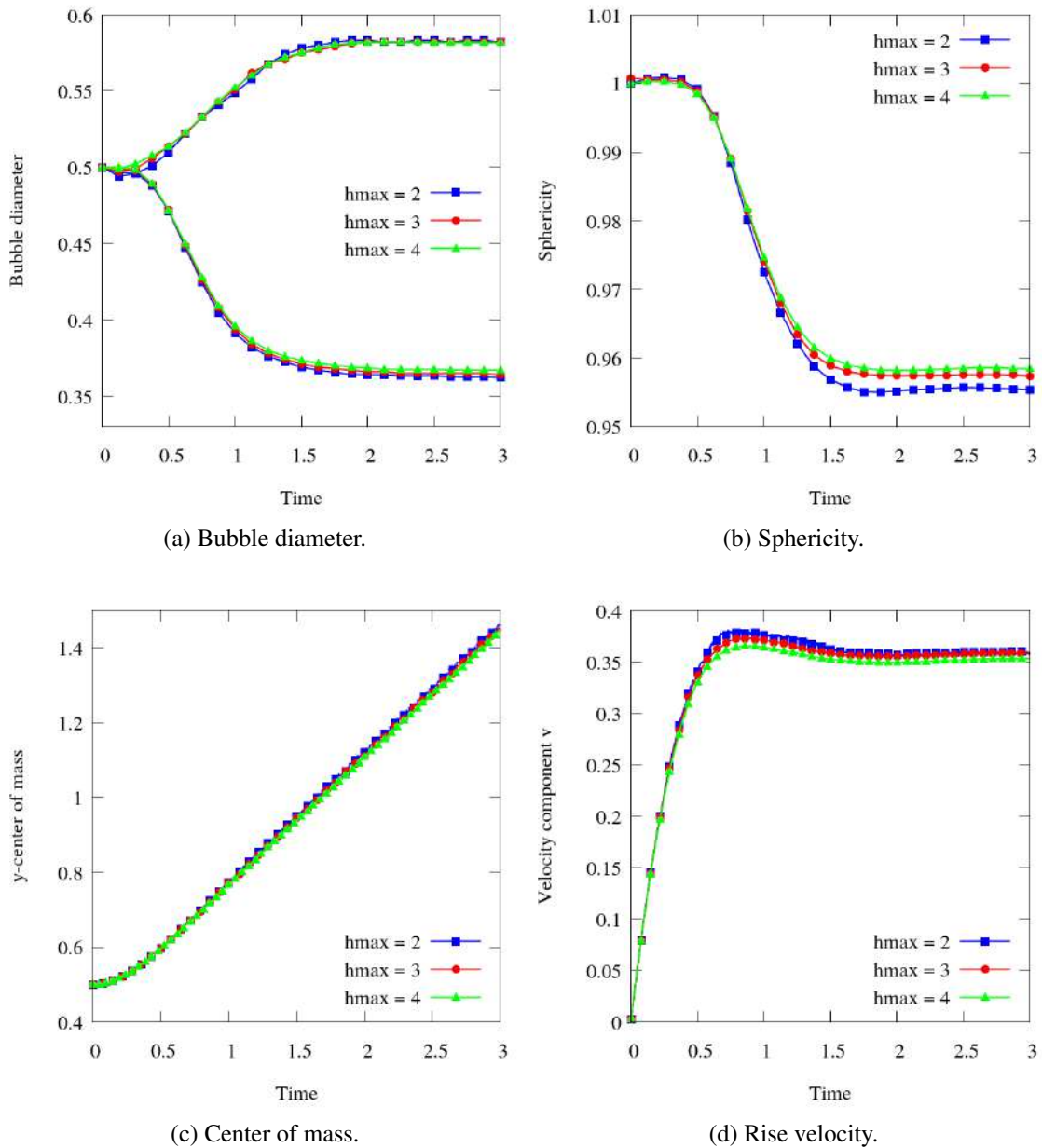


Figure 6.48: 3D rising bubble: Test A comparison of the results with different mesh refinements. The smallest element size for $h_{max(LS)} = 2$ is 0.025 m, for $h_{max(LS)} = 3$ is 0.0125 m and $h_{max(LS)} = 4$ is 0.00625.

6.5.4 Coalescence of two rising gas bubbles

Now, we simulate the coalescence of two rising bubbles in a quiescent liquid where the bubbles are initially spherical with their centers separated by three bubble radius. Here, we also simulate two configurations, (A) the co-axial coalescence and (B) the oblique coalescence. The Morton and Eötvös number for these cases are, respectively, equal to 2×10^{-4} and 16 (based on the individual bubbles). The data used in both cases is presented in Table 6.10.

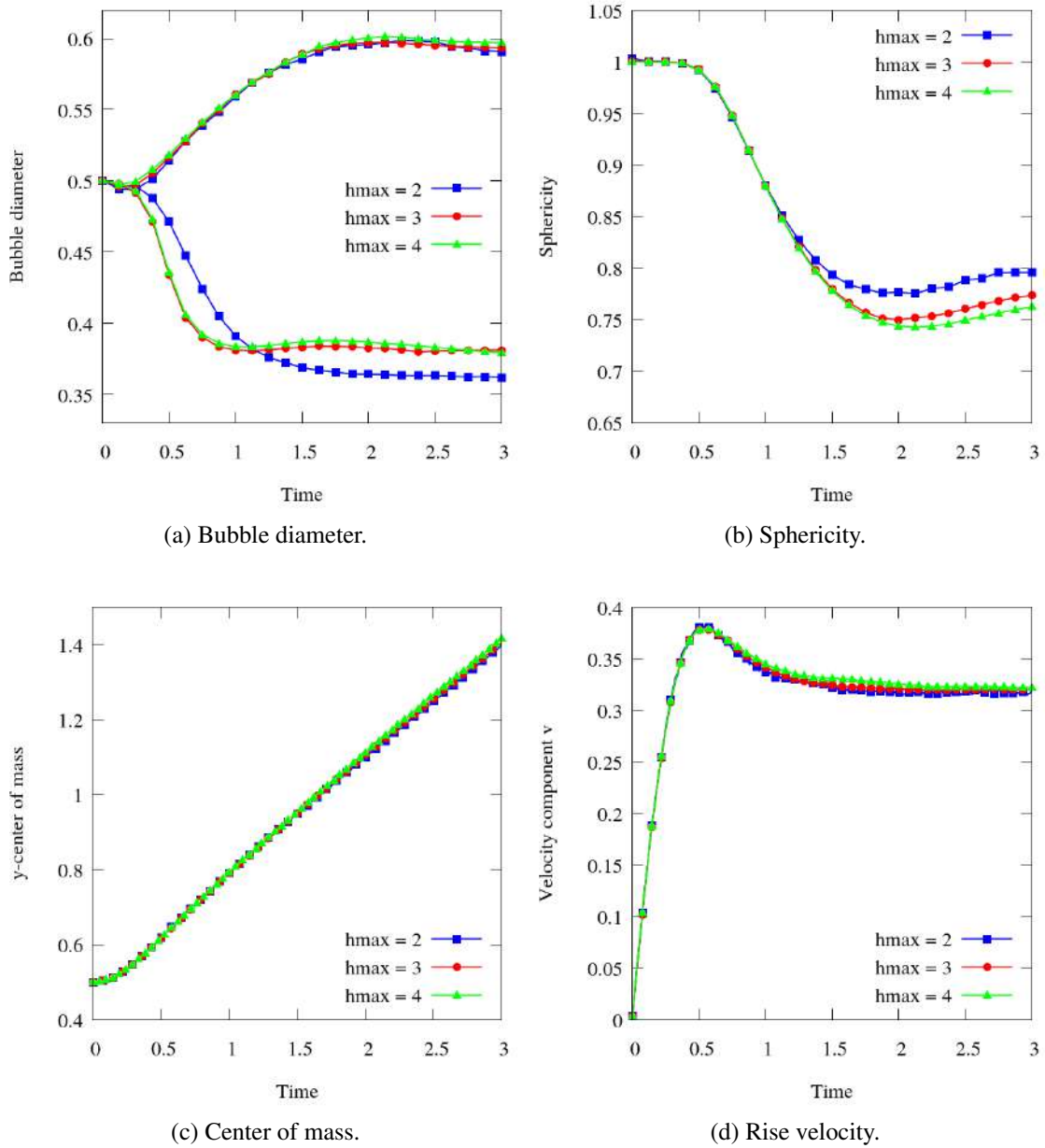










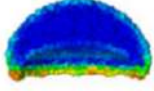



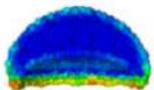






Figure 6.49: 3D rising bubble: Test B comparison of the results with different mesh refinements. The smallest element size for $h_{max(LS)} = 2$ is 0.025 m, for $h_{max(LS)} = 3$ is 0.0125 m and $h_{max(LS)} = 4$ is 0.00625.

In the co-axial case, as we can see in Fig. 6.55, the leading bubble deforms into the ellipsoidal shape quickly due to the considerable resistance force from the liquid above, while the trailing bubble deforms into a “bullet” shape due to the absorption effect from the flow wake of the leading bubble. At $t = 0.10$ s, the two bubbles touch each other and merge into one bigger bubble. The predicted shape of the bubbles may seem different from the experiment data, but this is because the initial bubble shapes in our model are spherical, which is different from the initial condition of an ellipsoid in the experiment. Figure 6.56 shows the oblique coalescence of the two rising bubbles. In this case, it is

Table 6.8: Grace's bubble diagram: Data.

Computational domain	0.02 × 0.02 × 0.08					(m)
Grid sizes	(1) 0.002 to 0.00025 and (2) 0.002 to 0.000125					(m)
Number of time-steps	500					(-)
Time-step	0.0005					s
Bubble radius	0.01					m
Initial bubble position	$(x, y, z) = (0.02, 0.02, 0.01)$					m
Case	A	B	C	D	E	
Liquid density	1000	1000	1000	1000	1000	kg/m ³
Liquid viscosity	1.872	0.039	0.102	0.205	0.023	kg/(ms)
Gas density	1	1	1	1	1	kg/m ³
Gas viscosity	0.01872	0.00039	0.00102	0.00205	0.00023	kg/(ms)
Surface tension	0.05535	0.03043	0.00289	0.00403	0.00852	N/m
Gravity	9.8					m/s ²

Table 6.9: Grace's bubble diagram: The terminal bubble shapes compared with the results of SPH [6], FTM [7] and the experimental data [5].

Case	Present work	SPH [6]	FTM [7]	Experiment [5]
(A) Eo = 17.7 M = 711	 Re = 0.172	No reference data -	 Re = 0.182	 Re = 0.232
(B) Eo = 32.2 M = 8.2 × 10 ⁻⁴	 Re = 51.86	 Re = 52.6	 Re = 54.8	 Re = 55.3
(C) Eo = 339 M = 43.1	 Re = 18.25	 Re = 18.1	 Re = 17.8	 Re = 18.3
(D) Eo = 243 M = 266	 Re = 7.86	 Re = 7.99	 Re = 7.6	 Re = 7.77
(E) Eo = 115 M = 4.63 × 10 ⁻³	 Re = 87.6	 Re = 88	 Re = 89.6	 Re = 94

possible to see more clearly the absorption effect of the leading bubble, which is stronger right below it, and it is weakened, as the horizontal distance increases. In this case, only at $t = 0.15$ s the two bubbles touch each other and merge into one. For both cases we

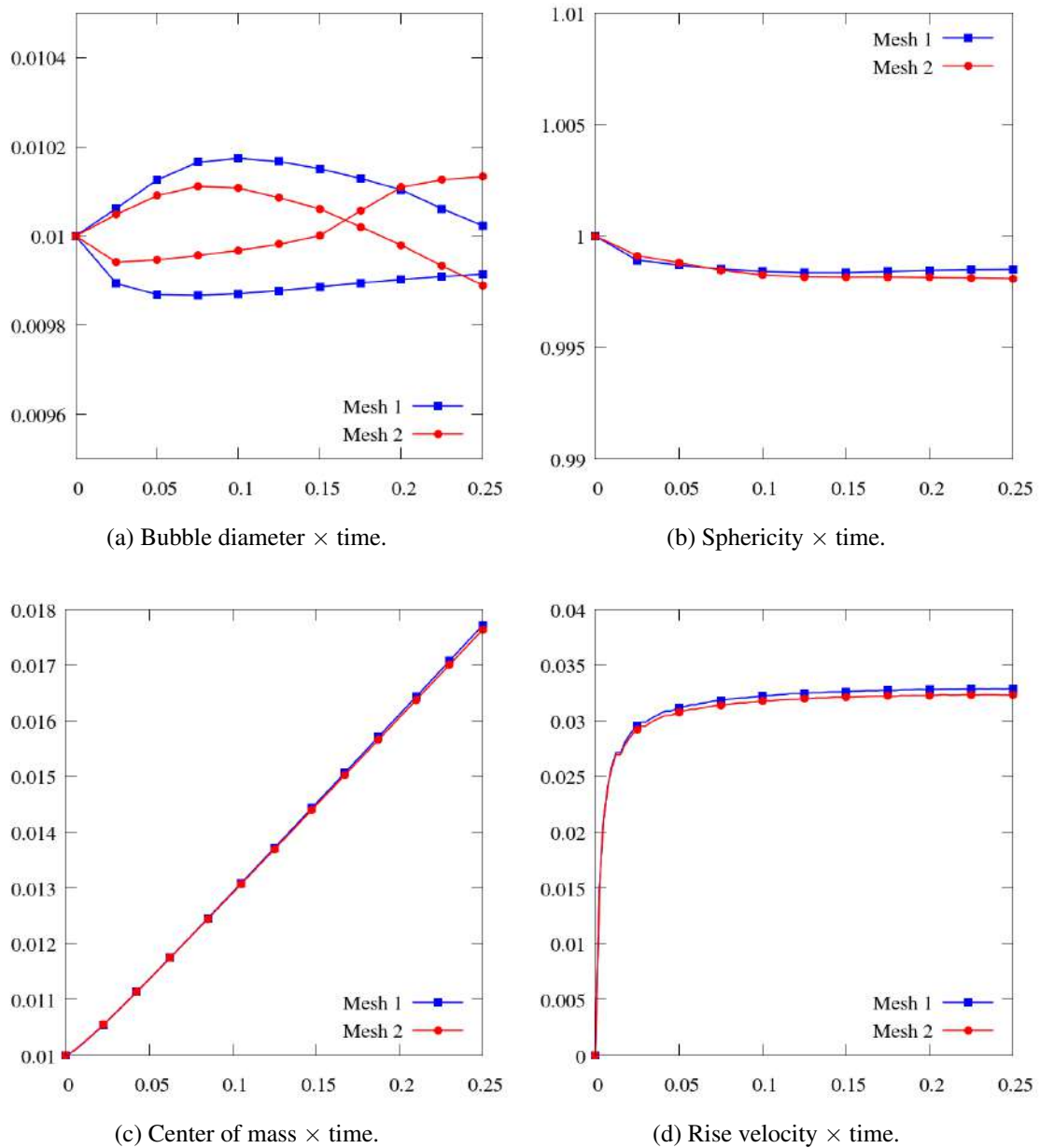


Figure 6.50: Grace's bubble diagram: Test A comparison of the results with different mesh refinements.

have good agreement with the experimental data of BRERETON e KOROTNEY [3], and with numerical results of VAN SINT ANNALAND *et al.* [68], ZHANG *et al.* [6] and XIE *et al.* [69].

6.6 Suspended load of sediments simulations

Two well-documented test cases where experimental data are available are used to validate the model employed in the present study. One involves the net entrainment of sediment from a loose channel bed, and the other deals with the net deposition of sedi-

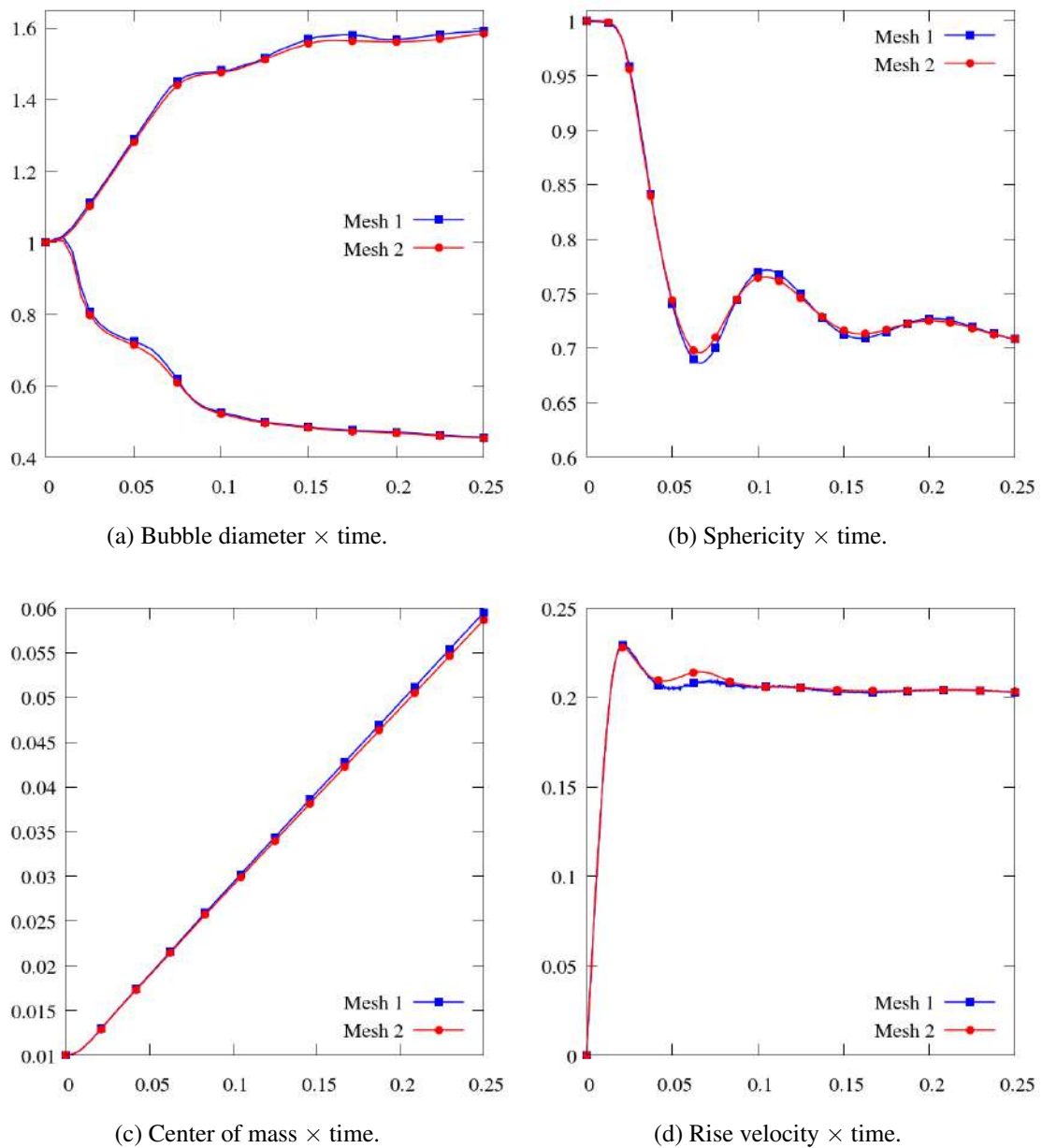


Figure 6.51: Grace's bubble diagram: Test B comparison of the results with different mesh refinements.

ment. These two test cases are essential to validate the implemented sediment boundary conditions. The bed-load transport of sediments is not included in these cases, and also the sediments are considered passive scalars to the flow, meaning that the flow carries them, but they do not change its behavior.

6.6.1 Net entrainment test case

In the van Rijn [146] entrainment test case, a fully-developed flow with clear water is introduced over the sediment bed and the sediment concentration is measured after

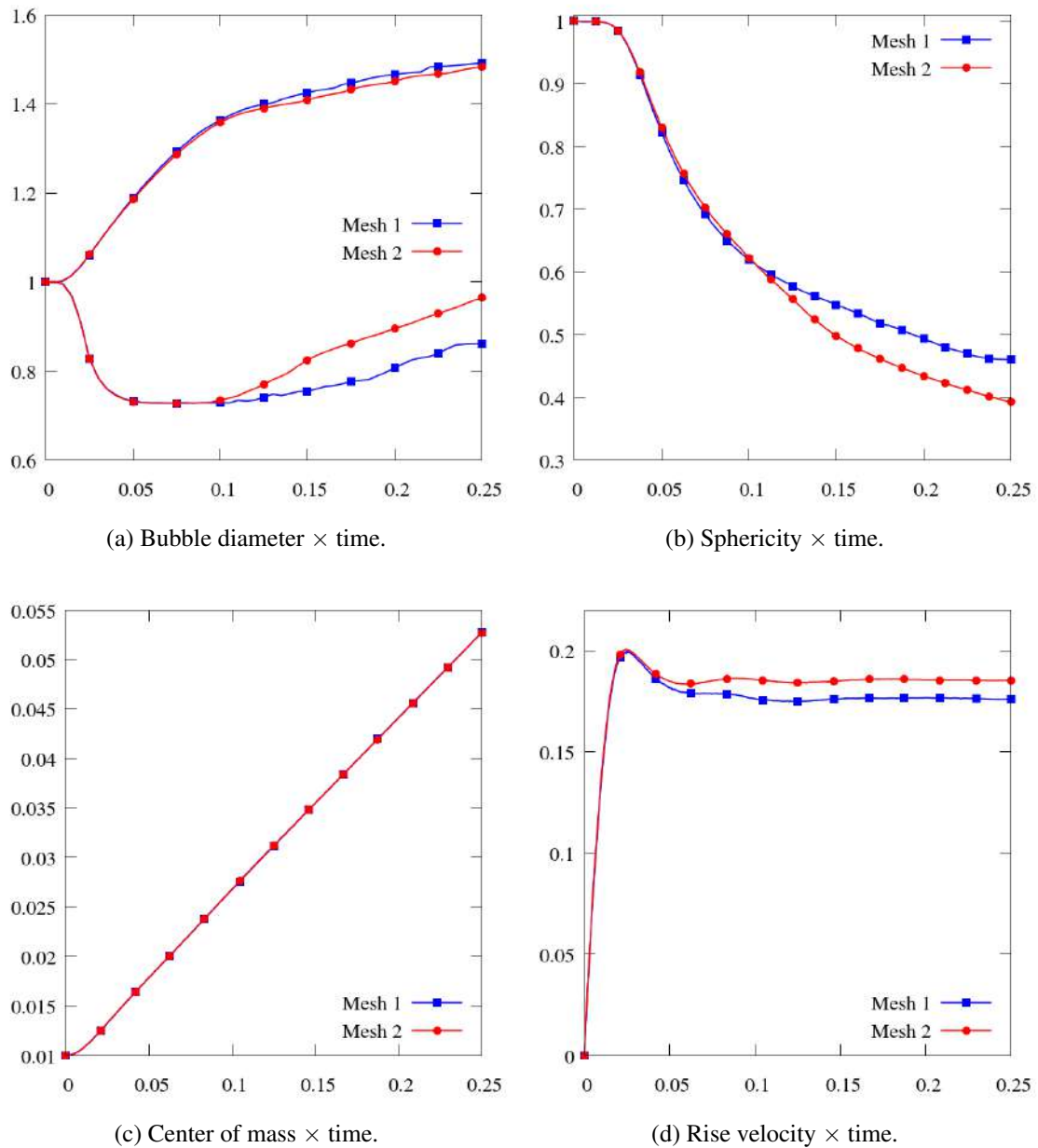


Figure 6.52: Grace's bubble diagram: Test C comparison of the results with different mesh refinements.

reaching the equilibrium condition (Figure 6.57). The present validation employs the original test conditions. The flow is steady and uniform, with a water depth of 0.25 m and a mean velocity \bar{u} of 0.67 m/s. The median diameter is 0.23 mm, d_{90} is 0.32 mm, and the corresponding settling velocity is set to be 0.022 m/s. The reference level above the bed is taken to be 0.005 m, and the predicted friction velocity is 0.047 m/s.

The numerical simulation is made with real dimensions, $\Delta t = 0.0001$ s and a mesh with 1640×40 bilinear quadrilateral elements. Figure 6.58 shows the flow boundary conditions which are: at the inlet, the velocity profile is specified to be the fully developed profile obtained from a separate calculation; for the rigid bed and the sediment bed, the

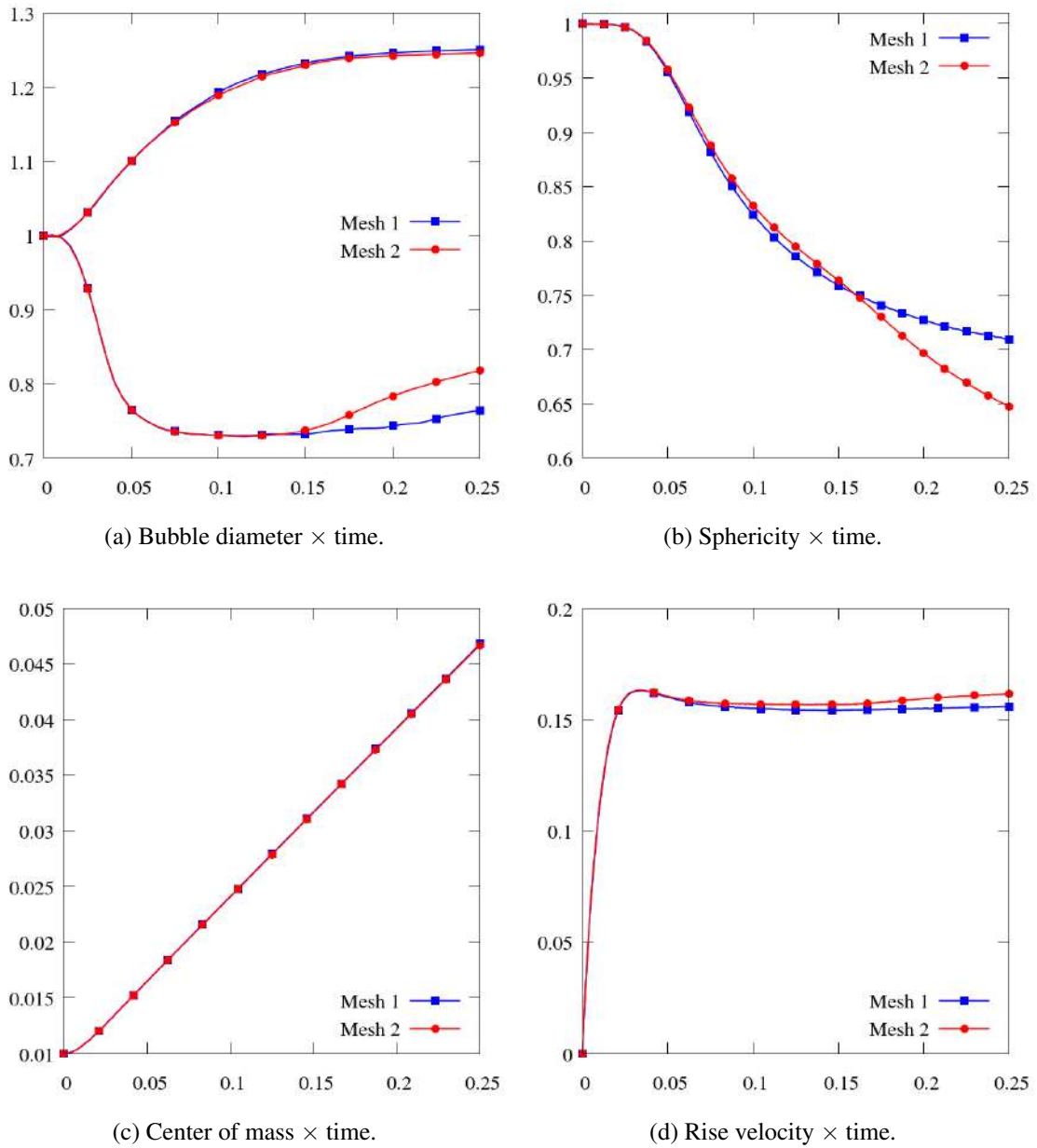


Figure 6.53: Grace's bubble diagram: Test D comparison of the results with different mesh refinements.

no-slip condition is applied; at atmosphere, the slip condition is set; finally, zero velocity gradient is set at the outlet. The sediment boundary conditions are defined in section 4.1.3 with the sediment concentration c_{in} set to zero at the inlet, which corresponds to clear water coming into the domain.

The initial conditions are zero sediment concentration in all domain and a fully-developed flow. In this study, we do not use adaptive mesh refinement. Figure 6.59 illustrates the comparison of computational and experimental sediment concentration distributions in four sections of the channel. We also show results from Liang *et al.* [30] and Wu *et al.* [38]. It can be observed that the simulation predicts the concentrations very

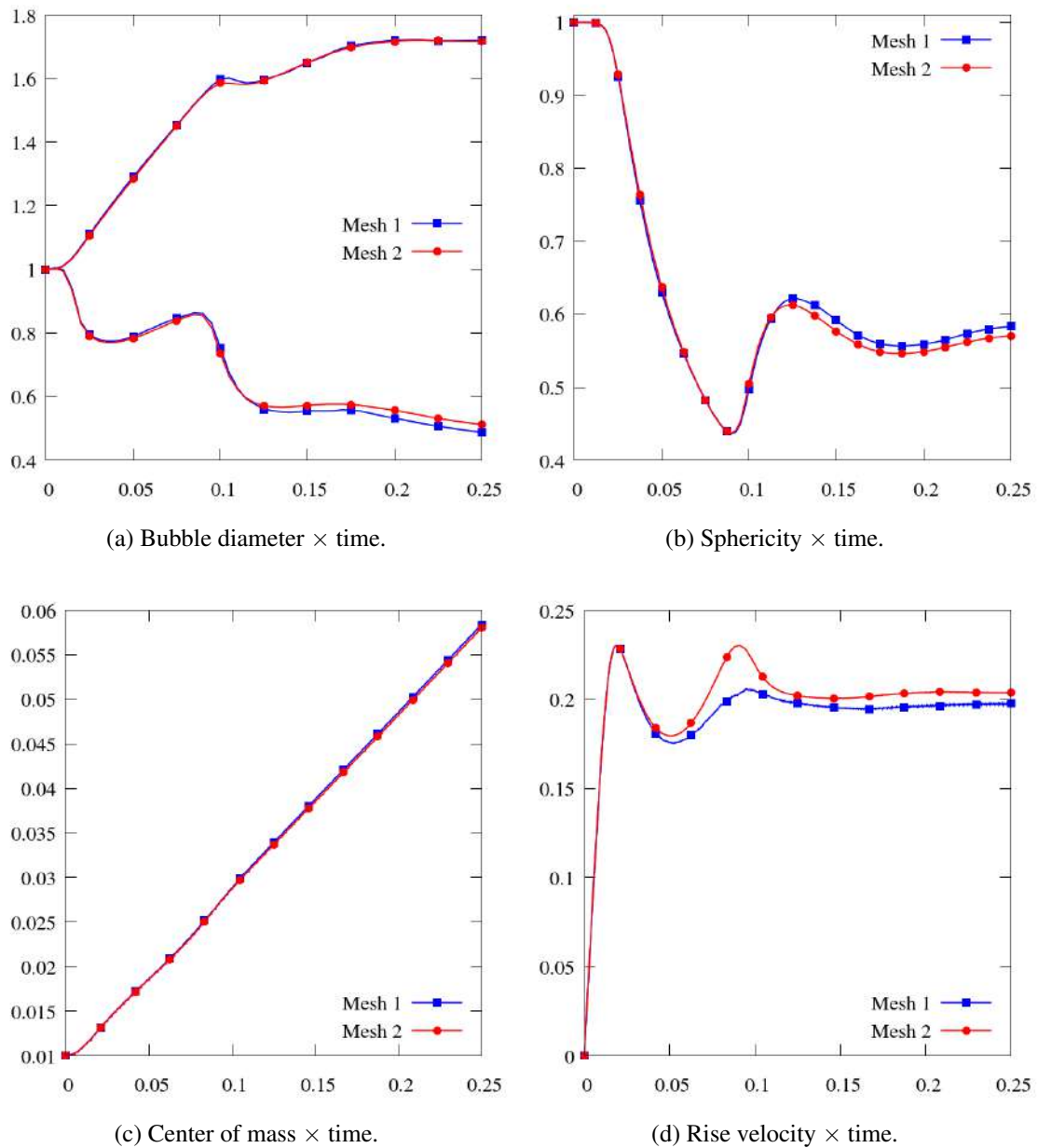


Figure 6.54: Grace's bubble diagram: Test E comparison of the results with different mesh refinements.

well.

6.6.2 Net deposition test case

In the second test case, Wang and Ribberink [147] discharged sand into a flume whose bed is changed from rigid to perforated, as shown in Figure 6.60. Once the sediments fall to the bed, they are trapped into these compartments through the perforated plate and could hardly be entrained again. Hence, this case represents net deposition, and the entrainment rate is considered zero at the bed. The flow is steady and uniform with a mean

Table 6.10: Coalescence of two rising gas bubbles: Data.

Computational domain	0.04 × 0.04 × 0.08		(m)
Grid sizes	0.0005 to 0.002		(m)
Number of time-steps	250		(-)
Time-step	0.001		s
Bubbles radius	0.005		m
Initial bubbles position (A)	$(x, y, z) = (0.02, 0.02, 0.01)$	$(0.02, 0.02, 0.025)$	m
Initial bubbles position (B)	$(x, y, z) = (0.028, 0.02, 0.01)$	$(0.02, 0.02, 0.025)$	m
Liquid density	1000		kg/m ³
Liquid viscosity	0.04655		kg/(ms)
Gas density	10		kg/m ³
Gas viscosity	0.0004655		kg/(ms)
Surface tension	0.06125		N/m
Gravity	9.8		m/s ²

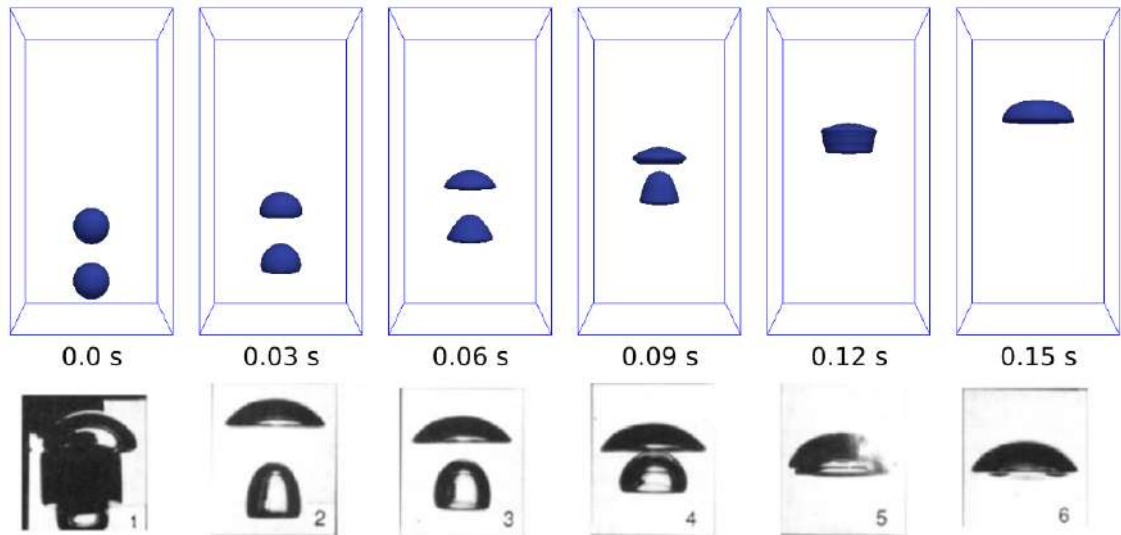


Figure 6.55: Coalescence of two rising gas bubbles: Snapshots of the co-axial coalescence together with the experimental data of [3].

velocity \bar{u} of 0.56 m/s and a water depth of 0.215 m. The median sediment diameter is 0.095 mm and its fall velocity is 0.007 m/s.

The numerical simulation is made with real dimensions, $\Delta t = 0.0001$ s and a mesh with 1600×22 bilinear quadrilateral elements. Figure 6.61 shows the flow boundary conditions. They are: at the inlet, the velocity profile is specified to be the fully developed profile obtained from a separate calculation; at the perforated bed, the no-slip condition is applied; at the atmosphere, the slip condition is set; lastly zero velocity gradient is set at the outlet. The sediment boundary conditions are defined in section 4.1.3. However, in this case, a measured sediment concentration profile is designated at the inlet, and the entrainment rate at the perforated bed is zero. This means that there is only deposition at

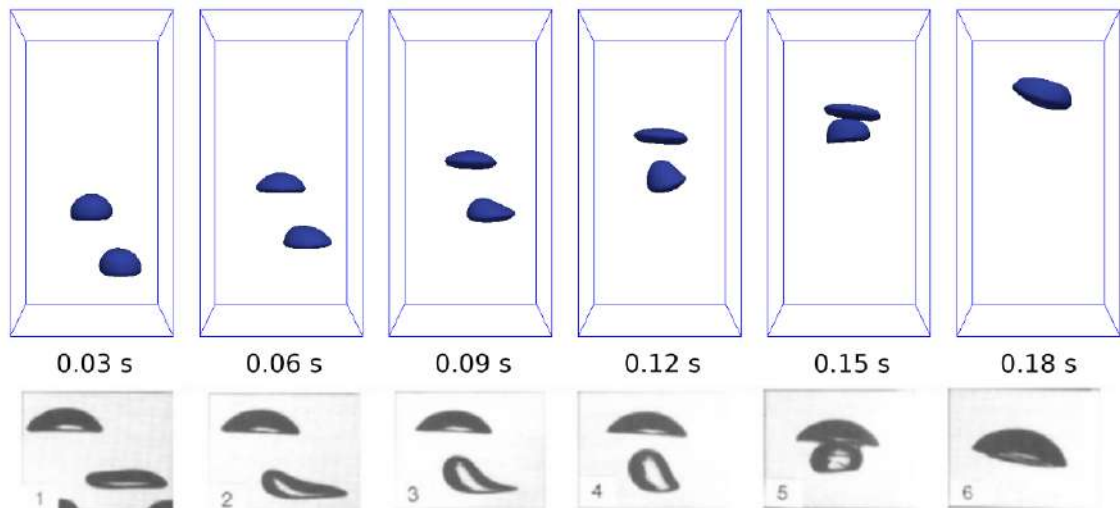


Figure 6.56: Coalescence of two rising gas bubbles: Snapshots of the oblique coalescence together with the experimental data of [3].

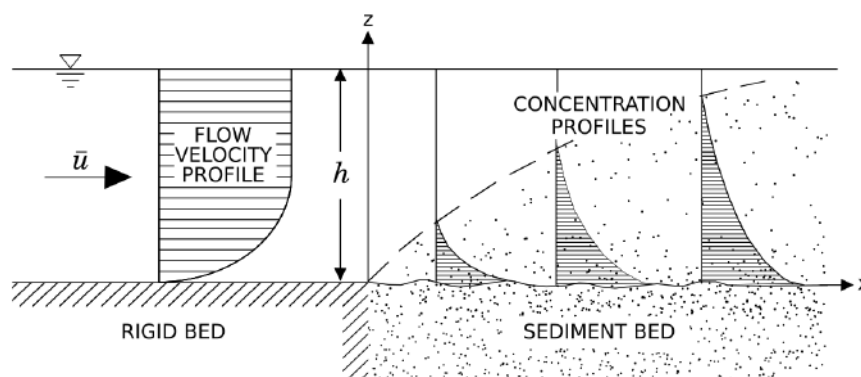


Figure 6.57: Net entrainment test case: Schematic definition of the problem.

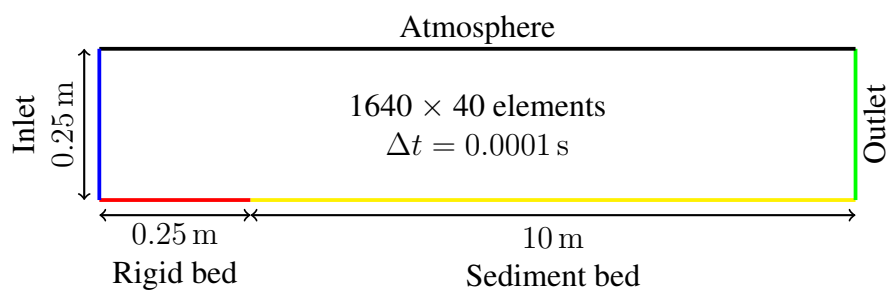


Figure 6.58: Net entrainment test case: Geometry, boundary conditions, time-step, and mesh.

this boundary.

The initial conditions are zero sediment concentration in all domain and a fully-developed flow whose velocities are calculated in a separated simulation. Here, we also do not use adaptive mesh refinement. Figure 6.62 illustrates the comparison of the com-

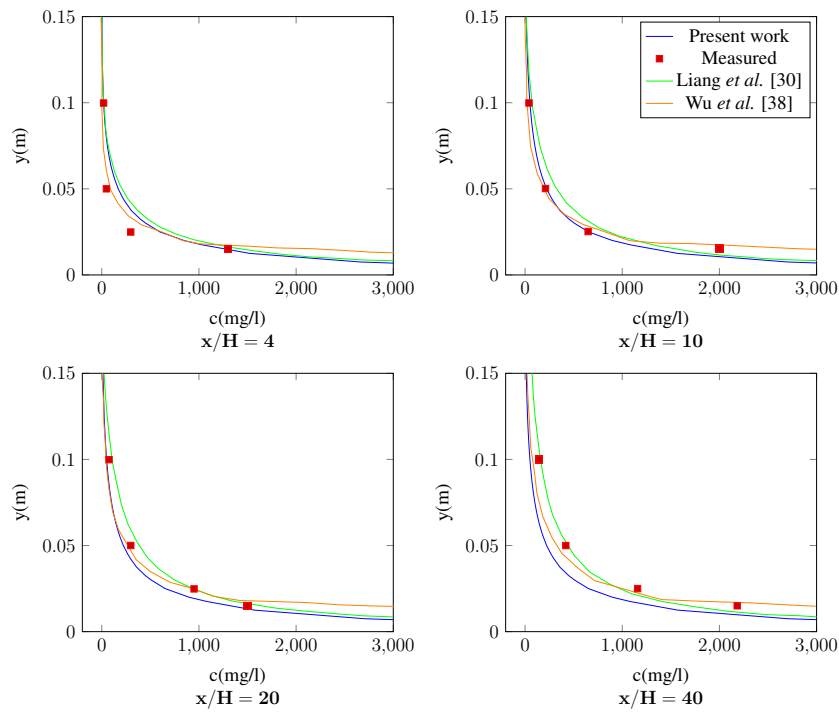


Figure 6.59: Net entrainment test case: Comparison of simulations (solid lines) and measured (squares) suspended sediment concentrations.

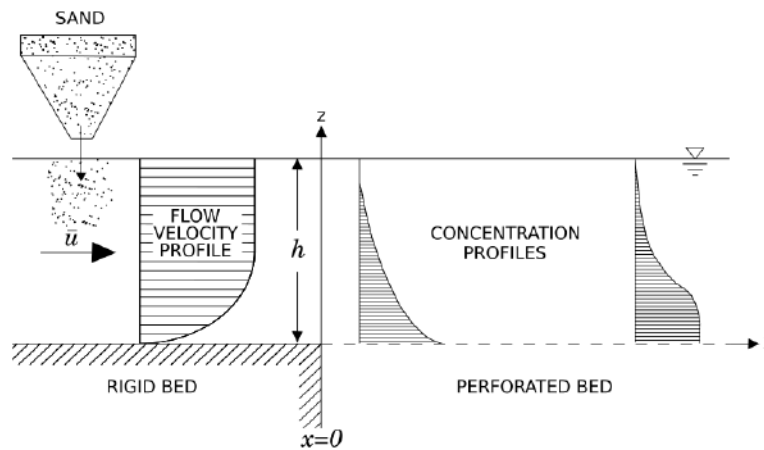


Figure 6.60: Net deposition test case: Schematic definition of the problem.

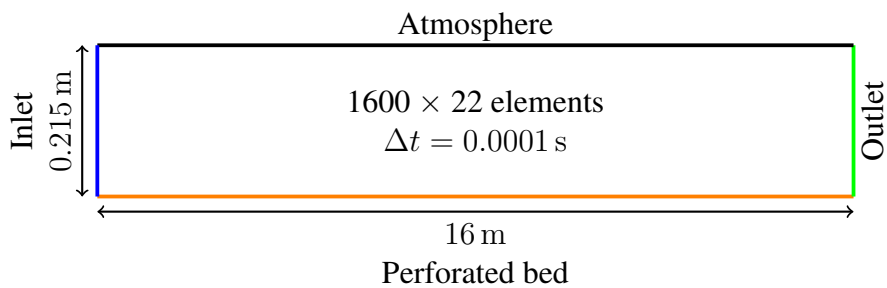


Figure 6.61: Net deposition test case: Geometry, boundary conditions, time-step, and mesh.

putational and experimental sediment concentration distributions for six sections of the channel. We also show results from Liang *et al.* [30] and Wu *et al.* [38]. Once again, it can be observed that our simulation predicts the experimental concentrations very well.

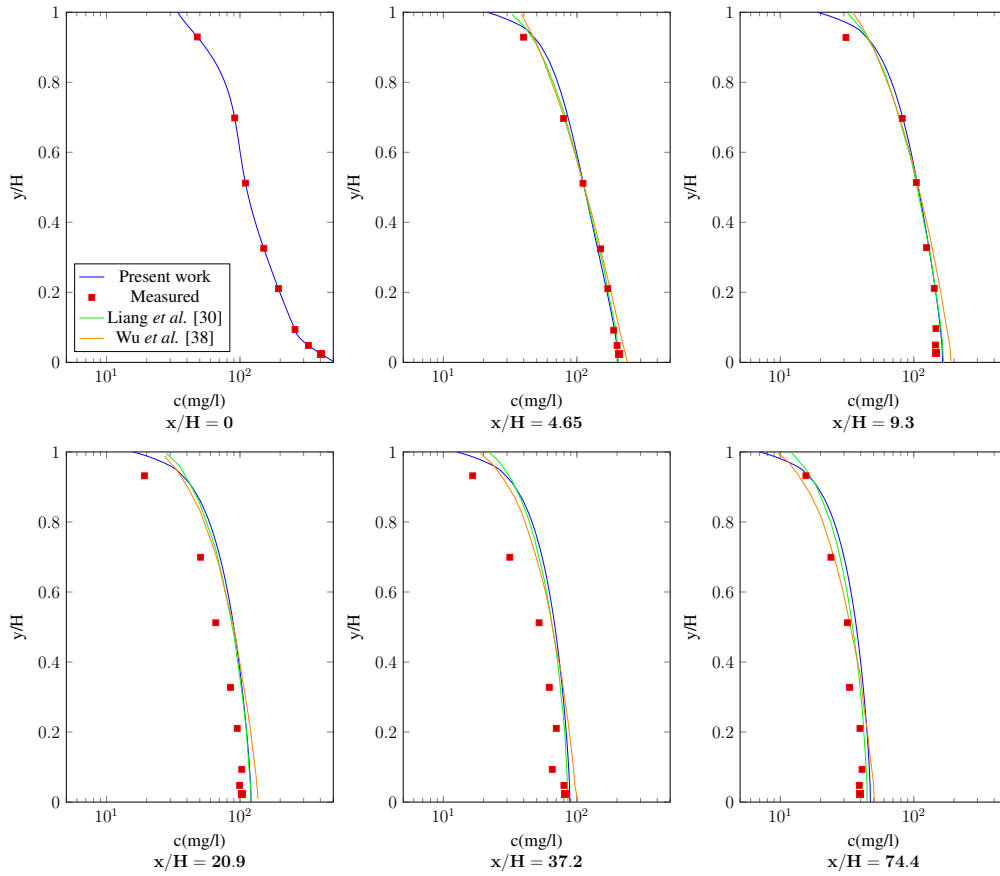


Figure 6.62: Net deposition test case: Comparison of simulations (solid lines) and measured (squares) suspended sediment concentrations.

6.7 Sediment-fluid interaction

Finally, we present two test cases considering bed-load transport and, consequently, sediment-fluid interaction. One reproduces the scour caused by a wall jet flowing over a rigid apron onto an erodible bed, and the other simulates the scour over a pipeline.

6.7.1 Wall Jet Scour test case

This case study investigates the flow field and sediment transport due to a submerged wall jet. The schematic representation of this problem is shown in Figure 6.63. The sediment bottom is initially plane, and due to the flow, it begins to move. The jet is formed by a small opening under a sluice gate and flows over a rigid apron onto an erodible bed. We compare our results with the experimental data of a turbulent wall jet scour by Chatterjee

and Ghosh [148] and Chatterjee *et al.* [149] and also with numerical results obtained by Liu and García [37]. The bed material is sand with $d_{50} = 0.76$ mm, submerged specific gravity 1.65, porosity 0.43, and angle of repose 29° . The apron length is 0.66 m and the jet inlet velocity resulting from the difference between upstream and downstream water depth is 1.56 m/s. We fix the downstream water depth as 0.291 m and the opening gate has 0.02 m. Therefore, the Reynolds number of this problem is equal to $Re = 31200$.

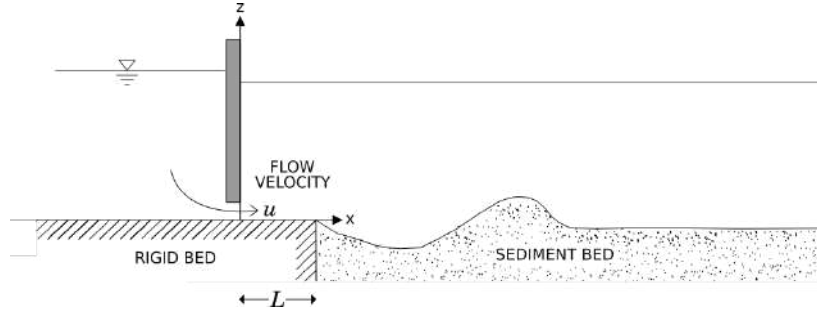


Figure 6.63: Wall Jet Scour test case: Schematic definition of the problem.

The numerical simulation is made with real dimensions, $\Delta t = 0.025$ s, and we use two different meshes, one fixed (i.e., unstructured) and one considering mesh adaptivity. The fixed mesh contains 34085 bilinear quadrilateral elements, while the mesh with adaptivity starts with 17889 elements and finishes with 24300. The fixed mesh is generated with an initial unstructured coarse mesh, and it is refined two times, to have a more dense mesh in the region near the sediment bed. The final mesh is non-conformal, with hanging nodes. The hanging nodes are treated naturally by `libMesh`. For the adaptive mesh refinement we use $h_{max} = 2$, a coarsening fraction $c_f = 0.001$ and a refining fraction $r_f = 0.99$.

Figure 6.64 shows the flow boundary conditions which are: at the inlet, a velocity of 1.56 m/s is applied; at the rigid wall, sediment bed and atmosphere, the slip condition is set; and zero velocity gradient is set at the outlet. The sediment boundary conditions are defined in section 4.1.3.

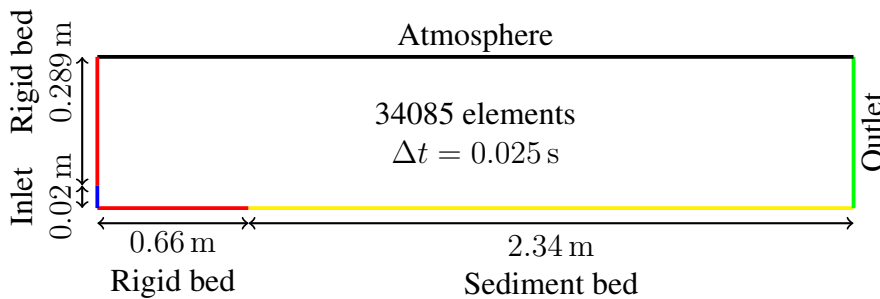


Figure 6.64: Wall Jet Scour test case: Geometry, boundary conditions, time-step, and number of elements in the unstructured mesh.

In this problem, some considerations are necessary. We define that the node between the rigid wall and sediment bottom has zero bed-load rate, since no sediments would leave

the rigid wall. After the sediment bottom reaches the critical sediment angle (30°) and when the bed-load rate increases this angle, we also consider that there is no bed-load rate on these elements. This means that there is bed-load rate only if the elements have the chance to move. Otherwise, the rate is null. This adjustment is necessary to keep the conservation of mass since the bed-load rate removes the sediment of one node to put it on its adjacent node. Meaning that if we cannot move a node because its movement would exceed the critical angle, but despite it, we keep adding sediments on its adjacent node, the result is the addition of mass to the system. We have verified that with our upwind sand-slide scheme, the difference between the initial and final integral of the coordinates of the vertical nodes in the region of interest is of the order of 10^{-7} , indicating that mass is conserved.

We plot the results of the fixed mesh after 60 seconds of simulation, then compare it with the experimental data and with the numerical results obtained by Liu and García [37] (Figure 6.65). They have used the $k - \epsilon$ method as a turbulence closure model, and the morphological changes are tracked within an ALE framework. Also, in Figure 6.66, we show the scour profile in different time-steps. Our results are in good agreement with the experimental and the numerical simulation of Liu and García [37].

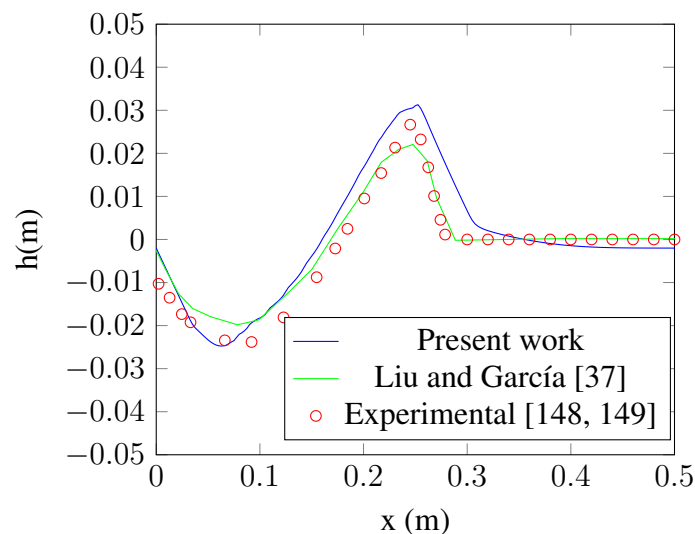


Figure 6.65: Wall Jet Scour test case: Comparison of simulations (solid lines) and measured (circles) scour profile at $t = 60$ s.

In Figure 6.67, we compare the results between the solution with the fixed and adapted meshes. The results are quite similar. Nevertheless, the meshes are considerably different, as we can observe in Figures 6.68 and 6.69. We save a lot of computational effort using the AMR procedure. We can note this when comparing the number of elements in the fixed mesh (34085) and the final number of elements of the adapted mesh (24300). To make this a fair comparison, we have used in the adapted meshes the same element sizes of the fixed mesh in the bed-load boundary. Also, to keep the mass conservation in the

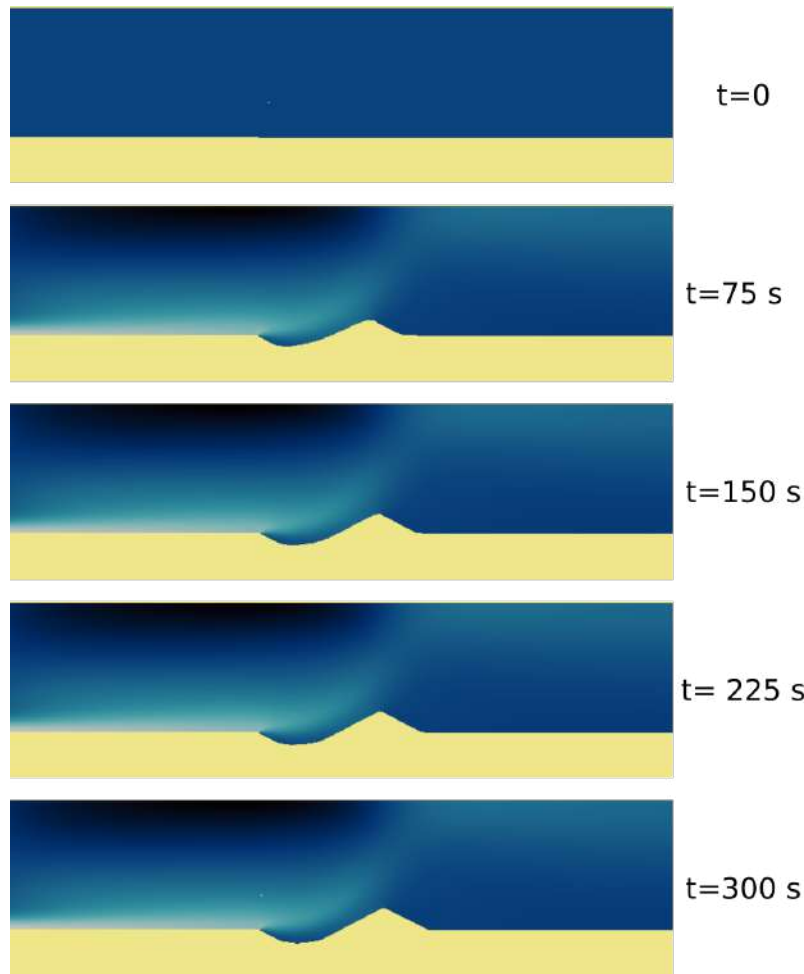


Figure 6.66: Wall Jet Scour test case: Scour profile evolution in different time-steps.

bed-load comparable, we cannot allow this first layer of elements to be refined, since it would change the horizontal distances between nodes. The number of elements of the adapted mesh has increased considerably between zero and 5 seconds of simulation. After that, it keeps oscillating close to 24000 elements.

6.7.2 Scour Over a Pipeline test case

The last test case investigates the scour over a pipe initially placed on a horizontal seabed. The water depth is 0.4 m and the pipeline has a diameter of 0.1 m. The sediment bed is composed of particles of $d_{50} = 0.36$ mm, which has an angle of repose 33° . The mean velocity at the inlet is equal to 0.37 m/s, and the porosity is equal to 0.43. The Reynolds number of this problem is equal to $Re = 37000$. We compare our results with the experimental ones obtained by Mao [150] and with the numerical results obtained by Liang *et al.* [30].

In the experiment, the pipeline is initially placed on the sediment bottom. However, for numerical purposes, we introduce a small hole between the pipeline and the sediment bottom. The gap has a bilinear shape with a depth of 0.01 m. Figure 6.70 shows the

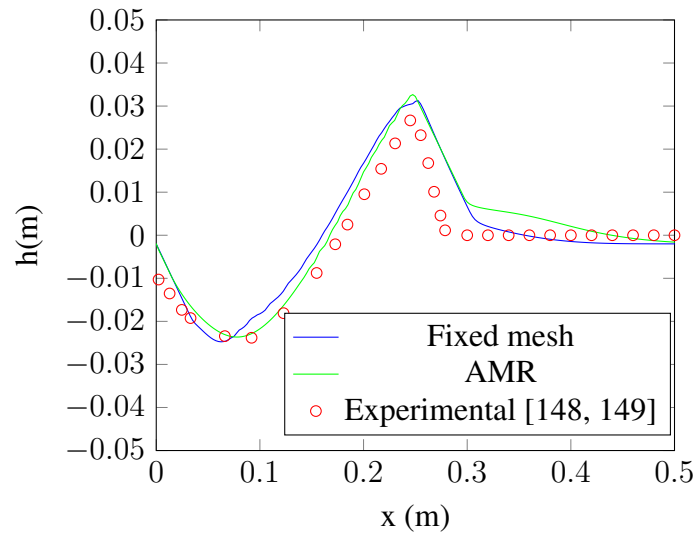


Figure 6.67: Wall Jet Scour test case: Comparison between the solutions for the fixed and the adapted meshes at $t = 60$ s.

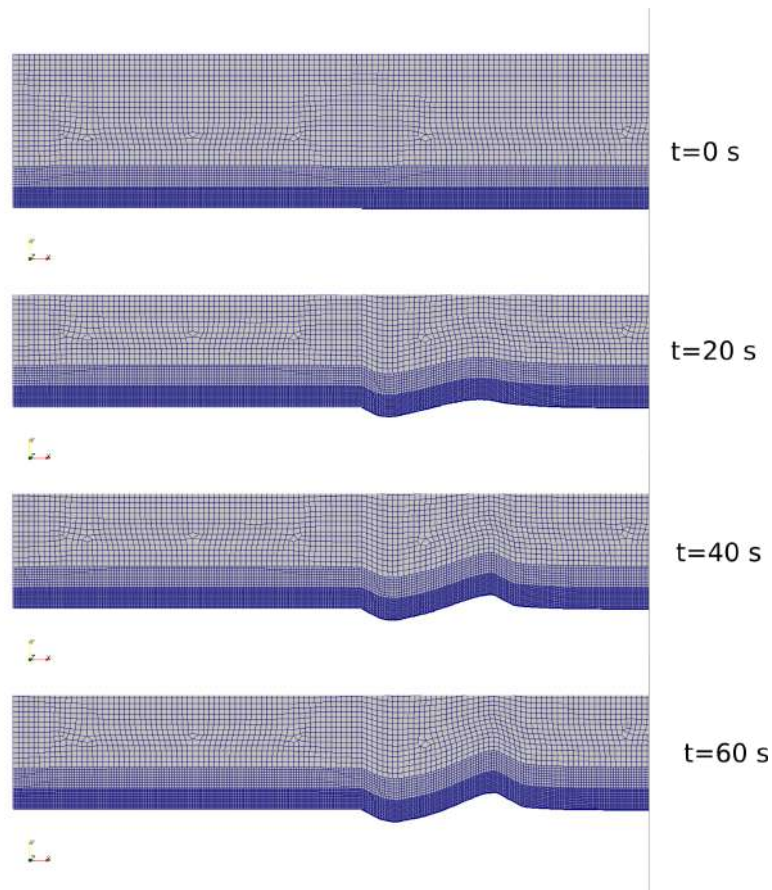


Figure 6.68: Wall Jet Scour test case: Detail of the initial and final unstructured quadrilateral fixed mesh with 34085 elements and hanging nodes at different times.

initial configuration and the flow boundary conditions. They are: at the inlet, a velocity of 0.37 m/s is applied; at the rigid wall, sediment bed and atmosphere, the slip condition is set; and zero velocity gradient is used at the outlet. The sediment boundary conditions

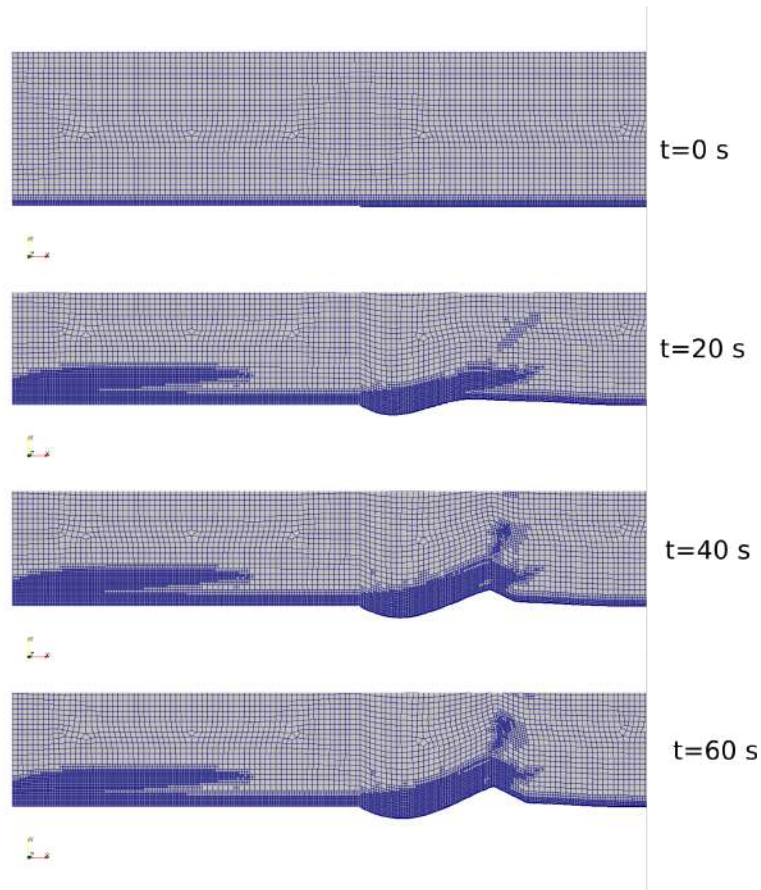


Figure 6.69: Wall Jet Scour test case: Detail of the unstructured quadrilateral adapted meshes at $t=0$, $t=20$, $t=40$ and $t=60$ seconds with respectively 17889, 23853, 24294 and 24300 elements and hanging nodes at different times.

are defined in section 4.1.3.

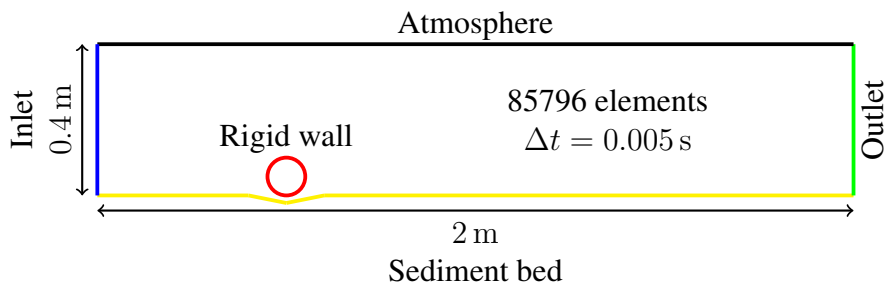


Figure 6.70: Scour over a pipeline test case: Geometry, boundary conditions, time-step, and number of elements in the unstructured mesh.

In this test case, we also use two different meshes, one fixed (i.e., unstructured) and one considering mesh adaptivity. The fixed mesh contains 85796 bilinear quadrilateral elements, while the mesh with adaptivity starts with 18820 elements and finishes with 26641. For the adaptive mesh refinement we use $h_{max} = 2$, a coarsening fraction $c_f = 0.001$ and a refining fraction $r_f = 0.99$.

In Figures 6.71 and 6.72, we can observe the scour profile after 600 seconds of the

fixed mesh simulation and the profiles evolution at different time-steps. In Figure 6.71 we also show the results obtained by Liang *et al.* [30]. The authors have studied two different turbulence models for this test case, the $k - \epsilon$ model, and the Smagorinsky subgrid-scale model. They also use an ALE framework to track the morphological changes. We can observe that our results, using the RBVMS formulation, are closer to the experimental ones. Again, we verify that with our upwind sand-slide scheme, the difference between the initial and final integral of vertical node coordinates in the region of interest is of the order of 10^{-7} , indicating that mass is conserved.

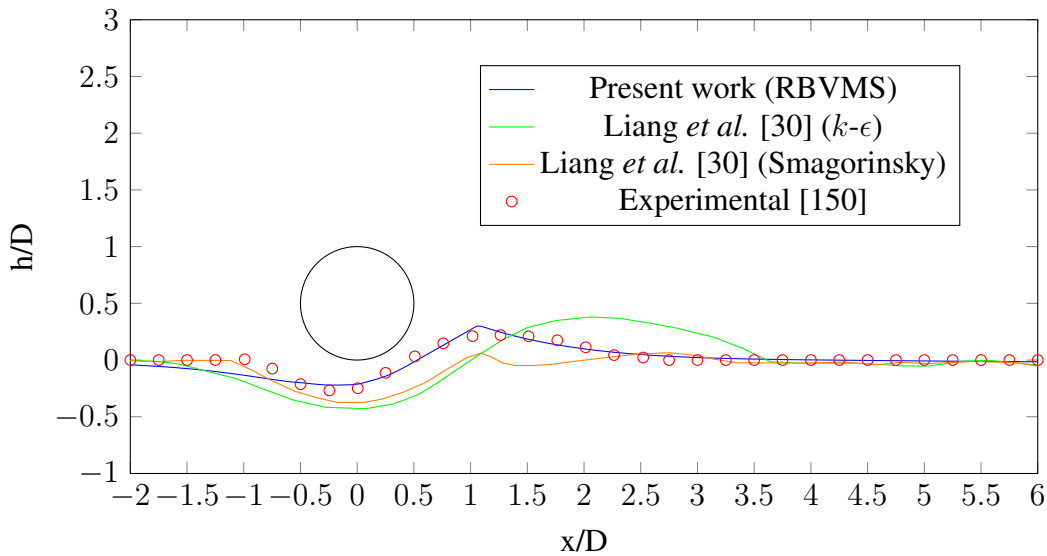


Figure 6.71: Scour over a pipeline test case: Comparison of the simulation (solid lines), with the measured (circles) scour profile at $t = 600s$.

In Figure 6.73, we compare the results between the solution with the fixed and adapted meshes. The results are quite similar, although the meshes are considerably different, as we can observe in Figure 6.74 and 6.75. The number of elements of the adapted mesh has increased considerably between zero and 20 seconds of simulation. Afterward, it keeps oscillating close to 26000 elements. Nevertheless, we saved a lot of computational effort by using adaptive mesh refinement. That can be seen by the number of elements we have used in the fixed mesh (85796) and the final number of elements of the adapted mesh (26641).

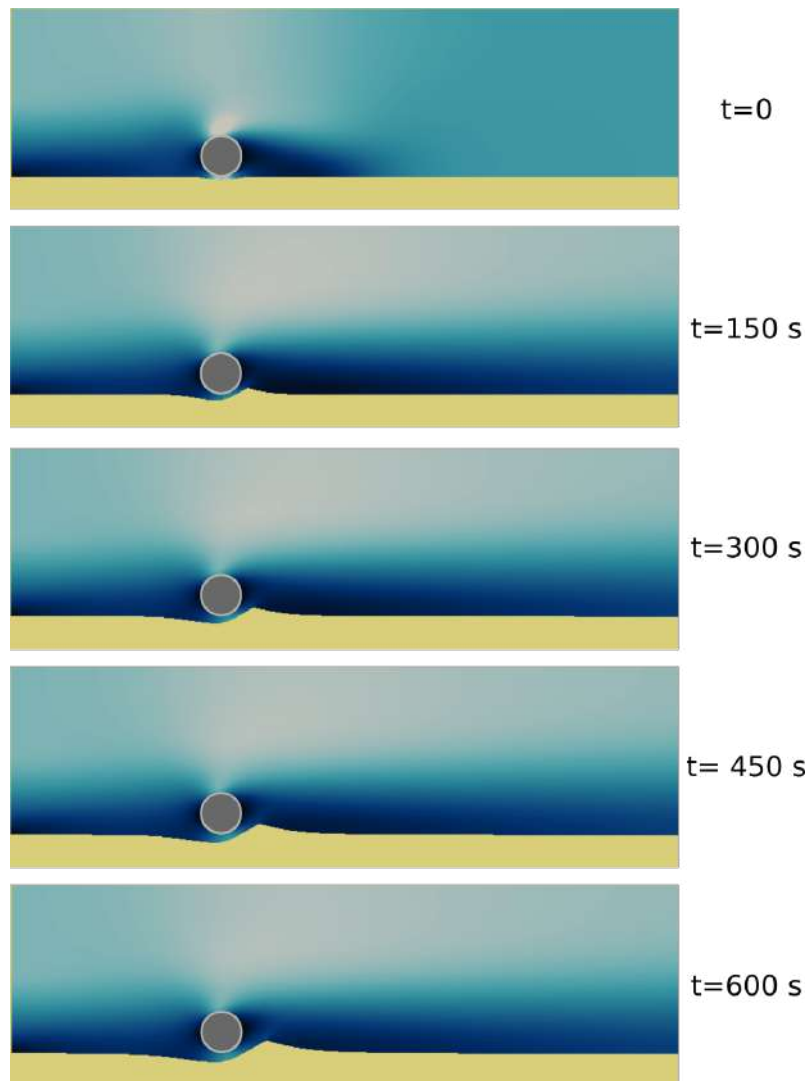


Figure 6.72: Scour over a pipeline test case: Scour profile evolution at different time-steps.

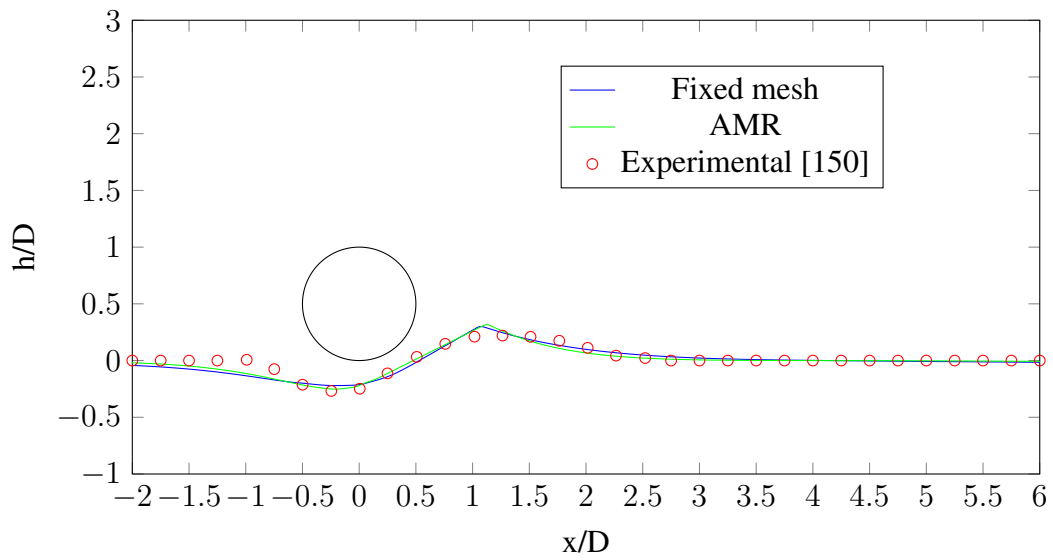


Figure 6.73: Scour over a pipeline test case: Comparison between the solutions for the fixed and the adapted meshes $t = 600s$.

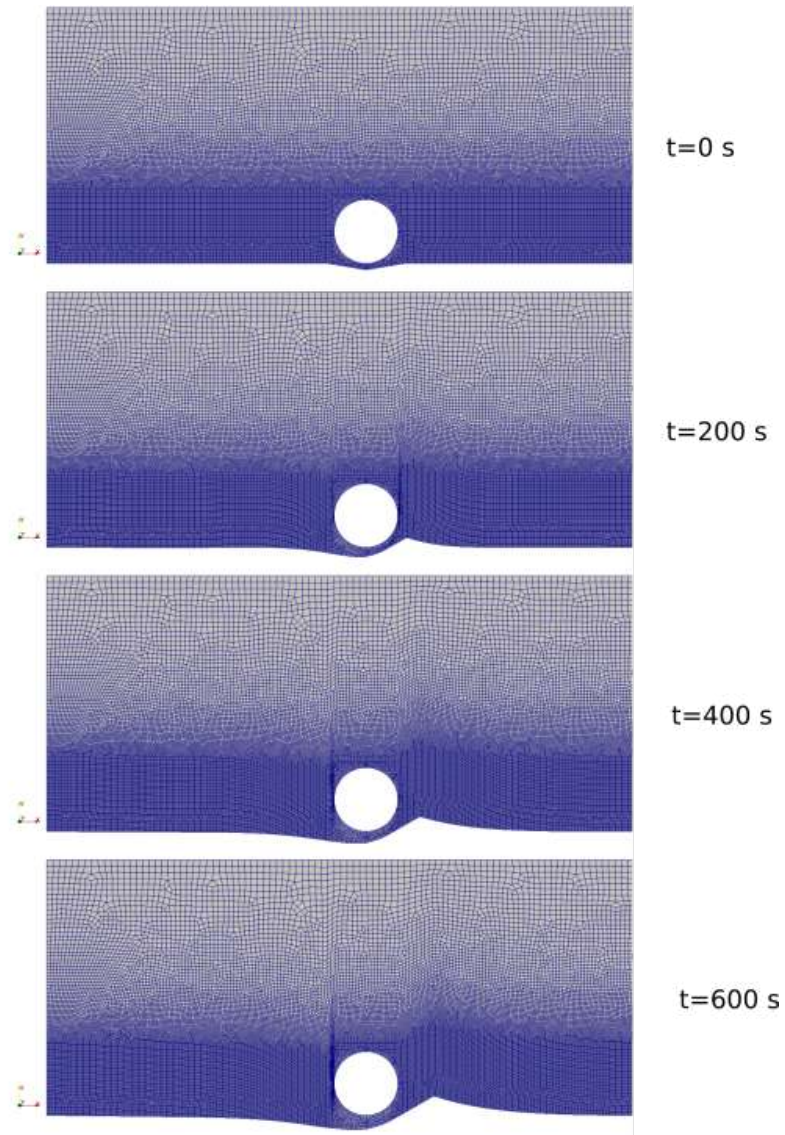


Figure 6.74: Scour over a pipeline test case: Detail of the initial and final unstructured quadrilateral fixed mesh with 85796 elements and hanging nodes at different times.

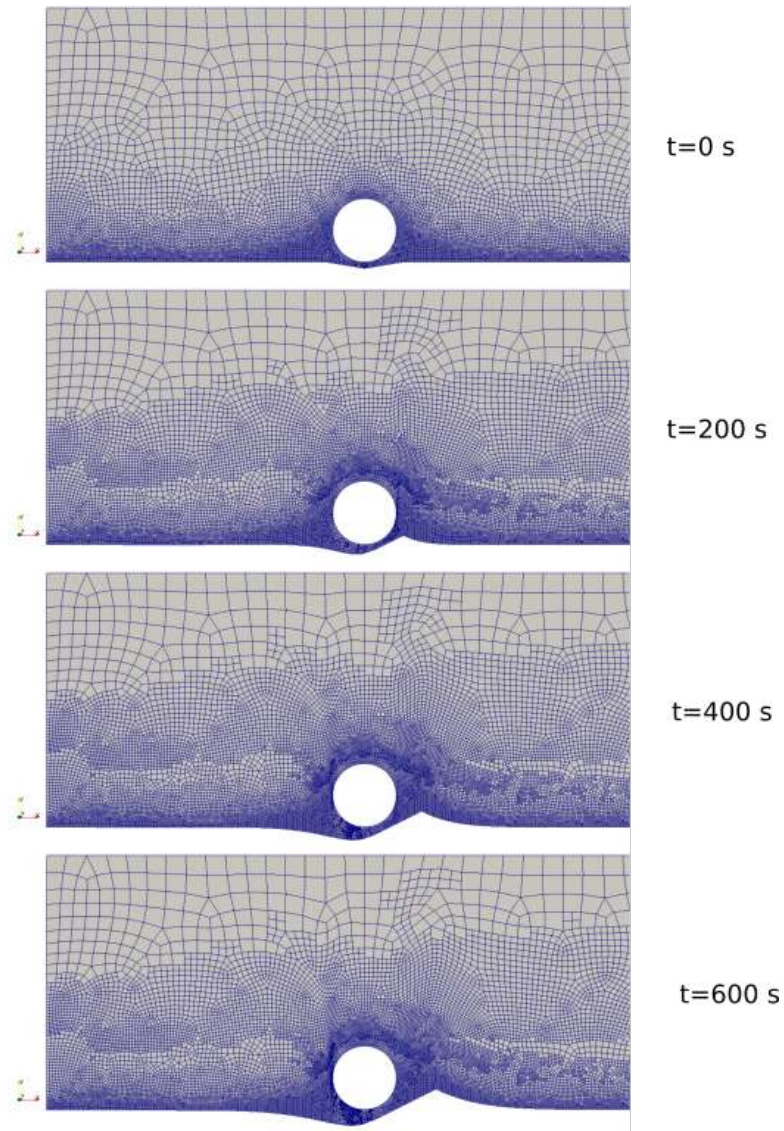


Figure 6.75: Scour over a pipeline test case: Detail of the unstructured quadrilateral adapted meshes at $t=0$, $t=200$, $t=400$ and $t=600$ seconds with respectively 18820, 26617, 25471 and 26641 elements and hanging nodes.

Chapter 7

Conclusions and future work

In this work, we have studied two methods of modeling moving boundaries and surfaces in CFD problems. Among the several methods used for this purpose, we choose to work with two methods quite different from each other. One is an interface tracking, and the other is an interface capturing method. This choice was made considering the type of problem under consideration: two-fluid and sediment-fluid interactions. Both situations are very distinct, the first one is related to the interaction between two fluids with different properties, like water and air, and the second one is about the interaction between a fluid and a sediment bottom that deforms as the flow evolves. Distinct problems that require distinct solutions to be modeled.

Therefore, we have implemented the convected level-set method to model two-fluid problems. The mathematical model results from the Navier-Stokes equations treated with the RBVMS formulation, together with the convected level-set method, used to track the interfaces between the different fluids. The convected level-set method includes the reinitialization step in the main equation, which avoids the extra step that appears in the original level-set formulation. Here, we introduced a new truncated signed distance function to get a smooth truncation far from the interface. We have used the CSF model [100] to model the surface tension between fluids, and we generalize the global mass conservation procedure introduced in [97] to enforce the mass balance, since the convected level-set method does not guarantee mass conservation between phases.

On the other hand, to model sediment-fluid interaction, we have presented a full approach within the ALE framework with mesh adaptivity. The mathematical model results from the Navier-Stokes equations combined with an advection-diffusion equation for suspended sediments, both treated with the RBVMS formulation. The morphological changes are calculated by empirical models used to represent the entrainment rate as well as the bed-load transport. The displacements, and consequently, the interface motion, are applied as mesh movement. We introduced an alternative smoothing technique called upwind sand-slide, in which we adjust the slope angle considering the flow direction. This adjustment is necessary to avoid unrealistic bed shapes that may occur in sediment trans-

port problems. We also applied a first-order Laplacian smoothing to improve numerical stability. It was verified that the upwind sand-slide is a robust method, and the mass is preserved when it is applied.

To validate the implementation of both methods, we have run a series of different problems. First, we have presented numerical results of classical CFD problems to check the implementation of the Navier-Stokes equations. Then, we have presented numerical results for classical interface-capturing benchmarks in order to validate the implementation of the convected-level set method. The convected level-set method showed to be simple to implement, and, together with the global mass enforcement procedure and BDF2 time integration, provided excellent results. The BDF2 method showed being more efficient than the BE for the simulations, and we also have noted the importance of the mesh size and the corresponding time-step. The truncated signed distance function introduced in this work worked quite well with the convected level-set formulation, and it was used as a marker for the adaptive mesh refinement algorithm.

Afterward, we have checked the implementation of the surface tension model in problems with analytical solutions. Then, to validate the two-fluid algorithm, we use two bubbles benchmarks that are available for comparison with other solvers, in 2D and 3D. The 2D benchmark is useful only to validate the code because it does not represent the bubbles rising process physically since it does not consider the contribution of the surface tension in the z -direction. Despite that, our results were in good agreement with other authors for this test case. In the 3D benchmark, we have shown the influence of the mesh refinement, and it was possible to see that the more refined is the mesh, the more accurate the results are. We also have presented comparisons with the bubble shapes and rise velocities of several experimental data, for both single and multiple bubbles. For the single bubbles, we have presented five bubbles regimes of Grace's diagram. For the multiple bubbles, we have presented the co-axial and oblique coalescence of two rising bubbles. Our results have a good qualitative agreement with the experimental ones. In all examples, mass is conserved to machine precision because the global mass enforcement procedure.

To model the sediment-fluid problems, we had to add some different routines of the moving mesh method. Thus, we have verified the mesh movement and the upwind sand-slide algorithms with simple problems. Afterward, the full model of sediment transport was verified against four controlled physical experiments, and it has shown good agreement with the experimental data, not only for the suspended load transport but also for the prediction of sediment profiles.

The RBVMS formulation has shown to provide excellent results for the flow field, particularly when turbulence is important, like in the scour over a pipeline test case and the wall jet scour problem. The sediment profile of the wall jet scour test case has a good prediction using our model with RBVMS as well as the results of Liu and García [37]

that have used the $k - \epsilon$ model. On the other hand, our predicted sediment profile of the scour over a pipeline test case is better than the results obtained by Liang *et al.* [30]. They have used two different turbulence closure models, the $k - \epsilon$ and the Smagorinsky subgrid-scale model. Therefore, it is possible to conclude that the RBVMS formulation is a good option to model sediment transport in fluid flows.

The main difficulty in the implementation of the presented methods is the definition of some sensitive parameters, as the penalty coefficient λ on the convected level-set formulation and the frequency of the application of the Laplacian smoothing procedure on the sediment-fluid boundary. The penalty coefficient λ defines the contribution of the reinitialization equation in the convection equation. A small λ may not be enough to correct the iso-surfaces to recover the signed distance properties adequately, and a large one may change the interface shape. In this work, we have adjusted the value of λ in each simulation. Although $\lambda = 1$ provides good results in almost all simulations, others needed a bigger value. As future work, we aim to study the influence of this parameter and then to define the best value of λ for each simulation.

The Laplacian smoothing has shown to be necessary, but it influences a lot the results of the sediment bottom profile. Depending on how frequently we apply this smoothing, the sediment bottom profile may have significant changes. This smoothing technique tends to let the boundary horizontal. Thus we cannot apply it in every time-step. We have defined an amount of sediment movement to be reached before we apply this procedure. The parameter that influences the sediment bottom profile is the definition of how much sediment movement has to be accumulated.

Other difficulties encountered were related to the stabilization of the methods. The discontinuity-capturing operator has shown to be essential to keep the stabilization of the convected level-set method. Without it, some oscillations around the interface appeared, and the simulation diverged. Also, other oscillations could appear depending on the λ combined with a very small time-step.

Although this work was written presenting the convected level-set before the ALE framework, at the beginning of this study, we started working with sediment transport problems, i.e., the ALE framework. We have developed the code in 2D, and we realized that our procedure was too geometrical, and the scalability for 3D problems would not be trivial. Thus, we have started to study other ways to follow an interface, as the convected level-set method. The convected level-set is easily scalable to 3D, and we started to study it with two objectives: to model two-fluid flows and to model the sediment-fluid interface. Until now, as it was presented, we only finished the two-fluid model. As future work, we aim to predict the sediment bottom profile with the convected level-set method as well. The idea is to model a full approach, including free-surface and bed-load transport, with all interfaces modeled with the convected level-set method.

References

- [1] SHYY, W., UDAYKUMAR, H., RAO, M. M. *Computational fluid dynamics with moving boundaries*. Taylor & Francis, 1996.
- [2] MARIEU, V., BONNETON, P., FOSTER, D. L., et al. “Modeling of vortex ripple morphodynamics”, *Journal of Geophysical Research: Oceans*, v. 113, n. C9, 2008.
- [3] BRERETON, G., KOROTNEY, D. *Coaxial and oblique coalescence of two rising bubbles. Dynamics of bubbles and vortices near a free surface*. ASME, 1991.
- [4] ADELSBERGER, J., ESSER, P., GRIEBEL, M., et al. “3D incompressible two-phase flow benchmark computations for rising droplets”. In: *Proceedings of the 11th world congress on computational mechanics (WCCM XI), Barcelona, Spain*, v. 179, 2014.
- [5] BHAGA, D., WEBER, M. “Bubbles in viscous liquids: shapes, wakes and velocities”, *Journal of fluid Mechanics*, v. 105, pp. 61–85, 1981.
- [6] ZHANG, A., SUN, P., MING, F. “An SPH modeling of bubble rising and coalescing in three dimensions”, *Computer Methods in Applied Mechanics and Engineering*, v. 294, pp. 189–209, 2015.
- [7] HUA, J., LOU, J. “Numerical simulation of bubble rising in viscous liquid”, *Journal of Computational Physics*, v. 222, n. 2, pp. 769–795, 2007.
- [8] NASTAC, L., TMS. *CFD Modeling and Simulation in Materials Processing*. Springer, 2016.
- [9] KAMAKOTI, R., SHYY, W. “Fluid–structure interaction for aeroelastic applications”, *Progress in Aerospace Sciences*, v. 40, n. 8, pp. 535–558, Nov 2004.
- [10] JAFARI, A., HASANI, M., HOSSEINI, M., et al. “Application of CFD technique to simulate enhanced oil recovery processes: current status and future opportunities”, *Petroleum Science*, Sep 2019. ISSN: 1995-8226.

- [11] TEZDUYAR, T. E. “Finite elements in fluids: stabilized formulations and moving boundaries and interfaces”, *International Journal of Numerical Methods for Heat and Fluid Flow*, v. 36, pp. 191–206, 2007.
- [12] HIRT, C. W., AMSDEN, A. A., COOK, J. L. “An arbitrary Lagrangian-Eulerian computing method for all flow speeds”, *Journal of Computational Physics*, v. 14, n. 3, pp. 227–253, 1974.
- [13] DONEA, J., HUERTA, A. *Finite element methods for flow problems*. John Wiley & Sons, 2003.
- [14] PESKIN, C. S. “Flow patterns around heart valves: a numerical method”, *Journal of Computational physics*, v. 10, n. 2, pp. 252–271, 1972.
- [15] PESKIN, C. S. “The immersed boundary method”, *Acta numerica*, v. 11, pp. 479–517, 2002.
- [16] KHALLOUFI, M., VALETTE, R., HACHEM, E. “Adaptive Eulerian framework for boiling and evaporation”, *Journal of Computational Physics*, p. 109030, 2019.
- [17] LUO, X.-W., JI, B., TSUJIMOTO, Y. “A review of cavitation in hydraulic machinery”, *Journal of Hydrodynamics*, v. 28, n. 3, pp. 335–358, 2016.
- [18] BESAGNI, G., INZOLI, F., ZIEGENHEIN, T. “Two-phase bubble columns: A comprehensive review”, *ChemEngineering*, v. 2, n. 2, pp. 13, 2018.
- [19] ARES, S., VOULGARAKIS, N., RASMUSSEN, K., et al. “Bubble nucleation and cooperativity in DNA melting”, *Physical review letters*, v. 94, n. 3, pp. 035504, 2005.
- [20] BENILOV, A. “AIR SEA INTERACTIONS — Surface Waves”. In: North, G. R., Pyle, J., Zhang, F. (Eds.), *Encyclopedia of Atmospheric Sciences (Second Edition)*, second edition ed., Academic Press, pp. 144 – 152, Oxford, 2015. ISBN: 978-0-12-382225-3.
- [21] LYONS, J. J., HANEY, M. M., FEE, D., et al. “Infrasound from giant bubbles during explosive submarine eruptions”, *Nature Geoscience*, v. 12, n. 11, pp. 952–958, 2019.
- [22] LIANG, F., BENNETT, C. R., PARSONS, R. L., et al. “A literature review on behavior of scoured piles under bridges”. In: *Contemporary Topics in In Situ Testing, Analysis, and Reliability of Foundations*, pp. 482–489, 2009.

- [23] BATUCA, D. G., JORDAAN JR, J. *Silting and desilting of reservoirs*. CRC Press, 2000.
- [24] MEIBURG, E., KNELLER, B. “Turbidity Currents and Their Deposits”, *Annual Review of Fluid Mechanics*, v. 42, n. 1, pp. 135–156, 2010.
- [25] KIRK, B. S., PETERSON, J. W., STOGNER, R. H., et al. “libMesh: a C++ library for parallel adaptive mesh refinement/coarsening simulations”, *Journal Engineering with Computers*, v. 22, n. 3, pp. 237–254, 2006.
- [26] ELGETI, S., SAUERLAND, H. “Deforming fluid domains within the finite element method: five mesh-based tracking methods in comparison”, *Archives of Computational Methods in Engineering*, v. 23, n. 2, pp. 323–361, 2016.
- [27] HIRT, C. W., NICHOLS, B. D. “Volume of fluid (VOF) method for the dynamics of free boundaries”, *Journal of Computational physics*, v. 39, n. 1, pp. 201–225, 1981.
- [28] LARSEN, B. E., FUHRMAN, D. R., SUMER, B. M. “Simulation of wave-plus-current scour beneath submarine pipelines”, *Journal of Waterway, Port, Coastal, and Ocean Engineering*, v. 142, n. 5, pp. 04016003, 2016.
- [29] BRØRS, B. “Numerical modeling of flow and scour at pipelines”, *Journal of hydraulic Engineering*, v. 125, n. 5, pp. 511–523, 1999.
- [30] LIANG, D., CHENG, L., LI, F. “Numerical modeling of flow and scour below a pipeline in currents: Part II. Scour simulation”, *Coastal Engineering*, v. 52, n. 1, pp. 43–62, 2005.
- [31] AJDEHAK, E., ZHAO, M., CHENG, L., et al. “Numerical investigation of local scour beneath a sagging subsea pipeline in steady currents”, *Coastal Engineering*, v. 136, pp. 106–118, 2018.
- [32] FAN, F., LIANG, B., LI, Y., et al. “Numerical investigation of the influence of water jumping on the local scour beneath a pipeline under steady flow”, *Water*, v. 9, n. 9, pp. 642, 2017.
- [33] FUHRMAN, D. R., BAYKAL, C., SUMER, B. M., et al. “Numerical simulation of wave-induced scour and backfilling processes beneath submarine pipelines”, *Coastal Engineering*, v. 94, pp. 10–22, 2014.
- [34] RATTIA, J. N., PERCIVAL, J., YEAGER, B., et al. “Numerical simulation of scour below pipelines using flexible mesh methods”. In: *Proceedings Eight International Conference on Scour and Erosion (ICSE2016)*, pp. 101–108, 2016.

- [35] BAYKAL, C., SUMER, B. M., FUHRMAN, D. R., et al. “Numerical simulation of scour and backfilling processes around a circular pile in waves”, *Coastal Engineering*, v. 122, pp. 87–107, 2017.
- [36] BAYKAL, C., SUMER, B. M., FUHRMAN, D. R., et al. “Numerical investigation of flow and scour around a vertical circular cylinder”, *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, v. 373, n. 2033, pp. 20140104, 2015.
- [37] LIU, X., GARCÍA, M. H. “Three-dimensional numerical model with free water surface and mesh deformation for local sediment scour”, *Journal of Waterway, Port, Coastal, and Ocean Engineering*, v. 134, n. 4, pp. 203–217, 2008.
- [38] WU, W., RODI, W., WENKA, T. “3D numerical modeling of flow and sediment transport in open channels”, *Journal of Hydraulic Engineering*, v. 126, n. 1, pp. 4–15, 2000.
- [39] SATTAR, A. M., JASAK, H., SKURIC, V. “Three dimensional modeling of free surface flow and sediment transport with bed deformation using automatic mesh motion”, *Environmental modelling & software*, v. 97, pp. 303–317, 2017.
- [40] CAMATA, J. J., ELIAS, R. N., COUTINHO, A. “FEM Simulation of coupled flow and bed morphodynamic interactions due to sediment transport phenomena”, *Journal of Computational Science and Technology*, v. 7, n. 2, pp. 306–321, 2013.
- [41] COUPEZ, T. “Convection of local level set function for moving surfaces and interfaces in forming flow”. In: *AIP Conference Proceedings*, v. 908, pp. 61–66. AIP, 2007.
- [42] VILLE, L., SILVA, L., COUPEZ, T. “Convected level set method for the numerical simulation of fluid buckling”, *International Journal for numerical methods in fluids*, v. 66, n. 3, pp. 324–344, 2011.
- [43] BAHBAH, C., KHALLOUFI, M., LARCHER, A., et al. “Conservative and adaptive level-set method for the simulation of two-fluid flows”, *Computers & Fluids*, v. 191, pp. 104223, 2019.
- [44] SAGAUT, P., DECK, S., TERRACOL, M. *Multiscale and Multiresolution Approaches in Turbulence*. London, Imperial College Press, 2006.
- [45] HUGHES, T. J. R., FEIJÓO, G. R., MAZZEI, L., et al. “The variational multi-scale method—a paradigm for computational mechanics”, *Computer methods in applied mechanics and engineering*, v. 166, n. 1-2, pp. 3–24, 1998.

- [46] CALO, V. M. *Residual-based multiscale turbulence modeling: finite volume simulations of bypass transition*. Tese de Doutorado, Stanford University USA, 2004.
- [47] RASTHOFER, U., GRAVEMEIER, V. “Recent Developments in Variational Multiscale Methods for large-Eddy Simulation of Turbulent Flow”, *Archives of Computational Methods in Engineering*, pp. 1–44, 2017.
- [48] TEZDUYAR, T. E., RAY, S. E., SHI, R. “Incompressible flow computations with stabilized bilinear and linear equal-order-interpolation”, *Computer Methods Applied in Mechanical Engineering*, v. 95, pp. 221–242, 1992.
- [49] HUGHES, T. J. R., SCOVAZZI, G., FRANCA, L. P. “Multiscale and stabilized methods”, *Encyclopedia of Computational Mechanics Second Edition*, 2004.
- [50] AHMED, N., REBOLLO, T. C., JOHN, V., et al. “A review of variational multiscale methods for the simulation of turbulent incompressible flows”, *Archives of Computational Methods in Engineering*, v. 24, n. 1, pp. 115–164, 2017.
- [51] CODINA, R., BADIA, S., BAIGES, J., et al. “Variational multiscale methods in computational fluid dynamics”, *Encyclopedia of Computational Mechanics Second Edition*, pp. 1–28, 2018.
- [52] ZHAO, Z., FERNANDO, H. J. S. “Numerical simulation of scour around pipelines using an Euler–Euler coupled two-phase model”, *Environmental Fluid Mechanics*, v. 7, n. 2, pp. 121–142, 2007.
- [53] QUEZADA, M., TAMBURRINO, A., NIÑO, Y. “Numerical simulation of scour around circular piles due to unsteady currents and oscillatory flows”, *Engineering Applications of Computational Fluid Mechanics*, v. 12, n. 1, pp. 354–374, 2018.
- [54] AHMAD, N., BIHS, H., KAMATH, A., et al. “Three-dimensional CFD modeling of wave scour around side-by-side and triangular arrangement of piles with REEF3D”, *Procedia Engineering*, v. 116, pp. 683–690, 2015.
- [55] AFZAL, M. S., BIHS, H., KAMATH, A., et al. “Three-dimensional numerical modeling of pier scour under current and waves using level-set method”, *Journal of Offshore Mechanics and Arctic Engineering*, v. 137, n. 3, pp. 032001, 2015.
- [56] LEE, C.-H., XU, C., HUANG, Z. “A three-phase flow simulation of local scour caused by a submerged wall jet with a water-air interface”, *Advances in Water Resources*, 2017.

- [57] KRAFT, S., WANG, Y., OBERLACK, M. “Large eddy simulation of sediment deformation in a turbulent flow by means of level-set method”, *Journal of Hydraulic Engineering*, v. 137, n. 11, pp. 1394–1405, 2011.
- [58] LEFTHERIOTIS, G. A., DIMAS, A. A. “Coupled Simulation of Oscillatory Flow, Sediment Transport and Morphology Evolution of Ripples Based on the Immersed Boundary Method”. In: *ASME 2014 33rd International Conference on Ocean, Offshore and Arctic Engineering*, pp. V08BT06A019–V08BT06A019. American Society of Mechanical Engineers, 2014.
- [59] KHOSRONEJAD, A., SOTIROPOULOS, F. “On the genesis and evolution of barchan dunes: morphodynamics”, *Journal of Fluid Mechanics*, v. 815, pp. 117–148, 2017.
- [60] LUMLEY, J. “Two-phase and non-Newtonian flows”. In: *Turbulence*, Springer, pp. 289–324, 1975.
- [61] VILLARET, C., DAVIES, A. G. “Modeling sediment-turbulent flow interactions”, *Applied mechanics reviews*, v. 48, n. 9, pp. 601–609, 1995.
- [62] GESENHUES, L., CAMATA, J. J., COUTINHO, A. “Simulation of a column collapse for dense granular flows”. In: *In: Proceedings of the XXXVIII Iberian Latin-American Congress on Computational Methods in Engineering*, 2017.
- [63] ZIO, S., DA COSTA, H. F., GUERRA, G. M., et al. “Bayesian assessment of uncertainty in viscosity closure models for turbidity currents computations”, *Computer Methods in Applied Mechanics and Engineering*, v. 342, pp. 653 – 673, 2018. ISSN: 0045-7825.
- [64] NECKER, F., HÄRTEL, C., KLEISER, L., et al. “High-resolution simulations of particle-driven gravity currents”, *International Journal of Multiphase Flow*, v. 28, n. 2, pp. 279–300, 2002.
- [65] LIU, L., YAN, H., ZHAO, G. “Experimental studies on the shape and motion of air bubbles in viscous liquids”, *Experimental Thermal and Fluid Science*, v. 62, pp. 109–121, 2015.
- [66] GRACE, J. “Shapes and velocities of bubbles rising in infinite liquid”, *Transactions of the Institution of Chemical Engineers*, v. 51, pp. 116–120, 1973.
- [67] MANGA, M., STONE, H. “Collective hydrodynamics of deformable drops and bubbles in dilute low Reynolds number suspensions”, *Journal of Fluid Mechanics*, v. 300, pp. 231–263, 1995.

- [68] VAN SINT ANNALAND, M., DEEN, N., KUIPERS, J. “Numerical simulation of gas bubbles behaviour using a three-dimensional volume of fluid method”, *Chemical Engineering Science*, v. 60, n. 11, pp. 2999–3011, 2005.
- [69] XIE, Z., PAVLIDIS, D., SALINAS, P., et al. “A balanced-force control volume finite element method for interfacial flows with surface tension using adaptive anisotropic unstructured meshes”, *Computers & Fluids*, v. 138, pp. 38–50, 2016.
- [70] VAN SINT ANNALAND, M., DIJKHUIZEN, W., DEEN, N., et al. “Numerical simulation of behavior of gas bubbles using a 3-D front-tracking method”, *AIChE Journal*, v. 52, n. 1, pp. 99–110, 2006.
- [71] HUA, J., STENE, J. F., LIN, P. “Numerical simulation of 3D bubbles rising in viscous liquids using a front tracking method”, *Journal of Computational Physics*, v. 227, n. 6, pp. 3358–3382, 2008.
- [72] CHENG, M., HUA, J., LOU, J. “Simulation of bubble–bubble interaction using a lattice Boltzmann method”, *Computers & Fluids*, v. 39, n. 2, pp. 260–270, 2010.
- [73] YU, Z., FAN, L.-S. “An interaction potential based lattice Boltzmann method with adaptive mesh refinement (AMR) for two-phase flow simulation”, *Journal of Computational Physics*, v. 228, n. 17, pp. 6456–6478, 2009.
- [74] AMAYA-BOWER, L., LEE, T. “Single bubble rising dynamics for moderate Reynolds number using Lattice Boltzmann Method”, *Computers & Fluids*, v. 39, n. 7, pp. 1191–1207, 2010.
- [75] YU, Z., YANG, H., FAN, L.-S. “Numerical simulation of bubble interactions using an adaptive lattice Boltzmann method”, *Chemical Engineering Science*, v. 66, n. 14, pp. 3441–3451, 2011.
- [76] SZEWC, K., POZORSKI, J., MINIER, J.-P. “Simulations of single bubbles rising through viscous liquids using smoothed particle hydrodynamics”, *International Journal of Multiphase Flow*, v. 50, pp. 98–105, 2013.
- [77] GRENIER, N., LE TOUZÉ, D., COLAGROSSI, A., et al. “Viscous bubbly flows simulation with an interface SPH model”, *Ocean Engineering*, v. 69, pp. 88–102, 2013.
- [78] KHANWALE, M. A., LOFQUIST, A. D., SUNDAR, H., et al. “Simulating two-phase flows with thermodynamically consistent energy stable Cahn-Hilliard

Navier-Stokes equations on parallel adaptive octree based meshes”, *arXiv preprint arXiv:1912.12453*, 2019.

- [79] CROCE, R., GRIEBEL, M., SCHWEITZER, M. A. “Numerical simulation of bubble and droplet deformation by a level set approach with surface tension in three dimensions”, *International Journal for numerical methods in fluids*, v. 62, n. 9, pp. 963–993, 2010.
- [80] QUEZADA DE LUNA, M., COLLINS, J. H., KEES, C. E. “An unstructured finite element model for incompressible two-phase flow based on a monolithic conservative level set method”, *arXiv preprint arXiv:1903.06919*, 2019.
- [81] HOSSEIN KAZEMINEZHAD, M., YEGANEH-BAKHTIARY, A., ETEMAD-SHAHIDI, A., et al. “Two-phase simulation of wave-induced tunnel scour beneath marine pipelines”, *Journal of Hydraulic Engineering*, v. 138, n. 6, pp. 517–529, 2011.
- [82] KHOSRONEJAD, A., KANG, S., BORAZJANI, I., et al. “Curvilinear immersed boundary method for simulating coupled flow and bed morphodynamic interactions due to sediment transport phenomena”, *Advances in Water Resources*, v. 34, n. 7, pp. 829–843, 2011.
- [83] VAN RIJN, L. C. “Sediment pick-up functions”, *Journal of Hydraulic Engineering*, v. 110, n. 10, pp. 1494–1502, 1984.
- [84] GARCIA, M., PARKER, G. “Entrainment of bed sediment into suspension”, *Journal of Hydraulic Engineering*, v. 117, n. 4, pp. 414–435, 1991.
- [85] VAN RIJN, L. C. “Sediment transport, part I: bed-load transport”, *Journal of Hydraulic engineering*, v. 110, n. 10, pp. 1431–1456, 1984.
- [86] SOULSBY, R. L., WHITEHOUSE, R. J. S. “Threshold of sediment motion in coastal environments”. In: *Pacific Coasts and Ports’ 97: Proceedings of the 13th Australasian Coastal and Ocean Engineering Conference and the 6th Australasian Port and Harbour Conference; Volume 1*, p. 145. Centre for Advanced Engineering, University of Canterbury, 1997.
- [87] ENGELUND, F., FREDSSØE, J. “A sediment transport model for straight alluvial channels”, *Hydrology Research*, v. 7, n. 5, pp. 293–306, 1976.
- [88] OSHER, S., SETHIAN, J. “Fronts propagating with curvature-dependent speed: algorithms based on hamilton-jacobi formulations”, *Journal of Computational Physics*, v. 79, pp. 12–49, 1988.

- [89] ELIAS, R. N., COUTINHO, A. L. “Stabilized edge-based finite element simulation of free-surface flows”, *International Journal for Numerical Methods in Fluids*, v. 54, n. 6-8, pp. 965–993, 2007.
- [90] TOURÉ, M. K., SOULAÏMANI, A. “Stabilized finite element methods for solving the level set equation without reinitialization”, *Computers & Mathematics with Applications*, v. 71, n. 8, pp. 1602–1623, 2016.
- [91] BUTCHER, J. C. *Numerical methods for ordinary differential equations*. John Wiley & Sons, 2016.
- [92] HUGHES, T. J. “Recent progress in the development and understanding of SUPG methods with special reference to the compressible Euler and Navier-Stokes equations”, *International journal for numerical methods in fluids*, v. 7, n. 11, pp. 1261–1275, 1987.
- [93] BAZILEVS, Y., CALO, V. M., TEZDUYAR, T. E., et al. “ γ discontinuity capturing for advection-dominated processes with application to arterial drug delivery”, *International Journal for Numerical Methods in Fluids*, v. 54, n. 6-8, pp. 593–608, 2007.
- [94] GALEÃO, A. C., CARMO, E. G. D. D. “A consistent approximate upwind Petrov-Galerkin method for convection-dominated problems”, *Computer Methods in Applied Mechanics and Engineering*, v. 68, n. 1, pp. 83–95, 1988.
- [95] CHANG, Y.-C., HOU, T., MERRIMAN, B., et al. “A level set formulation of Eulerian interface capturing methods for incompressible fluid flows”, *Journal of computational Physics*, v. 124, n. 2, pp. 449–464, 1996.
- [96] SUSSMAN, M., FATEMI, E. “An efficient, interface-preserving level set redistancing algorithm and its application to interfacial incompressible fluid flow”, *SIAM Journal on scientific computing*, v. 20, n. 4, pp. 1165–1191, 1999.
- [97] SMOLIANSKI, A. *Numerical modeling of two-fluid interfacial flows*. University of Jyväskylä, 2001.
- [98] MASUD, A., HUGHES, T. J. R. “A space-time Galerkin/least-squares finite element formulation of the Navier-Stokes equations for moving domain problems”, *Computer Methods in Applied Mechanics and Engineering*, v. 146, n. 1-2, pp. 91–126, 1997.
- [99] YOKOI, K. “A density-scaled continuum surface force model within a balanced force formulation”, *Journal of Computational Physics*, v. 278, pp. 221–228, 2014.

- [100] BRACKBILL, J. U., KOTHE, D. B., ZEMACH, C. “A continuum method for modeling surface tension”, *Journal of computational physics*, v. 100, n. 2, pp. 335–354, 1992.
- [101] BUSSMANN, M., CHANDRA, S., MOSTAGHIMI, J. “Modeling the splash of a droplet impacting a solid surface”, *Physics of Fluids*, v. 12, n. 12, pp. 3121–3132, 2000.
- [102] CODINA, R., PRINCIPE, J., AVILA, M. “Finite element approximation of turbulent thermally coupled incompressible flows with numerical sub-grid scale modelling”, *International Journal of Numerical Methods for Heat and Fluid Flow*, v. 20, n. 5, pp. 492–516, 2010.
- [103] GRAVEMEIER, V., WALL, W. “Residual-based variational multiscale methods for laminar, transitional and turbulent variable density flow at low mach number”, *International Journal of Numerical Methods for Fluids*, v. 65, pp. 1260–1278, 2011.
- [104] PRINCIPE, J., CODINA, R., HENKE, F. “The dissipative structure of variational multiscale methods for incompressible flows”, *Computer Methods Applied in Mechanical Engineering*, v. 199, pp. 791–801, 2010.
- [105] WADELL, H. “Sphericity and roundness of rock particles”, *The Journal of Geology*, v. 41, n. 3, pp. 310–331, 1933.
- [106] BABUŠKA, I. “The finite element method with penalty”, *Mathematics of Computation*, v. 27, n. 122, pp. 221–228, 1973.
- [107] BAZILEVS, Y., MICHLER, C., CALO, V. M., et al. “Weak Dirichlet boundary conditions for wall-bounded turbulent flows”, *Computer Methods in Applied Mechanics and Engineering*, v. 196, n. 49-52, pp. 4853–4862, 2007.
- [108] SPALDING, D. B. “A Single Formula for the “Law of the Wall””, *Journal of Applied Mechanics*, v. 28, n. 3, pp. 455–458, 1961.
- [109] PAPANASTASIOU, T. C., MALAMATARIS, N., ELLWOOD, K. “A new outflow boundary condition”, *International journal for numerical methods in fluids*, v. 14, n. 5, pp. 587–608, 1992.
- [110] GÜLER, I., BEHR, M., TEZDUYAR, T. “Parallel finite element computation of free-surface flows”, *Computational Mechanics*, v. 23, n. 2, pp. 117–123, 1999.
- [111] CHENG, N.-S. “Simplified settling velocity formula for sediment particle”, *Journal of hydraulic engineering*, v. 123, n. 2, pp. 149–152, 1997.

- [112] SHIELDS, A. *Application of similarity principles and turbulence research to bed-load movement*. Pasadena, California Institute of Technology, 1936.
- [113] MEYER-PETER, E., MÜLLER, R. “Formulas for bed-load transport”. In: *IAHSR 2nd meeting, Stockholm, appendix 2*. IAHR, 1948.
- [114] RIJN, L. C. V. “Sediment transport, Part I: Bed load transport”, *Journal of hydraulic engineering*, v. 110, n. 10, pp. 1431–1456, 1984.
- [115] VANONI, V. A., BROOKS, N. H. *Laboratory studies of the roughness and suspended load of alluvial streams*. Pasadena, California Institute of Technology, 1957.
- [116] PAOLA, C., VOLLER, V. R. “A generalized Exner equation for sediment mass balance”, *Journal of Geophysical Research: Earth Surface*, v. 110, n. F4, 2005.
- [117] ROULUND, A., SUMER, B. M., FREDSSØE, J., et al. “Numerical and experimental investigation of flow and scour around a circular pile”, *Journal of Fluid Mechanics*, v. 534, pp. 351–401, 2005.
- [118] DU PONT, C. S., GONDRET, P., PERRIN, B., et al. “Granular avalanches in fluids”, *Physical Review Letters*, v. 90, n. 4, pp. 044301, 2003.
- [119] STROUSTRUP, B. *The C++ programming language*. India, Pearson Education, 2000.
- [120] SKALICKY, T. “LASPack reference manual”, *Dresden University of Technology*, 1996.
- [121] BALAY, S., ABHYANKAR, S., ADAMS, M. F., et al. “PETSc Web page”. 2017.
- [122] KARYPIS, G., KUMAR, V. “METIS-Serial graph partitioning and fill-reducing matrix ordering”. 2012.
- [123] KARYPIS, G., SCHLOEGEL, K., KUMAR, V. “Parmetis: Parallel graph partitioning and sparse matrix ordering library”, *Version 1.0, Dept. of Computer Science, University of Minnesota*, p. 22, 1997.
- [124] CAMATA, J. J. *Estratégias computacionais de alto desempenho aplicadas ao método dos elementos finitos*. Tese de Doutorado, Federal University of Rio de Janeiro, 2011.
- [125] ROSSA, A. *Simulação paralela e adaptativa de escoamentos incompressíveis de fluidos viscosos pelo método estabilizado dos elementos finitos*. Tese de Doutorado, Federal University of Rio de Janeiro, 2012.

- [126] AINSWORTH, M., ODEN, J. T. *A posteriori error estimation in finite element analysis*, v. 37. John Wiley & Sons, 2011.
- [127] PETERSON, J. W. *Parallel Adaptive Finite Element Methods for Problems in Natural Convection*. Tese de Doutorado, The University of Texas at Austin, 2008.
- [128] ROSSA, A. L., COUTINHO, A. L. “Parallel adaptive simulation of gravity currents on the lock-exchange problem”, *Computers & Fluids*, v. 88, pp. 782–794, 2013.
- [129] KELLY, D., DE SR GAGO, J., ZIENKIEWICZ, O., et al. “A posteriori error analysis and adaptive processes in the finite element method: Part I - error analysis”, *International Journal for Numerical Methods in Engineering*, v. 19, n. 11, pp. 1593–1619, 1983.
- [130] GEUZAIN, C., REMACLE, J. F. “Gmsh: A 3-D finite element mesh generator with built-in pre-and post-processing facilities”, *International journal for numerical methods in engineering*, v. 79, n. 11, pp. 1309–1331, 2009.
- [131] GHIA, U. K. N. G., GHIA, K. N., SHIN, C. T. “High-Re solutions for incompressible flow using the Navier-Stokes equations and a multigrid method”, *Journal of Computational physics*, v. 48, n. 3, pp. 387–411, 1982.
- [132] ERTURK, E., CORKE, T. C., GÖKÇÖL, C. “Numerical solutions of 2-D steady incompressible driven cavity flow at high Reynolds numbers”, *International journal for Numerical Methods in fluids*, v. 48, n. 7, pp. 747–774, 2005.
- [133] COUPEZ, T., HACHEM, E. “Solution of high-Reynolds incompressible flow with stabilized finite element and adaptive anisotropic meshing”, *Computer methods in applied mechanics and engineering*, v. 267, pp. 65–85, 2013.
- [134] GARTLING, D. K. “A test problem for outflow boundary conditions—flow over a backward-facing step”, *International Journal for Numerical Methods in Fluids*, v. 11, n. 7, pp. 953–967, 1990.
- [135] ZALESAK, S. T. “Fully multidimensional flux-corrected transport algorithms for fluids”, *Journal of computational physics*, v. 31, n. 3, pp. 335–362, 1979.
- [136] BELL, J. B., COLELLA, P., GLAZ, H. M. “A second-order projection method for the incompressible Navier-Stokes equations”, *Journal of Computational Physics*, v. 85, n. 2, pp. 257–283, 1989.

- [137] LEVEQUE, R. J. “High-resolution conservative algorithms for advection in incompressible flow”, *SIAM Journal on Numerical Analysis*, v. 33, n. 2, pp. 627–665, 1996.
- [138] KUMAR, R., CHENG, L., XIE, B., et al. “THINC-scaling scheme that unifies VOF and level set methods”, *arXiv preprint arXiv:1908.04529*, 2019.
- [139] SHEPEL, S. V., SMITH, B. L. “On surface tension modelling using the level set method”, *International journal for numerical methods in fluids*, v. 59, n. 2, pp. 147–171, 2009.
- [140] HYSING, S., TUREK, S., KUZMIN, D., et al. “Proposal for quantitative benchmark computations of bubble dynamics”, *International Journal for Numerical Methods in Fluids*, 2007.
- [141] TUREK, S., BECKER, C. “FEATFLOW - Finite element software for the incompressible Navier-Stokes equations.” *User Manual*, 1999.
- [142] PAROLINI, N., BURMAN, E. “A finite element level set method for viscous free-surface flows”. In: *Applied and Industrial Mathematics in Italy*, pp. 416–427. World Scientific, Jun 2005.
- [143] JOHN, V., MATTHIES, G. “MooNMD—a program package based on mapped finite element methods”, *Computing and Visualization in Science*, v. 6, n. 2-3, pp. 163–170, 2004.
- [144] GROSS, S., REUSKEN, A. *Numerical Methods for Two-phase Incompressible Flows*. Springer Berlin Heidelberg, 2011.
- [145] OPENFOAM. *The Open Source CFD Toolbox, User Guide Version 2.2.2*. Relatório técnico, 2013. Disponível em: <<http://www.openfoam.org>>.
- [146] VAN RIJN, L. C. “Application of Sediment Pick-up Function”, *J. Hydraul. Eng.*, v. 112, n. 9, pp. 867–874, 1986.
- [147] WANG, Z. B., RIBBERINK, J. S. “The validity of a depth-integrated model for suspended sediment transport”, *Journal of Hydraulic Research*, v. 24, n. 1, pp. 53–67, 1986.
- [148] CHATERJEE, S. S., GHOSH, S. N. “Submerged horizontal jet over erodible bed”, *Journal of the Hydraulics Division*, v. 106, n. 11, pp. 1765–1782, 1980.
- [149] CHATERJEE, S. S., GHOSH, S. N., CHATTERJEE, . “Local scour due to submerged horizontal jet”, *Journal of Hydraulic Engineering*, v. 120, n. 8, pp. 973–992, 1994.

[150] MAO, Y. *The interaction between a pipeline and an erodible bed*. Tese de Doutorado, Institute of Hydrodynamics and Hydraulic Engineering, Technical University of Denmark, 1986.

Appendix A

RBVMS formulation applied to the Navier-Stokes equations

In the following, we present the terms of the Navier-Stokes equations resulting from applying the RBVMS formulation. The temporal discretization used is the backward Euler method, and Newton's method is used to treat the non-linearity of the problem, being k Newton's iteration counter. On the left-hand side, we gather the terms containing an unknown, whereas all the other terms are taken to the right-hand-side. Here, we are going to consider that,

$$\mathbf{f} = \rho \mathbf{g} + \mathbf{F}_{st}. \quad (\text{A.1})$$

First, we introduce the formulation excluding the partial terms coming from $-\sum_e \frac{1}{\rho} (\nabla \mathbf{w}^h, \tau_m \mathcal{R}_m \otimes \tau_m \mathcal{R}_m)_{\Omega_e}$.

Left-Hand-Side:

$$\begin{aligned}
& (\mathbf{w}^h, \rho \mathbf{u}_{k+1}^h)_\Omega + \Delta t \left((\mathbf{w}^h, \rho \mathbf{u}_{M_{k+1}}^h \cdot \nabla \mathbf{u}_k^h)_\Omega + (\mathbf{w}^h, \rho \mathbf{u}_{M_k}^h \cdot \nabla \mathbf{u}_{k+1}^h)_\Omega \right. \\
& \left. + (\nabla \mathbf{w}^h, \mu \nabla \mathbf{u}_{k+1}^h)_\Omega - (\nabla \cdot \mathbf{w}^h, p_{k+1}^h)_\Omega + (q^h, \nabla \cdot \mathbf{u}_{k+1}^h)_\Omega \right) \\
& + \sum_e \left\{ \left(\nabla \mathbf{w}^h, \rho \mathbf{u}_{k+1}^h \otimes \tau_m \left(\mathbf{u}_k^h - \mathbf{u}_{old} + \Delta t \left(-\nu \Delta \mathbf{u}_k^h + \frac{\nabla p_k^h}{\rho} - \frac{\mathbf{f}}{\rho} \right) \right) \right) \right\}_{\Omega_e} \\
& + \left(\nabla \mathbf{w}^h, \rho \mathbf{u}_k^h \otimes \tau_m \left(\mathbf{u}_{k+1}^h + \Delta t \left(-\nu \Delta \mathbf{u}_{k+1}^h + \frac{\nabla p_{k+1}^h}{\rho} \right) \right) \right)_{\Omega_e} \\
& + \left(\nabla \mathbf{w}^h, \rho \mathbf{u}_{k+1}^h \otimes \tau_m \Delta t (\mathbf{u}_{M_k}^h \cdot \nabla) \mathbf{u}_k^h \right)_{\Omega_e} \\
& + \left(\nabla \mathbf{w}^h, \rho \mathbf{u}_k^h \otimes \tau_m \Delta t (\mathbf{u}_{M_{k+1}}^h \cdot \nabla) \mathbf{u}_k^h \right)_{\Omega_e} \\
& + \left(\nabla \mathbf{w}^h, \rho \mathbf{u}_k^h \otimes \tau_m \Delta t (\mathbf{u}_{M_k}^h \cdot \nabla) \mathbf{u}_{k+1}^h \right)_{\Omega_e} \\
& + \left(\nabla \mathbf{w}^h, \tau_m \left(\mathbf{u}_k^h - \mathbf{u}_{old} + \Delta t \left(-\nu \Delta \mathbf{u}_k^h + \frac{\nabla p_k^h}{\rho} - \frac{\mathbf{f}}{\rho} \right) \right) \otimes \rho \mathbf{u}_{M_{k+1}}^h \right)_{\Omega_e} \\
& + \left(\nabla \mathbf{w}^h, \tau_m \left(\mathbf{u}_{k+1}^h + \Delta t \left(-\nu \Delta \mathbf{u}_{k+1}^h + \frac{\nabla p_{k+1}^h}{\rho} \right) \right) \otimes \rho \mathbf{u}_{M_k}^h \right)_{\Omega_e} \\
& + \left(\nabla \mathbf{w}^h, \tau_m \Delta t (\mathbf{u}_{M_k}^h \cdot \nabla) \mathbf{u}_k^h \otimes \rho \mathbf{u}_{M_{k+1}}^h \right)_{\Omega_e} \\
& + \left(\nabla \mathbf{w}^h, \tau_m \Delta t (\mathbf{u}_{M_{k+1}}^h \cdot \nabla) \mathbf{u}_k^h \otimes \rho \mathbf{u}_{M_k}^h \right)_{\Omega_e} \\
& + \left(\nabla \mathbf{w}^h, \tau_m \Delta t (\mathbf{u}_{M_k}^h \cdot \nabla) \mathbf{u}_{k+1}^h \otimes \rho \mathbf{u}_{M_k}^h \right)_{\Omega_e} \\
& + \Delta t \left(\nabla \cdot \mathbf{w}^h, \rho \tau_c \nabla \cdot \mathbf{u}_{k+1}^h \right)_{\Omega_e} \\
& + \left(\nabla q^h, \tau_m \left(\mathbf{u}_{k+1}^h + \Delta t \left(-\nu \Delta \mathbf{u}_{k+1}^h + \frac{\nabla p_{k+1}^h}{\rho} \right) \right) \right)_{\Omega_e} \\
& + \left(\nabla q^h, \tau_m \Delta t (\mathbf{u}_{M_{k+1}}^h \cdot \nabla) \mathbf{u}_k^h \right)_{\Omega_e} \\
& + \left. \left(\nabla q^h, \tau_m \Delta t (\mathbf{u}_{M_k}^h \cdot \nabla) \mathbf{u}_{k+1}^h \right)_{\Omega_e} \right\}
\end{aligned} \tag{A.2}$$

Right-Hand-Side:

$$\begin{aligned}
& \left(\mathbf{w}^h, \rho \mathbf{u}_{old} \right)_{\Omega} + \Delta t \left(\mathbf{w}^h, \rho \mathbf{u}_{M_k^h} \cdot \nabla \mathbf{u}_k^h \right)_{\Omega} + \Delta t \left(\mathbf{w}^h, \mathbf{f} \right)_{\Omega} + \left(\mathbf{w}^h, \mathbf{t}' \right)_{\bar{\Gamma}} \\
& + \sum_e \left\{ \left(\nabla \mathbf{w}^h, \rho \mathbf{u}_k^h \otimes \tau_m \left(\mathbf{u}_k^h + \Delta t \left(-\nu \Delta \mathbf{u}_k^h + \frac{\nabla p_k^h}{\rho} \right) \right) \right)_{\Omega_e} \right. \\
& + 2 \left(\nabla \mathbf{w}^h, \rho \mathbf{u}_k^h \otimes \tau_m \Delta t (\mathbf{u}_{M_k^h} \cdot \nabla) \mathbf{u}_k^h \right)_{\Omega_e} \\
& + \left. \left(\nabla \mathbf{w}^h, \tau_m \left(\mathbf{u}_k^h + \Delta t \left(-\nu \Delta \mathbf{u}_k^h + \frac{\nabla p_k^h}{\rho} \right) \right) \otimes \rho \mathbf{u}_{M_k^h} \right)_{\Omega_e} \right. \tag{A.3} \\
& + 2 \left(\nabla \mathbf{w}^h, \tau_m \Delta t (\mathbf{u}_{M_k^h} \cdot \nabla) \mathbf{u}_k^h \otimes \rho \mathbf{u}_{M_k^h} \right)_{\Omega_e} \\
& + \left. \left(\nabla q^h, \tau_m \Delta t (\mathbf{u}_{M_k^h} \cdot \nabla) \mathbf{u}_k^h \right)_{\Omega_e} \right\} \\
& + \left(\nabla q^h, \tau_m \mathbf{u}_{old} \right)_{\Omega_e} \\
& + \Delta t \left(\nabla q^h, \tau_m \frac{\mathbf{f}}{\rho} \right)_{\Omega_e} \left. \right\}
\end{aligned}$$

Now, we present the partial terms coming from the $-\sum_e \frac{1}{\rho} (\nabla \mathbf{w}^h, \tau_m \mathcal{R}_m \otimes \tau_m \mathcal{R}_m)_{\Omega_e}$. We introduce this term separately to keep the formulation clear because it is composed by several partial terms.

Left-Hand-Side:

$$\begin{aligned}
& \Delta t \sum_e \left\{ - \left(\nabla \mathbf{w}^h, \rho \tau_m \frac{\mathbf{u}_{k+1}^h}{\Delta t} \otimes \tau_m \frac{\mathbf{u}_k^h}{\Delta t} \right)_{\Omega_e} - \left(\nabla \mathbf{w}^h, \rho \tau_m \frac{\mathbf{u}_k^h}{\Delta t} \otimes \tau_m \frac{\mathbf{u}_{k+1}^h}{\Delta t} \right)_{\Omega_e} \right. \\
& + \left(\nabla \mathbf{w}^h, \rho \tau_m \frac{\mathbf{u}_{k+1}^h}{\Delta t} \otimes \tau_m \frac{\mathbf{u}^{\text{old}}}{\Delta t} \right)_{\Omega_e} - \left(\nabla \mathbf{w}^h, \rho \tau_m \frac{\mathbf{u}_{k+1}^h}{\Delta t} \otimes \tau_m \mathbf{u}_{M_k}^h \cdot \nabla \mathbf{u}_k^h \right)_{\Omega_e} \\
& - \left(\nabla \mathbf{w}^h, \rho \tau_m \frac{\mathbf{u}_k^h}{\Delta t} \otimes \tau_m \mathbf{u}_{M_{k+1}}^h \cdot \nabla \mathbf{u}_k^h \right)_{\Omega_e} - \left(\nabla \mathbf{w}^h, \rho \tau_m \frac{\mathbf{u}_k^h}{\Delta t} \otimes \tau_m \mathbf{u}_{M_k}^h \cdot \nabla \mathbf{u}_{k+1}^h \right)_{\Omega_e} \\
& - \left(\nabla \mathbf{w}^h, \rho \tau_m \frac{\mathbf{u}_{k+1}^h}{\Delta t} \otimes \tau_m \frac{\nabla p_k^h}{\rho} \right)_{\Omega_e} - \left(\nabla \mathbf{w}^h, \rho \tau_m \frac{\mathbf{u}_k^h}{\Delta t} \otimes \tau_m \frac{\nabla p_{k+1}^h}{\rho} \right)_{\Omega_e} \\
& + \left(\nabla \mathbf{w}^h, \rho \tau_m \frac{\mathbf{u}_{k+1}^h}{\Delta t} \otimes \tau_m \nu \Delta \mathbf{u}_k^h \right)_{\Omega_e} + \left(\nabla \mathbf{w}^h, \rho \tau_m \frac{\mathbf{u}_k^h}{\Delta t} \otimes \tau_m \nu \Delta \mathbf{u}_{k+1}^h \right)_{\Omega_e} \\
& + \left(\nabla \mathbf{w}^h, \rho \tau_m \frac{\mathbf{u}_{k+1}^h}{\Delta t} \otimes \tau_m \frac{\mathbf{f}}{\rho} \right)_{\Omega_e} + \left(\nabla \mathbf{w}^h, \rho \tau_m \frac{\mathbf{u}^{\text{old}}}{\Delta t} \otimes \tau_m \frac{\mathbf{u}_{k+1}^h}{\Delta t} \right)_{\Omega_e} \\
& + \left(\nabla \mathbf{w}^h, \rho \tau_m \frac{\mathbf{u}^{\text{old}}}{\Delta t} \otimes \tau_m \mathbf{u}_{M_{k+1}}^h \cdot \nabla \mathbf{u}_k^h \right)_{\Omega_e} + \left(\nabla \mathbf{w}^h, \rho \tau_m \frac{\mathbf{u}^{\text{old}}}{\Delta t} \otimes \tau_m \mathbf{u}_{M_k}^h \cdot \nabla \mathbf{u}_{k+1}^h \right)_{\Omega_e} \\
& + \left(\nabla \mathbf{w}^h, \rho \tau_m \frac{\mathbf{u}^{\text{old}}}{\Delta t} \otimes \tau_m \frac{\nabla p_{k+1}^h}{\rho} \right)_{\Omega_e} - \left(\nabla \mathbf{w}^h, \rho \tau_m \frac{\mathbf{u}^{\text{old}}}{\Delta t} \otimes \tau_m \nu \Delta \mathbf{u}_{k+1}^h \right)_{\Omega_e} \\
& - \left(\nabla \mathbf{w}^h, \rho \tau_m \mathbf{u}_{M_{k+1}}^h \cdot \nabla \mathbf{u}_k^h \otimes \tau_m \frac{\mathbf{u}_k^h}{\Delta t} \right)_{\Omega_e} \\
& - \left(\nabla \mathbf{w}^h, \rho \tau_m \mathbf{u}_{M_k}^h \cdot \nabla \mathbf{u}_{k+1}^h \otimes \tau_m \frac{\mathbf{u}_k^h}{\Delta t} \right)_{\Omega_e} - \left(\nabla \mathbf{w}^h, \rho \tau_m \mathbf{u}_{M_k}^h \cdot \nabla \mathbf{u}_k^h \otimes \tau_m \frac{\mathbf{u}^{k+1}}{\Delta t} \right)_{\Omega_e} \\
& + \left(\nabla \mathbf{w}^h, \rho \tau_m \mathbf{u}_{M_{k+1}}^h \cdot \nabla \mathbf{u}_k^h \otimes \tau_m \frac{\mathbf{u}^{\text{old}}}{\Delta t} \right)_{\Omega_e} + \left(\nabla \mathbf{w}^h, \rho \tau_m \mathbf{u}_{M_k}^h \cdot \nabla \mathbf{u}^{k+1} \otimes \tau_m \frac{\mathbf{u}^{\text{old}}}{\Delta t} \right)_{\Omega_e} \\
& - \left(\nabla \mathbf{w}^h, \rho \tau_m \mathbf{u}_{M_{k+1}}^h \cdot \nabla \mathbf{u}_k^h \otimes \tau_m \mathbf{u}_{M_k}^h \cdot \nabla \mathbf{u}_k^h \right)_{\Omega_e} \\
& - \left(\nabla \mathbf{w}^h, \rho \tau_m \mathbf{u}_{M_k}^h \cdot \nabla \mathbf{u}_{k+1}^h \otimes \tau_m \mathbf{u}_{M_k}^h \cdot \nabla \mathbf{u}_k^h \right)_{\Omega_e} \\
& - \left(\nabla \mathbf{w}^h, \rho \tau_m \mathbf{u}_{M_k}^h \cdot \nabla \mathbf{u}_k^h \otimes \tau_m \mathbf{u}_{M_{k+1}}^h \cdot \nabla \mathbf{u}_k^h \right)_{\Omega_e} \\
& - \left(\nabla \mathbf{w}^h, \rho \tau_m \mathbf{u}_{M_k}^h \cdot \nabla \mathbf{u}_k^h \otimes \tau_m \mathbf{u}_{M_k}^h \cdot \nabla \mathbf{u}_{k+1}^h \right)_{\Omega_e} \\
& \left. - \left(\nabla \mathbf{w}^h, \rho \tau_m \mathbf{u}_{M_{k+1}}^h \cdot \nabla \mathbf{u}_k^h \otimes \tau_m \frac{\nabla p_k^h}{\rho} \right)_{\Omega_e} \right\}
\end{aligned} \tag{A.4}$$

$$\begin{aligned}
& +\Delta t \sum_e \left\{ - \left(\nabla \mathbf{w}^h, \rho \tau_m \mathbf{u}_{M_k}^h \cdot \nabla \mathbf{u}_{k+1}^h \otimes \tau_m \frac{\nabla p_k^h}{\rho} \right)_{\Omega_e} - \left(\nabla \mathbf{w}^h, \rho \tau_m \mathbf{u}_{M_k}^h \cdot \nabla \mathbf{u}_k^h \otimes \tau_m \frac{\nabla p_{k+1}^h}{\rho} \right)_{\Omega_e} \right. \\
& + \left(\nabla \mathbf{w}^h, \rho \tau_m \mathbf{u}_{M_{k+1}}^h \cdot \nabla \mathbf{u}_k^h \otimes \tau_m \nu \Delta \mathbf{u}_k^h \right)_{\Omega_e} + \left(\nabla \mathbf{w}^h, \rho \tau_m \mathbf{u}_{M_k}^h \cdot \nabla \mathbf{u}_{k+1}^h \otimes \tau_m \nu \Delta \mathbf{u}_k^h \right)_{\Omega_e} \\
& + \left(\nabla \mathbf{w}^h, \rho \tau_m \mathbf{u}_{M_k}^h \cdot \nabla \mathbf{u}_k^h \otimes \tau_m \nu \Delta \mathbf{u}_{k+1}^h \right)_{\Omega_e} + \left(\nabla \mathbf{w}^h, \rho \tau_m \mathbf{u}_{M_{k+1}}^h \cdot \nabla \mathbf{u}_k^h \otimes \tau_m \frac{\mathbf{f}}{\rho} \right)_{\Omega_e} \\
& + \left(\nabla \mathbf{w}^h, \rho \tau_m \mathbf{u}_{M_k}^h \cdot \nabla \mathbf{u}_{k+1}^h \otimes \tau_m \frac{\mathbf{f}}{\rho} \right)_{\Omega_e} - \left(\nabla \mathbf{w}^h, \rho \tau_m \nabla p_{k+1}^h \otimes \tau_m \rho \frac{\mathbf{u}_k^h}{\Delta t} \right)_{\Omega_e} \\
& - \left(\nabla \mathbf{w}^h, \rho \tau_m \frac{\nabla p_k^h}{\rho} \otimes \tau_m \rho \frac{\mathbf{u}_{k+1}^h}{\Delta t} \right)_{\Omega_e} + \left(\nabla \mathbf{w}^h, \rho \tau_m \frac{\nabla p_{k+1}^h}{\rho} \otimes \tau_m \frac{\mathbf{u}_{\text{old}}}{\Delta t} \right)_{\Omega_e} \\
& - \left(\nabla \mathbf{w}^h, \rho \tau_m \frac{\nabla p_{k+1}^h}{\rho} \otimes \tau_m \mathbf{u}_{M_k}^h \cdot \nabla \mathbf{u}_k^h \right)_{\Omega_e} - \left(\nabla \mathbf{w}^h, \rho \tau_m \frac{\nabla p_k^h}{\rho} \otimes \tau_m \mathbf{u}_{M_{k+1}}^h \cdot \nabla \mathbf{u}_k^h \right)_{\Omega_e} \\
& - \left(\nabla \mathbf{w}^h, \rho \tau_m \frac{\nabla p_k^h}{\rho} \otimes \tau_m \mathbf{u}_{M_k}^h \cdot \nabla \mathbf{u}_{k+1}^h \right)_{\Omega_e} - \left(\nabla \mathbf{w}^h, \rho \tau_m \frac{\nabla p_{k+1}^h}{\rho} \otimes \tau_m \frac{\nabla p_k^h}{\rho} \right)_{\Omega_e} \\
& - \left(\nabla \mathbf{w}^h, \rho \tau_m \frac{\nabla p_k^h}{\rho} \otimes \tau_m \frac{\nabla p_{k+1}^h}{\rho} \right)_{\Omega_e} + \left(\nabla \mathbf{w}^h, \rho \tau_m \frac{\nabla p_{k+1}^h}{\rho} \otimes \tau_m \nu \Delta \mathbf{u}_k^h \right)_{\Omega_e} \\
& + \left(\nabla \mathbf{w}^h, \rho \tau_m \frac{\nabla p_k^h}{\rho} \otimes \tau_m \nu \Delta \mathbf{u}_{k+1}^h \right)_{\Omega_e} + \left(\nabla \mathbf{w}^h, \rho \tau_m \frac{\nabla p_{k+1}^h}{\rho} \otimes \tau_m \frac{\mathbf{f}}{\rho} \right)_{\Omega_e} \\
& + \left(\nabla \mathbf{w}^h, \rho \tau_m \nu \Delta \mathbf{u}_{k+1}^h \otimes \tau_m \frac{\mathbf{u}_k^h}{\Delta t} \right)_{\Omega_e} + \left(\nabla \mathbf{w}^h, \rho \tau_m \nu \Delta \mathbf{u}_k^h \otimes \tau_m \frac{\mathbf{u}_{k+1}^h}{\Delta t} \right)_{\Omega_e} \\
& - \left(\nabla \mathbf{w}^h, \rho \tau_m \nu \Delta \mathbf{u}_{k+1}^h \otimes \tau_m \frac{\mathbf{u}_{\text{old}}}{\Delta t} \right)_{\Omega_e} + \left(\nabla \mathbf{w}^h, \rho \tau_m \nu \Delta \mathbf{u}_{k+1}^h \otimes \tau_m \mathbf{u}_{M_k}^h \cdot \nabla \mathbf{u}_k^h \right)_{\Omega_e} \\
& + \left(\nabla \mathbf{w}^h, \rho \tau_m \nu \Delta \mathbf{u}_k^h \otimes \tau_m \mathbf{u}_{M_{k+1}}^h \cdot \nabla \mathbf{u}_k^h \right)_{\Omega_e} + \left(\nabla \mathbf{w}^h, \rho \tau_m \nu \Delta \mathbf{u}_k^h \otimes \tau_m \mathbf{u}_{M_k}^h \cdot \nabla \mathbf{u}_{k+1}^h \right)_{\Omega_e} \\
& + \left(\nabla \mathbf{w}^h, \rho \tau_m \nu \Delta \mathbf{u}_{k+1}^h \otimes \tau_m \frac{\nabla p_k^h}{\rho} \right)_{\Omega_e} + \left(\nabla \mathbf{w}^h, \rho \tau_m \nu \Delta \mathbf{u}_k^h \otimes \tau_m \frac{\nabla p_{k+1}^h}{\rho} \right)_{\Omega_e} \\
& - \left(\nabla \mathbf{w}^h, \rho \tau_m \nu \Delta \mathbf{u}_{k+1}^h \otimes \tau_m \nu \Delta \mathbf{u}_k^h \right)_{\Omega_e} - \left(\nabla \mathbf{w}^h, \rho \tau_m \nu \Delta \mathbf{u}_k^h \otimes \tau_m \nu \Delta \mathbf{u}_{k+1}^h \right)_{\Omega_e} \\
& - \left(\nabla \mathbf{w}^h, \rho \tau_m \nu \Delta \mathbf{u}_{k+1}^h \otimes \tau_m \frac{\mathbf{f}}{\rho} \right)_{\Omega_e} + \left(\nabla \mathbf{w}^h, \rho \tau_m \frac{\mathbf{f}}{\rho} \otimes \tau_m \frac{\mathbf{u}_{k+1}^h}{\Delta t} \right)_{\Omega_e} \\
& + \left(\nabla \mathbf{w}^h, \rho \tau_m \frac{\mathbf{f}}{\rho} \otimes \tau_m \mathbf{u}_{M_{k+1}}^h \cdot \nabla \mathbf{u}_k^h \right)_{\Omega_e} + \left(\nabla \mathbf{w}^h, \rho \tau_m \frac{\mathbf{f}}{\rho} \otimes \tau_m \mathbf{u}_{M_k}^h \cdot \nabla \mathbf{u}_{k+1}^h \right)_{\Omega_e} \\
& \left. + \left(\nabla \mathbf{w}^h, \rho \tau_m \frac{\mathbf{f}}{\rho} \otimes \tau_m \frac{\nabla p_{k+1}^h}{\rho} \right)_{\Omega_e} - \left(\nabla \mathbf{w}^h, \rho \tau_m \frac{\mathbf{f}}{\rho} \otimes \tau_m \nu \Delta \mathbf{u}_{k+1}^h \right)_{\Omega_e} \right\}
\end{aligned}$$

Right-Hand-Side:

$$\begin{aligned}
& \Delta t \sum_e \left\{ - \left(\nabla \mathbf{w}^h, \rho \tau_m \frac{\mathbf{u}_k^h}{\Delta t} \otimes \tau_m \frac{\mathbf{u}_k^h}{\Delta t} \right)_{\Omega_e} - 2 \left(\nabla \mathbf{w}^h, \rho \tau_m \frac{\mathbf{u}_k^h}{\Delta t} \otimes \tau_m \mathbf{u}_{M_k}^h \cdot \nabla \mathbf{u}_k^h \right)_{\Omega_e} \right. \\
& - \left(\nabla \mathbf{w}^h, \rho \tau_m \frac{\mathbf{u}_k^h}{\Delta t} \otimes \tau_m \frac{\nabla p_k^h}{\rho} \right)_{\Omega_e} + \left(\nabla \mathbf{w}^h, \rho \tau_m \frac{\mathbf{u}_k^h}{\Delta t} \otimes \tau_m \nu \Delta \mathbf{u}_k^h \right)_{\Omega_e} \\
& + \left(\nabla \mathbf{w}^h, \rho \tau_m \frac{\mathbf{u}^{\text{old}}}{\Delta t} \otimes \tau_m \frac{\mathbf{u}^{\text{old}}}{\Delta t} \right)_{\Omega_e} + \left(\nabla \mathbf{w}^h, \rho \tau_m \frac{\mathbf{u}^{\text{old}}}{\Delta t} \otimes \tau_m \mathbf{u}_{M_k}^h \cdot \nabla \mathbf{u}_k^h \right)_{\Omega_e} \\
& + \left(\nabla \mathbf{w}^h, \rho \tau_m \frac{\mathbf{u}^{\text{old}}}{\Delta t} \otimes \tau_m \frac{\mathbf{f}}{\rho} \right)_{\Omega_e} - 2 \left(\nabla \mathbf{w}^h, \rho \tau_m \mathbf{u}_{M_k}^h \cdot \nabla \mathbf{u}_k^h \otimes \tau_m \frac{\mathbf{u}_k^h}{\Delta t} \right)_{\Omega_e} \\
& + \left(\nabla \mathbf{w}^h, \rho \tau_m \mathbf{u}_{M_k}^h \cdot \nabla \mathbf{u}_k^h \otimes \tau_m \frac{\mathbf{u}^{\text{old}}}{\Delta t} \right)_{\Omega_e} - 3 \left(\nabla \mathbf{w}^h, \rho \tau_m \mathbf{u}_{M_k}^h \cdot \nabla \mathbf{u}_k^h \otimes \tau_m \mathbf{u}_{M_k}^h \cdot \nabla \mathbf{u}_k^h \right)_{\Omega_e} \\
& - 2 \left(\nabla \mathbf{w}^h, \rho \tau_m \mathbf{u}_{M_k}^h \cdot \nabla \mathbf{u}_k^h \otimes \tau_m \frac{\nabla p_k^h}{\rho} \right)_{\Omega_e} + 2 \left(\nabla \mathbf{w}^h, \rho \tau_m \mathbf{u}_{M_k}^h \cdot \nabla \mathbf{u}_k^h \otimes \tau_m \nu \Delta \mathbf{u}_k^h \right)_{\Omega_e} \\
& + \left(\nabla \mathbf{w}^h, \rho \tau_m \mathbf{u}_{M_k}^h \cdot \nabla \mathbf{u}_k^h \otimes \tau_m \frac{\mathbf{f}}{\rho} \right)_{\Omega_e} - \left(\nabla \mathbf{w}^h, \rho \tau_m \frac{\nabla p_k^h}{\rho} \otimes \tau_m \frac{\mathbf{u}_k^h}{\Delta t} \right)_{\Omega_e} \\
& - 2 \left(\nabla \mathbf{w}^h, \rho \tau_m \frac{\nabla p_k^h}{\rho} \otimes \tau_m \mathbf{u}_{M_k}^h \cdot \nabla \mathbf{u}_k^h \right)_{\Omega_e} - \left(\nabla \mathbf{w}^h, \rho \tau_m \frac{\nabla p_k^h}{\rho} \otimes \tau_m \frac{\nabla p_k^h}{\rho} \right)_{\Omega_e} \\
& + \left(\nabla \mathbf{w}^h, \rho \tau_m \frac{\nabla p_k^h}{\rho} \otimes \tau_m \nu \Delta \mathbf{u}_k^h \right)_{\Omega_e} + \left(\nabla \mathbf{w}^h, \rho \tau_m \nu \Delta \mathbf{u}_k^h \otimes \tau_m \frac{\mathbf{u}_k^h}{\Delta t} \right)_{\Omega_e} \\
& + 2 \left(\nabla \mathbf{w}^h, \rho \tau_m \nu \Delta \mathbf{u}_k^h \otimes \tau_m \mathbf{u}_{M_k}^h \cdot \nabla \mathbf{u}_k^h \right)_{\Omega_e} + \left(\nabla \mathbf{w}^h, \rho \tau_m \nu \Delta \mathbf{u}_k^h \otimes \tau_m \frac{\nabla p_k^h}{\rho} \right)_{\Omega_e} \\
& - \left(\nabla \mathbf{w}^h, \rho \tau_m \nu \Delta \mathbf{u}_k^h \otimes \tau_m \nu \Delta \mathbf{u}_k^h \right)_{\Omega_e} + \left(\nabla \mathbf{w}^h, \rho \tau_m \frac{\mathbf{f}}{\rho} \otimes \tau_m \frac{\mathbf{u}^{\text{old}}}{\Delta t} \right)_{\Omega_e} \\
& \left. + \left(\nabla \mathbf{w}^h, \rho \tau_m \frac{\mathbf{f}}{\rho} \otimes \tau_m \mathbf{u}_k^h \cdot \nabla \mathbf{u}_k^h \right)_{\Omega_e} + \left(\nabla \mathbf{w}^h, \rho \tau_m \frac{\mathbf{f}}{\rho} \otimes \tau_m \frac{\mathbf{f}}{\rho} \right)_{\Omega_e} \right\} \tag{A.5}
\end{aligned}$$

Appendix B

RBVMS formulation applied to the sediment transport equations

Finally, the terms for the suspended load equation, resulting from applying the RB-VMS formulation, are presented as follows.

Right-Hand-Side:

$$\left(v^h, c_{old}\right)_{\Omega} + \sum_e \left(\left(\mathbf{u}_{M_k}^h + u_s \mathbf{e}^g\right) \cdot \nabla v^h, \tau_t c_{old}\right)_{\Omega} \quad (\text{B.1})$$

Left-Hand-Side:

$$\begin{aligned} & \left(v^h, c_{k+1}^h\right)_{\Omega} - \Delta t \left(\nabla v^h, \left(\mathbf{u}_{M_k}^h + u_s \mathbf{e}^g\right) c_{k+1}^h\right)_{\Omega} + \Delta t \left(\nabla v^h, \epsilon_s \nabla c_{k+1}^h\right)_{\Omega} \\ & + \sum_e \left(\left(\mathbf{u}_{M_k}^h + u_s \mathbf{e}^g\right) \cdot \nabla v^h, \tau_t c_{k+1}^h\right)_{\Omega_e} \\ & + \Delta t \sum_e \left(\left(\mathbf{u}_{M_k}^h + u_s \mathbf{e}^g\right) \cdot \nabla v^h, \tau_t \left(\mathbf{u}_{M_k}^h + u_s \mathbf{e}^g\right) \cdot \nabla c_{k+1}^h\right)_{\Omega_e} \\ & - \Delta t \sum_e \left(\left(\mathbf{u}_{M_k}^h + u_s \mathbf{e}^g\right) \cdot \nabla v^h, \tau_t \epsilon_s \Delta c_{k+1}^h\right)_{\Omega_e} \end{aligned} \quad (\text{B.2})$$