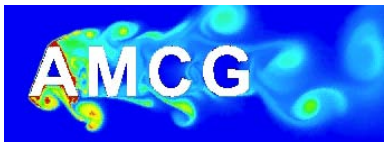
Imperial College
London

Interface Tracking and Solid-Fluid Coupling Techniques with Coastal Engineering Applications

by

Julian Eduardo Mindel

A thesis submitted in fulfilment of the requirements for the degree of Doctor of Philosophy and the Diploma of Imperial College



Applied Modelling and Computational Group

Department of Earth Science and Engineering

Imperial College of Science, Technology, and Medicine

September 2008

For being a continuous source of inspiration, I dedicate this thesis

to my parents, Eduardo and Susana,

my brother Darío,

and the loving memory of my grandparents.

Declaration of originality

The material presented in this thesis is entirely the result of my own independent research under the supervision of Dr. John Paul Latham, Professor Christopher Pain, and Professor Antonio Munjiza. All published or unpublished material used in this thesis has been given full acknowledgement.

Name: Julian Eduardo Mindel Date: 26th February 2009

Signature:

List of Publications

Journal Papers:

Latham, J.-P., Munjiza, A., Mindel , J., Xiang, J., Guises, R., Garcia, X. Pain, C.C., Gorman, G. and Piggott, M. 2008. Towards the modelling of massive particulates for breakwater engineering using coupled FEMDEM and CFD. *Particuology* (Volume 6, Issue 5, October 2008, In Press: Corrected Proof available on line.)

Latham, J.-P., Mindel , J., Xiang, J., Guises, R., Garcia, X. Pain, C.C., Gorman, G. and Munjiza, A. 2008. Coupled FEMDEM/Fluids for coastal engineers with special reference to armour stability and breakage. *Geomechanics and Geoengineering* (submitted April 2008).

Conference Proceedings:

Mindel, J.E., Collins, G.S., Latham, J.-P., Pain C.C. and Munjiza, A. 2007. Towards a numerical wave simulator using the two-fluid interface tracking approach combined with a novel ALE scheme. Fifth International Conference on Coastal Structures, Venice, Italy, July 2-4, 2007. Published by COPRI on behalf of ASCE.

Xiang, J., Mindel, J.E., Munjiza, A., Latham, J.-P., Guises, R., Garcia, X. and Pain C.C. 2007. A combined FEM and DEM fluids-particulates model of waves interacting with granular media. Fifth International Conference on Coastal Structures, Venice, Italy, July 2-4, 2007. Published by COPRI on behalf of ASCE.

Latham, J.-P., Mindel, J., Guises, R., Garcia, X., Xiang, J., Pain. C and Munjiza A. Coupled DEM and CFD for coastal structures: application to armour stability and breakage. Fifth International Conference on Coastal Structures, Venice, Italy, July 2-4, 2007. Published by COPRI on behalf of ASCE.

Latham, J.-P., Munjiza, A., Mindel , J., Xiang, J., Guises, R., Pain, C.C., Gorman, G. and Garcia, X. 2007. Towards simulation of concrete armour unit breakage under wave loads using coupled DEM, FEM/DEM and CFD. Proceedings of Fifth International Conference on Discrete Element Methods, Brisbane, Aug 27-29, 2007.

Latham, J.-P., Mindel, J., Xiang, J., R., Garcia, X. Pain, C.C., Gorman, G. and Munjiza, A. 2008. Development of a modelling approach for large granular media used in coastal applications. Proceedings of International Symposium/RCUK-China Summer School on Discrete Element methods and Numerical Modelling of Discontinuum Mechanics (Beijing DEM'08), Ed. Li S.H. and Feng, Y. T. pp 122-130. Beijing China, 24-30 Sept. 2008.

Xiang, J., Latham, J.-P., Munjiza, A. and Mindel, J. 2008. Applications of the combined finite-discrete element method. Proceedings of International Symposium/RCUK-China Summer School on Discrete Element methods and Numerical Modelling of Discontinuum Mechanics (Beijing DEM'08), Ed. Li S.H. and Feng, Y. T. pp 223-231 Beijing China, 24-30 Sept. 2008.

Abstract

Multi-material physics arise in an innumerable amount of engineering problems. A broadly scoped numerical model is developed and described in this thesis to simulate the dynamic interaction of multi-fluid and solid systems. It is particularly aimed at modelling the interaction of two immiscible fluids with solid structures in a coastal engineering context; however it can be extended to other similar areas of research. The Navier Stokes equations governing the fluids are solved using a combination of finite element (FEM) and control volume finite element (CVFE) discretisations. The sharp interface between the fluids is obtained through the compressive transport of material properties (e.g. material concentration). This behaviour is achieved through the CVFE method and a conveniently limited flux calculation scheme based on the Hyper-C method by Leonard (1991). Analytical and validation test cases are provided, consisting of steady and unsteady flows. To further enhance the method, improve accuracy, and exploit Lagrangian benefits, a novel moving mesh method is also introduced and tested. It is essentially an Arbitrary Lagrangian Eulerian method in which the grid velocity is defined by semi-explicitly solving an iterative functional minimisation problem.

A multi-phase approach is used to introduce solid structure modelling. In this approach, solution of the velocity field for the fluid phase is obtained using Model B as explained by Gidaspow (1994, page 151). Interaction between the fluid phase and the solids is achieved through the means of a source term included in the fluid momentum equations. The interacting force is calculated through integration of this source term and adding a buoyancy contribution. The resulting force is passed to an external solid-dynamics model such as the Discrete Element Method (DEM), or the combined Finite Discrete Element Method (FEMDEM).

The versatility and novelty of this combined modelling approach stems from its ability to capture the fluid interaction with particles of random size and shape. Each of the three main components of this thesis: the advection scheme, the moving mesh method, and the solid interaction are individually validated, and examples of randomly shaped and sized particles are shown. To conclude the work, the methods are combined together in the context of coastal engineering applications, where the complex coupled problem of waves impacting on breakwater amour units is chosen to demonstrate the simulation possibilities. The three components developed in this thesis significantly extend the application range of already powerful tools, such as Fluidity, for fluids-modelling and finite discrete element solids-modelling tools by bringing them together for the first time.

Acknowledgements

Firstly, I would like to thank my supervisors Dr. John-Paul Latham and Professor Christopher Pain for allowing me to research the areas of coastal engineering and numerical modelling. Most of the important work achieved in this thesis could not have been done without their encouragement, support, and guidance. Both of them have provided me with sound advice and in many cases overflowing enthusiasm for my work. Their time and efforts are deeply appreciated. I would also like to mention the help of Professor Antonio Munjiza who, as another one of my supervisors, provided most of the code relating to the combined finite discrete element method along with interestingly insightful coffee chats.

My special thanks go as well to people who have strongly influenced me during my research, and they include Dr. Gerard "G" Gorman, Dr. Matthew Piggott, and Dr. Gareth Collins. Their guidance, insight, and the fact that they consistently went out of their way to help me is also greatly appreciated.

I particularly appreciate all the help, good will, and company provided by the different members of the AMCG group. They were very patient when I was the cause of many a build bot failure.

I would also like to thank the members of the VGW group, Dr. Jiansheng Xiang and Xavier Garcia, for their help and guidance with solid mechanics modelling, among many other shared moments.

No part of this research experience would have been the same without the help, attention, patience, and care of my good old friends who have been by my side even before the beginning, for better or for worse: Guillermo Gonzalo Schiava D'Albano, Mariano Sallito, and Mathias Bertrand. Additionally, I have acquired some very good friends along the way which also deserve mentioning, and they are Romain Guises, Karl Charvin, Martin Putz, and Reinhard Schuetz.

I couldn't thank my girlfriend Michela Quaranta enough for her love and patience, especially during turbulent thesis-writing times.

Lastly but most importantly, I wish to thank my parents, Susana R. Cortese and Eduardo A. Mindel. They bore me, raised me, support me, and love me. To them I dedicate this thesis.

The author is also particularly grateful for funding from EPSRC under grant GR/S42699/01 and industrial support from Sogreah, CLI and Baird Associates.

Contents

1 Introduction **29**

1.1 Background 29

1.1.1 Climate change, sea level rise, and coastal protection 30

1.1.2 The wave hydraulic-load problem 32

1.2 VGW - Virtual Geoscience Workbench 35

1.3 Fluidity 36

1.4 Contributions of this work 37

1.4.1 Interface tracking 38

1.4.2 Mesh movement through an ALE scheme 38

1.4.3 Solid-fluid interaction 39

1.5 Thesis outline 39

2 Interface modelling **43**

2.1 Background 44

2.1.1 Level Set Method 45

2.1.2 Volume Of Fluid method 46

2.1.3 Pseudo Concentration Method 47

2.2 The interface tracking problem 48

2.2.1 Example 1: Forward time first order upwinding 50

2.2.2 Example 2: Second order centred (Lax-Wendroff) 52

2.2.3 Example 3: Water collapse problem (2D) 53

2.3 Compressive advection 54

2.4 Discretisation formulation and solution 58

2.4.1 Governing equations 58

2.4.2 Galerkin weighted residual approach 60

2.4.3 Piecewise-linear basis functions 62

2.4.4 Discretisation of the momentum equations 64

2.4.5 CVFEM method 65

2.4.6 Implementing Hyper-C 67

2.4.7 Solution mechanism 69

2.5 Analytical validation tests . . . 70
2.5.1 Zalesak's problem . . . 71
2.5.2 Shearing flow . . . 73
2.6 Experimental validation tests . . . 76
2.6.1 Collapsing water column (2D) . . . 76
2.6.2 Collapsing water column with obstruction (2D) . . . 87
2.6.3 Collapsing water column with obstruction (3D) . . . 94
2.7 Conclusions . . . 103

3 Mesh Movement 105
3.1 Grids and Adaptation . . . 106
3.1.1 Grid optimisation techniques . . . 108
3.2 Lagrangian and Eulerian descriptions of motion . . . 110
3.2.1 ALE formulation . . . 114
3.2.2 Implementation of grid velocity into the solution algorithm. . . . 116
3.3 Calculating grid velocity . . . 117
3.3.1 The Lagrangian functional . . . 118
3.3.2 The Grid Quality functional . . . 118
3.3.3 The Interface Proximity functional . . . 119
3.3.4 Minimisation process . . . 120
3.4 Test problems . . . 123
3.4.1 GCL test . . . 123
3.4.2 Static interface capture . . . 125
3.4.3 Sloshing tank . . . 126
3.5 Conclusions . . . 133

4 Fluid structure interaction 135
4.1 Background . . . 136
4.2 Multiphase approach and coupled solvers . . . 138
4.2.1 Mass conservation law . . . 141
4.2.2 Momentum conservation law . . . 141
4.3 Geometry mapping and friction coefficients . . . 142
4.3.1 Sub-grid scale particles . . . 143
4.3.2 Resolved particles . . . 145
4.3.2.1 Spheres and cylinders . . . 146
4.3.2.2 External meshes . . . 147
4.4 Interacting forces and solution algorithm . . . 148
4.5 Test cases . . . 152
4.5.1 Flow past a cylinder . . . 152
4.5.2 Flow past a 3D cylinder . . . 153

4.5.3 Flow past a 3D square section tube 160
4.5.4 Flow past a sphere 164
4.5.5 Sphere drop in a viscous fluid 166
4.5.6 Two spheres in viscous fluid 171
4.5.7 2D FEMDEM one way coupling 174
4.6 Conclusions 180

5 Applications to coastal defence design 181
5.1 Design techniques 182
5.1.1 Wave attack parameters 184
5.2 Numerical modelling 186
5.3 Results 189
5.3.1 Two dimensional wave breaking at different slope angles 189
5.3.2 Three dimensional wave impact on breakwater amour units 195
5.4 Conclusions 197

6 Conclusion 209
6.1 Concluding remarks 209
6.2 Future work 210

References 213

A Numerical methods used in FSI tests 233

List of Figures

1.1 Examples of coastal defence structures 32
1.2 Types of concrete armour blocks. (Tsinker, 1997) 33
1.3 Concrete armour block types (new generation). 34

2.1 VOF representations of a fluid interface (a), by (b) the volume fraction values, (c) the SLIC method, and (d) the PLIC method. 46
2.2 1D control volume centred at node i. 49
2.3 Results for first order upwinding when $t = 45\Delta x$, at CFL number of 0.05 (Top) and 0.5 (Bottom) 51
2.4 Results for the Lax-Wendroff method when $t = 45\Delta x$, at CFL number of 0.05 (Top) and 0.5 (Bottom). 52
2.5 Results for the Lax-Wendroff method limited by the ULTIMATE strategy when $t = 45\Delta x$, at CFL number of 0.05 (Top) and 0.5 (Bottom). 53
2.6 Two dimensional breaking dam problem. Lower density fluid is shown in blue, while the higher density fluid is shown in red. 54
2.7 Results for first order downwinding when $t = \Delta x$, at CFL number of 0.05 (Top) and 0.5 (Bottom) 55
2.8 Definition of upwind, downwind, and central nodes, depending on the face and flow direction. 56
2.9 NVD diagrams for (a) first order upwinding and downwinding, and (b) Lax Wendroff second order centred methods 56
2.10 NVD Diagram for Hyper-C. 57
2.11 Results for limited first order downwinding (Hyper-C) when $t = \Delta x$, at CFL number of 0.05 (Top) and 0.5 (Bottom) 57
2.12 Partial volume (shaded region) corresponding to local node i used to calculate volume coordinates at point p. 63
2.13 Tetrahedron element being subdivided via the volumetric centre, face centroids, and edge midpoints. 66
2.14 Control volume (i.e. shaded region) generated by contributions from all elements which contain a given node. 66

2.15 Control volume configuration and naming conventions, where the scenario for nodes (a) in the interior of the domain, and (b) in the boundary of the domain are shown. A two dimensional version is show for clarity purposes, however it is straightforwardly extensible to three dimensions. 68
2.16 Initial Conditions for Zalesak's problem given by (a) a schematic, and (b) the simulation output. 71
2.17 Sequence of results for Zalesak's problem at (a) $t = 0.798$ seconds, (b) $t = 1.60$ seconds, (c) $t = 2.40$ seconds, and (d) $t = 3.15$ seconds. 71
2.18 Comparison of results for Zalesak's problem after one full rotation using (a) SLIC, (b) Hirt-Nichols, (c) FCT-VOF, (d) Youngs, (e) LVIRA, and (f) this thesis's Hyper-C methods. LVIRA's results were obtained from Pilliod and Pucket (2004), all others except Hyper-C were obtained from Rudman (1997) 72
2.19 Conservation graph for the Zalesak problem using Hyper-C. 73
2.20 Initial conditions for the shearing flow problem given by (a) a schematic, and (b) simulation output. 74
2.21 Result comparison between different methods (i.e. obtained from Rudman (1997)) including Hyper-C, for the shearing flow problem. Rows (a) and (b) show the advection process at reversal time and at the end of the simulation, respectively, for $T = 8.86$. Rows (c) and (d) analogously show the results for $T = 15.72$. 75
2.22 Conservation graph for the shearing problem using Hyper-C 76
2.23 Initial configuration for collapsing water column problem. Measurements are in meters [m]. 77
2.24 Comparison of results for the collapsing water column problem using (a) Hyper-C with (b) experiments by Koshizuka et al. (1995b), (c) a moving particle semi-implicit method (MPS) from Koshizuka et al. (1995a), and (d) a VOF combined with adaptive grids by Greaves (2006). Results correspond to timeframe $\tau = 0$. 79
2.25 Continued from Figure 2.24, with frames for $\tau = 1.617$. 80
2.26 Continued from Figure 2.24, with frames for $\tau = 3.233$. 81
2.27 Continued from Figure 2.24, with frames for $\tau = 4.850$. 82
2.28 Continued from Figure 2.25, with frames for $\tau = 6.466$. 83
2.29 Continued from Figure 2.25, with frames for $\tau = 8.083$. 84
2.30 Graph of advancing water front vs. normalised time. 85
2.31 Graph of water height vs. normalised time. 85
2.32 Mass conservation graph for 2D water collapse problem, showing results from adaptive and non-adaptive tests. 86
2.33 Mass conservation graph for 2D water collapse with obstruction problem, showing results from adaptive and non-adaptive tests. 88

2.34 Comparison of results for the obstructed collapsing water column using Hyper-C (a), experiments by Koshizuka et al. (1995b) (b), the VOF approach from Greaves (2006)(c) . Time frames are at a normalised value of $\tau = 0.809$. 89
2.35 Continued from Figure 2.34, with frames for $\tau = 1.607$. 90
2.36 Continued from Figure 2.34, with frames for $\tau = 2.426$. 91
2.37 Continued from Figure 2.34, with frames for $\tau = 3.233$. 92
2.38 Continued from Figure 2.34, with frames for $\tau = 4.043$. 93
2.39 Experimental setup for the 3D collapsing water column with an obstruction as used by Kleefsman et al. (2005), showing a (a) side, (b) isometric, and (c) top view. 94
2.40 Pressure sensor locations at the obstacle Kleefsman et al. (2005) 95
2.41 Water height sensor locations at the obstacle Kleefsman et al. (2005) 96
2.42 Result comparison of the 3D collapsing water column with an obstruction, comparing (a) Hyper-C to (b) experiments, and (c) the VOF method by Kleefsman et al. (2005). Time frames are, by column, for $t = 0.2$ and $t = 0.4$. 97
2.43 Continued from Figure 2.42, for time frames $t = 0.6$ and $t = 0.8$. 98
2.44 Continued from Figure 2.42, for time frames $t = 1.0$ and $t = 1.2$. 99
2.45 Continued from Figure 2.42, for time frames $t = 1.4$ and $t = 1.6$. 100
2.46 Continued from Figure 2.42, for time frames $t = 1.8$ and $t = 2.0$. 101
2.47 Pressure sensor comparison between results from this thesis's Hyper-C and experiments at MARIN facilities. Spikes that can be observed in the darker curve are attributed to solution remapping in adaptive steps. 102
2.48 Height sensor comparison between results from this thesis's Hyper-C and experiments at MARIN facilities. Spikes that can be observed in the darker curve are attributed to solution remapping in adaptive steps. 103

3.1 Comparison between meshed and meshless geometric discretisations. Meshless domain discretisation can be viewed as spheres with a given radius that depends on the method. Meshed domain pictured here is a linear tetrahedral discretisation of the cube, of which the surface mesh can be seen. 106
3.2 Examples of 2D mesh suitability for the represented results. 108
3.3 Frame sequence (a) to (c) describing a node in the mesh moving until it goes through an edge and turns the element "inside out". 109
3.4 Subdivision of a tetrahedron (a), into (b) two, (c) four, and (d) eight sub-tetrahedra. 109
3.5 A two dimensional schematic of the minimisation process for node *i* in the computational mesh. Blue arrow shows the calculated functional gradient direction, while the red arrow depicts the line search direction. A graph is also shown indicating the choice of the new location for the node (circled in green). 121
3.6 Graph of initial mesh for the GCL test. 124

3.7 Graph of final mesh for the GCL test. 124

3.8 Scalar field plot of density (a) and vector field plot of the velocity norm (b) for the initial mesh and final mesh. 125

3.9 Initial mesh and concentration scalar field. 126

3.10 Solution after several timesteps for (a) a mesh adaptive case, and (b) an ALE case. 126

3.11 Cross section graph of the concentration scalar field across a line given by equation $y = 0.5$. 127

3.12 Initial concentration field for the sloshing tank problem 128

3.13 Sloshing sequence of results showing frames at (a) $t = 1$, (b) $t = 2$, (c) $t = 3$, and $t = 4$ seconds. 129

3.14 Mass conservation graph obtained for the sloshing simulation with $Re = 250$. . 130

3.15 Low amplitude sloshing sequence of results showing frames at (a) $t = 0.0$, (b) $t = 0.2$, (c) $t = 0.4$, (d) $t = 0.6$, (e) $t = 0.8$, and (f) $t = 1.0$ seconds. 131

3.16 Graph of accumulated real time versus simulated time, for the sloshing simulation with $Re = 3.5 \times 10^6$. 132

3.17 Graph of computational real time spent per timestep, for the sloshing simulation with $Re = 3.5 \times 10^6$. 132

4.1 Dynamically deposited pack of DEM particles. Each particle is represented as a cluster of simpler shapes, such as spheres or other more sophisticated shapes (i.e. parametric superquadrics). Frames are courtesy of Xavier Garcia. 139

4.2 Frames depicting a two dimensional FEMDEM simulation of an avalanche process. (courtesy of Dr. Jiansheng Xiang using y2d code originally developed by Professor Antonio Munjiza) 140

4.3 Three dimensional FEMDEM simulation of random interaction of different particle shapes. (courtesy of Dr. Jiansheng Xiang) 140

4.4 Schematic of a sub-grid scale particle case 144

4.5 Particle shape (i.e. thick black line) being resolved by the computational mesh. Blue region corresponds to $\alpha = 1$, while lighter coloured region has intermediate has $0 < \alpha < 1$. Outside of these two regions, $\alpha = 0$. 145

4.6 Particle shape being resolved by the computational mesh. In (a) sphere shape is resolved by a coarse mesh, while in (b) sphere shape is resolved by an optimised mesh. 147

4.7 Rotating cylinder subjected to fluid flow. 148

4.8 External mesh mapping example. Sub- figure (a) shows the original shape of the particle while (b) shows the corresponding particle mesh, (c) and (d) provide different views on the 0.5 value isosurface produced by the solid volume fraction scalar field, and finally (d) and (e) provide a volumetric cutaway of the unstructured grid, showing the scalar solid concentration values along with the optimised resolution of the mesh. 151
4.9 Schematic of the the two dimensional cylinder in flow test case. 153
4.10 Streamlines obtained from results of flow past a cylinder. 153
4.11 Streamlines obtained from results of flow past a cylinder (close view). 154
4.12 Drag coefficient evolution for the two dimensional cylinder test case. 156
4.13 Lift coefficient evolution for the two dimensional cylinder test case. 156
4.14 Schematic of the the three dimensional cylinder in flow test case. 157
4.15 Streamlines obtained from results of flow past a cylinder. Middle cut planes are presented parallel to the 3D plot of a group of streamlines. Left slice gives detail of the mesh, while the right slice shows a scalar field plot of solid concentration. 157
4.16 Streamlines past the cylinder in close view. Grey isosurface is shown corresponding to a solid volume fraction value of 0.5. 158
4.17 Drag coefficient evolution for the three dimensional cylinder test case. 159
4.18 Lift coefficient evolution for the three dimensional cylinder test case. 159
4.19 Schematic of the the three dimensional square section tube in flow test case. . . 160
4.20 Streamlines obtained from results of flow past a square section tube. 161
4.21 Streamlines obtained from results of flow past a square section tube. (close view) 161
4.22 Velocity isocontours on vertical and horizontal cut planes. 162
4.23 Close up view of velocity field vector plot (vectors not in magnitude scale) showing the detail captured in the wake behind the square section. 162
4.24 Drag coefficient evolution for the three dimensional square section test case. . . 163
4.25 Lift coefficient evolution for the three dimensional square section test case. . . 164
4.26 Domain schematic of the sphere drag test case. The sphere is placed closer to the inflow surface boundary. A large domain was used to suppress any possibility of wall effects. 165
4.27 Result plots for flow past a rigid sphere, showing (a) the solid isosurface, and (b) the corresponding flow streamlines. 165
4.28 Sphere drag coefficient vs. Reynolds number plot of numerical results obtained through the coupling method presented in this thesis. The experimental curve used for comparison was obtained from Brown and Lawler (2003). 166
4.29 Schematic of the domain used in the sphere drop simulation. 167
4.30 Graph of dropped sphere velocity vs. time elapsed. It can be observed that the sphere reaches terminal velocity. 168

4.31 Sequence of frames for the sphere drop simulation. The time frames included are for t =0.5, 1.0, 1.5, 2.0, 2.5, 3.0, 3.5, and 4.0 . Data shown are velocity magnitude contours for a middle cut-plane across the sphere. 169

4.32 (Continued from Figure 4.31) Sequence of frames for the sphere drop simulation. The time frames included are for t =4.5, 5.0, 5.5, and 6.0. 170

4.33 Sequence of frames for the two spheres in tandem drop simulation. The time frames included are for t =0.5, 1.0, 1.5, 2.0, 2.5, 3.0, 3.5, 4.0, 4.5, 5.0, 5.5, and 6.0. Data shown are velocity magnitude contours for a middle cut-plane across the spheres. 172

4.34 Sequence of frames showing the impact of two spheres initially set off against each other in an otherwise quiescent fluid. The time frames included are for t =0.02, 0.04, 0.06, 0.08, 0.1, 0.12, 0.14, and 0.16 seconds. Data shown are velocity magnitude contours for a middle cut-plane across the spheres. 173

4.35 Sequence of frames showing the two dimensional triangle impacting a horizontal base and splitting into its smallest components. Time frames, in order from top to bottom, are for t =5, 55, 105, and 155 milliseconds 175

4.36 (Continued from Figure 4.35) Sequence of frames showing the two dimensional triangle impacting a horizontal base and splitting into its smallest components. Time frames, in order from top to bottom, are for t =205, 255, 305, and 355 milliseconds. 176

4.37 Sequence of frames showing the two dimensional triangle impacting a horizontal base and splitting into its smallest components. Time frames, in order from top to bottom, are for t =5, 15, 25, and 35 milliseconds 177

4.38 (Continued from Figure 4.37) Sequence of frames showing the two dimensional triangle impacting a horizontal base and splitting into its smallest components. Time frames, in order from top to bottom, are for t =45, 55, 65, and 75 milliseconds. 178

4.39 (Continued from Figure 4.38) Sequence of frames showing the two dimensional triangle impacting a horizontal base and splitting into its smallest components. Time frames, in order from top to bottom, are for t =85, 95, 105, and 110 milliseconds. 179

5.1 Typical breakwater cross section (diagram obtained from Palmer and Christian (1998, page 21)) 183

5.2 Setup of a multi- layer breakwater for a scale model test in a wave basin; colouring is designed for ease of interpretation of damage patterns. (picture obtained from **?**) 183

5.3 Diagram of breaker types (Battjes, 1974). 185

5.4 Sketch of the setup for the 2D simulations of wave run-up Height of the domain is 0.5 m while length is 2.5 m. Initial still water depth was set to 0.2 m, while the slope pictured here is of 0.7. 189
5.5 Velocity vectors within a plunging breaker. Slope angle used here is 1:5. 191
5.6 (Continued from Figure 5.5) 192
5.7 Sequence of frames corresponding to wave run up simulations at two different slope angles. For the gentle slope case of 1:5 (i.e. left column), (a) $t = 1.3\,s$, (b) $t = 1.45\,s$, (c) $t = 1.65\,s$, (d) $t = 1.80\,s$, (e) $t = 1.95\,s$, and (f) $t = 2.3\,s$. For the steep slope case of (1:1.4), (a) $t = 3.05\,s$, (b) $t = 3.30\,s$, (c) $t = 3.55\,s$, (d) $t = 3.80\,s$, (e) $t = 4.05\,s$, and (f) $t = 4.30\,s$. 193
5.8 3D version of the gravity generated solitary wave breaking on a slope. Dimensions of the tank in this case were : height=0.2 m, width 0.5 m, length=0.75 m. The slope used here was of 0.3. Still frames were taken at (a) $t = 0.375$, (b) $t = 0.435$, (c) $t = 0.459$, (d) $t = 0.555$, (e) $t = 0.615$, and (f) $t = 0.663$ seconds. 194
5.9 Frames depicting free surface evolution and one way interaction with the solid armour units. Still frames were taken at (a) $t = 0.1$, (b) $t = 0.2$, (c) $t = 0.3$, (d) $t = 0.4$, (e) $t = 0.5$, (f) $t = 0.6$, (g) $t = 0.7$, (h) $t = 0.8$, (i) $t = 0.9$, (j) $t = 1.0$, (k) $t = 1.1$, and (l) $t = 1.2$. 198
5.10 Result frames depicting a group of scalar cut planes across different sections of the fluid domain. The three left hand side ones correspond to horizontal cuts, while the four right hand side ones are vertical cuts. Time frames depicted here are $t = 0.1$ (top), and $t = 0.2$ (bottom). Plots are of velocity magnitude. 199
5.11 Continued from Figure 5.10. Time frames depicted here are $t = 0.3$ (top), and $t = 0.4$ (bottom). 200
5.12 Continued from Figure 5.11. Time frames depicted here are $t = 0.5$ (top), and $t = 0.6$ (bottom). 201
5.13 Continued from Figure 5.12. Time frames depicted here are $t = 0.7$ (top), and $t = 0.8$ (bottom). 202
5.14 Result frames depicting a group of scalar cut planes across different sections of the fluid domain, analogous to Figure 5.10. The three left hand side ones correspond to horizontal cuts, while the four right hand side ones are vertical cuts. Time frames depicted here are $t = 0.1$ (top), and $t = 0.2$ (bottom). Plots are of vorticity magnitude. 203
5.15 Continued from Figure 5.14. Time frames depicted here are $t = 0.3$ (top), and $t = 0.4$ (bottom). 204
5.16 Continued from Figure 5.15. Time frames depicted here are $t = 0.5$ (top), and $t = 0.6$ (bottom). 205
5.17 Continued from Figure 5.16. Time frames depicted here are $t = 0.7$ (top), and $t = 0.8$ (bottom). 206

5.18 Graph of the total horizontal force acting on both units due to the wave impact. 207
5.19 Graph of the total vertical force acting on both units due to the wave impact. . . 207

List of Tables

2.1 Error for Zalesak's test after one full rotation. Data obtained from Rudman (1997) and López et al. (2004). 73
2.2 Error calculation for shearing flow test case. 74
2.3 Calculation parameters for the water collapse simulation 77
2.4 Parameters for 3D water collapse problem 95

3.1 Table of parameters for the sloshing test case. 129
3.2 Table of parameters for the sloshing test case. 130

4.1 Comparison table for the 2D cylinder test case using different methods (See Appendix A for details of each the methods). 155
4.2 Comparison table for the 3D cylinder test case using different methods (See Appendix A for details of each the methods). 158
4.3 Comparison table for the 3D square section test case using different methods (See Appendix A for details of each the methods). 163
4.4 Drag coefficient values at different Reynolds numbers for the flow submerged sphere test case. 166

A.1 Numerical methods used for comparison in 2D and 3D steady flow tests. 234
A.2 (Continued from Table A.1) Numerical methods used for comparison in 2D and 3D steady flow tests. 235

List of Algorithms

1 Solution algorithm for incompressible Navier Stokes equations. 70
2 Solution algorithm for incompressible Navier Stokes equations, modified to include ALE reference frame. 116
3 Procedure for calculating grid velocity. 122
4 Sphere and cylinder shape mapping to the computational fluid domain. 147
5 Generic shape field value mapping algorithm with fall back on brute force technique 149
6 Construction of sorting bins for elements in the originating mesh. 150
7 Brute force search algorithm used as a fall back in FNN search. 150
8 Calculation of particle force and friction coefficient terms. 150
9 Solution algorithm for incompressible Navier Stokes equations, taking into account the presence of solids. 152

Chapter 1

Introduction

The research developments reported in this thesis are in the field of numerical modelling for fluid-structure interaction (FSI) problems. The chosen field of application of such numerical methods, namely coastal engineering, is extremely active today as there is an urgency among policy makers for scientists and engineers to provide solutions for adapting to climate change. In the first part of this introductory chapter, a broader context is introduced and explored together with enormous potential and motivation for better modelling of coastal processes. The thesis is then narrowed to consider wave-structure interaction and modelling requirements. The research project is part of a broader objective for modelling capability. It attempts to bridge many solid-based and fluid-based numerical methods already operational or under simultaneous development within the host research group, the Applied Modelling and Computational Group (AMCG). This context is also introduced before the distinct contributions of this research project are outlined.

1.1 Background

Coastal engineering involves the practise of civil engineering, as well as the sciences of oceanography and coastal geology, to control erosion; place, construct, and monitor coastal structures; nourish beaches; and develop and maintain ports, harbours, and related navigation facilities. Within this context, predicting the effects of wave energy dissipating on an armoured coast is a main priority. Physical models have been developed and combined with theory to produce a number of empirical formulae which provide a limited amount of design parametrisation for armoured coastlines. Good sources of details for these parameters can be found in Sawaragi (1995); Dean and Dalrymple (1991). However, influences such as scale, wave obliquity, multi-directionality, wave irregularity, and many other factors on the damage mechanisms of coastal protection is a long way from being fully studied and understood. With the potential to by-pass scale effects, model fluid behaviour realistically, and its relatively low cost with respect to physical (i.e. experimental) modelling, computational simulation is becoming an important design

tool for coastal defence. This thesis relates the development of such a numerical modelling technique that allows the representation of interaction between multi- scale particles and fluids, to be used for coastal engineering applications.

1.1.1 Climate change, sea level rise, and coastal protection

Climate change is at our doorstep, and it is one of the most neglected threats humanity has ever faced. The most alarming fact is that it not only represents an environmental change, but also a change in global economy and sociology. Many nations appear to be committed to mitigation and adaptation strategies, and many are taking daring if not ambitious measures. The warming up of the globe is evident from measurements of rise in averaged air and ocean temperature. However a more tangible way of realising it, is the yearly rate of melting of snow and ice, specially in polar zones. This, together with thermal expansion of seawater (e.g. see Wigley and Raper (1987)) and other factors results in a yearly increase of the mean sea level.

The Earth's average surface temperature has risen during the twentieth century. In a recent report by Bates et al. (2008), the Intergovernmental Panel on Climate Change (IPCC) predicts by extrapolation, that the global average surface temperature will rise by a further 1.8-4.0°C this century, and by up to 6.4°C in the worst case scenario. If even the lowest of the these values becomes true, a change of this magnitude in global average temperature could have irreversible results which, in turn, are likely to trigger serious consequences for mankind and other life forms. One of these consequences is the increase of sea levels. From the report by Bates, the observed average rate of increase was 1.7 ± 0.5 mm/yr for the 20th century, 1.8 ± 0.5 mm/yr for 1961–2003, and 3.1 ± 0.7 mm/yr for 1993–2003. Projecting these changes over the next twenty to fifty years yields that coastal areas and small islands are in danger.

Storm frequency and magnitude is also a major consequence of climate change. Seen as the major contributor to changes in coastal morphology (Heinze et al., 2001), these weather events have the ability to re-shape coastal features in a number of hours, while under normal conditions it could take years. Research has shown that storms dominate barrier wash overs, inlet breaches, and sedimentation in estuaries, bays, and lagoons (Hayes, 1978). Very strong winds are associated with the passage of storms, and the term 'storminess' has been used to encompass both the frequency and the intensity of storms; see, for instance, Carnell et al. (1996). A good review on storminess measurements can be found by Clarke and Rendell (2008).

Extreme weather events and sea level, as they increase due to climate change, have a strong impact potential on the habitability of the coastline (Tribbia and Moser, 2008; Baquerizo and Losada, 2008). Given that human population tends to gather around coastal zones, both re-curring or singular events can have a catastrophic consequences on daily life (Srinivas and Nakagawa, 2008). It is stipulated that about 50% of the world's population (that is ~3.1 billion people) resides within 200 kilometres of the sea (Masalu, 2008). These zones are, in general, very economically rich areas where goods are produced and industries reside. It is estimated

that in the European Union, most of the economic wealth comes from within areas 50 kilometres of the sea. Industries such as fishing, shipping, and tourism share approximately 90,000 kilometres of coastline, which is also home to some of Europe's most frail flora and fauna. For all the above reasons, it is imperative that these regions are properly protected once the policy makers have decided to "hold the line".

Coastal protection structures, whose objective is to decelerate erosion and repel the effects of flooding, have a long history, specially in regions of the globe where the elevation is very close or even below the average sea level, such as Venice, New Orleans, the Nagara river in Japan, The Netherlands, and the Caspian Sea. Up until the mid-twentieth century, most of the shore protection consisted of static structures built to stand in between sea and land. Nowadays, strategies for defence have become more sophisticated having assimilated the fact that simply building structures in an ad-hoc manner will not guarantee expected results, and may even have a strong impact on the surrounding ecosystems; see Airoldi et al. (2005). Therefore, coastal defence measures have now become more dynamically and naturally adjusting when possible, and environmentally friendly approaches are adopted.

Design approaches used to prevent the erosion of the coast line can be divided into two main categories: hard structures and soft structures. Hard structures have long been the instrument of the coastal engineer in his/her attempt to control coastline erosion. They include groynes, jetties, breakwaters, and sea walls. Groynes are structures built perpendicular to the shoreline and are usually made up of wood, concrete, or large rocks. Their main purpose is to slow the movement of sediment along the coastline. Jetties are placed at entrances or harbour inlets to prevent build-up of sediment. The main purpose of off-shore breakwaters is to reduce the wave action in a certain area of the coastline, such as harbours. Normal breakwaters, seawalls, or revetments are constructed to prevent the erosion or flooding of the land behind. Examples of these structures are shown in Figure 1.1.

Soft structures are those that arise essentially from the management of natural sediments. Sand and shingle are used, as well as vegetation, rubble, and small structures all to reduce the effect of erosion by providing an absorbing buffer for wave action or a well maintained volume of beach. An example of this management is beach nourishment, which consists of the artificial placement of sand on a beach; see, for example Mohan et al. (2003) for more details. It is a relatively short-term measure as it does not fix the cause of the erosion; however, it is an alternative when there are economic, political, or environmental issues to consider. Other examples include sand scrapping, relocation of natural inlets, groundwater control, regional sediment management, and sand berms.

The coastal zone is a complex and highly sensitive environment that is in a constant state of change. Defending against sea level and extreme weather during this century will be crucial, given their alarming rate of increase, and it will most definitely pose a challenge to coastal engineers around the world. Protective structures such as sea walls and breakwaters are expensive, take time to build, and must provide safe and reliable protection. Waves, currents, and tidal



(a) Groyne (b) Jetty

(c) Breakwater (d) Seawall

Figure 1.1: Examples of coastal defence structures

conditions can vary widely over short distances, so careful parameters have to be set for the design of these structures, which can have a profound impact on the physical balance of the coastline.

1.1.2 The wave hydraulic-load problem

Waves are generated essentially by the wind, and propagate from the oceans to the shorelines across the continental shelves. A typical wave undergoes phenomena such as refraction, diffraction, shoaling, and breaking, as it moves towards and along the surf zone, to finally run up on the beach slope. The design of coastal defence structures is currently dependant on empirical formulae obtained from scale experiments, and linear wave theory (Sawaragi, 1995; Dean and Dalrymple, 1991). However, wind-generated waves are random, and dissipation mechanisms due to wave breaking, turbulence, and generation of eddies in the fluid region as well as turbulence and friction within the porous armour material is still not well established (Losada et al., 2005).

Wave climate[1], combined with currents, tides and storm surges, are the main driving factors of coastal erosion problems. A "rubble mound" is the term often used to describe breakwater and revetment slopes that are armoured with loose and typically randomly oriented blocks of rock known as "armourstone" which have to be sufficiently massive and durable to resist wave action. For breakwaters, there are two sides sloping into the sea, and for revetments, which

[1]Wave climate is defined as the probability of occurrence of a typical wave parameters, such as wave height, period, and direction.

are like a sloping rubble sea wall, there is only one side. Advancements in their construction have led to complex structures consisting not only of quarry rock, but also of engineered and manufactured concrete blocks. Apart from their engineered shape, these large blocks are very common and viable option for economical and logistical reasons. Most of these units are designed to rely on friction, weight, and interlock in order to achieve stability. Examples of typical units that can be seen in coasts around the world are shown in Figure 1.2, although further units have been invented since; some of them are shown in Figure 1.3. The weight and shape of these armour units was calculated to give adequate protection using a design wave condition from empirical derivations obtained from scaled physical model tests; see, for example Melby and Turk (1997). Concrete armour blocks tend to provide better stability results than quarry rock armour. The armour layer is basically a granular pack with high permeability which allows for a better absorption of the wave energy. High interlocking ability is often linked to lower structural strength, but advanced armour units with high interlocking ability and high strength, like the Accropode^{TM}, Core-loc^{TM}, X-bloc^{TM}, Ecopode^{TM} designs, are preferred in high wave-energy environments.

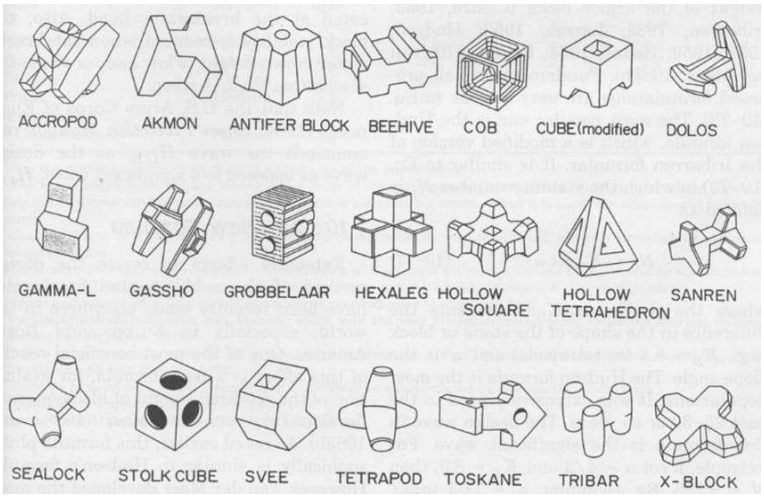


Figure 1.2: Types of concrete armour blocks. (Tsinker, 1997)

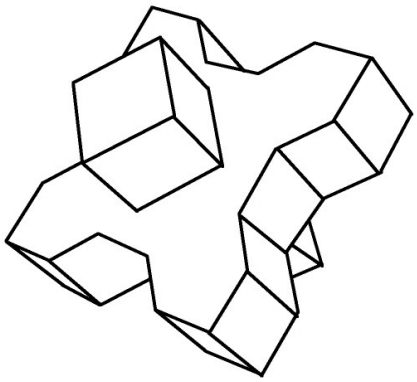
Many empirical formulae have been proposed to optimise armoured coastlines. Losada and Gimenez-Curto (1981) proposed an empirical reflection coefficient of a permeable structure with a uniform slope in regular waves, and another for the amount of distance that the wave runs up and down the slope. These are important design requirements for the hydraulic efficiency of the structure. Regarding stability of armour units or rock, the best known empirical equations are those by Hudson and Jackson (1953) and by Van der Meer (1988). Hudson's formula is given by,

$$W = \frac{1}{K_D cot(\theta)} \frac{H_I^3 \rho_r}{\left(\frac{\rho_s}{\rho_w - 1}\right)^3} \tag{1.1}$$

where the mass of the mound W is determined by using factors such as incident wave height (H_I), density of the water (ρ), density of the stone or rock (ρ_r), seaward slope angle of the



(a) Core-loc (TM) unit



(b) X-bloc(TM) unit



(c) Accropode II(TM) unit

Figure 1.3: Concrete armour block types (new generation).

rubble mound (θ), and a stability coefficient (K_D). Van der Meer (1988) proposed a more comprehensive range of design parameters with which to determine the unit mass of armourstone for two different types of wave conditions, breaking and plunging. These formulae include parameters to account for expected permeability inside the mound, random waves, and hydraulic scale effects. For a more detailed guide to rock armour, concrete armour units, and up to date stability parameters (**?**).

Unfortunately, scaled physical tests are unable to consistently predict aspects such as breakage mechanisms involving material strength, hydraulic forces, and the underlying rubble mound permeability. An extensive list of detailed examples of design failures can be obtained from Maddrell (2005). It is somewhat evident then, given that Hudson's formula dates back half a century, that fundamental understanding of the forces generated during wave-structure interaction remains poor and continues to challenge designers. Additionally, physical model tests require very large tank facilities that must approach prototype sizes if scale effects are to be avoided. These facilities are expensive while smaller scale models can be unreliable due to force relations not scaling correctly.

As numerical modelling does not suffer from these problems, and even though it does suffer from numerically inherited effects such as discretisation error and computational cost, simulations with multi- physics and multi-scale capabilities will play an important complementary

role to physical laboratory tests and on-site measurements. This will only aid the search for understanding important design parameters, such as the effects of dissipation through turbulence generation, armour unit interlock, mass and strength, and underlying rubble-mound permeability.

1.2 VGW - Virtual Geoscience Workbench

The Virtual Geo-science Workbench for discontinuous systems (see Munjiza and Latham, 2004 for more details) is a 5-year project funded by EPSRC and is under development, collaboratively, at Imperial College London and Queen Mary, University of London. Principal investigators are Dr. John Paul Latham[2], from Imperial College, and Professor Antonio Munjiza[3], from Queen Mary.

The project consists on creating a software environment to simulate multi- physical phenomena for discontinuous or particulate systems for geoscience and geoengineering. Particles (i.e. in coastal engineering these range from sand grains up to massive concrete armour blocks) may be modelled in one of two ways, depending on user resources such as time, computer power, and the type of problem modelled.

The two options for particles are the discrete element method (DEM), and the combined finite element/discrete element method (FEMDEM). DEM involves modelling discontinua as rigid particles with rheological contact mechanics laws. The shapes are typically spheres, although recent development accounts for non-spherical particles represented by superquadric functions (Preece et al., 1999)), clustered spheres (Hubbard, 1996)), or polyhedral particles (Nezami et al., 2006). When large number of particles are modelled, DEM simulations require ingenious algorithms for detecting and resolving contacts to produce results within a reasonable time (e.g. see, for instance Hubbard (1995), Munjiza (2004), Song et al. (2006), or Nezami et al. (2004). The model for interaction between particles is normally chosen depending on the type of system being modelled. In the case of granular matter, for example, it is common to assume that particles interact via contact forces. Several models of contact interaction have been proposed in the literature, such as those by Schäfer et al. (1996) and Poshel T. (1993).

The second, and more sophisticated method, is FEMDEM. This method, pioneered by Professor Munjiza, combines all the aspects of DEM, with an internal discretisation of each particle, allowing the calculation of internal stresses and deformation. Contact detection and force calculation also takes place through complex algorithms, as explained by Munjiza (2004).

For the set of tools needed to bridge these solid methods with computational fluid dynamics (CFD), a considerable effort (i.e. this thesis project) within the VGW research project has been focused towards fluid-structure interaction to expand the range of particulate physics that VGW

[2]e-mail: j.p.latham@imperial.ac.uk
[3]e-mail: a.munjiza@qmul.ac.uk

can be applied to. The CFD code "Fluidity"[4], written by the Applied Modelling and Computational Group within the Earth Science and Engineering department of Imperial College, was developed further, modified, and interfaced with both DEM and FEMDEM codes, allowing important multi- physics applications envisaged by Latham et al. (2007) to be modelled using a combination of VGW tools and Fluidity tools.

1.3 Fluidity

Fluidity is a general purpose multi- phase open source CFD code written to solve the incompressible Navier-Stokes equations (NSE) on arbitrary unstructured finite element meshes. It uses projection type methods, methods that derive a Poisson equation for pressure (PPE), artificial compressible methods (also known as the penalised momentum equation), and other solution methods including Uzawa's method. See Gresho et al. (1998) for details on projection methods, PPE, and PME, and see Robichaud et al. (1990) for details of the Uzawa method. Itemised below are some of the most important reasons for choosing Fluidity as the generic CFD code to progress these objectives for fluid-structure modelling:

- Fluidity solves the incompressible Navier-Stokes equations, including an arbitrary number of extra tracer fields. These fields can be used to describe physical phenomena in areas of research such as heat transfer, sediment transport, radiation, ocean modelling, and environmental pollution.
- The usage of Fluidity's mesh adaptive capabilities allows for a refined and computationally optimised discretised domain (Pain et al., 2001a). This capability is a key factor when the application requires the description of a sharp interface. Without this optimisation, computational cost would be high, and detail level would be limited.
- Error estimation and rate of convergence[5] is well known in Finite Element Methods (FEM), especially when dealing with boundaries (Zienkiewicz and Taylor, 1991).

A wide range of fluid dynamics problems have been solved using Fluidity. Applications include ocean modelling (Piggott et al., 2008), heat transfer (Pain et al., 2001b), fluidised beds, and radiation transport (Pain et al., 2003) to name a few. The code includes powerful parallel multiblock explicit domain decomposition solution methods (see Gorman et al. (2007)) which enable simulations to be run across a number of work stations over a standard network, or a dedicated parallel computer such as the HPC cluster[6], using either Message Passing Interface (MPI) or

[4]See webpage: http://amcg.ese.ic.ac.uk/ (accessed September 30th, 2008)

[5]The speed at which a convergent sequence approaches its limit is called the rate of convergence. Although this limit does not give information about any finite first part of the sequence, it is of practical importance when dealing with a sequence of successive approximations for an iterative method. Typically, fewer iterations are needed to yield a useful approximation if the rate of convergence is higher.

[6]See webpage: http://www3.imperial.ac.uk/ict/services/teachingandresearchservices/highperformancecomputing (accessed September 30th, 2008)

Parallel Virtual Machine (PVM) interfaces. It also forms the CFD module of the transient criticality code FETCH[7] for the modelling of fissile solutions and criticality accidents.

Perhaps one of its most interesting features, Fluidity also possesses the capability to optimise tetrahedral meshes through a number of operations including edge collapsing, edge splitting, face-edge swapping, and node movement. The optimisation method utilises a memory management procedure containing two linked lists of node, edge, and element information, which guarantees an efficient use of computational resources (i.e.: RAM), and results in high quality anisotropic meshes. The algorithm is robust, as it does not fail to produce valid elements. Details of the tetrahedral mesh optimisation method can be found in Pain et al. (2001a).

A variety of stabilisation methods are available such as: least squares methods for first order PDE's, Petrov-Galerkin methods such as Streamline Upwind Petrov Galerkin (SUPG) (see , for example Tezduyar, 1998), as well as non-oscillatory high-resolution methods that achieve globally, formal high order accuracy in space and time (see, for instance Drikakis, 2003; Harten, 1997; Hirsch, 2002). Furthermore, Large Eddy Simulation (LES) models have been incorporated as well to account for turbulence, namely a Smagorinsky sub-grid scale model, and recently a fourth order dissipation model. Stability and accuracy are a main objective, and hence a range of time-stepping methods are implemented oriented towards the use of implicit methods. Stability is of particular importance when using adaptive mesh optimisation (AMO), since element size changes and small ones might be created.

For solution of non-symmetric linear systems, conjugate gradient squared, restarted GMRES, or bi-conjugate gradient stabilised are implemented options, while conjugate gradient is used for symmetric systems. Preconditioner options include incomplete factorisations, m-step, SSOR, parametrised forward-backward Gauss-Seidel, Jacobi diagonal preconditioning, along with many others provided by the Portable, Extensible Toolkit for Scientific computation (PETSc) package[8]. Theory behind these methods is thoroughly explained by Saad (2003).

The fact that Fluidity also solves any number of extra scalar or "tracer fields" equations of advective and/or diffusive characteristics, allows it to create the so called multi- fluid, or multi- material method which is at the heart of the developments of this thesis. The multi-fluid or multi- material method is technically a multi-phase approach, as all fluids are modelled as continuous phases. More information on the multi- fluid approach is given in Chapter 2.

1.4 Contributions of this work

In this thesis, a new set of tools is developed for the VGW project. These tools are aimed towards describing the interaction between structures (i.e. particles) and multiple fluids. To achieve this, a combination of algorithms were devised and coded to allow the interfacing of the CFD code Fluidity with DEM and FEMDEM codes. These algorithmic tools can be divided

[7] See webpage: http://amcg.ese.ic.ac.uk/ (accessed September 30th, 2008)

[8] See webpage: http://acts.nersc.gov/petsc

into three parts, which are described below in sections 1.4.1, 1.4.2, and 1.4.3.

The algorithms and corresponding code generated through the work presented in this thesis have the ultimate goal of solving one of the main concerns of the coastal engineering community: breakwater armour stability under wave attack. At the onset of this research, Fluidity had no means with which to simulate an energetic water free surface required for wave simulations.

1.4.1 Interface tracking

Given the intent to model the dynamics of two immiscible fluids, it is necessary to simulate the evolution of the interface between them in a sharp, conservative, and efficient way. A concentration scalar field is defined on the computational domain, and its evolution is governed by a pure advection equation. Even though pure advection equations involve no diffusion terms, smearing of the interface is still observed due to numerical diffusion. The methods developed within this thesis, fundamentally described in Chapter 2, are intended to overcome this obstacle. Leonard's Hyper-C advection scheme (see Leonard, 1991) is implemented into Fluidity through a Control Volume Finite Element (CVFEM) discretisation. The resulting method was found to successfully allow for sharper and more accurate interface definitions at a low expense when compared to other methods such as volume of fluid (VOF), or level set methods (LSM). A detailed review of these methods is provided in Chapter 2.

1.4.2 Mesh movement through an ALE scheme

The mesh movement method developed and implemented herein has the characteristics of Arbitrary Lagrangian Eulerian (ALE) motion. It is aimed to reduce the interpolation error introduced by adaptive mesh optimisation through reduction of the adapt frequency. At each time step, the mesh is moved in such a way as to enhance the representation of the dynamics, minimise numerical dissipation, and prevent tangling. Summarising, the three main purposes of the new mesh movement scheme are:

1. To reduce any further diffusive behaviour that can't be handled by the compressive advection scheme. (i.e sections of a fluid or material that become sub-grid scale in size[9])

2. To allow the mesh resolution to follow the interfaces and therefore reduce the amount of AMO needed. This is also aimed to reduce computational time while maintaining enhanced energetic free surface capturing.

3. To prolong grid quality according to error metric values as described by Pain et al. (2001a), thus protecting optimal resolution until AMO is required.

A detailed review, description, and results, are provided in Chapter 3.

[9]Sub-grid scale size in this case refers to particles or sections of fluid that are smaller than the local size of the elements used for the discretisation. It can be assumed that these particles or sections of fluid "fit" inside any of these elements.

1.4.3 Solid-fluid interaction

The intent is to also model the presence of a solid in the fluid (i.e. multi- fluid) domain. The method developed in this thesis follows a combined disperse and two-fluid multi- phase approach, as presented by Prosperetti and Tryggvason (2007); Gidaspow (1994); Brennen (2005). Fluid and solid phases coexist in the same computational domain. However, instead of solving the fluid equations to determine the velocity field of the solid phase, the equations of motion are solved by DEM or FEMDEM methods. Resulting solid material velocities and configurations from the discrete methods are then mapped to the computational domain for the solution of the fluid phase velocity field. Thus, the hydraulic drag on a particle is computed directly as well as particle-particle contact forces, to obtain the fully coupled motion.

Large particles (i.e. relative to the grid size) are exceptionally well resolved with the help of Fluidity's AMO. However, representing the shape and size of an arbitrary number of smaller particles through a complex fully resolved computational grid would be extremely difficult, as this would require high amounts of resolution and the mesh to evolve with the motion of these particles. It seems a logical cost-effective solution to approximate and parameterise effective forces acting on particles which, in comparison to other dimensions in the problem to be solved (i.e. minimum element size), are much smaller in size. Furthermore, if the larger particles were to somehow break or fragment, the method should be able to transition from the non-parameterised model to the parameterised model in a smooth straightforward manner.

Similar approaches have been used with great success using sub-grid scale particle-particle interaction sub-models based on kinetic theory (e.g. see Gidaspow (1994)), or rheology Astarita et al. (1997), and even particle-wall interaction sub-models (e.g. see Jenkins and Savage (1983)). Glowinski et al. (2000) applied a method similar to the one presented here named the fictitious domain method to model particles larger than grid-scale. More detail on fluid structure interaction models is given in Chapter 4.

The flexibility of this approach is important as it may be used to adopt the parameterisation of particle flow, as opposed to resolving each individual particle, and for modelling both stationary and moving particles independent of size. Adaptive mesh optimisation within the fluid domain allows for greater resolution in areas of interest such as those that define the geometry of the larger particles, as well as those of dynamic importance, such as fluid vortices and boundary layers.

1.5 Thesis outline

The project described by this thesis has at its core the development of a numerical modelling technique that is able to describe the interaction between structures and fluids, and their behaviour under certain physically representative conditions. The main applications of this work are related to coastal engineering, and hence the modelled fluids, unless stated otherwise, are

air and water. The results presented were obtained keeping in mind four fundamental goals:

1. The shape of the free surface between air and water must be resolved, including effects such as breaking or merging of fluids.

2. Calculation of forces between the fluids and structure must be accurate.

3. Given that solid structures may break and become smaller, or different sizes might need to be modelled at the same time, the method must be dynamically flexible and stable.

4. Overall, the method must be as efficient and accurate as possible.

Specifically, the primary problem focus within the broader context of coastal engineering is the class of problems that involve wave interaction with coastal structures such as breakwaters. As a secondary class of problems that are no less important, sediment transport and erosive flow problems related to wave and current attack are always not far from consideration for application of the simulation methods under development. The contents of this thesis are described below.

Chapter 2 begins with an introduction to the discretisation formulation used, including finite element (FEM) and control volume finite element (CVFEM) methods. Three main areas of research related to interface tracking are introduced, which are the Volume Of Fluid method (VOF), Level Set Method (LSM), and the Pseudo Concentration Method (PCM). The multi-material approach is then introduced. Hyper-C, and its implementation into Fluidity is detailed. Analytical validation tests are provided and compared to other methods found in the literature. A series of comparisons to laboratory experiments are also provided as test cases.

Chapter 3 first gives a general description and review on meshes and meshed methods use. An overview of adaptive concepts and methods is also given. Introduction and formulation for a generic ALE scheme is introduced, followed by a review of different solution methods used since the 1960's. A description of the moving mesh method developed in this thesis is presented, and validation cases are given for 2D cases. A final discussion is also presented.

Chapter 4 begins with a general description and background review of solid-fluid coupling. Approaches are shown, and the objective and direction of the work is presented. The multi-phase approach to solid modelling is described, followed by a detailed description of the method used for fluid force calculation. Description of the formulation including different approaches for modelling of sub-grid scale and larger-than-grid scale particles is given, followed by formulation for time evolution of the coupled system. Sample validation tests are provided, including "wind-tunnel" type tests of different static shapes, along with comparisons with experimental work found in the literature. Full coupling is shown between Fluidity and a DEM model in 3D simulations, as well as one way coupling of Fluidity with a 2D FEMDEM model. Discussion and conclusions end the chapter.

Chapter 5 begins with background and literature review showing contrast between physical modelling and numerical modelling methods used for coastal defence design. Different ap-

proaches, both empirical and numerical, are described. Applied tests are shown which involve the full usage of the implementations presented in this thesis.

Chapter 6 is the closing chapter, providing summing up conclusions and plans for future work.

Chapter 2

Interface modelling

Free moving surfaces or interfaces in fluid flows are encountered in a variety of physical phenomena. For example, bubble dynamics is of particular interest for chemical engineering, propulsion of liquid–metal jets constitutes the main part of metal forming processes, and most importantly for this thesis ocean waves are under thorough investigation in marine and coastal engineering. Each one of these few examples is of great importance for industrial applications.

In numerical modelling, interfaces are considered to be surfaces on which discontinuities exist in one or more variables. Clear examples are immiscible fluid interfaces (e.g. air-water), shock waves, or interfaces between solids and fluids. A good review of numerical methods for modelling interfacial flow can be found in papers by Scardovelli and Zaleski (1999) , McKee et al. (2008), and the book by Osher and Fedkiw (2003).

The general setting for modelling ocean waves and their behaviour is the unsteady flow of two immiscible fluids, namely air and water. In this thesis, air is also modelled as a fluid, since its effects cannot be neglected, and both fluids are assumed to be incompressible, viscous, and Newtonian. The flow is assumed isothermal, thus neglecting the viscosity and density variations due to changes of temperature. A thorough discussion on the impact of these assumptions on the numerical model is given in the book by Gresho and Sani (2000, pp. 1-12). Furthermore, assuming that the fluids are homogeneous, the densities and viscosities are constant within each fluid. Thus, the interface is defined by a discontinuous jump in these variables (see, for instance, books by Batchelor (2000), and Osher and Fedkiw (2003, pp. 149-238)).

This Chapter begins by providing background to the problems associated with interface tracking, namely their discrete representation and their evolution in time, and popular methods in the literature used to overcome them. A new interface capturing scheme is then presented, consisting of a semi-implicit combined Finite Element (FEM) and Control Volume Finite Element (CVFEM) discretisation including a highly compressive advection method. The CVFEM advection method is one of the core developments of this thesis, and is based on the Hyper-C scheme originally developed by Leonard (1991). The outcome is a powerful fluid to fluid interface modelling tool designed to obtain sharp, conservative results in a computationally efficient manner. Its versatility and relative ease of implementation in three dimensions are distinct ad-

vantages when compared to other methods. Furthermore, the coupling of this newly developed technique with adaptive tetrahedral mesh optimisation greatly enhances its potential for resolving multi-scale flows. Verification based upon analytical test cases, as well as validation against experiments and results from other numerical methods are given at the end of the Chapter.

2.1 Background

Even though significant progress has been made over the years, numerical modelling of interfacial flow problems remains a challenging task. One of the main challenges is to accurately determine the location and orientation of the interface, as it evolves in time. Many methods have been proposed and are currently under development to solve this conundrum. The ideas can broadly be classified into three categories: Lagrangian, Eulerian, and Lagrangian-Eulerian methods.

In Lagrangian methods, which are very popular in computational solid mechanics, calculation points (i.e. nodes) are fixed or attached to the material being modelled. This leads to the inexistence of convective terms in the governing equations of the model. The code is then conceptually simpler and faster in this aspect, in contrast to Eulerian approaches. Ideally, if a node is placed in a material surface, it will remain on it and hence interface tracking should be trivial. Unfortunately for grid-based[1] methods, the Lagrangian approach is difficult to apply with materials that suffer large deformations, as this incurs severe mesh distortion leading to inaccuracy and timestep size reduction. Meshless[2] Lagrangian methods exist and have recently become very popular, such as the Smoothed Particle Hydrodynamics (SPH) method (Gingold and Monaghan, 1977; Cleary and Prakash, 2004) , Finite Point Method (Fang and Parriaux, 2008; Cheng and Liu, 2002; Oñate et al., 1996), and the Point Interpolation Method (Wang et al., 2001; Liu and Yang, 1999); a detailed list of methods can be obtained from Fries and Matthies (2004). However, difficulties are encountered when establishing Dirichlet boundary conditions, artificial compressibility is introduced due to truncation of kernel functions, and it is yet not clear how particle irregularity affects the accuracy of the solutions (see, for example, Liu and Liu, 2003).

Eulerian methods are the common choice for the fluid modeller. Points that define the numerical grid are fixed in space and the flow of material is calculated through them from an inertial reference frame. The main disadvantage of this is that, in general, the solution of the governing equations will contain numerical dissipation associated with the existence of advective (also known as convective) terms, leading in some cases to unrealistic results. Furthermore, if the mesh is fixed in space, the resolution must be fine enough everywhere in the domain so the details of the flow may be predicted accurately. This, in turn, could lead to a

[1] Grid based methods are those which are based on domain discretisation through a lattice of points or nodes where the variables are evaluated. These nodes are interconnected to form a mesh or grid.

[2] Meshless methods use a domain discretisation in which only nodes are used, and an evolving list of nodal neighbours is used instead of a fixed connectivity.

high computational cost through excessive resolution in areas of the domain where the flow is not very complex.

An interesting set of methods, which sprouted from the conception that a Lagrangian mesh could be used, followed by a re-mapping step, are the Lagrangian- Eulerian methods. The results from the moving Lagrangian mesh are interpolated to another mesh which is moved to an optimised position. These methods are widely used in interface tracking, and a detailed review and analysis is done in Chapter 3.

Leaving movable meshes aside, the three most popular examples of the Eulerian approach are the level set method (LSM), the volume of fluid (VOF) method, and the pseudo-concentration method (PCM), which are described below.

2.1.1 Level Set Method

The level set method has its roots in the seminal paper by Osher and Sethian (1988), in which a Hamilton Jacobi approach is used to solve the time dependent equation of a moving implicit surface. As also described in Osher and Fedkiw (2003), the level set method adds dynamics to implicit surfaces. This implicit surface is typically chosen to be a signed distance function χ given by the minimum distance from a given point in the domain to the material interface. Hence, $\chi > 0$ on one side of the interface, $\chi = 0$ at the interface, and $\chi < 0$ on the other side of the interface. The function χ is then transported using the advection equation,

$$\frac{\partial \chi}{\partial t} + \mathbf{u} \cdot \nabla \chi = 0 \tag{2.1}$$

where $\mathbf{u}$ is the fluid velocity obtained externally, for example from the solution of the fluid momentum equations. Function χ is also used to evaluate the physical properties ϑ of the fluid, such as the viscosity or the density, by calculating,

$$\vartheta = \vartheta_1 \mathscr{F}(\chi) + (1 - \mathscr{F}(\chi))\vartheta_2 \tag{2.2}$$

where ϑ_1 and ϑ_2 denote the properties of fluids 1 and 2 respectively, and $\mathscr{F}(\chi)$ transforms the sign of function χ into a step function that varies from 0 to 1. This method has been implemented successfully by Sussman et al. (1994), who used it to model incompressible two-phase flow. Sukumar et al. (2001), used LSM in combination with the extended finite element method (XFEM) to represent holes and material interfaces with application to elastostatics. Kölke et al. (2003) also applied the LSM in combination with stabilised finite elements to solve problems involving two immiscible fluids interacting with lightweight structures.

Despite some graphically exciting results, one of the main difficulties associated with using LSM is the tendency to suffer from mass loss wherever the flow is not properly resolved. Foster and Fedkiw (2001) devised a method that combined the LSM with a particle method, using the local interface curvature as a diagnostic. By monitoring the curvature, it would identify

regions that were potentially not well resolved and then used the particles to reconstruct the level set function. Smolianski (2005) performed correction and re-initialisation steps by allowing the level set function to be artificially modified with corrections of the order of, at most, the interpolation error. Both of these approaches, although considerably efficient at solving the problem, are still in some way case dependant, and hence non-generalisable.

2.1.2 Volume Of Fluid method

Another method which is widely accepted for its success in interface tracking, is the volume of fluid (VOF) method. In the VOF approach, each individual element or cell contains the value of a marker or concentration function C, which varies from 0 to 1, defining if the material is one fluid or another (i.e or a a mix of both) much like function $\mathscr{F}(\chi)$ in equation (2.2). The fluid interface is then reconstructed, rather than tracked, from the values of C. Once the location and orientation of the interface is found, a partially filled cell contains the interface which divides it into two regions and consequently the amount of one of the fluids which passes on to the surrounding cells is determined by the internal geometry and by the velocity of the flow. A sample representation of these concepts is shown in Figure 2.1. The fact that the interface is reconstructed from the movement of material volume rather than the interface itself leads to VOF-type methods sometimes being referred to as volume-tracking methods.

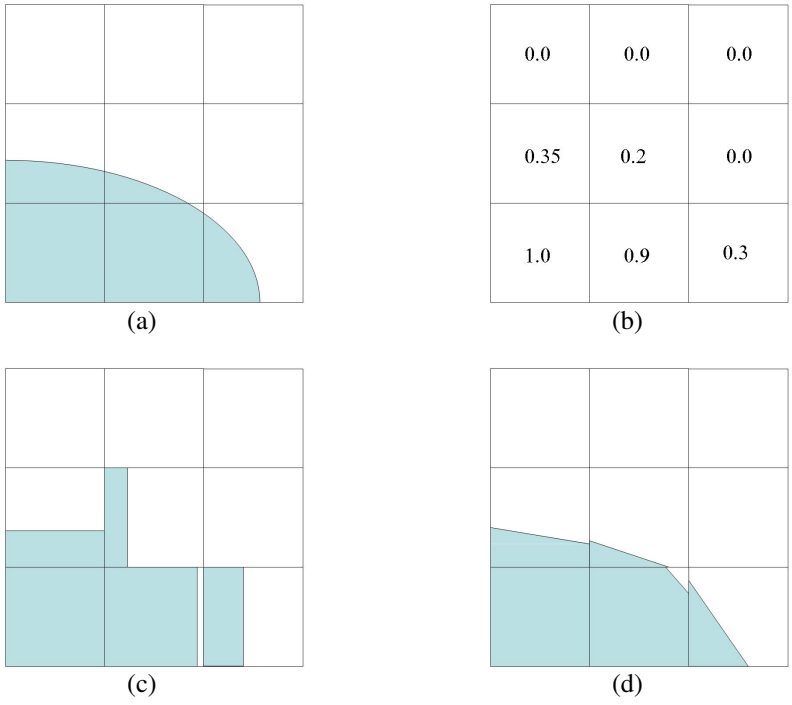


Figure 2.1: VOF representations of a fluid interface (a), by (b) the volume fraction values, (c) the SLIC method, and (d) the PLIC method.

One of the earliest implementations was performed by DeBar (1974), where a two dimensional Eulerian method was used to model compressible multi-phase flow. Noh and Woodward

(1976) developed a simple line interface reconstruction method (SLIC) relying on a piecewise constant representation of the interface. This idea was later widely used, namely by Colella et al. (1989) to model shock wave refraction at a gas interface, and by Chorin (1980) to model flame propagation and combustion. Whitaker (1990) used Chorin's variant to model Hele-Shaw flow. Rudman (1997) combined a limited flux corrected transport (FCT) scheme originally developed by Zalesak (1979) with a VOF technique, yielding a method called FCT-VOF.

Another important VOF implementation is that of Hirt and Nichols (1981), which was used on Eulerian, and Arbitrary Lagrangian- Eulerian meshes. This implementation began a series of codes, namely SOLA-VOF (Hirt and Nichols, 1981), NASA-VOF2D (Torrey et al., 1985), NASA-VOF3D (Torrey et al., 1987), RIPPLE (Kothe et al., 1991b,a), and FLOW3D (Hirt, 1988).

More recent VOF-type schemes include piece-wise linear interface calculation (PLIC) to reconstruct the material interface (see, for instance, López et al., 2005). One of the most successful of these methods is the one developed by Youngs (1982), however the original paper contained few details on the interface reconstruction and flux calculation, which were later published by Rudman (1997) in a review of volume tracking method. Pilliod and Pucket (2004) proved that the PLIC method of interface reconstruction does not guarantee second order accuracy, as was previously expected, and used the least squares volume-of-fluid interface reconstruction algorithm (LVIRA) (see Puckett (1991)) and the efficient least squares volume-of-fluid interface reconstruction algorithm (ELVIRA) (see Pilliod (1992)) which are claimed to possess second order accuracy. A thorough review of VOF-type methods can be obtained from Gopala and van Wachem (2008).

In conclusion, volume-of-fluid methods have been widely used to model the evolution of interfaces and can be formulated naturally in conservative finite difference form. This ensures that the mass of each material is conserved, and that density discontinuities as well as those of other fluid properties coincide with the location of the interface. The algorithm complexity is low when compared to actual flow solvers. However, the the main disadvantage is that the interface reconstruction is time consuming and its extension to three dimensions is quite complicated. This fact is further emphasised for unstructured grids, where the geometrically conditioned fluxing of material is not easily calculated.

2.1.3 Pseudo Concentration Method

An interface capturing method, perhaps slightly less popular than VOF and LSM, is the pseudo-concentration method (PCM). One of the earliest references on the method is the one by Thompson (1986) where creeping viscous flows are modelled through the usage of pseudo-concentrations which define material positions. These concentrations, which resemble the volume fractions of VOF and are assumed to be transported only by pure convection, serve as material markers. The evolution of the interface, defined by discontinuities in the concentration field C, is solved

directly through the advection equation,

$$\frac{\partial C}{\partial t} + \nabla \cdot (\mathbf{u}C) = 0 \tag{2.3}$$

where $\mathbf{u}$ is the flow velocity field. Concentration values vary from 0 to 1, and a typical value of 0.5 is used to define the interface through isocontours in 2D and isosurfaces in 3D. A fluid property ϑ such as the density and viscosity, is then defined by,

$$\vartheta = \vartheta_1 C + (1 - C)\vartheta_2 \tag{2.4}$$

where ϑ_1 and ϑ_2 denote the properties of fluids 1 and 2, respectively.

One of the advantages of PCM is that the modelling of multiple fluids is fairly straightforward. In general, if the modelling of m fluids is sought then it is necessary to solve $m-1$ advection equations. Natural phenomena such as merging and breaking of individual fluids does not require any special treatment, although the inclusion of surface tension forces must be carefully examined as it is not a simple task[3]. A successful implementation of surface tension has been performed by Brackbill et al. (1992), and Dufour and Pelletier (1998).

Dufour and Pelletier (2001) combined this method with a stabilised Galerkin Least Squares 2D finite element formulation and adaptive mesh strategies, while Devals et al. (2005) combined it with a streamline upwind Petrov Galerkin (SUPG) stabilised finite element formulation. Both authors encountered spurious oscillations in their concentration solution near the interface, which was solved by applying shock capturing cross-wind diffusion schemes. Kaceniauskas (2005) also combined PCM with GLS finite elements, however a simple capping of the concentration function was applied to solve the issue of unphysical oscillations.

In conclusion, PCM appears to have the potential to be more efficient than VOF and LSM. It does suffer from numerical diffusion, which is an inherent outcome of the numerical solution of the pure convection equation, and it is mesh size dependent if implemented within an Eulerian scheme. However it does not require any interface reconstruction, and can be easily implemented on unstructured meshes (see Kaceniauskas (2005)).

2.2 The interface tracking problem

In this thesis, the technique developed for interface modelling falls in the category of pseudo concentration methods. As was explained in section 2.1.3, PCM requires the solution of the advection equation for a scalar field C, given by equation (2.3). Since field C defines two immiscible fluids, it contains abrupt changes in its values between 0 and 1. These abrupt changes or jumps are directly linked to fluid properties through equation (2.4), and therefore define the fluid interface. It is clear then, that the accurate calculation of the solution to this scalar advec-

[3] Surface tension modelling remains out of the scope of this thesis.

tion equation is the key element in a realistic model of the evolution of a fluid to fluid interface.

Unfortunately, solving the purely advective equation is no easy task, and is plagued with problems such as unwanted numerical diffusion and spurious oscillations. This section exposes the issues faced when attempting to solve the purely advective equation by different methods, which leads directly into section 2.3 which describes the solution implemented in this thesis.

Perhaps the most conveniently simple way to expose the problems linked to finding the numerical solution of equation (2.3), is to perform a one dimensional analysis. Hence, let there be a one dimensional domain along the x axis, of length $\Omega = 2$. Within this domain, several functions set the shape of the initial condition for a concentration scalar field $C = C(x,t)$ which varies between 0 and 1. The equation that describes the transport of scalar field $C(x,t)$ across the domain (i.e. assuming incompressible flow) with constant velocity u is,

$$\frac{\partial C}{\partial t} + u\frac{\partial C}{\partial x} = 0 \quad \begin{cases} 0 \leq x \leq 2 \\ t \geq 0 \end{cases} \tag{2.5}$$

The advection equation can be solved numerically by discretising the domain into $N-1$ contiguous line elements, where N is the number of nodes that compose them. That is,

$$\begin{gathered} x_i = (i-1)\Delta x \quad i = 1,N \\ \Delta x = \frac{\Omega}{N-1} \end{gathered} \tag{2.6}$$

where x_i corresponds to the position of each node, and Δx is defined as the spatial step or element size used to discretise the domain. Each node is assumed to be the centre of a control volume cell, as shown in Figure 2.2.

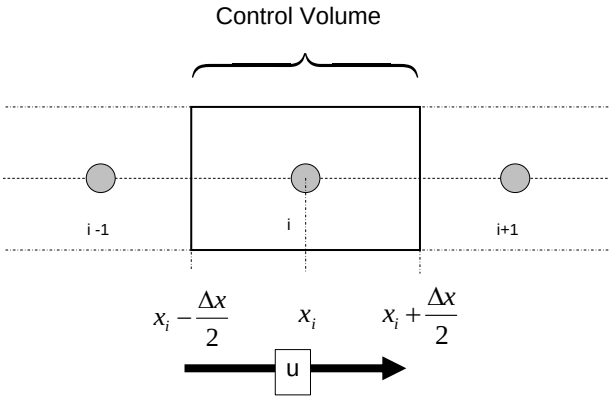


Figure 2.2: 1D control volume centred at node i.

Three functions will serve as initial conditions for the one dimensional analysis, and they are a step function wave, a sine-squared function, and half an ellipse, given by,

$$C\left(x,t^0\right)=\begin{cases}1 & x<0.1\\ 0 & x\geq 0.1\\ \sin^2\left(\frac{x-0.4}{20\Delta x}\right) & 0.4\leq x\leq 0.4+20\Delta x\\ \sqrt{1-\left(\frac{x-1.1}{0.1}\right)^2} & 1\leq x\leq 1+0.2\end{cases} \tag{2.7}$$

where t is discretised into increments of Δt such that $t=n\Delta t$, $\forall n\geqq 0$, n is an integer denoting each time iteration, and $t^0$is assumed to be the initial time of the numerical simulation.

Using a control volume approach (see, for instance Fox et al., 2003), equation (2.3) must be integrated with respect to time and space to render a solution. Two basic examples are shown as solution methods for this problem, which are first order up-winding and the Lax-Wendroff scheme, followed by a two dimensional example to further emphasise the problems associated with numerical advection. These examples lead directly into the main focus of this Chapter, which is the implementation of Leonard's Hyper-C scheme.

2.2.1 Example 1: Forward time first order upwinding

As stated by Leonard (1991), conservative explicit schemes of arbitrarily high order can be obtained from transient interpolation modelling (TIM). This means that by establishing a normalised local coordinate inside control volume i, given by ξ where,

$$\xi=\frac{2(x-x_i)}{\Delta x} \tag{2.8}$$

then the value of scalar field C within the control volume i can be calculated from the equation

$$C(\xi,t^n)=C_i^n+\xi f(\xi)\quad -1\leq\xi\leq 1 \tag{2.9}$$

First order upwinding, also known as the "Donor Cell" scheme, interpolates values in the following way,

$$C(\xi,t^n)=C_i^n+\xi(C_i^n-C_{i-1}^n) \tag{2.10}$$

which leads to the values at the control volume's faces, C_l for the left side and C_r for the right side, given by,

$$\begin{aligned}C\left(1,t^n\right)&=C_r=C_i^n\\ C\left(-1,t^n\right)&=C_l=C_{i-1}^n\end{aligned} \tag{2.11}$$

where it is useful to note that information about the amount of material to be advected into the neighbouring control volume is upwind biased. The final discretisation of the problem then becomes,

$$C_i^{n+1}=C_i^n-\gamma(C_r-C_l)^n=C_i^n-\gamma(C_i-C_{i-1})^n \tag{2.12}$$

where $\gamma = \frac{u\Delta t}{\Delta x}$ is the Courant-Friedrichs-Lewy number[4] as first described in Courant et al. (1928).

Most TIM methods can be written in the form of equation (2.9). The different variations are obtained by altering the manner in which the face values are calculated. Conservation is ensured by the fact that the calculated flux F_f at the faces of control volume i satisfies $F_r(i) = F_l(i+1)$, where $F_r = \gamma_r C_r$. In the case of higher order time discretisation schemes, the face values themselves become dependant on the Courant number γ.

Results from the first order upwinding scheme can be seen in Figure 2.3. Diffusion affects the three shapes advected to the right at constant velocity, even though the governing equation itself possesses no diffusive terms. In simple terms, the donor cell i passing material to the right has no information about the content of the downwind cell (immediately right), and therefore material is fluxed without restraint. These unwanted fluxes grow in magnitude as the diffusive effect propagates, which is at a faster rate than the shapes should travel. This effect is known as numerical diffusion.

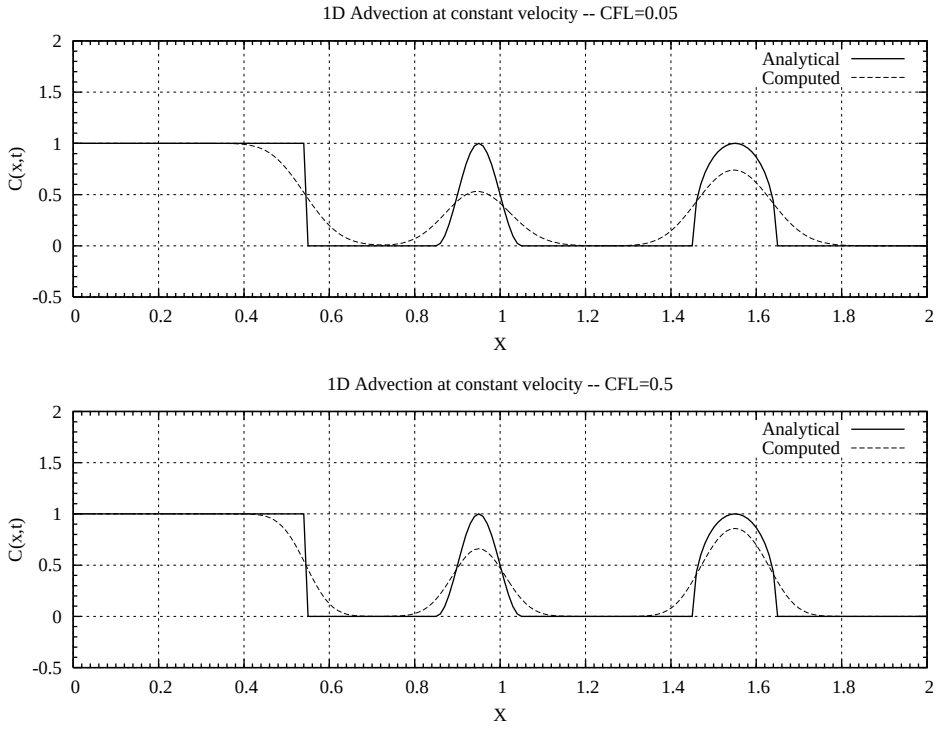


Figure 2.3: Results for first order upwinding when $t = 45\Delta x$, at CFL number of 0.05 (Top) and 0.5 (Bottom)

[4]CFL is a condition for certain algorithms for solving partial differential equations to be convergent (not to be confused with numerically stable, though for an explicit approach it often is within a small constant factor of the stability condition). As a consequence, the time step must be restricted in many explicit time-marching computer simulations, otherwise the simulation will produce incorrect results. The condition is named after Richard Courant, Kurt Friedrichs and Hans Lewy who described it in their 1928 paper Courant et al. (1928).

2.2.2 Example 2: Second order centred (Lax-Wendroff)

An attempt to obtain higher accuracy can be done by increasing the order of the approximation, in the hope that this will help to cancel diffusive effects. The Lax-Wendroff method is second order accurate in space and time, and the corresponding face values are given by,

$$\begin{aligned} C(1,t^n) = C_r &= \tfrac{1}{2}\left((C_{i+1}+C_i) - \gamma(C_{i+1}-C_i)\right) \\ C(-1,t^n) = C_l &= \tfrac{1}{2}\left((C_i+C_{i-1}) - \gamma(C_i-C_{i-1})\right) \end{aligned} \tag{2.13}$$

and the numerical solution by this method yields the results shown in Figure 2.4.

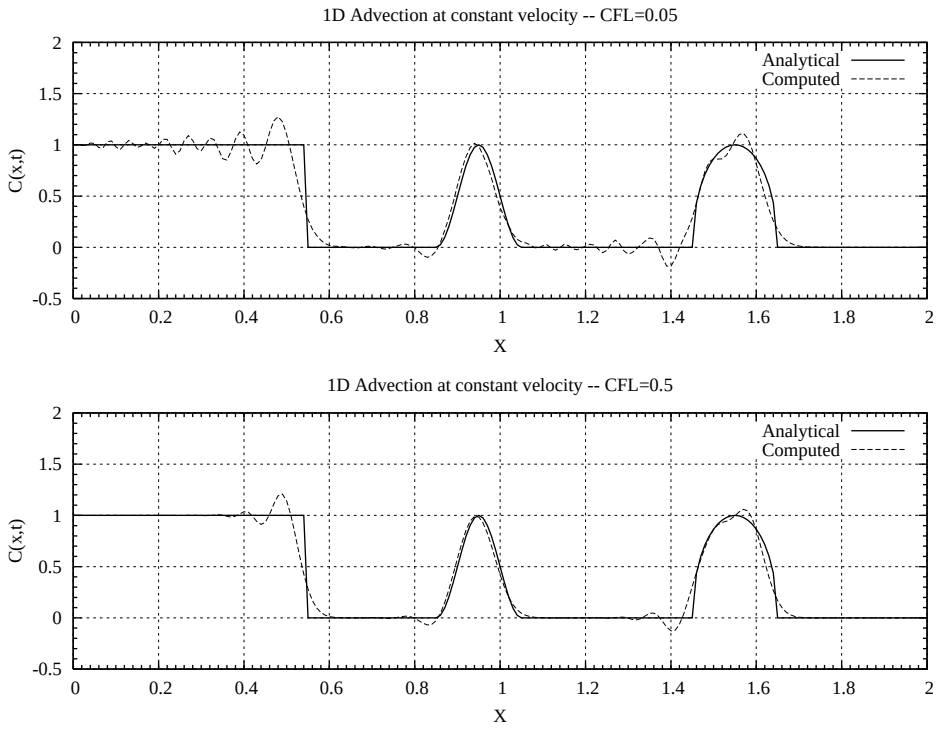


Figure 2.4: Results for the Lax-Wendroff method when $t = 45\Delta x$, at CFL number of 0.05 (Top) and 0.5 (Bottom).

Although this method seems to fare better than first order upwinding with respect to diffusive behaviour there are now two main drawbacks. The solution oscillates when representing the advection of sharp variations (e.g. the step function), and also seems to lag behind the exact solution. Improvements can be made by implementing a limiter yielding a method such as ULTIMATE (Universal Limiter for Transient Interpolation Modelling of the Advective Transport Equations), as described by Leonard (1991). Results of limiting the fluxes generated by the Lax-Wendroff scheme are shown in Figure 2.5. The results no longer contain any oscillations, however the step function still poses a problem since a certain level of diffusion can be observed.

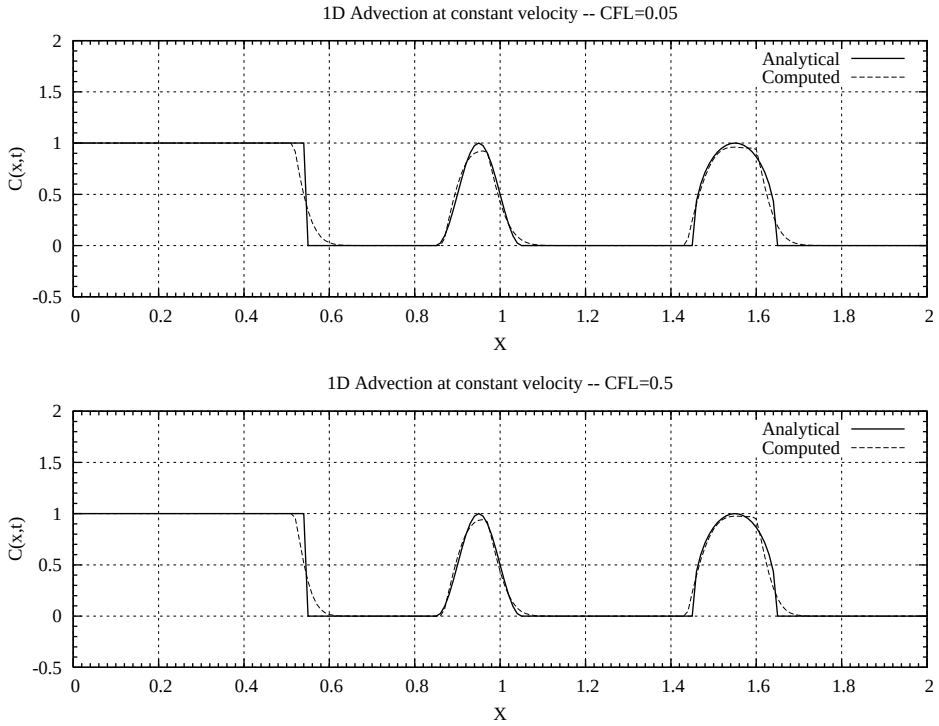
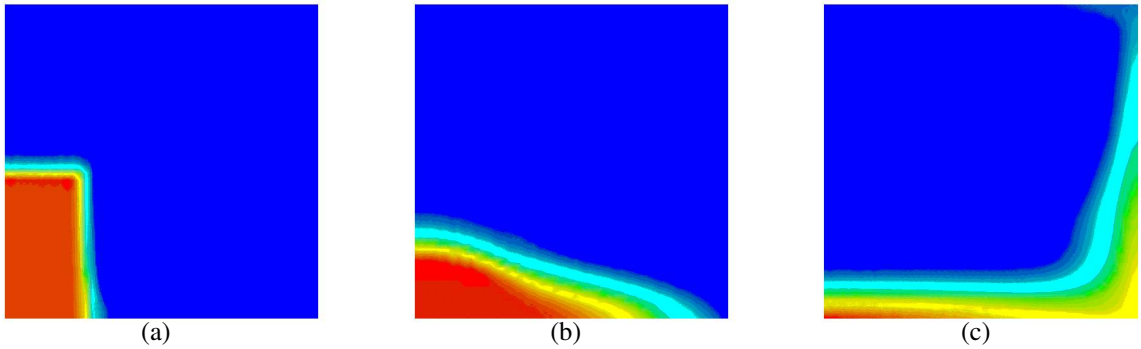


Figure 2.5: Results for the Lax-Wendroff method limited by the ULTIMATE strategy when $t = 45\Delta x$, at CFL number of 0.05 (Top) and 0.5 (Bottom).

2.2.3 Example 3: Water collapse problem (2D)

To further illustrate the implications of numerical diffusion, the sequence of frames from Figure 2.6 shows the results from a simulation intended to represent the breaking dam problem, also known as the collapsing water column, in two dimensions (see, for instance, Zhao et al. (2002a)). The scalar advection equation for concentration was solved using a first order upwinding scheme, while the flow equations were solved using the finite element method (see Gresho and Sani (2000)). The initial conditions for the concentration field were defined to represent a column of water in a container full of air, with boundary conditions of free-slip on all walls, using an unstructured triangular mesh. Inside the water column, $C = 1$, while outside, $C = 0$. Gravitational forces acting in the downward direction collapse the column and cause water to rise against the opposite wall.

Even though the scalar equation being solved involves no diffusive terms, smearing of the interface is observed. This smearing is not representative of realistic behaviour of immiscible fluids. Section 2.3 introduces the core solution technique used to develop the method presented in this thesis to eradicate diffusion from low order schemes (as seen in Figure 2.3) and oscillations from higher order ones (as seen in Figure 2.4). Hyper-C, a compressive advection scheme described by Leonard (1991), was implemented in a multi-dimensional incompressible flow solution method to obtain the desired sharp, conservative, and efficient interface tracking



(a) (b) (c)

Figure 2.6: Two dimensional breaking dam problem. Lower density fluid is shown in blue, while the higher density fluid is shown in red.

technique. Section 2.4 provides in-depth details of the discretisation and solution technique for the Navier- Stokes equations, which lead into the details for the method of scalar advection of the concentration function using Hyper-C. Sections 2.5 and 2.6 present results from analytical and experimental validation tests, respectively.

2.3 Compressive advection

A method that would take into account material information from the node directly downwind is called first order downwinding. For this method, the corresponding face values are,

$$\begin{aligned} C_r &= C^n_{i+1} \\ C_l &= C^n_i \end{aligned} \quad \forall n \geqq 0 \tag{2.14}$$

Results presented in Figure 2.7, show that after just a few timesteps ($t = \Delta x$) this method is unstable due to the overshoots and undershoots. The fluxes calculated to transfer material on to the downwind cell do not possess any information about the content in the upwind cell, and thus the method causes control volumes to be over filled and over emptied. Qualitatively, while the flow tends to carry the material to the right, the method tends to force it in the opposite direction in a compressive manner. This compressive behaviour can be combined with a limiting process in order to prevent the unwanted amount of fluxing, and thus to obtain a stable method.

To achieve correct limiting, the face values must be controlled. Since there are several combinations of cases regarding the values of C at the nodes, the sign of the velocities, and values of the gradient of C, a very useful concept of normalised variables should be introduced. In general, control volume face values are dependant on the upwind, downwind, and central node value and sign. Figure 2.8 expresses the definition of these nodes, which depends on the direction of the velocity at the face.

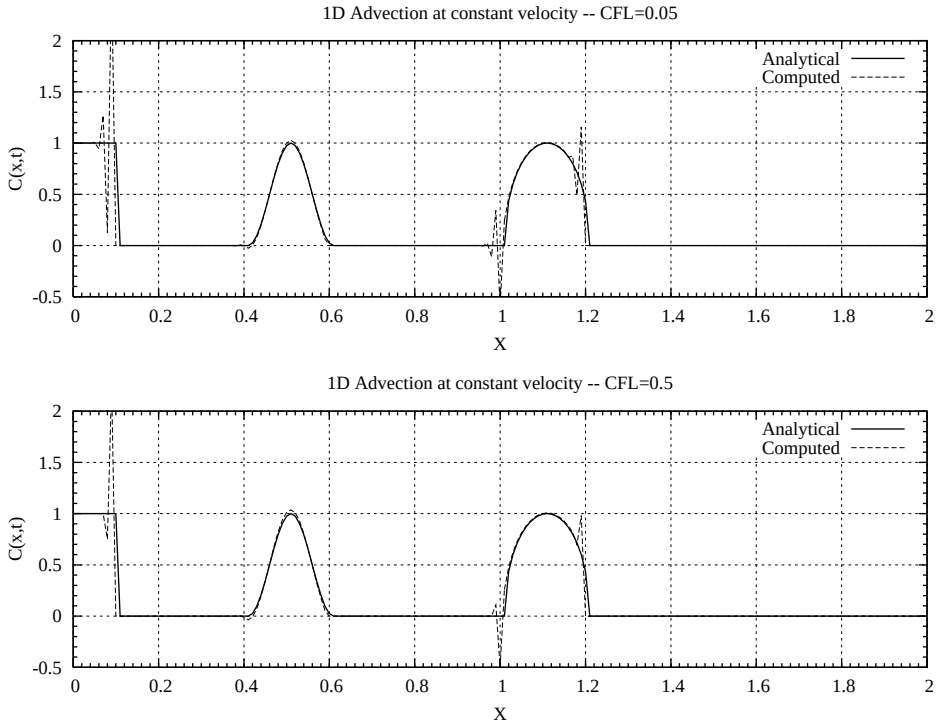


Figure 2.7: Results for first order downwinding when $t = \Delta x$, at CFL number of 0.05 (Top) and 0.5 (Bottom)

Let $\bar{C}$ be the normalised value of the concentration at any point in the control volume, then

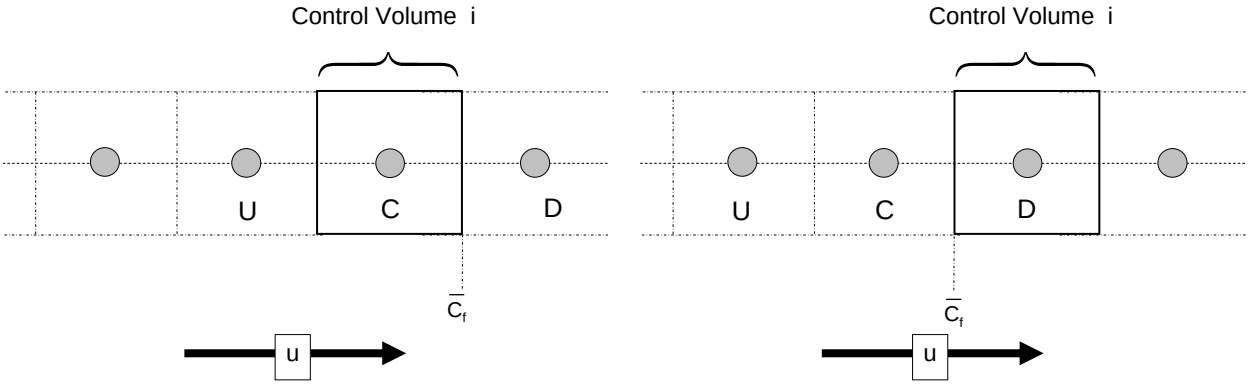
$$\bar{C} = \frac{C - C_U}{C_D - C_U} \tag{2.15}$$

where subindexes U and D imply values at the upwind and downwind node, respectively. In unnormalised terms, the face value C_f depends not only on the Courant number γ, but also on the upwind, downwind, and central values (for higher order schemes other nodes must also be included). In normalised terms, the face value $\bar{C}_f$ depends only on the Courant number and the central value,

$$\bar{C}_f = \frac{C_f - C_U}{C_D - C_U} = f\left(\bar{C}_c, \gamma\right) \tag{2.16}$$

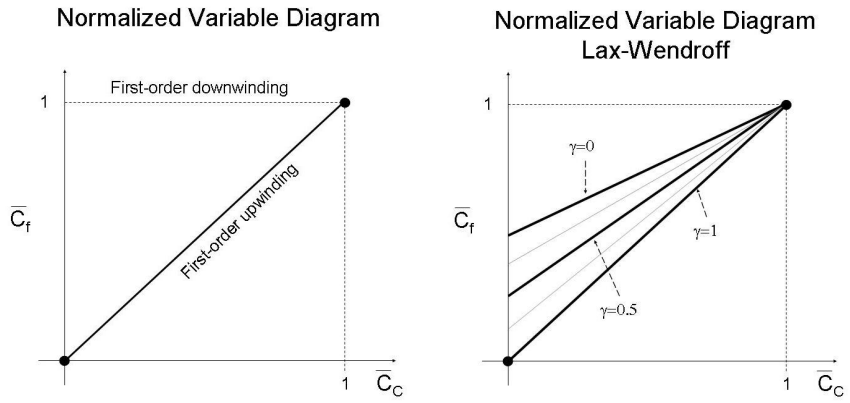
A Normalised Variable Diagram (NVD) can be constructed representing this function, and a curve is obtained for each advection scheme and each Courant number, where applicable. Both first order schemes follow a fixed curve in this diagram, as noted by the dark lines in Figure 2.9(a), while the Lax-Wendroff scheme follows curves that vary with the Courant number γ, as shown in Figure 2.9(b). Many combinations may be built by constructing different curves on these diagrams.

In the case where compressive behaviour is desired, then the normalised face values are



(a) Convention for right face (b) Convention for left face

Figure 2.8: Definition of upwind, downwind, and central nodes, depending on the face and flow direction.



(a) (b)

Figure 2.9: NVD diagrams for (a) first order upwinding and downwinding, and (b) Lax Wendroff second order centred methods

conditioned by,

$$\begin{aligned} \bar{C}_f &= \frac{\bar{C}_c}{\gamma} \qquad \bar{C}_c \leq \gamma \\ \bar{C}_f &= 1 \qquad \gamma < \bar{C}_c \leq 1 \end{aligned} \tag{2.17}$$

which can be described by the NVD diagram shown in Figure 2.10. This method is called Hyper-C and was developed in Leonard (1991) and also by Depres and Lagoutiere (2001). Its highly compressive characteristics can be appreciated in the results shown in Figure 2.11 for the advection of the three functions defined in (2.7).

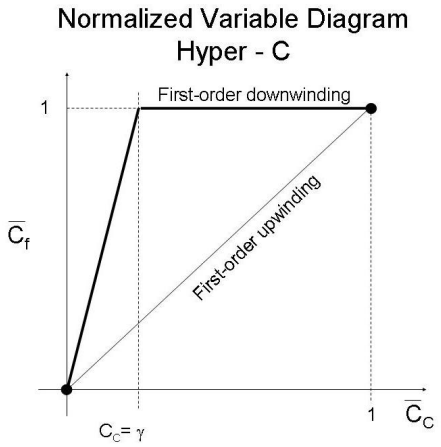


Figure 2.10: NVD Diagram for Hyper-C.

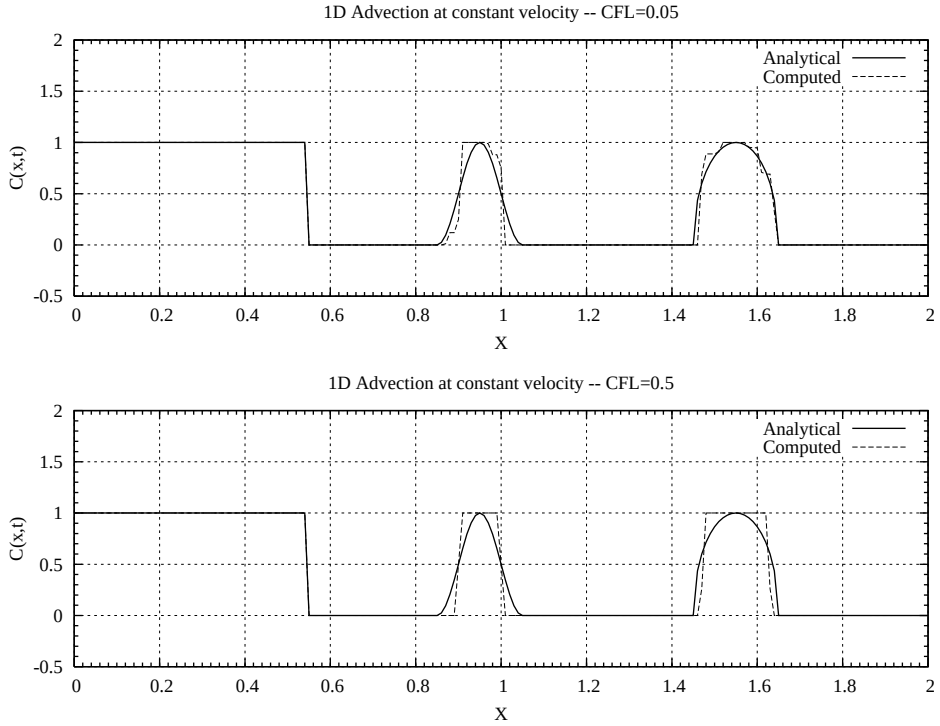


Figure 2.11: Results for limited first order downwinding (Hyper-C) when $t = \Delta x$, at CFL number of 0.05 (Top) and 0.5 (Bottom)

This method not only has the rare capability to sharpen smooth functions, but also has the highly sought after ability to maintain sharpness in other functions such as the step function. The sine and half-ellipse functions have been distorted into square waves, which is an example of the method's compressive behaviour. Even though the functions have changed shape there is no lag present in their movement, as the centres of the respective exact and approximated solutions have moved the same distance. The step function in one dimension is representative

of the discontinuous jumps in fluid properties across interfaces in two and three dimensions and hence the use of this method for interface modelling at this stage has a strong potential.

Ubbink (1997) implemented what is referred to as CICSAM (Compressive Interface Capturing Scheme for Arbitrary Meshes), which is based on Leonard's Hyper-C scheme, into a finite volume formulation. Ubbink used a smoothed volume fraction field to obtain the curvature of the free surface, and consequently model surface tension.

Details of the implementation of Hyper-C into a multi-dimensional finite element incompressible flow solver, Fluidity, are detailed in the next section.

2.4 Discretisation formulation and solution

One of the fundamental steps in computational fluid dynamics is the spatial discretisation of the fluid domain. The geometric subdivision brings along a formulation that assumes how the different variables are described within the elements. Depending on different factors such as computational cost, required accuracy, or simplicity, an element type will be chosen from tetrahedra, hexahedra, or prism elements, to name a few. These elements, in turn, could be linear, quadratic, or any other given order. A good summary of element types is given by Zienkiewicz and Taylor (1991). The formulation and solution methods are strongly influenced by the choice of element. Gresho and Sani (2000) performed a thorough analysis on choices of element analysing different discretisation techniques. This analysis is crucial as not all choices yield stable methods.

The aim of this section is to portray the workspace in which the developments have taken place. It contains a summary of concepts, starting from the Galerkin finite element method (GFEM) formulation, and how it is linked in the solution method to the Control Volume Finite Element Method formulation (CVFEM), which is where Hyper-C is applied. For the sake of completeness, some basic approximation techniques for solution variables within a tetrahedron are also given. The section culminates with the portrayal of the solution algorithm and some of its features.

2.4.1 Governing equations

The governing equations for a compressible fluid is given by the mathematical description of the laws of conservation of mass, conservation of momentum, and conservation of energy,

$$\frac{d\rho}{dt} + \rho\nabla\cdot\mathbf{u} = \frac{\partial\rho}{\partial t} + \nabla\cdot(\rho\mathbf{u}) = \frac{\partial\rho}{\partial t} + \mathbf{u}\cdot\nabla\rho + \rho\nabla\cdot\mathbf{u} = 0 \tag{2.18}$$

$$\rho\frac{d\mathbf{u}}{dt} = \rho\left(\frac{\partial\mathbf{u}}{\partial t} + (\mathbf{u}\cdot\nabla)\,\mathbf{u}\right) = \nabla\cdot\Theta + \mathbf{f} \tag{2.19}$$

$$\rho\frac{dE}{dt} = \rho\left(\frac{\partial E}{\partial t} + \mathbf{u}\cdot\nabla E\right) = \nabla\cdot(\Theta\cdot\mathbf{u}) + \mathbf{u}\cdot\mathbf{f} \tag{2.20}$$

where ρ represents fluid density, $\mathbf{u} = (u, v, w)$ the three dimensional flow velocity vector, $\mathbf{f}$ is a vector containing external body and/or surface forces (i.e. gravity), E is the fluid's internal energy, and Θ is the stress tensor containing the effects of pressure and viscous forces, given by,

$$\Theta = \begin{bmatrix} -p+\frac{2}{3}\mu\left(2\frac{\partial u}{\partial x}-\frac{\partial v}{\partial y}-\frac{\partial w}{\partial z}\right) & \mu\left(\frac{\partial u}{\partial y}+\frac{\partial v}{\partial x}\right) & \mu\left(\frac{\partial u}{\partial z}+\frac{\partial w}{\partial x}\right) \\ \mu\left(\frac{\partial u}{\partial y}+\frac{\partial v}{\partial x}\right) & -p+\frac{2}{3}\mu\left(2\frac{\partial v}{\partial y}-\frac{\partial u}{\partial x}-\frac{\partial w}{\partial z}\right) & \mu\left(\frac{\partial v}{\partial z}+\frac{\partial w}{\partial y}\right) \\ \mu\left(\frac{\partial u}{\partial z}+\frac{\partial w}{\partial x}\right) & \mu\left(\frac{\partial v}{\partial z}+\frac{\partial w}{\partial y}\right) & -p+\frac{2}{3}\mu\left(2\frac{\partial w}{\partial z}-\frac{\partial u}{\partial x}-\frac{\partial v}{\partial y}\right) \end{bmatrix} \tag{2.21}$$

Finally, there is an equation of state, which rules the relationship between density, pressure, and temperature,

$$p = p(\rho, T) \tag{2.22}$$

Assuming incompressible, isothermal, flow of a Newtonian fluid, and adding an equation for the advection of the scalar field C, yields new governing equations, given by,

$$\nabla \cdot \mathbf{u} = 0 \tag{2.23}$$

$$\rho\left(\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u}\cdot\nabla)\,\mathbf{u}\right) = \nabla\cdot\sigma + \mathbf{f} \tag{2.24}$$

$$\frac{\partial C}{\partial t} + \nabla\cdot(\mathbf{u}C) = 0 \tag{2.25}$$

where C is the material concentration or volume fraction of fluid, and σ is the incompressible stress tensor, given by,

$$\sigma = \begin{bmatrix} -p+2\mu\frac{\partial u}{\partial x} & \mu\left(\frac{\partial u}{\partial y}+\frac{\partial v}{\partial x}\right) & \mu\left(\frac{\partial u}{\partial z}+\frac{\partial w}{\partial x}\right) \\ \mu\left(\frac{\partial u}{\partial y}+\frac{\partial v}{\partial x}\right) & -p+2\mu\frac{\partial v}{\partial y} & \mu\left(\frac{\partial v}{\partial z}+\frac{\partial w}{\partial y}\right) \\ \mu\left(\frac{\partial u}{\partial z}+\frac{\partial w}{\partial x}\right) & \mu\left(\frac{\partial v}{\partial z}+\frac{\partial w}{\partial y}\right) & -p+2\mu\frac{\partial w}{\partial z} \end{bmatrix} \tag{2.26}$$

Equation (2.25) literally defines an equation of state, so at any point in the fluid domain,

$$\rho = \rho_1 C + \rho_2(1-C) \tag{2.27}$$

$$\mu = \mu_1 C + \mu_2(1-C) \tag{2.28}$$

where ρ_1, ρ_2, μ_1, and μ_2 are the prescribed densities and viscosities for each fluid. It is also important to keep in mind, that,

$$C_1 + C_2 = 1 \tag{2.29}$$

where C_1 and C_2 are the respective volume fractions of each fluid. In general, the solution will be sought for one of the fluids, obtaining the concentration values for the other one by subtracting from 1.

Equations (2.23), (2.24), and (2.25) will be used as the fundamental model governing fluid equations throughout this thesis. Even though work is in progress to allow the modelling of compressible multi- material dynamics, the main assumption in this thesis is that fluids are incompressible.

The methods used in this section to obtain a numerical solution of the mentioned governing equations are part of what is referred to as the finite element method. Characteristics of the method involve the division of the problem's domain into subdomains, or elements, which are defined by points or nodes which are interconnected. The connectivity information is essential in describing and solving the behaviour of the different properties of the continuum by linking elemental contributions from each subdomain into a global one. To approximate variables such as fluid properties within a sub-domain, interpolation functions are defined. These functions are part of a discretisation method which, coupled to different solution techniques, provides the means to resolve the problem placed by the governing equations, associated boundary conditions, and geometry. Linear tetrahedrons were the elements used to develop the methods described in this thesis and they will be assumed by default, unless otherwise stated, in the remainder of this text. The choice was made based on simplicity for analysis and the element's ability to describe complex geometrical domains. It is appropriate to note, however, that many descriptions and methodologies are applicable analogously to different kinds of elements which vary geometrically (e.g. hexahedra, prisma, etc.) and/or in order (e.g. quadratic elements), although formulations might prove somewhat much more complex (see, for instance Gresho and Sani, 2000) .

2.4.2 Galerkin weighted residual approach

The method of weighed residuals is the basis for FEM, and consists in transforming the governing equations into an equivalent integral expression. To illustrate this, let there be a set of differential equations, defined by an operator $\mathbf{g}$ on a given domain Ω such that,

$$\mathbf{g}(\vartheta) = \mathbf{0} = \left\{ \begin{array}{c} g_1(\vartheta) \\ g_2(\vartheta) \\ \vdots \end{array} \right\} = \left\{ \begin{array}{c} \mathbf{0} \\ \mathbf{0} \\ \vdots \end{array} \right\} \tag{2.30}$$

coupled to a corresponding set of boundary conditions defined by operator $\mathbf{h}$ on the contour Γ of domain Ω,

$$\mathbf{h}(\vartheta) = \mathbf{0} \tag{2.31}$$

where ϑ is a solution variable. The equivalent integral form of the governing set of equations (2.30) and (2.31), is given by,

$$\int_{\Omega} \mathbf{w}(\mathbf{x})^T \, \mathbf{g}(\vartheta) \, d\Omega + \oint_{\Gamma} \bar{\mathbf{w}}(\mathbf{x})^T \, \mathbf{h}(\vartheta) \, d\Gamma = \mathbf{0} \tag{2.32}$$

where $\mathbf{w}$ and $\bar{\mathbf{w}}$ are sets of arbitrary functions, normally referred to as weight, test, or trial functions. It is useful to note at this point, that if equations (2.30) and (2.31) are satisfied, then so must (2.32), and vice-versa. One of the main advantages of the integral form resides in the fact that if the domain Ω was subdivided, it can be assembled with contributions from each element e belonging to the subdivided domain,

$$\sum_e \left\{ \int_{\Omega_e} \mathbf{w}(\mathbf{x})^T \, \mathbf{g}(\vartheta) \, d\Omega + \oint_{\Gamma_e} \bar{\mathbf{w}}(\mathbf{x})^T \, o\mathbf{h}(\vartheta) \, d\Gamma \right\} = \mathbf{0} \tag{2.33}$$

where $\sum_e$ implies the assembled sum. Considering that the exact generalised solution to the problem stated by equation (2.33) is impossible, the next step is to approximate the variable ϑ. As will be explained further in depth in Section 2.4.3, ϑ may be approximated through a linear combination of functions N_i, such as,

$$\vartheta \cong \hat{\vartheta}(\mathbf{x}) = \sum_{i=0}^{n_n} N_i(\mathbf{x}) \, \bar{\vartheta}_i \tag{2.34}$$

where $\mathbf{x}$ is the vector of coordinates of a point in Ω, n_n are the number of points corresponding to functions $N_i(\mathbf{x})$, and $\bar{\vartheta}_i$ are the unknown approximated values of ϑ at those points. Substituting the approximated version $\hat{\vartheta}$ into (2.33) causes equations (2.30) and (2.31) to be no longer satisfied, hence

$$\mathbf{g}\left(\hat{\vartheta}\right) = \left\{ \begin{array}{c} g_1\left(\hat{\vartheta}\right) \\ g_2\left(\hat{\vartheta}\right) \\ \vdots \end{array} \right\} = \left\{ \begin{array}{c} r_1 \\ r_2 \\ \vdots \end{array} \right\} \tag{2.35}$$

and,

$$\mathbf{h}\left(\hat{\vartheta}\right) = r_\Gamma \tag{2.36}$$

where r is known as the residual from the approximation. However, due to the arbitrariness of the weight functions, equation (2.33) still holds true, and so

$$\sum_e \left\{ \int_{\Omega_e} \mathbf{w}(\mathbf{x})^T \, \mathbf{g}\left(\hat{\vartheta}\right) \, d\Omega + \oint_{\Gamma_e} \bar{\mathbf{w}}(\mathbf{x})^T \, \mathbf{h}\left(\hat{\vartheta}\right) \, d\Gamma \right\} = \mathbf{0} \tag{2.37}$$

Equation (2.37) is referred to as the weighted residual formulation of the set of governing equations described by $\mathbf{g}$ and $\mathbf{h}$ in equations (2.35) and (2.36), and obtaining an approximate solution for ϑ implies choosing a finite number of weight functions, along with a finite number of points where $\bar{\vartheta}_i$ is to be sought. Weight functions may be defined either globally or locally, and the choices lead to different discretisation methods. For example, choosing a set of Dirac delta functions leads to the point collocation method, while minimising the square of the residual is known as the Least Squares method. In this thesis, the Galerkin variant of the weighted residual approach is used.

Also known as the Galerkin projection, in the Galerkin weighed residual method the weight functions chosen are basis functions N_i for describing a variable's variation within a given element. Hence $\mathbf{w} = \mathbf{N}$, where $\mathbf{N}$ is a set of the basis, or shape, functions, described in Section 2.4.3. In general, the Galerkin weighted residual approach takes the following form, assuming $\bar{\mathbf{w}} = -\mathbf{w}$,

$$\sum_e \int_{\Omega_e} \mathbf{N}(\mathbf{x})^T \mathbf{g}\left(\hat{\vartheta}\right) d\Omega - \oint_{\Gamma_e} \mathbf{N}(\mathbf{x})^T \mathbf{h}\left(\hat{\vartheta}\right) d\Gamma = \mathbf{0} \tag{2.38}$$

Further details of the weighted residual approach and projection methods can be found in any good Finite Element book, such as the ones by Zienkiewicz and Taylor (1991), Gresho and Sani (2000), and Hughes (2000).

2.4.3 Piecewise-linear basis functions

As was mentioned in Section 2.4.2, they key to the discretisation lies in how a variable is approximated within a subdomain, or element. Let there be a domain $\Omega \in \Re^3$ in which the conservation equations laws for Newtonian fluids, described by equations (2.23-2.25) are to be solved. If this domain is subdivided into non-overlapping elements[5] such that,

$$\Omega = \sum_{e=1}^{n_{ele}} \Omega_e \tag{2.39}$$

where n_{ele} is the domain's total number of elements, and Ω_e is the volume of each element e. Let ϑ be a fluid property defined globally as $\vartheta = \vartheta(\mathbf{x},t)$, where $\mathbf{x}$ is the vector of global coordinates of the continuum, and t is the time variable. Property ϑ may have either scalar, vectorial, or tensorial characteristics, however for notation simplicity it will be considered a scalar unless otherwise stated. The value of ϑ at any point p within Ω can be approximated in terms of the values at the nodes that define the element e which contains p through a set of interpolation functions N_i, hence,

$$\hat{\vartheta}(\mathbf{x}(p),t) = \sum_{i=1}^{n_l} \bar{\vartheta}_i N_i^e(\mathbf{x}(p)) \quad \mathbf{x}(p) \in \Omega_e \tag{2.40}$$

where $\mathbf{x}(p)$ are the coordinates of point p, n_l is the number of nodes that make up the element shape, N_i are the shape functions[6] and $\bar{\vartheta}_i$ the variable's nodal values defined for local node i. Functions N_i for a lineal tetrahedral are obtained from the element geometry through,

[5]This affirmation does not necessarily imply that the elements perfectly fit the shape of domain Ω. This is generally untrue for complicated or rounded boundary domains described by, for example, linear tetrahedra. Explanation of the effects of domain geometry description by the discretisation is not the intent of this Chapter, and is therefore neglected.

[6]Also often referred to as test, or basis functions. These functions characteristics (i.e: order) is normally determined by the number of nodes that compose the shape of the element. In general, the more the nodes that form the shape, the higher order the shape functions may be. (see Zienkiewicz and Taylor, 1991)

$$N_i(\mathbf{x}) = \frac{1}{6\Omega_e}\left(a_i + b_i x + c_i y + d_i z\right) \tag{2.41}$$

where x, y, and z are the coordinates of a given point within element e (i.e. point p), and Ω_e is the volume of the element.. Coefficients a_i, b_i, c_i, and d_i are given by the relationship,

$$a_i = \begin{vmatrix} x_j & y_j & z_j \\ x_k & y_k & z_k \\ x_l & y_l & z_l \end{vmatrix}; \; b_i = -\begin{vmatrix} 1 & y_j & z_j \\ 1 & y_k & z_k \\ 1 & y_l & z_l \end{vmatrix} \tag{2.42}$$

$$c_i = \begin{vmatrix} x_j & 1 & z_j \\ x_k & 1 & z_k \\ x_l & 1 & z_l \end{vmatrix}; \; d_i = -\begin{vmatrix} x_j & y_j & 1 \\ x_k & y_k & 1 \\ x_l & y_l & 1 \end{vmatrix} \tag{2.43}$$

where subindexes i, j, k, l imply the respective circular rotation of the local node numbers. The volume of the element can be obtained from,

$$\Omega_e = \frac{1}{6}\begin{vmatrix} 1 & x_i & y_i & z_i \\ 1 & x_j & y_j & z_j \\ 1 & x_k & y_k & z_k \\ 1 & x_l & y_l & z_l \end{vmatrix} \tag{2.44}$$

It is also typical with tetrahedrons to use volumetric coordinates. The volumetric coordinate of a given point p within an element Ω_e is given by the ratio of the partial volume Ω_p defined by the tetrahedron formed by point p and the face opposing node i, as seen in Figure 2.12 ,

$$\upsilon_i = \frac{\Omega_p}{\Omega_e} \;\; i = 1, 2, 3, 4 \tag{2.45}$$

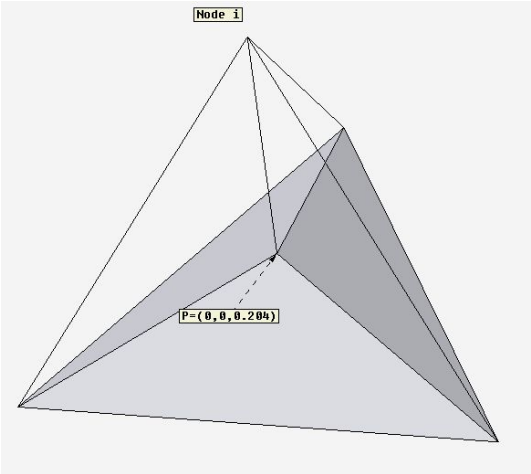


Figure 2.12: Partial volume (shaded region) corresponding to local node i used to calculate volume coordinates at point p.

Equation (2.45) is convenient when used numerically to calculate values of form functions, as it is easy to prove that,

$$\upsilon_i = N_i \tag{2.46}$$

From equations (2.45) and (2.46) it can be deduced that,

$$N_1 + N_2 + N_3 + N_4 = 1 \tag{2.47}$$

It is then convenient to show at this point that the derivative of the shape functions can be obtained, for a linear tetrahedron, from

$$\frac{\partial N_i}{\partial x} = \frac{b_i}{6\Omega_e}; \frac{\partial N_i}{\partial y} = \frac{c_i}{6\Omega_e}; \frac{\partial N_i}{\partial z} = \frac{d_i}{6\Omega_e} \tag{2.48}$$

2.4.4 Discretisation of the momentum equations

Applying the Galerkin projection method to the incompressible momentum equation (2.24), and expressing it in weak form[7] yields,

$$\sum_{e=1}^{n_e} \left\{ \int_{\Omega_e} \rho \mathbf{N} \left(\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla) \mathbf{u} \right) d\Omega_e + \int_{\Omega_e} \sigma \cdot \nabla \mathbf{N} d\Omega_e - \oint_{\Gamma_e} \mathbf{N} (\sigma \cdot \mathbf{n}) \, d\Gamma_e - \int_{\Omega_e} \mathbf{N} \rho \mathbf{f} d\Omega_e \right\} = 0 \tag{2.49}$$

The $\sum_{e=1}^{n_e} \{\}$ operator will be dropped from here on, however an e sub index implies that it is still in effect unless otherwise stated. Substituting the approximate solutions $\tilde{\mathbf{u}} = \mathbf{N}^T \mathbf{U}$ for velocity, and $\tilde{p} = \mathbf{N}^T \mathbf{p}$ for pressure into equation (2.49), renders the following matrix form,

$$\mathbf{M} \frac{\partial \mathbf{U}}{\partial t} + \mathbf{A}\mathbf{U} + \mathbf{K}\mathbf{U} + \mathbf{Q}\mathbf{p} = \mathbf{F} \tag{2.50}$$

where,

$$\mathbf{U} = [u_1, u_2, u_3, u_4, v_1, v_2, v_3, v_4, w_1, w_2, w_3, w_4]^T \quad \mathbf{p} = [p_1, p_2, p_3, p_4]^T \tag{2.51}$$

where superscript T implies the transpose, and sub-indexes $1-4$ imply values at each one of the four nodes of a tetrahedron. The block matrices are defined by,

$$\mathbf{M} = \begin{bmatrix} M & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & M & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & M \end{bmatrix} \quad \mathbf{A} = \begin{bmatrix} A & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & A & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & A \end{bmatrix} \quad \mathbf{F} = \begin{bmatrix} f_x \\ f_y \\ f_z \end{bmatrix} \quad \mathbf{Q} = \begin{bmatrix} Q_x \\ Q_y \\ Q_z \end{bmatrix}, \tag{2.52}$$

and,

[7] The weak form is obtained when a lower order of continuity for $\mathbf{u}$ is needed for integration, in exchange for a higher continuity requirement for $\mathbf{N}$. (See Zienkiewicz and Taylor (1991) for details.) In this case it was obtained using the Divergence Theorem.

$$\mathbf{K} = \begin{bmatrix} 2K_{xx}+K_{yy}+K_{zz} & K_{yx} & K_{zx} \\ K_{xy} & K_{xx}+2K_{yy}+K_{zz} & K_{zy} \\ K_{xz} & K_{yz} & K_{xx}+K_{yy}+2K_{zz} \end{bmatrix} \tag{2.53}$$

The components of the different block matrices and vectors used are given by,

$$M = \int_{\Omega_e} \rho \mathbf{N}\mathbf{N}^T \, \mathbf{d}\Omega_e \quad A = \int_{\Omega_e} \rho \mathbf{N}\mathbf{N}^T \mathbf{U} \nabla \mathbf{N}^T \, d\Omega_e \tag{2.54}$$

$$\mathbf{K} = \int_{\Omega_e} \mu \nabla \mathbf{N} \nabla \mathbf{N}^T \mathbf{d}\Omega_e \quad \mathbf{Q} = \int_{\Omega_e} \nabla \mathbf{N}\mathbf{N}^T \mathbf{d}\Omega_e \tag{2.55}$$

$$\mathbf{F} = \int_{\Omega_e} \rho \mathbf{N}\mathbf{f} d\Omega_e + \oint_{\Gamma_e} \mathbf{N}\sigma \cdot \hat{\mathbf{n}} \, d\Gamma_e \tag{2.56}$$

2.4.5 CVFEM method

As mentioned earlier, the method implemented for the solution of the scalar advection equation in conjunction with the incompressible Navier Stokes equations is the Control Volume Finite Element method (CVFEM). Baliga and Patankar (1980) originally introduced this numerical method for discretising convection diffusion equations, applying it on two dimensional triangular elements after constructing polygonal control volumes from the underlying mesh. This method continues to be the subject of much research, and modifications and extensions to the original one have been published. Swaminathan et al. (1993) combined the aspects of Petrov-Galerkin stabilisation methods for standard finite elements and CVFEM to obtain a discretisation for coupled fluid flow and heat transfer. Jones et al. (1997) designed a CVFEM method for quadrilateral grids to model flows on faults, fractures, and other geological irregularities. Omri and Nasrallah (1999) employed this method on triangular elements to investigate ventilation and cooling quality by analysing mixed convective flows in an air-cooled cavity. CVFEM is often chosen for its greater accuracy, robustness, and flexibility when handling problems with high gradients or complex geometry.

In three dimensions, CVFEM spatial discretisation is obtained through the construction of a grid underlying the original tetrahedral mesh. To visualise this, a tetrahedron belonging to a given mesh may be subdivided taking into account its centre of mass $\mathbf{x}_m$, the centroids of each of the faces $\mathbf{x}_{f_i}$, and the midpoint of its edges $\mathbf{x}_e$ as is shown in Figure 2.13. Adding the individual volumetric contributions (i.e.: grey shaded area of tetrahedron in Figure 2.13) from each element that is connected to one node yields a control volume described in Figure 2.14.

Formulation of CVFEM is node centred, hence, for a typical advection equation for a scalar $C = C(\mathbf{x},t)$,

$$\int_{\Omega_{cv}} \left(\frac{\partial C}{\partial t} + \nabla \cdot (C\mathbf{u}) \right) d\Omega_{CV} = 0 \tag{2.57}$$

Using vector identities, equation (2.57) results in,

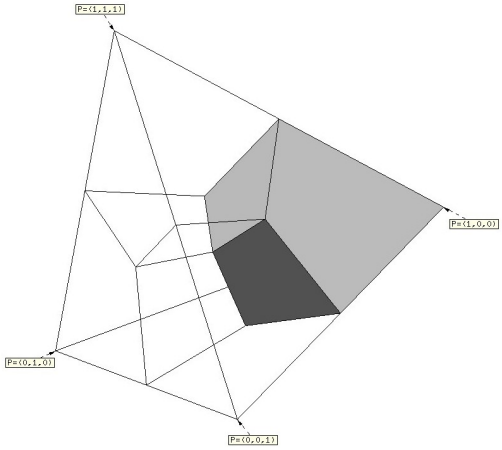


Figure 2.13: Tetrahedron element being subdivided via the volumetric centre, face centroids, and edge midpoints.

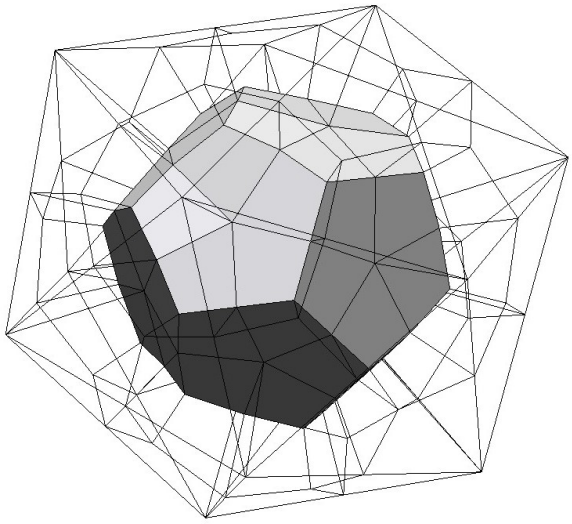


Figure 2.14: Control volume (i.e. shaded region) generated by contributions from all elements which contain a given node.

$$\int_{\Omega_{CV}} \frac{\partial C}{\partial t} d\Omega_{CV} + \int_{\Omega_{CV}} C\,(\nabla \cdot \mathbf{u}) + \mathbf{u} \cdot \nabla C\, d\Omega_{CV} = 0 \tag{2.58}$$

and after taking into account a solenoidal velocity field $\mathbf{u}$ and applying the Divergence Theorem yields,

$$\int_{\Omega_{CV}} \frac{\partial C}{\partial t} d\Omega_{CV} + \int_{\Gamma_{CV}} C_\Gamma \mathbf{u}_\Gamma \cdot \hat{\mathbf{n}}\, d\Gamma_{CV} = 0 \tag{2.59}$$

where C_Γ and $\mathbf{u}_\Gamma$ are the control volume surface values of C and flow velocity, respectively. Given that the control volumes are node centred, there must be one equation per node, hence using the weighed residual approach through arbitrary weight function $\mathbf{w}$ yields,

$$\int_{\Omega_{CVi}} \mathbf{w} \frac{\partial C}{\partial t} d\Omega_{CVi} + \int_{\Gamma_{CVi}} \mathbf{w} C_\Gamma \mathbf{u}_\Gamma \cdot \hat{\mathbf{n}}\, d\Gamma_{CVi} = 0 \tag{2.60}$$

In contrast to the Galerkin projection approach which was shown in Section 2.4.2, the method of subdomain collocation is used in this case. This basically means that w is considered to be equal to 1 inside a given control volume and 0 everywhere else.

2.4.6 Implementing Hyper-C

Backward Euler is a typical finite difference time discretisation scheme, found in many books related to differential equations. This method is used here to allow for larger timesteps than those that could be used in an explicit scheme (e.g. Euler forward). Discretising equation (2.60) through the backward Euler implicit scheme, and assuming the value of C to be constant in time during Δt inside the control volume yields,

$$\Omega_{CV_i}\frac{C_i^{n+1}-C_i^n}{\Delta t}+\int_{\Gamma_{CVi}} C_\Gamma^{n+1}\mathbf{u}_\Gamma\cdot\hat{\mathbf{n}}\,d\Gamma_{CVi}=0 \tag{2.61}$$

where superscripts n and $n+1$ denote actual and future time levels, respectively. In a similar manner to the one dimensional case in Section 2.3, the success of the advection of the sharp interface relies on the selection of the correct control volume face value C_Γ. Assuming an iterative solution method within a time step, given by,

$$C_\Gamma^{n+1}\big|_k = C_\Gamma^L\big|_k + (C_{\alpha,k+1}-C_{\alpha,k})\;\; k=1,2,\ldots m_{iter} \tag{2.62}$$

where k is the iterative solution step, m_{iter} is the number of iterative steps, C_Γ^L is the limited face value, and $C_{\alpha,k}$ is a guess at the value of $C_{\alpha,k+1}$. Index α represents the choice of position value, given by the central position value i if the flow is outgoing, or by the downwind position D value, if the flow is incoming. The value of C_α may be approximated by a low order guess, given by,

$$C_\alpha=\begin{cases} C_i^{n+1} & \mathbf{u}\cdot\hat{\mathbf{n}}>0 \\ C_D^{n+1} & \mathbf{u}\cdot\hat{\mathbf{n}}<0 \end{cases} \tag{2.63}$$

where C_D is value of concentration at the node directly downwind of node i as seen in Figure 2.15. At each iterative step k of equation (2.62), the limited face value C_Γ^L at the face shared by control volume i and control volume D, can be obtained analogously to Leonard's Hyper-C scheme recalling equation (2.17),

$$\overline{C_\Gamma^L}=min\left(1,\frac{\bar{C}_i}{\gamma}\right) \tag{2.64}$$

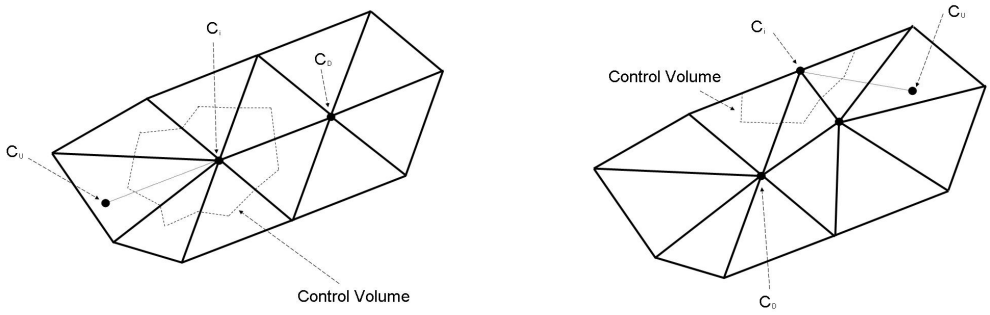
which, in terms of unnormalised variables yields,

$$C_\Gamma^L=min\left(C_D,\frac{C_i-C_U}{\gamma}+C_U\right) \tag{2.65}$$

where C_U is the upwind value, as explained earlier in Section 2.3, and γ is the Courant number defined in this case by

$$\gamma = \hat{\mathbf{u}} \cdot \mathbf{n} \frac{\Delta t}{\Delta x} \tag{2.66}$$

where Δx is the direct distance from node i to node D. Given the unstructured nature of the discretised domain, the definition of the upwind value C_U is not straightforward. The value position is extrapolated by extending the line connecting nodes i and D by Δx in the upwind direction (see Figure 2.15(a)) and interpolating accordingly. In situations where the extrapolated point is outside Ω then the line is reflected at the boundary Γ as seen in Figure 2.15 (b).



(a) Internal nodes (b) Boundary nodes

Figure 2.15: Control volume configuration and naming conventions, where the scenario for nodes (a) in the interior of the domain, and (b) in the boundary of the domain are shown. A two dimensional version is show for clarity purposes, however it is straightforwardly extensible to three dimensions.

If equation (2.62) is substituted into equation (2.61) then,

$$\Omega_{CV_i} \frac{C_i^{n+1} - C_i^n}{\Delta t} + \int_{\Gamma_{CVi}} C_\Gamma^L\big|_k + (C_{\alpha,k+1} - C_{\alpha,k}) \mathbf{u}_\Gamma \cdot \hat{\mathbf{n}} \, d\Gamma_{CVi} = 0 \tag{2.67}$$

which, after some rearrangement yields,

$$\begin{aligned} &\Omega_{CV_i} C_i^{n+1} + \Delta t \int_{\Gamma_{CVi}} C_{\alpha,k+1} \mathbf{u}_\Gamma \cdot \hat{\mathbf{n}} \, d\Gamma_{CVi} \\ &= \Omega_{CV_i} C_i^n + \Delta t \int_{\Gamma_{CVi}} \left(C_{\alpha,k} - C_\Gamma^L\big|_k \right) \mathbf{u}_\Gamma \cdot \hat{\mathbf{n}} \, d\Gamma_{CVi} \end{aligned} \tag{2.68}$$

A matrix system may be constructed from equation (2.68) of the type,

$$\left(\mathbf{M}_{CV} + \mathbf{B}|_k \right) \mathbf{C}^{n+1}\big|_{k+1} = R|_k \tag{2.69}$$

where $\mathbf{M}_{CV}$ is a diagonal matrix containing the individual volumes of the control volumes, $\mathbf{B}|_k$ is the k^{th} iterate of the advective flux matrix, $R|_k$ is made up of the right hand side components of equation (2.68), and $\mathbf{C}^{n+1}\big|_{k+1}$ is the array of values of concentration which is solved for during

the k iterations. Matrix $\mathbf{B}$ can be assembled by summing the flux contributions related to control volumes j which are connected to control volume i. Effectively, if control volume i is fluxing material to connected control volume j, then $C_{\alpha,k+1} = C_i^{n+1}$ and the corresponding positive contribution is added to the diagonal of matrix $\mathbf{B}$ for node i (i.e. $\mathbf{B}_{i,i}$). An equal contribution of opposite sign is added to matrix element $\mathbf{B}_{j,i}$. On the other hand, if the flow is the opposite way, then $C_{\alpha,k+1} = C_j^{n+1}$, $\mathbf{B}_{j,i}$ is added to the diagonal $\mathbf{B}_{j,j}$, and $B_{i,j} = -B_{j,i}$.

The matrix system in (2.69) is solved iteratively, and as convergence is reached, $\mathbf{C}^{n+1}\big|_{k+1} \rightarrow \mathbf{C}^{n+1}\big|_k$and so $\mathbf{C}_\Gamma^{n+1} \rightarrow C_\Gamma^L$. The latter implies that the solution reached is of a compressive nature observed in Hyper-C.

2.4.7 Solution mechanism

The overall numerical model description comes to a close with the following description of the solution mechanism, which was designed to combine both finite element and control volume finite element discretisations. The mechanism used here for the solution of the incompressible Navier- Stokes equations was also presented by Ford et al. (2004), and by Piggott et al. (2008) , and is here combined with a Hyper-C limited control volume finite element formulation. Solution for velocity and pressure is obtained through a projection method, in which they are both solved separately. Starting with equation (2.50) , and using implicit time stepping, yields,

$$\mathbf{M}\frac{\mathbf{U}^{n+1}-\mathbf{U}^n}{\Delta t} + \mathbf{A}\mathbf{U}^{n+1} + \mathbf{K}\mathbf{U}^{n+1} + \mathbf{Q}\mathbf{p}^{n+1} = \mathbf{F}^{n+1} \tag{2.70}$$

which, after some rearrangement of terms,

$$\left(\mathbf{M} + \Delta t\left(\mathbf{A}+\mathbf{K}\right)\right)\mathbf{U}^{n+1} + \Delta t\mathbf{Q}\mathbf{p}^{n+1} = \mathbf{M}\mathbf{U}^n + \Delta t\mathbf{F}^{n+1} \tag{2.71}$$

where it should be noted that $\mathbf{F}^{n+1}$contains the buoyancy terms which are calculated through the use of $\mathbf{C}^{n+1}$ obtained through the method shown in section 2.4.6. Equation (2.71) is solved to obtain an initial approximation $\mathbf{U}_*^{n+1}$ to $\mathbf{U}^{n+1}$using an initial guess for the pressure $\mathbf{p}_*^{n+1} = \mathbf{p}^n$. This initial approximation of velocity is used to solve for the pressure change through a discretised elliptic problem [8] given by,

$$\mathbf{Q}^T\mathbf{M}^{-1}\mathbf{Q}\left(\mathbf{p}^{n+1} - \mathbf{p}_*^{n+1}\right) = -\frac{\mathbf{Q}^T\mathbf{U}_*^{n+1}}{\Delta t} \tag{2.72}$$

Finally, $\mathbf{U}^{n+1}$can be obtained from the Helmholtz decomposition derived equation (see Piggott et al. (2008)),

$$\mathbf{M}\mathbf{U}^{n+1} = \mathbf{M}\mathbf{U}_*^{n+1} + \Delta t\mathbf{Q}\left(\mathbf{p}^{n+1} - \mathbf{p}_*^{n+1}\right) \tag{2.73}$$

It should be noted, that at the very beginning of the simulation, there is no information on

[8] This elliptic problem is obtained from the Helmholtz decomposition of the velocity approximation, to which the continuity equation is the applied. See Ford et al. (2004) for more details.

any previous timestep pressure, and hence an estimate must be obtained to start the process. This initial pressure value is obtained from solving the pressure Poisson equation (PPE), as described by Gresho and Sani (2000), in which the divergence of the momentum equations is taken and the continuity equation is applied. Details of this procedure and others, such as an increase in memory efficiency through mass matrix lumping used for $\mathbf{M}$ and its implications, is provided by Ford et al. (2004).

Summarising details over the whole scheme can be seen in Algorithm 1. It is important to restate at this point that the implemented method of CVFEM-Hyper-C combined with the finite element discretisation and solution mechanism satisfies the characteristics of being efficient, easy to implement, and achieves total conservation.

Algorithm 1 Solution algorithm for incompressible Navier Stokes equations.

loop {timestep}
  loop {Picard iterations (k)}
    Obtain solution for $\mathbf{C}_k^{n+1}$by solving equation (2.69) in m_{iter} iterations
    Use $\mathbf{C}_k^{n+1}$to calculate f^{n+1}
    Solve equation (2.71) using $\mathbf{p}_*^{n+1} = \mathbf{p}^n$ or the solution of the PPE equation if on the very first timestep.
    Use the intermediate solution $\mathbf{U}_*^{n+1}$ to calculate the pressure correction through equation (2.72).
    Use the pressure correction obtained in previous step to obtain $\mathbf{U}_k^{n+1}$ from equation (2.73).
  end loop
end loop

2.5 Analytical validation tests

The following are typical tests intended to confirm the effectiveness of interface preserving methods. The tests consist of different shapes, defined by an initial condition on the concentration scalar field, which are transported by a prescribed velocity field. The common objective is to recover the initial shape as accurately as possible. The solution error for the concentration scalar field C between the initial shape and the final shape can be calculated by the following relation of L_2 norms over the nodal values,

$$E = \frac{\sum_{i=1}^{n_n} \left\| C^e - C^0 \right\|}{\sum_{i=1}^{n_n} C^0} \tag{2.74}$$

where C^0 are the scalar values of the initial shape, and C^e are the values at the end of the simulation. Cases below depict the Hyper-C implementation in action, being compared to other advection methods found in the literature.

2.5.1 Zalesak's problem

In this test, a slotted circle of fluid rotates in a uniform velocity field (see Zalesak, 1979). In order to compare with other results found in the literature, namely López et al. (2004), Rudman (1997), and Hirt and Nichols (1981), the domain has been designed as a two dimensional square of unit edges. The mesh used in this case is uniformly unstructured and has a reference length of 0.01[9], a total amount of nodes equal to 11621, and 22840 triangular elements. The centre of rotation of the velocity field is located at the very middle of the domain, and the slotted circle shape is located at coordinates $(0.5, 0.75)$ within the square domain. As defined by Rudman (1997), the flow angular velocity is equal to 0.5.

Initial conditions are presented in Figure 2.16, and results are presented in Figure 2.17 for the simulation of one full revolution of the slotted circle shape. Figure 2.18 shows a result comparison between Hyper-C and other methods in the literature which were mentioned in Section (2.1). Table 2.1 presents a summary of the errors for the methods compared. These numbers were obtained using equation (2.74)..

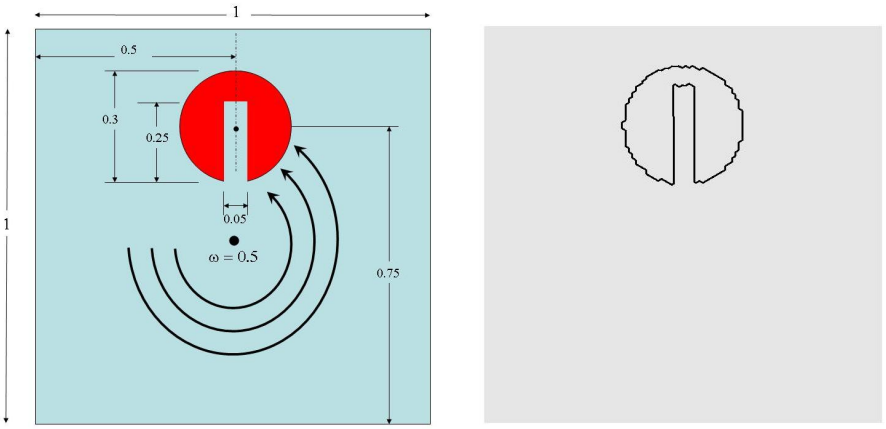


Figure 2.16: Initial Conditions for Zalesak's problem given by (a) a schematic, and (b) the simulation output.



Figure 2.17: Sequence of results for Zalesak's problem at (a) $t = 0.798$ seconds, (b) $t = 1.60$ seconds, (c) $t = 2.40$ seconds, and (d) $t = 3.15$ seconds.

[9]This means that the mesh generator will subdivide the domain into elements of a size that should remain close to that length

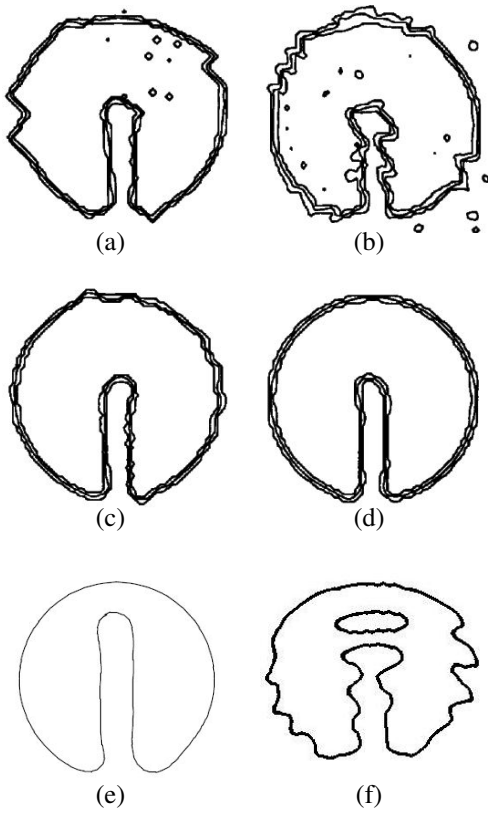


Figure 2.18: Comparison of results for Zalesak's problem after one full rotation using (a) SLIC, (b) Hirt-Nichols, (c) FCT-VOF, (d) Youngs, (e) LVIRA, and (f) this thesis's Hyper-C methods. LVIRA's results were obtained from Pilliod and Pucket (2004), all others except Hyper-C were obtained from Rudman (1997)

From values in Table 2.1 and the resulting shapes in Figure 2.18, it is evident that Youngs method still performs best for this problem, however the method presented in this thesis fares reasonably well when compared to the rest. It is useful to note that some of the main disadvantages of the other methods shown here are the fact that they are not easily extended to three dimensions, and that they are difficult to implement in unstructured meshes.

As it happens in nature, it is of crucial importance that the method satisfies mass conservation laws. Figure 2.19 shows a graph of the calculated mass of one of the fluids throughout the simulation compared to the initial one. Mass conservation, keeping in mind that density is constant for each fluid and that values of C range from 0 to 1, is calculated by,

$$\%\, mass = \frac{\int_\Omega C^n \, d\Omega}{\int_\Omega C^0 \, d\Omega} \times 100 \tag{2.75}$$

where C^0 is the initial concentration field, and $C^n$is the concentration field at the nth time level.

Evidently from these results, and as will be shown again in other simulations presented herein, the method performs extremely well in this task, satisfying yet another of the established goals which is conservation. After one full revolution material quantities are conserved well within 1.0×10^{-6} percent.

Method	Error
SLIC	$8.38 \cdot 10^{-2}$
Hirt-Nichols	$9.62 \cdot 10^{-2}$
FCT-VOF	$3.29 \cdot 10^{-2}$
Youngs	$1.09 \cdot 10^{-2}$
LVIRA	$1.7 \cdot 10^{-2}$
This thesis (Hyper-C)	$3.17 \cdot 10^{-1}$

Table 2.1: Error for Zalesak's test after one full rotation. Data obtained from Rudman (1997) and López et al. (2004).

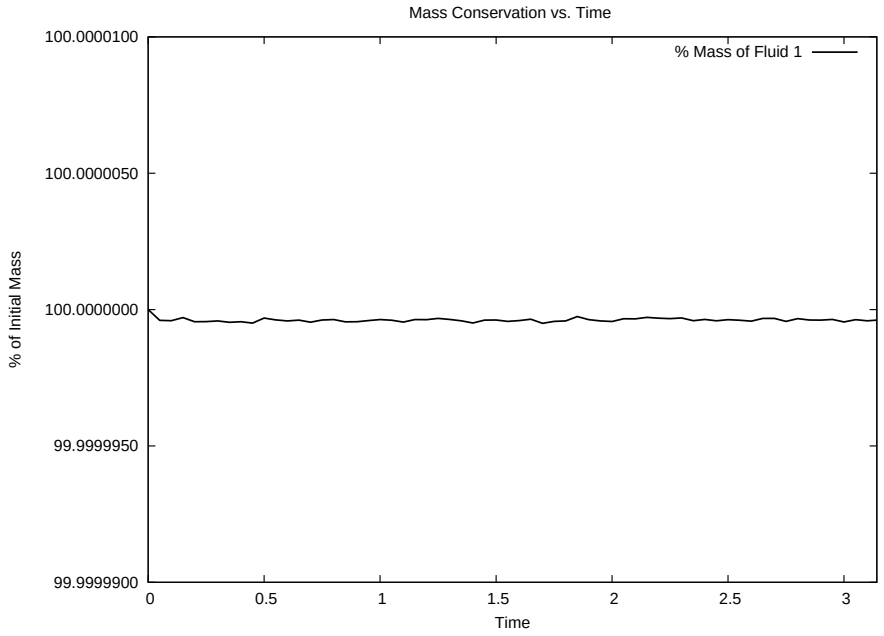


Figure 2.19: Conservation graph for the Zalesak problem using Hyper-C.

2.5.2 Shearing flow

The simple rotations are not sufficient tests due to the lack of topological change in the solution. The flow is divergence free, and furthermore each component of the divergence is null as well. The latter point leads to the fact that the shape does not deform as it is advected. In reality, natural flows tend to shear, stretch, break, break and merge. Therefore, a substantial element missing from being tested is the existence of fluid shear.

The two dimensional test domain is a square with opposite corners in $(0,0)$, and (π, π). A rotational velocity field $\mathbf{u}$ is defined to represent a single vortex, which is given by,

$$\mathbf{u} = \begin{pmatrix} sin(x) \cdot cos(y) \\ -cos(x) \cdot sin(y) \end{pmatrix} \tag{2.76}$$

and applied in the whole domain. The direction of the velocity field is later reversed at time

$t = T$. A circle shape of radius $\frac{\pi}{5}$ is located at coordinates $(\frac{\pi}{2}, \frac{\pi}{4})$ in the domain. A perfect advection scheme should return the initial conditions after the velocity field is reversed. Initial configuration for the problem is shown in Figure 2.20, and the mesh used in this case is uniformly unstructured and has a reference length of 0.03, a total amount of 25018 elements, and 12720 nodes.

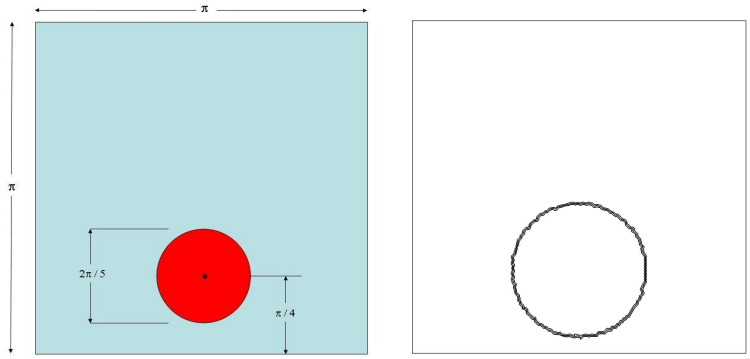


Figure 2.20: Initial conditions for the shearing flow problem given by (a) a schematic, and (b) simulation output.

The problem was run for several values of T, as shown in Table 2.2. Figure 2.21 shows results for $T = 7.86\,s$ and $T = 15.72\,s$.

T	SLIC	Hirt-Nichols	FCT-VOF	Youngs	Hyper-C
1.96	$2.72 \cdot 10^{-2}$	$3.24 \cdot 10^{-2}$	$1.94 \cdot 10^{-2}$	$2.61 \cdot 10^{-3}$	$3.68 \cdot 10^{-2}$
3.93	$3.30 \cdot 10^{-2}$	$4.00 \cdot 10^{-2}$	$2.35 \cdot 10^{-2}$	$5.12 \cdot 10^{-3}$	$4.86 \cdot 10^{-2}$
7.86	$4.59 \cdot 10^{-2}$	$6.66 \cdot 10^{-2}$	$3.14 \cdot 10^{-2}$	$8.60 \cdot 10^{-3}$	$6.99 \cdot 10^{-2}$
15.72	$9.02 \cdot 10^{-2}$	$1.09 \cdot 10^{-1}$	$1.44 \cdot 10^{-1}$	$3.85 \cdot 10^{-2}$	$1.30 \cdot 10^{-1}$

Table 2.2: Error calculation for shearing flow test case.

As was presented for the Zalesak problem, Figure 2.22 shows the mass conservation calculation throughout the calculation, for the toughest case in this respect, $T = 15.71$.

It is apparent that after continuous shearing flow all methods eventually begin to breakdown a certain point. Hyper-C performed extremely well in this case, specially for the longest case, in which it managed to keep its error below several of the other compared methods. An interesting observation can be made for some of the VOF methods in this simulation, where it seems that some material is left lying around in small lumps. As can be observed also from the results, Hyper-C does not exhibit this unwanted behaviour.

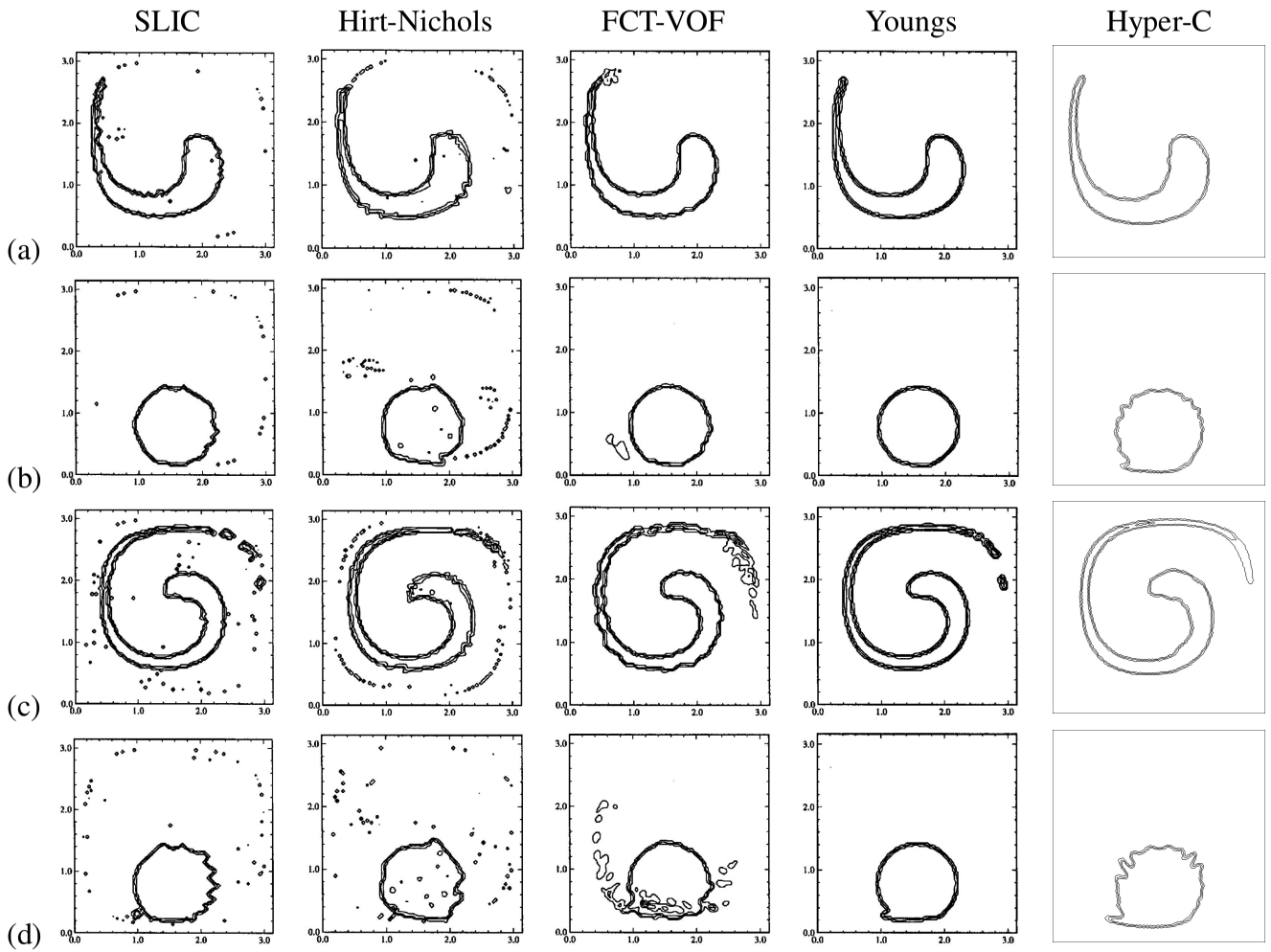


Figure 2.21: Result comparison between different methods (i.e. obtained from Rudman (1997)) including Hyper-C, for the shearing flow problem. Rows (a) and (b) show the advection process at reversal time and at the end of the simulation, respectively, for $T = 8.86$. Rows (c) and (d) analogously show the results for $T = 15.72$.

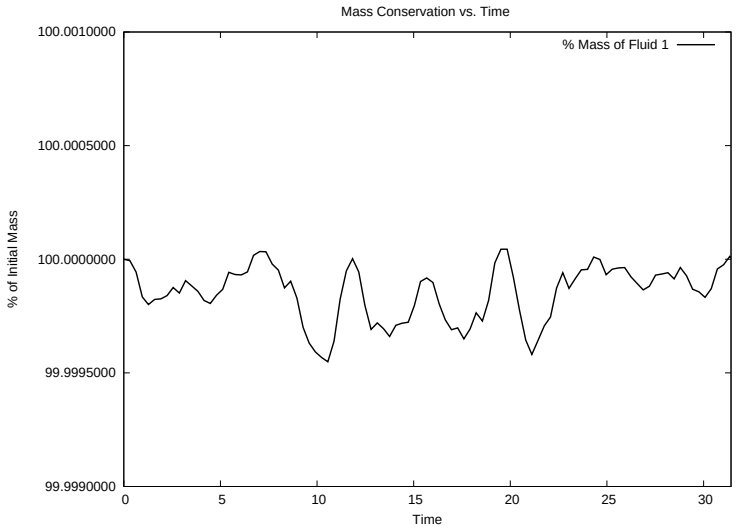


Figure 2.22: Conservation graph for the shearing problem using Hyper-C

2.6 Experimental validation tests

These are tests which have a more realistic value, further proving the method's worth as an interface tracking scheme. In these simulations a more realistic flow is represented through the solution of the incompressible Navier Stokes equations with an extra scalar field advection equation solved using the Hyper-C method, as described in Section 2.4.

2.6.1 Collapsing water column (2D)

With regard to testing numerical methods that deal with free surfaces and interface tracking, the collapsing water column is one of the most frequently used naturally meaningful simulations found in the literature. Hirt and Nichols (1981) performed a comparison of a hydrodynamics code SOLA-VOF which is based on the volume of fluid methodology for tracking interfaces, with experimental measurements obtained from Martin and Moyce (1952). Similarly Koshizuka et al. (1995a) compared a moving particle semi-implicit method (MPS) for viscous incompressible fluids, to experiments he carried out by Koshizuka et al. (1995b) and Koshizuka and Oka (1996), as well as those by Martin and Moyce (1952). Kohno and Tanahashi (2004) tested a FEM method combined with a level set scheme reinitialised by a pseudo-particle method comparing them to Koshizuka's experiments, while Greaves (2006) compared these same experiments to a VOF approach coupled to adaptive quadtree grids.

The water column is initially located in a tank, as shown in Figure 2.23, and the parameters and measurements of the tank are given in Table 2.3. A free slip[10] boundary condition was

[10]A free slip condition implies that the only requisite for a given boundary is that there exists no velocity component in the direction normal to that boundary. This also implies that there is no mass-flow across the boundary. Tangential flow, however, is allowed and hence the name "slip". It is normally applied in cases where

applied on all walls.

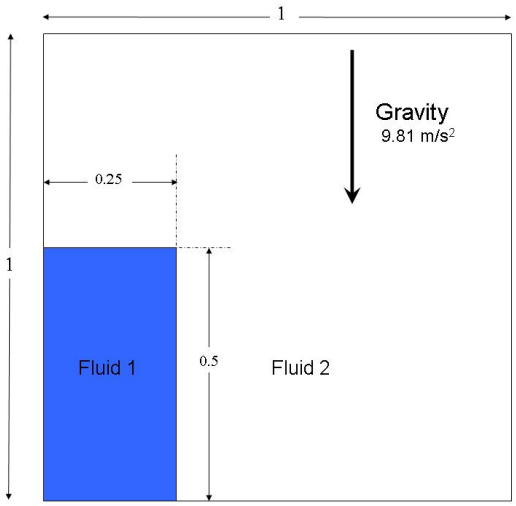


Figure 2.23: Initial configuration for collapsing water column problem. Measurements are in meters [m].

Parameter	Value
Density, Fluid 1	$1000.0\ \frac{Kg}{m^3}$
Density, Fluid 2	$1.205\ \frac{Kg}{m^3}$
Viscosity, Fluid 1	$1.0 \times 10^{-3}\ \frac{Ns}{m^2}$
Viscosity, Fluid 2	$1.7 \times 10^{-5}\ \frac{Ns}{m^2}$
Gravity	$9.81\ \frac{m}{s^2}$
Simulation-time	$0.0s \Rightarrow 2s$
Timestep size	$0.0004\,s$
Min. elem. size	$0.005\,m$
Max. elem. size	$0.05\,m$

Table 2.3: Calculation parameters for the water collapse simulation

Much of the result comparison and validation can be done graphically, through careful observation of instantaneous photographs of experiments with numerical results at equal time frames. Such results are presented in Figures 2.24, 2.25, and 2.26, where an adaptive mesh Hyper-C solution is presented compared to physical experiments, MPS, and VOF, respectively. A normalised variable is used to describe time t in these frames, calculated by $\tau = t\sqrt{\frac{g}{a}}$, where a is the initial width of the water column, and g is the gravitational acceleration.

Another way of verifying the validity of results is through the comparison with measurements of the advancing front of the water column. The distance x from the leading edge to the left wall is normalised by dividing with the initial width of the water column a, whereas time t

the inertia forces are much greater than the viscous forces at the mentioned boundary. See, for instance, Batchelor (2000) for more details.

is normalised by calculating $\tau = t\sqrt{\frac{2g}{a}}$, where the coefficient of 2 appears to take into account the shape of the column, as described by Martin and Moyce (1952). A graph of these results is shown in Figure 2.30. Zhao et al. (2002a) compared the descending water height in a similar manner, by normalising it with the initial column height b. Experiments for this were also carried out by Martin and Moyce (1952) for different column sizes. Results for this comparison are shown in Figure 2.31.

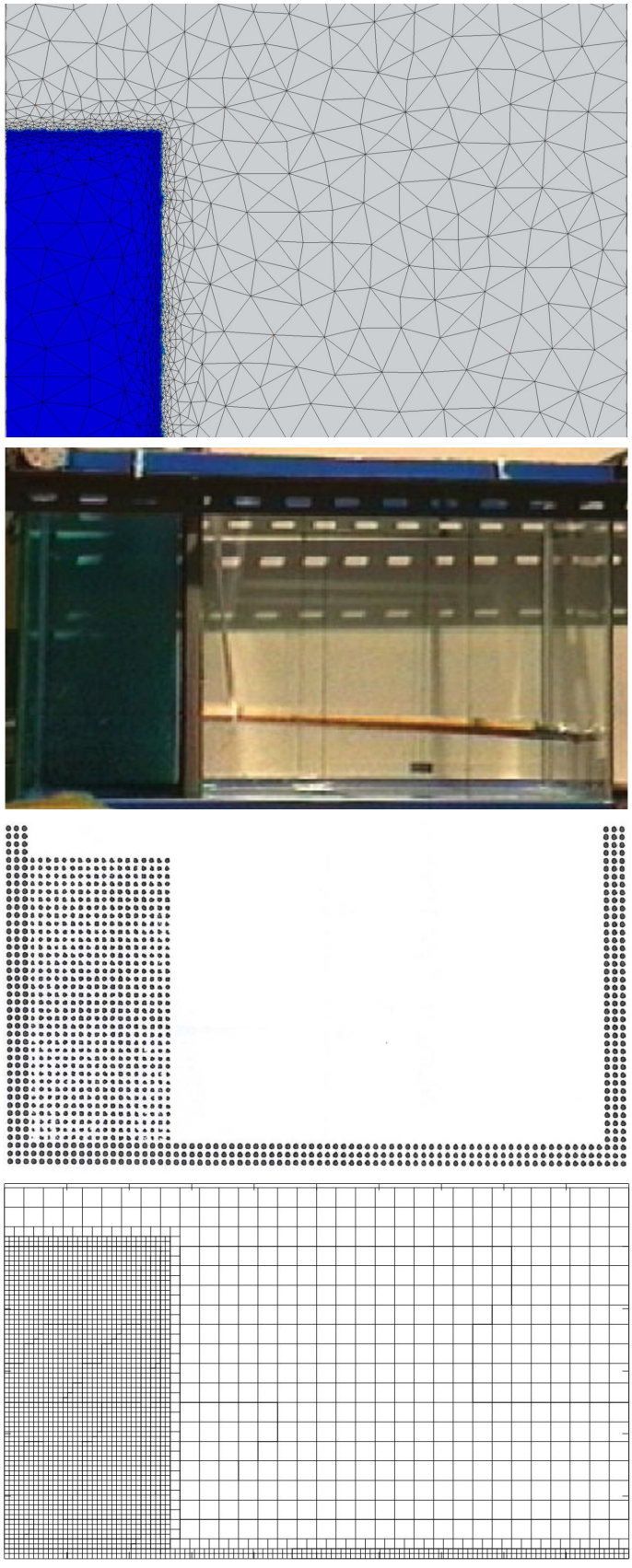
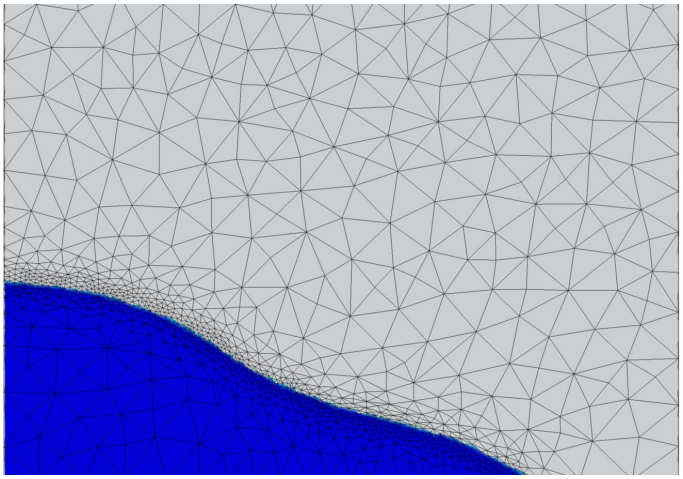
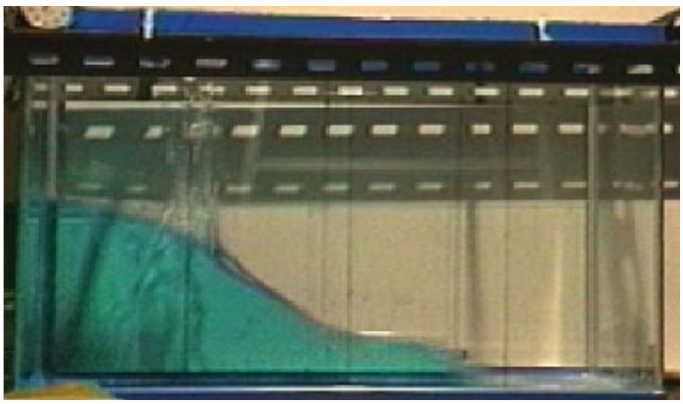


Figure 2.24: Comparison of results for the collapsing water column problem using (a) Hyper-C with (b) experiments by Koshizuka et al. (1995b), (c) a moving particle semi-implicit method (MPS) from Koshizuka et al. (1995a), and (d) a VOF combined with adaptive grids by Greaves (2006). Results correspond to timeframe $\tau = 0$.

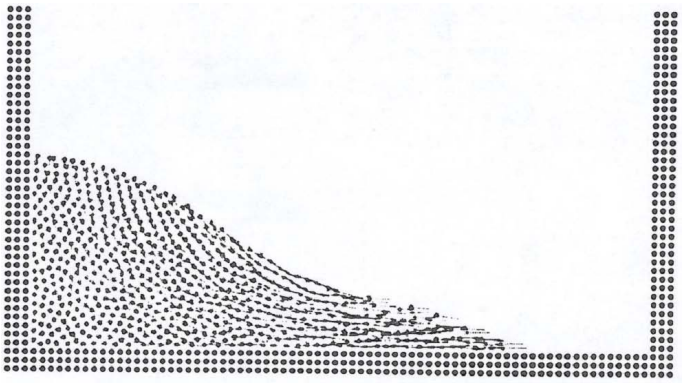
(a)



(b)



(c)



(d)

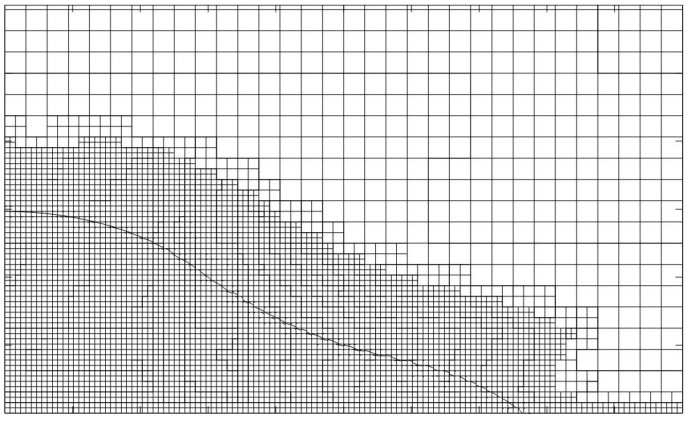
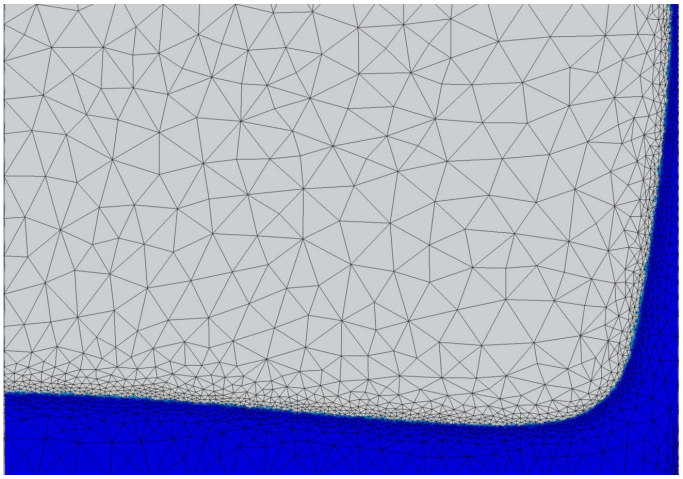
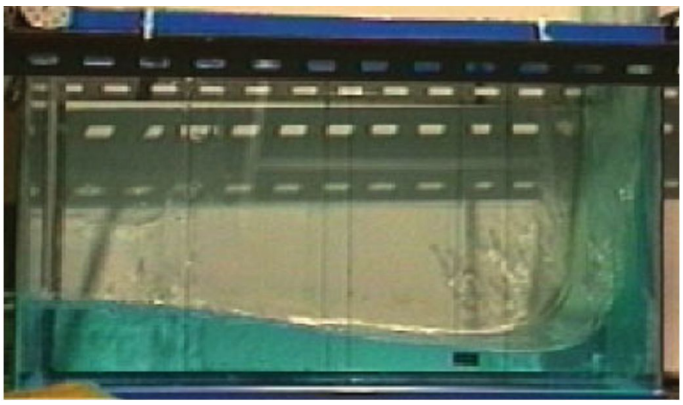


Figure 2.25: Continued from Figure 2.24, with frames for $\tau = 1.617$.

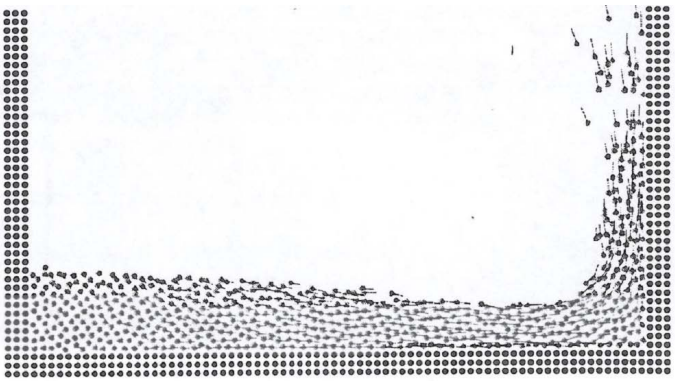
(a)



(b)



(c)



(d)

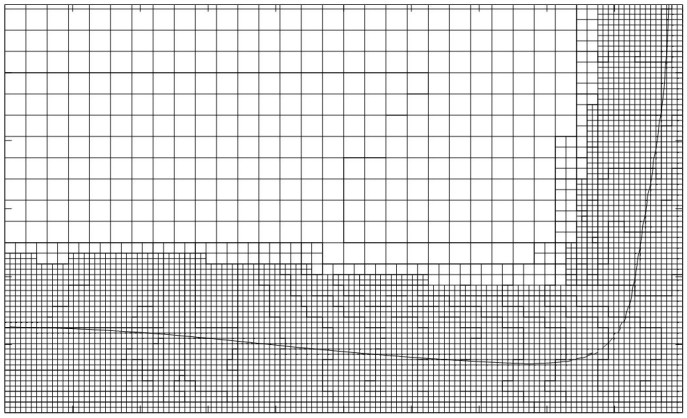
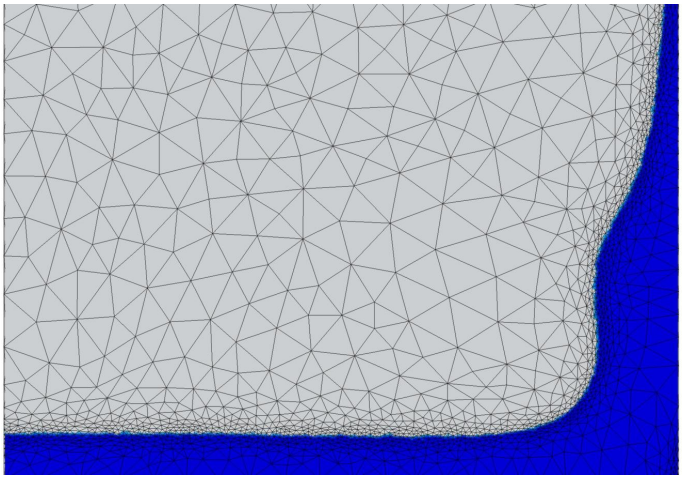
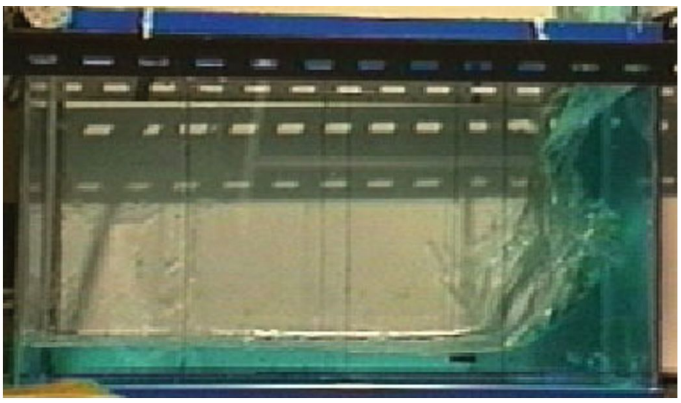


Figure 2.26: Continued from Figure 2.24, with frames for $\tau = 3.233$.

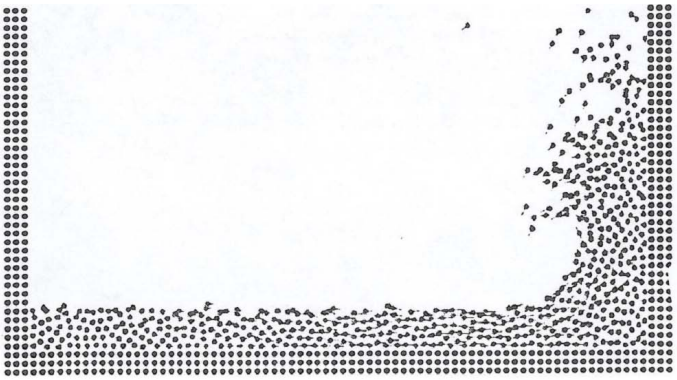
(a)



(b)



(c)



(d)

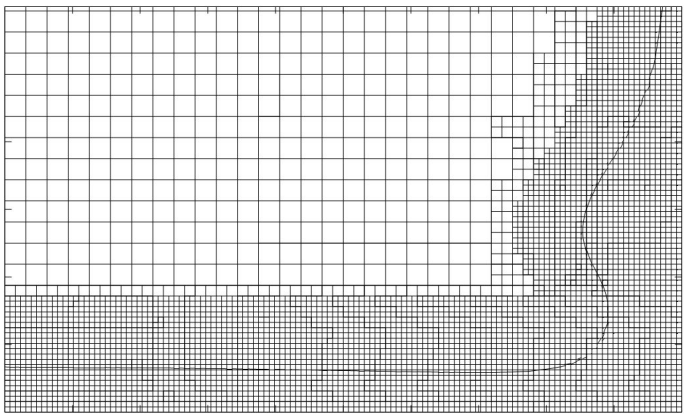
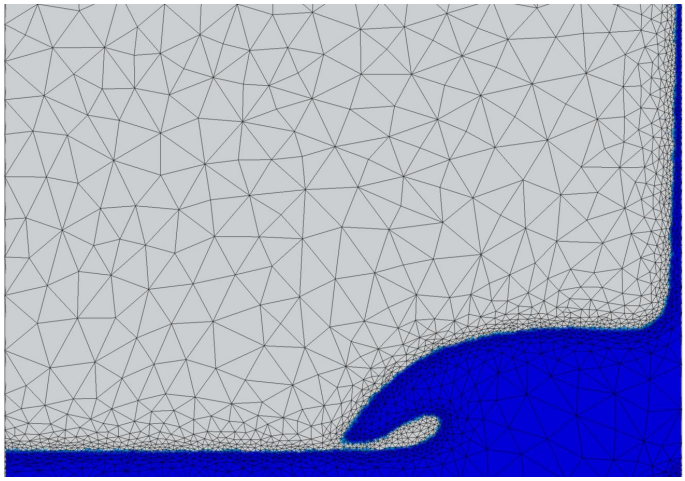
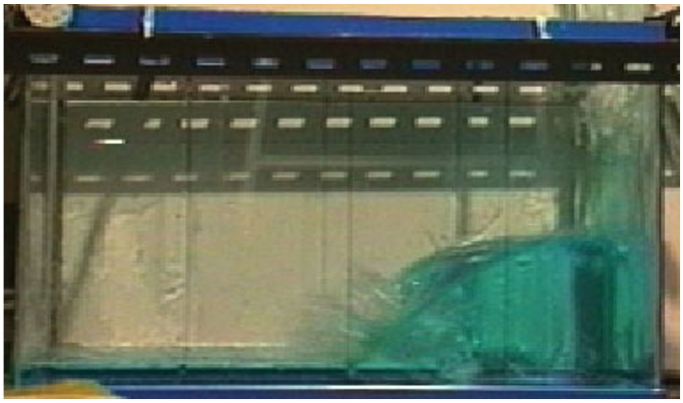


Figure 2.27: Continued from Figure 2.24, with frames for $\tau = 4.850$.

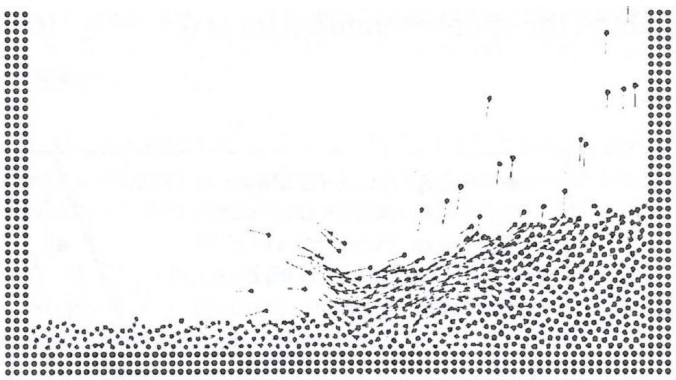
(a)



(b)



(c)



(d)

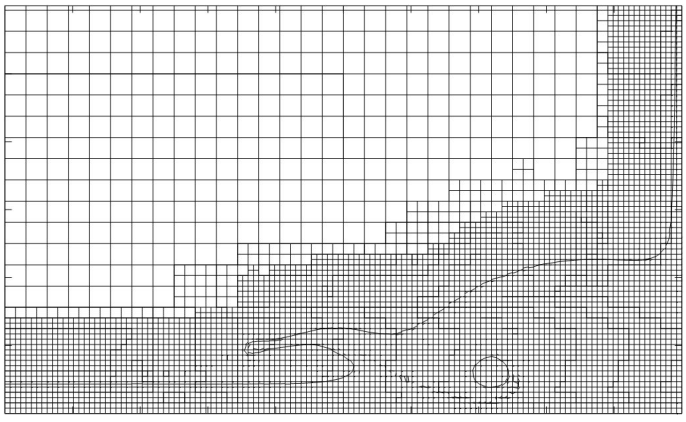
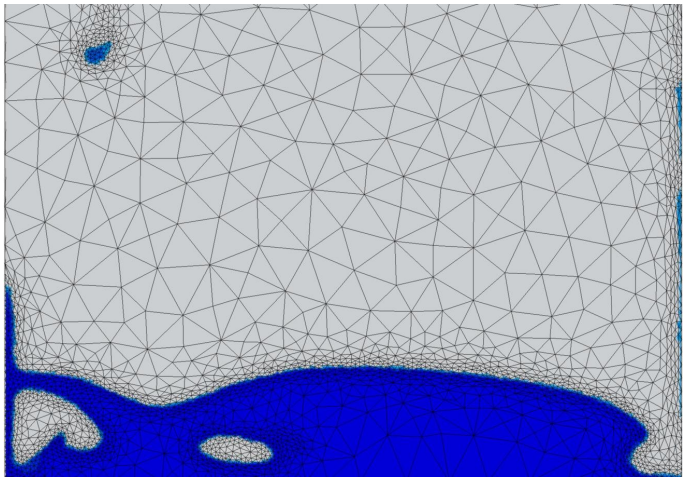
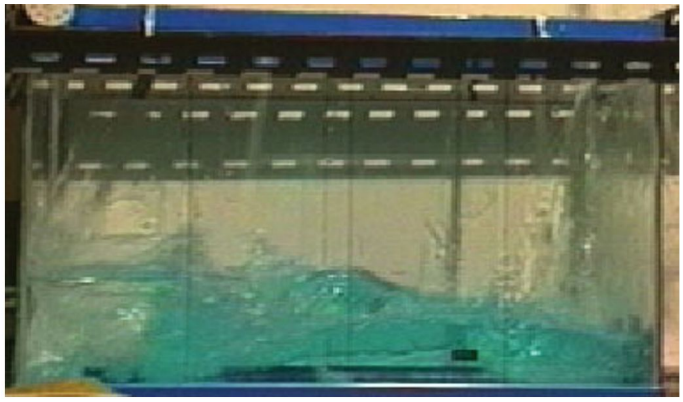


Figure 2.28: Continued from Figure 2.25, with frames for $\tau = 6.466$.

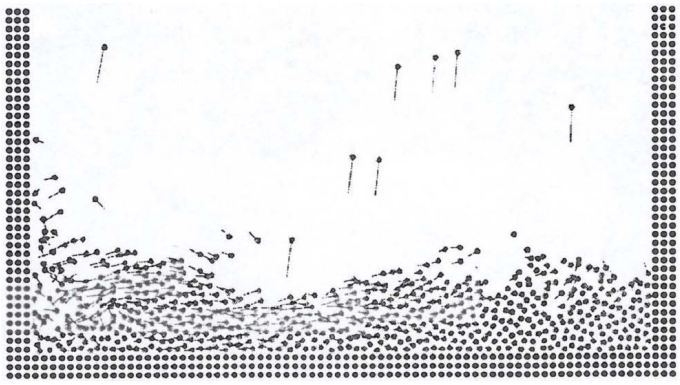
(a)



(b)



(c)



(d)

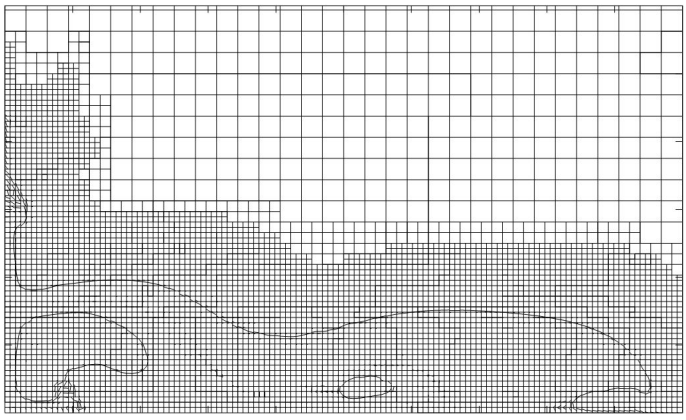


Figure 2.29: Continued from Figure 2.25, with frames for $\tau = 8.083$.

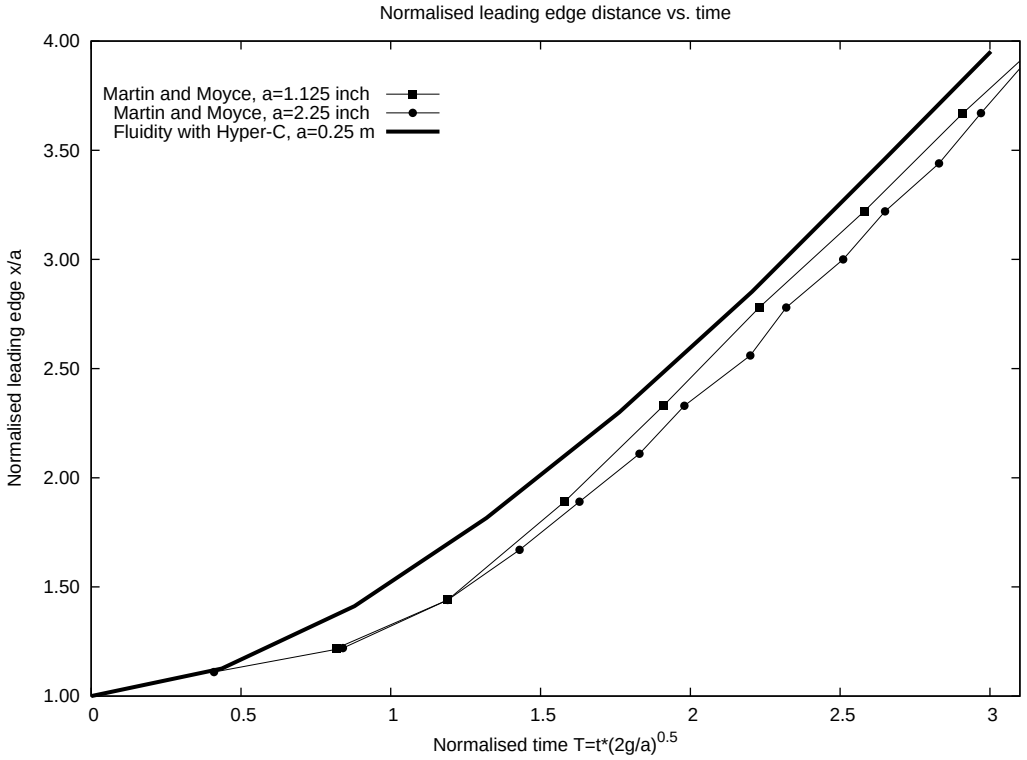


Figure 2.30: Graph of advancing water front vs. normalised time.

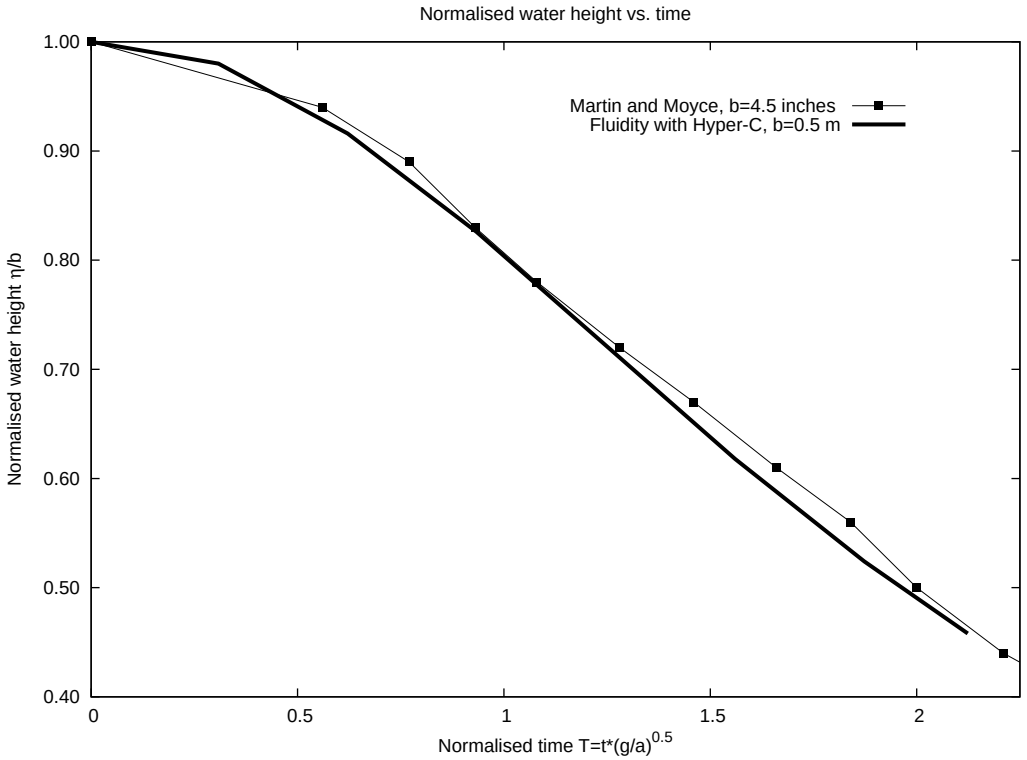


Figure 2.31: Graph of water height vs. normalised time.

From Figures 2.30 and 2.31 it can be observed that Hyper-C performs extremely well when compared to the experiments. It must be noted, though, that the determination of the points that define the leading edge or the water height are difficult to obtain considering the influence of the resolution. The method implemented in this thesis, however, possesses the advantage of increased resolution around the interfaces, and so a relatively high degree of precision when measuring is practically guaranteed.

The results presented for Hyper-C in Figure 2.24 were obtained using an adaptive mesh optimisation technique mentioned in Chapter 1 as being one of Fluidity's main characteristics. Although significant improvement is made on interface definition, the use of adaptive mesh optimisation has effects on the overall conservation in the problem at hand, and hence a graph of this evolution is presented in Figure 2.32 compared the same exact simulation run without adaptivity. Conservation results show that the effects of solution mapping after adaptivity account for a 5% mass variation for this problem, while the non-adaptive test is completely mass conservative. This mass augmentation effect is caused by non-conservative grid to grid interpolation used in adaptivity. Conservative interpolation methods exist, but are generally time inefficient and complex. Further research of this problem is a matter of current research and development within AMCG.

This non-conservation of mass is further proof that a method should be devised to prolong the much needed mesh optimisation steps, thus reducing the amount of mass loss in the simulation. The development of such a method is also a main part of this thesis and details are provided in Chapter 3.

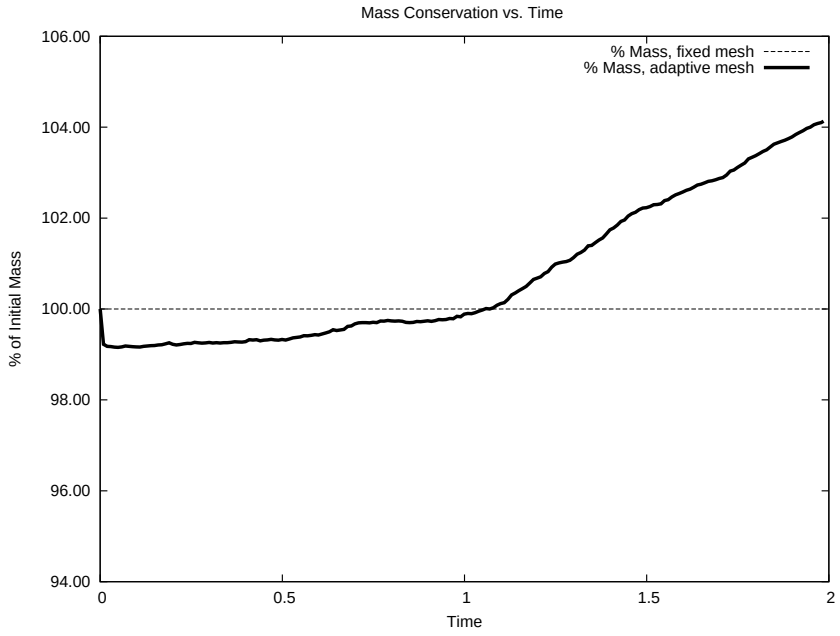


Figure 2.32: Mass conservation graph for 2D water collapse problem, showing results from adaptive and non-adaptive tests.

2.6.2 Collapsing water column with obstruction (2D)

A more extreme version of the previous test, is the case of the same water tank, but with an obstruction in the way of the collapsing water column. This problem was also studied by several authors, namely Koshizuka et al. (1995b), Ubbink (1997), Andrillon and Alessandrini (2004), and Greaves (2006). The setup is the same as the one presented in Section 2.6.1, with the difference of the existence of a rectangular obstruction with its leading edge set halfway along the tank, 0.04 units wide, and 0.08 units high. Figure 2.34 depicts a comparison of adaptive mesh Hyper-C results with experiments from Koshizuka et al. (1995b) and numerical results using VOF by Greaves (2006). Normalised time is calculated in a similar manner to the water collapse test presented in section 2.6.1, using $\tau = t\sqrt{\frac{g}{a}}$, where a is the width of the initial water column.

Figure 2.34 shows that Hyper-C behaves as well as the VOF method, and even better in several aspects. At normalised time $\tau = 0.809$, both VOF and Hyper-C results seem to predict well when compared to experimental results. Hyper-C performs well at predicting the the water tongue at times $\tau = 1.607$ and $\tau = 2.426$, as does VOF, however both methods are not able to capture some spray detail that is observed around the main tongue. At time $\tau = 3.233$, VOF resolves well the secondary tongue that appears right above the obstacle, which seems to be missing in the Hyper-C results. However, in the last frame for $\tau = 4.043$, the secondary tongue has grown considerably and VOF predicts that this tongue has split into several smaller ones. The shape of the secondary tongue is predicted considerably better by Hyper-C at this point. This particular test demonstrates that the combination of Hyper-C and AMO is able to resolve interfaces and model their evolution with a high level of detail. For completeness, a conservation graph is also presented for this test in Figure 2.33, showing again that mass conservation is still kept within 5% difference, and that it is caused by grid to grid interpolation error in the adaptive process.

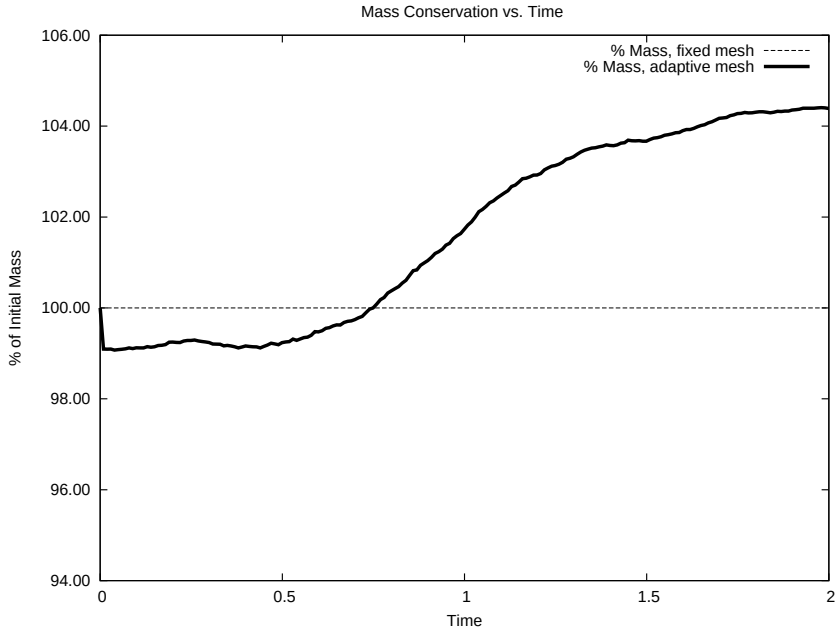


Figure 2.33: Mass conservation graph for 2D water collapse with obstruction problem, showing results from adaptive and non-adaptive tests.

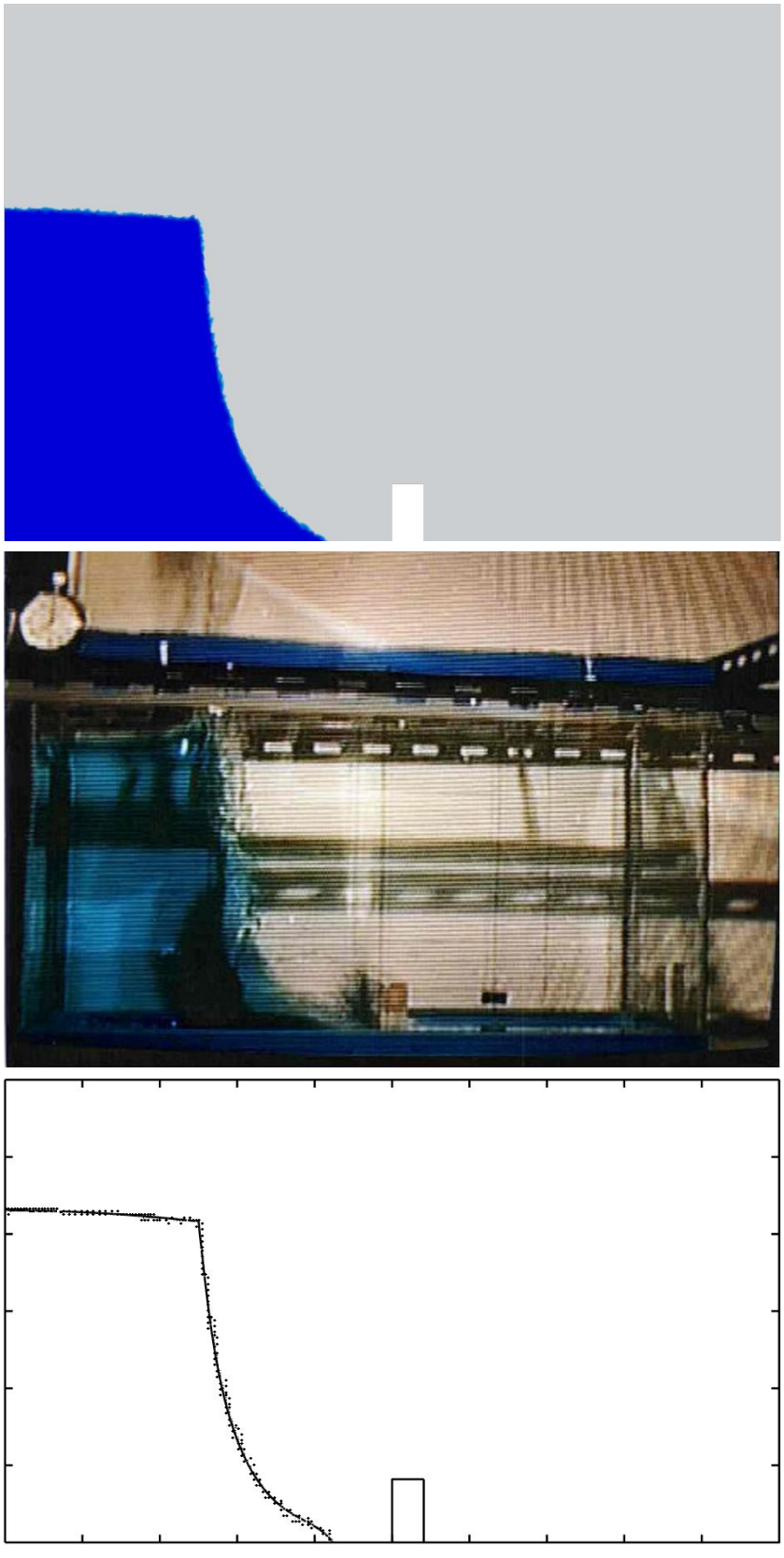


Figure 2.34: Comparison of results for the obstructed collapsing water column using Hyper-C (a), experiments by Koshizuka et al. (1995b) (b), the VOF approach from Greaves (2006)(c) . Time frames are at a normalised value of $\tau = 0.809$.

(a)



(b)



(c)

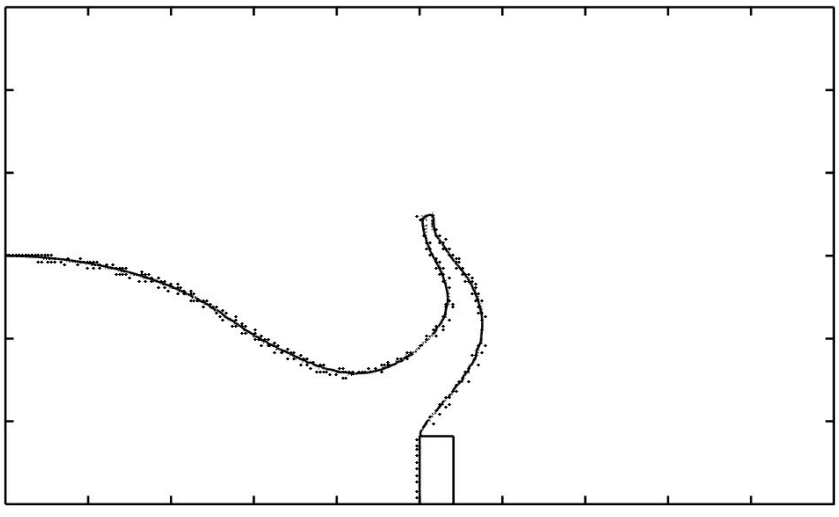


Figure 2.35: Continued from Figure 2.34, with frames for $\tau = 1.607$.

(a)



(b)



(c)

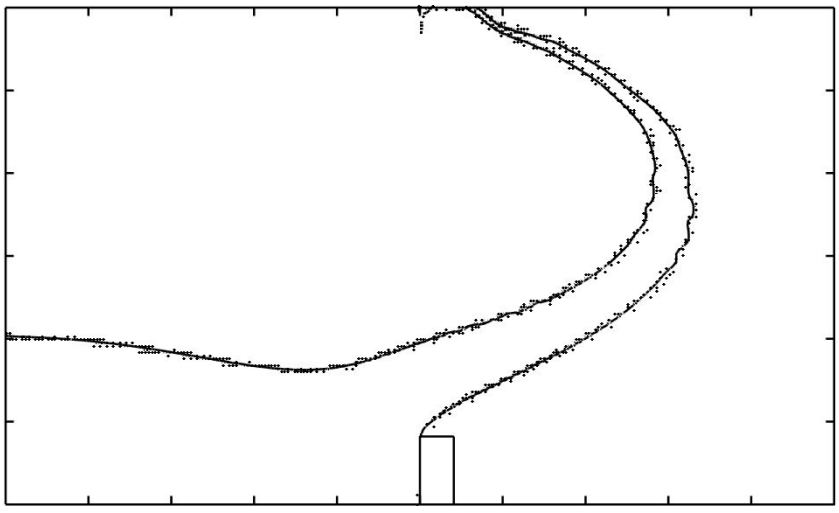


Figure 2.36: Continued from Figure 2.34, with frames for $\tau = 2.426$.

(a)



(b)



(c)

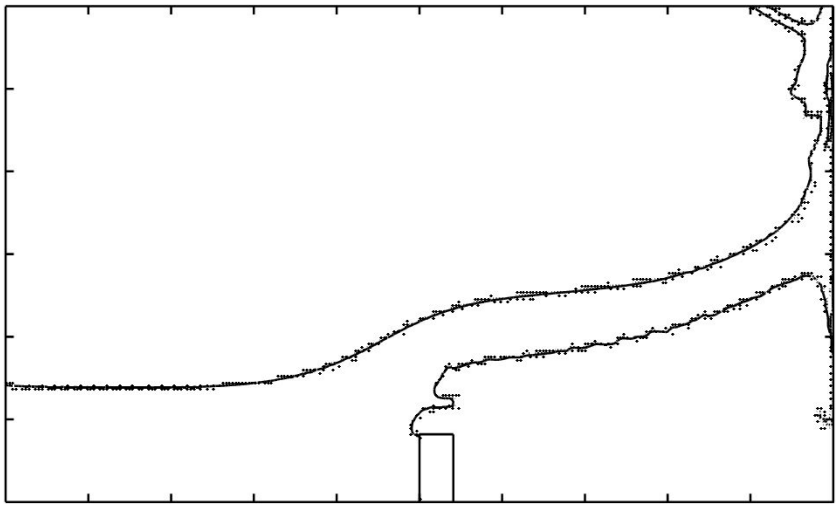
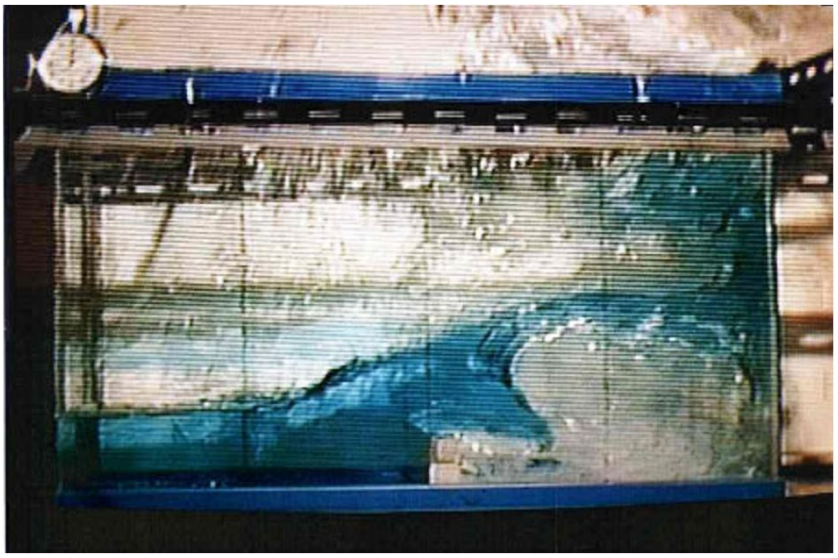


Figure 2.37: Continued from Figure 2.34, with frames for $\tau = 3.233$.

(a)



(b)



(c)

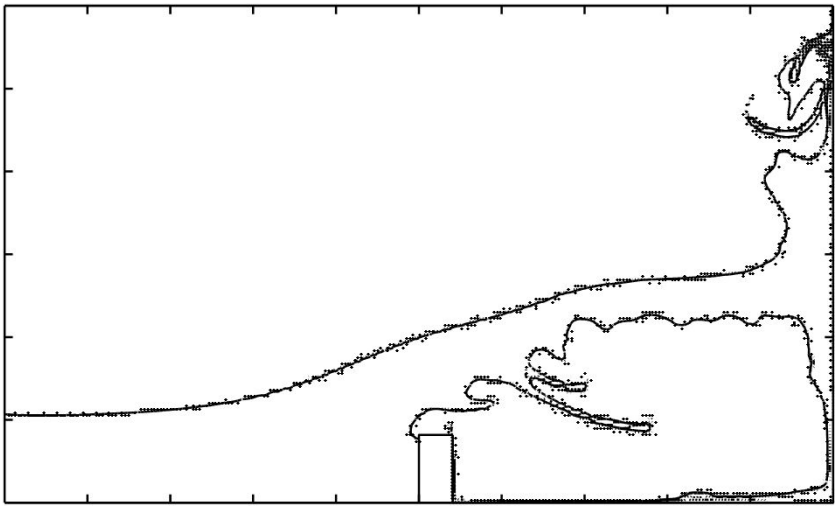


Figure 2.38: Continued from Figure 2.34, with frames for $\tau = 4.043$.

2.6.3 Collapsing water column with obstruction (3D)

This test is fundamentally a three dimensional version of the collapsing water column from Section 2.6.2. A column of water is released from one end of a closed tank and the advancing front is allowed to impact a prismatic obstacle. A schematic of the experimental setup is shown in Figure 2.39. The original experiment was carried out at the Marine Research Institute Netherlands (MARIN) and was meant to emulate water flow on the deck of a ship, which is also known as green water. Kleefsman et al. (2005) compared these experiments to results from a numerical simulation using the code COMFLOW implemented with a VOF method. A list of parameters used for the simulation is provided in Table 2.4.

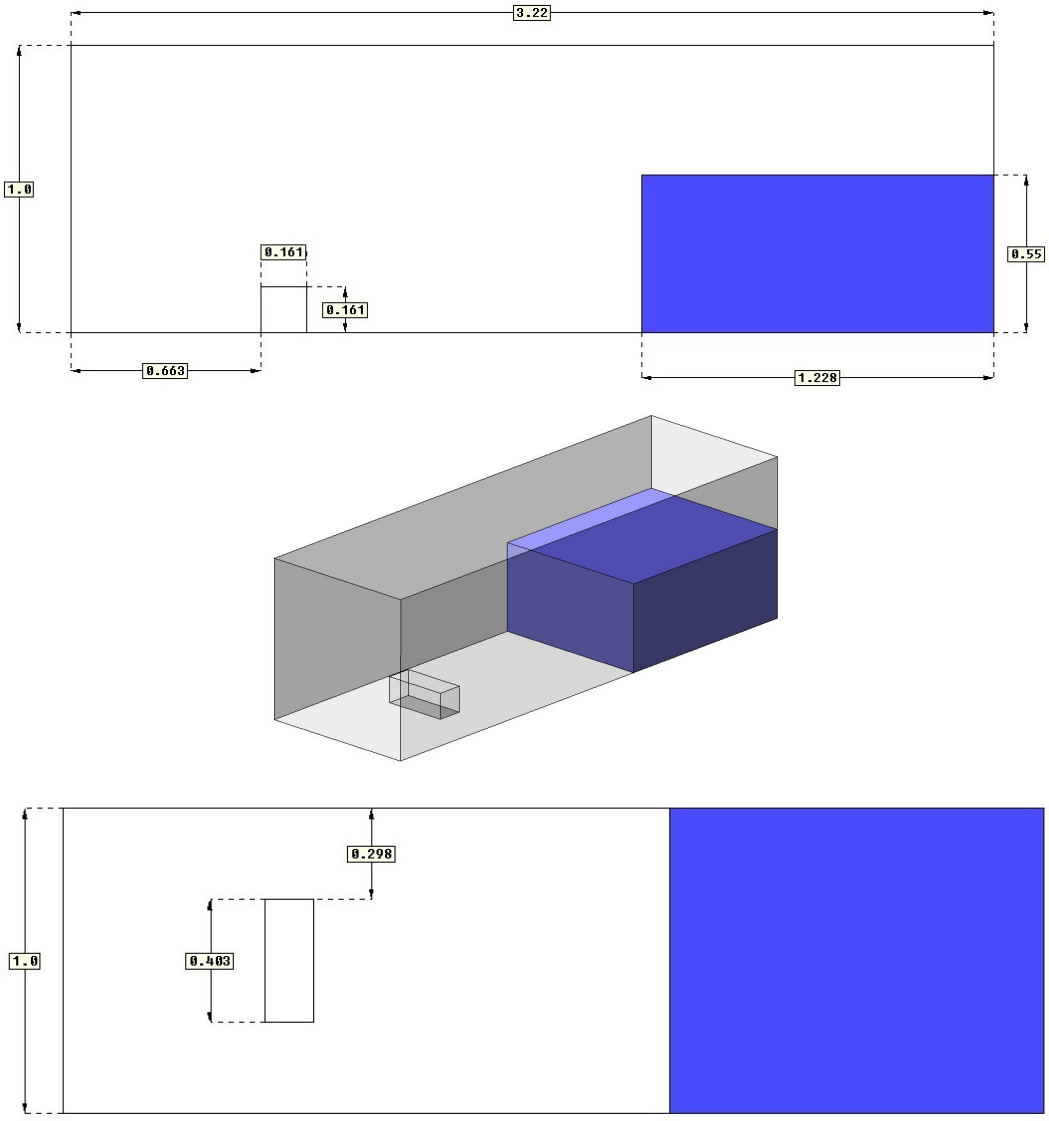


Figure 2.39: Experimental setup for the 3D collapsing water column with an obstruction as used by Kleefsman et al. (2005), showing a (a) side, (b) isometric, and (c) top view.

A qualitative comparison of results between Hyper-C, VOF, and experiments is shown in

Parameter	Value
Density, Fluid 1	$1000.0\ \frac{Kg}{m^3}$
Density, Fluid 2	$1.205\ \frac{Kg}{m^3}$
Viscosity, Fluid 1	$1.0 \times 10^{-3}\ \frac{Ns}{m^2}$
Viscosity, Fluid 2	$1.7 \times 10^{-5}\ \frac{Ns}{m^2}$
Gravity	$9.81\ \frac{m}{s^2}$
Simulation-time	$0.0s \Rightarrow 6.0s$
Timestep size	$0.0001s$
Min. elem. size	$0.015m$
Max. elem. size	$0.15m$

Table 2.4: Parameters for 3D water collapse problem

Figure 2.42. A quantitative comparison is obtained from eight pressure sensors located at different points on the obstacle, as well as four water height sensors placed along the domain bottom. A schematic of the location of the pressure sensors is shown in Figure 2.40, while for the height sensors it is shown in Figure 2.41 . Experimental data from the pressure and height sensors was compared to data from the numerical simulation with Hyper-C. Graphs depicting this comparison are shown in Figure 2.47 for pressure sensors and in Figure 2.48 for height sensors.

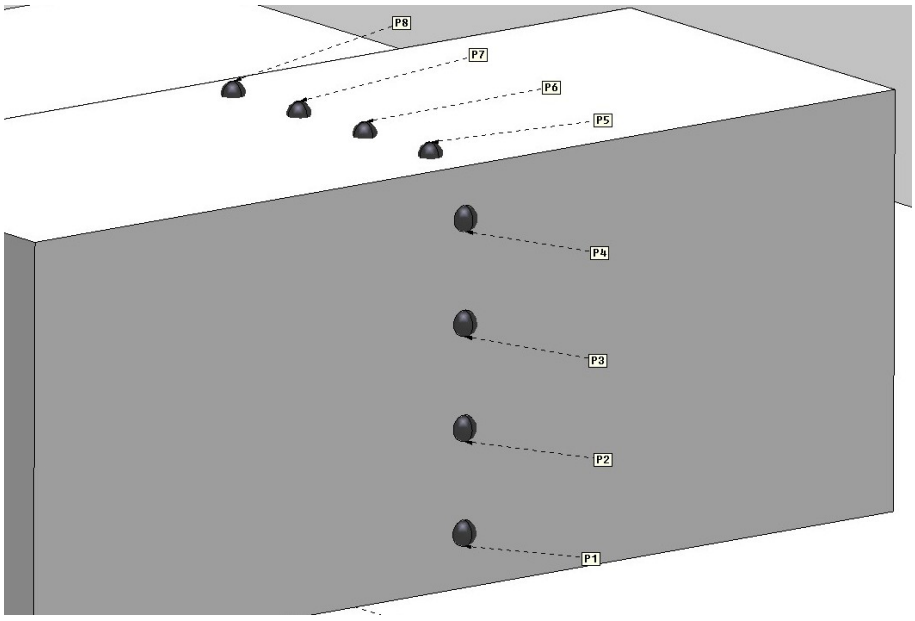


Figure 2.40: Pressure sensor locations at the obstacle Kleefsman et al. (2005)

Effectively, results obtained from this simulation portray the excellent capabilities of interface modelling contained in the Hyper-C method implementation developed in this thesis. As was also observed in results from the shearing flow simulation, results from the VOF simulation of this test (Figures 2.42 through 2.46) show small traces of material being left stranded in

mid-air. This is not observed for Hyper-C. Excellent agreement is also observed for each of the height and pressure sensors, with only minor discrepancies. This test provides further proof that CVFEM-Hyper-C is a more than excellent candidate for fluid interface modelling. It is able to maintain interface sharpness to the limit of the resolution and is inherently conservative.

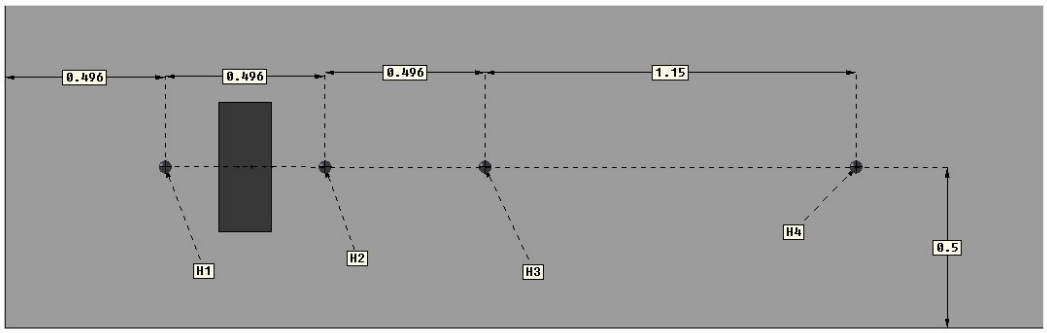


Figure 2.41: Water height sensor locations at the obstacle Kleefsman et al. (2005)

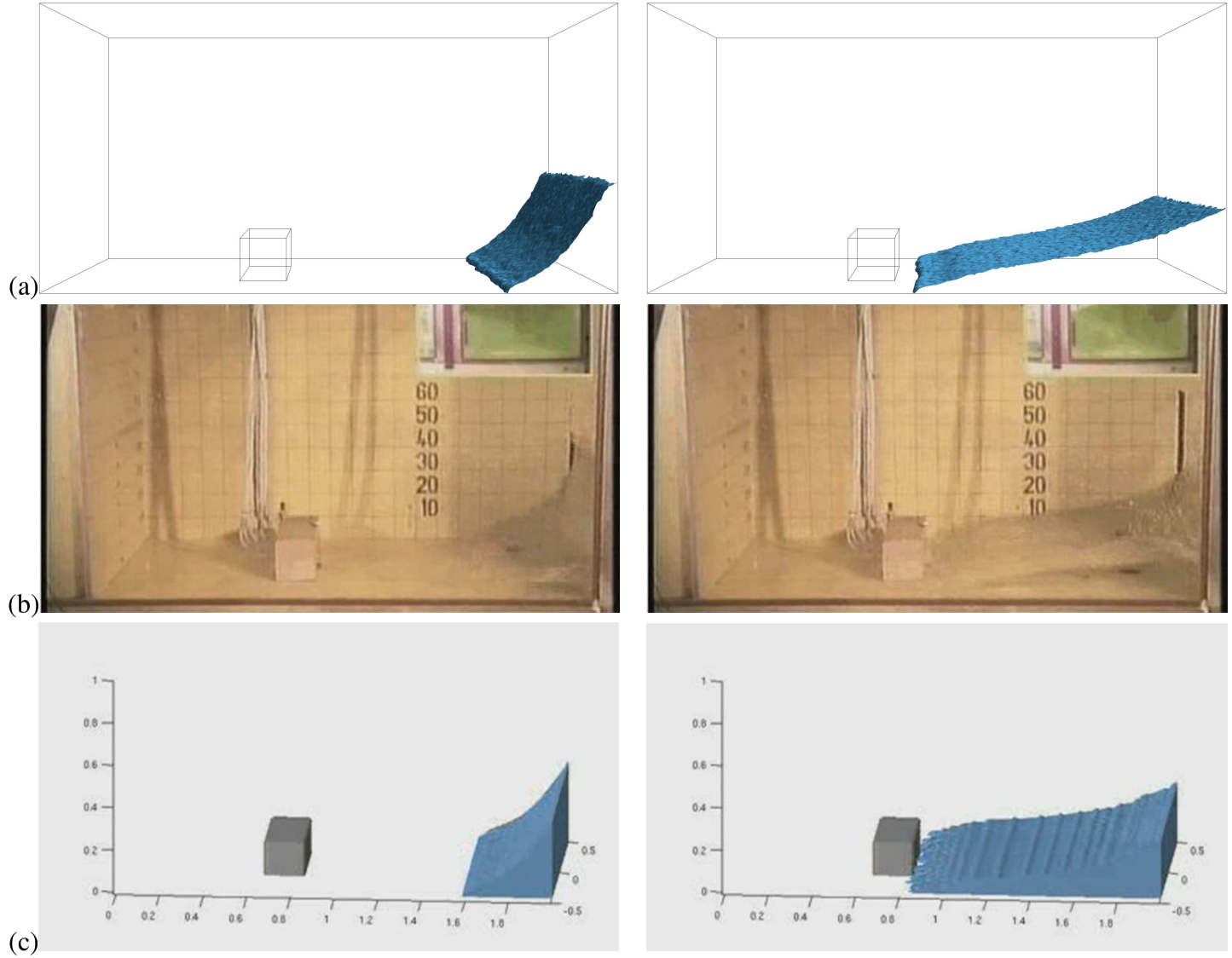


Figure 2.42: Result comparison of the 3D collapsing water column with an obstruction, comparing (a) Hyper-C to (b) experiments, and (c) the VOF method by Kleefsman et al. (2005). Time frames are, by column, for $t = 0.2$ and $t = 0.4$.

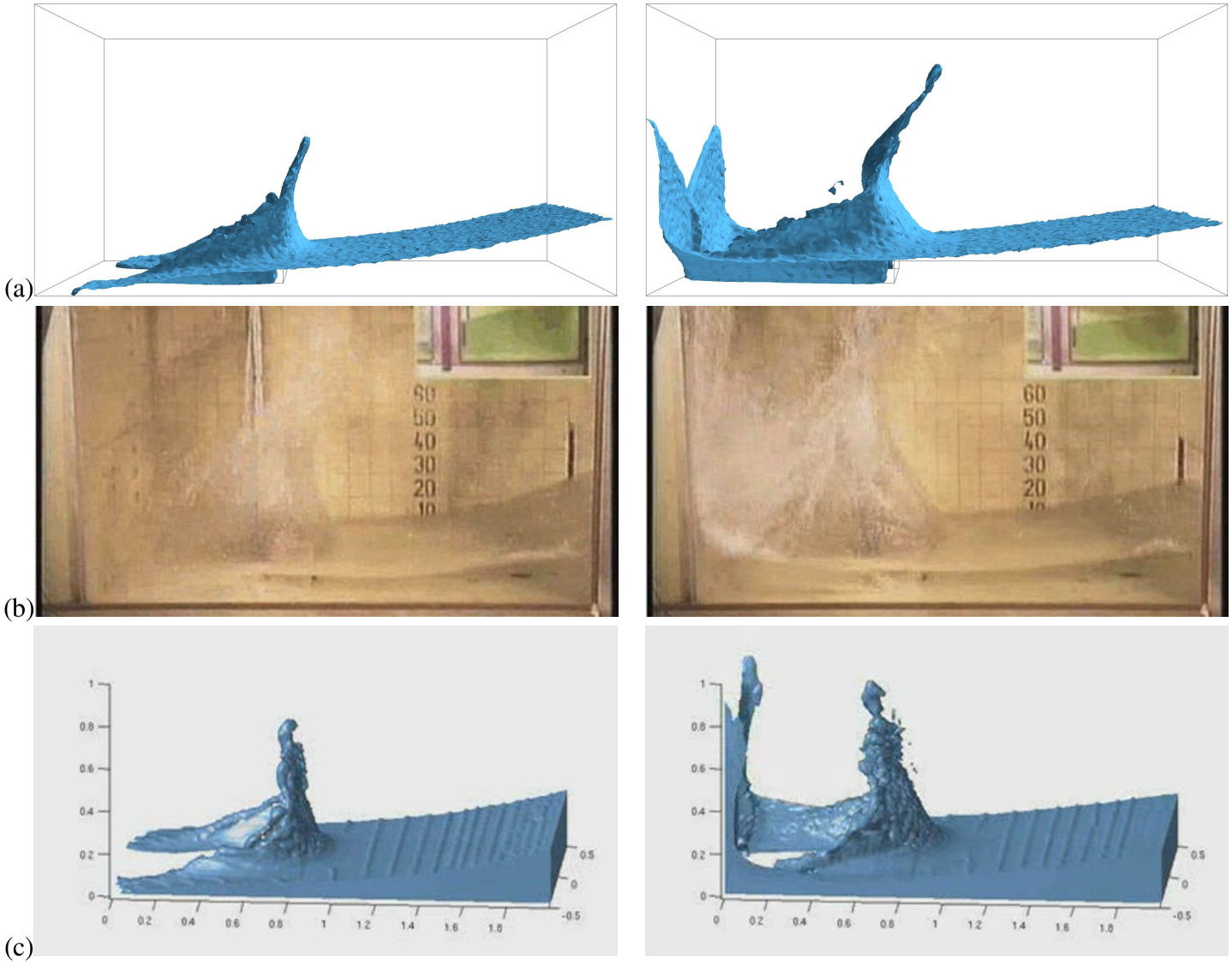


Figure 2.43: Continued from Figure 2.42, for time frames $t = 0.6$ and $t = 0.8$.

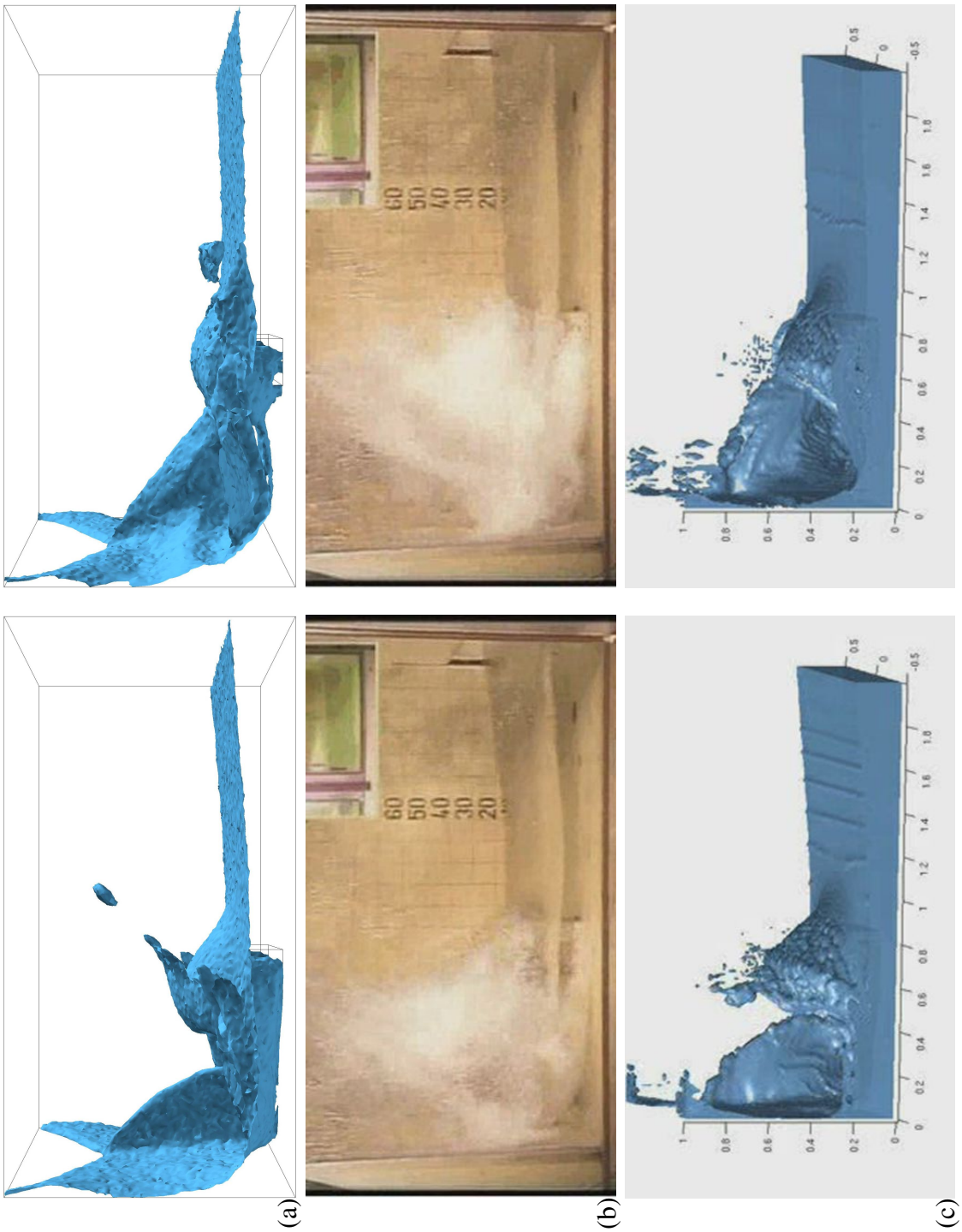


Figure 2.44: Continued from Figure 2.42, for time frames $t = 1.0$ and $t = 1.2$.

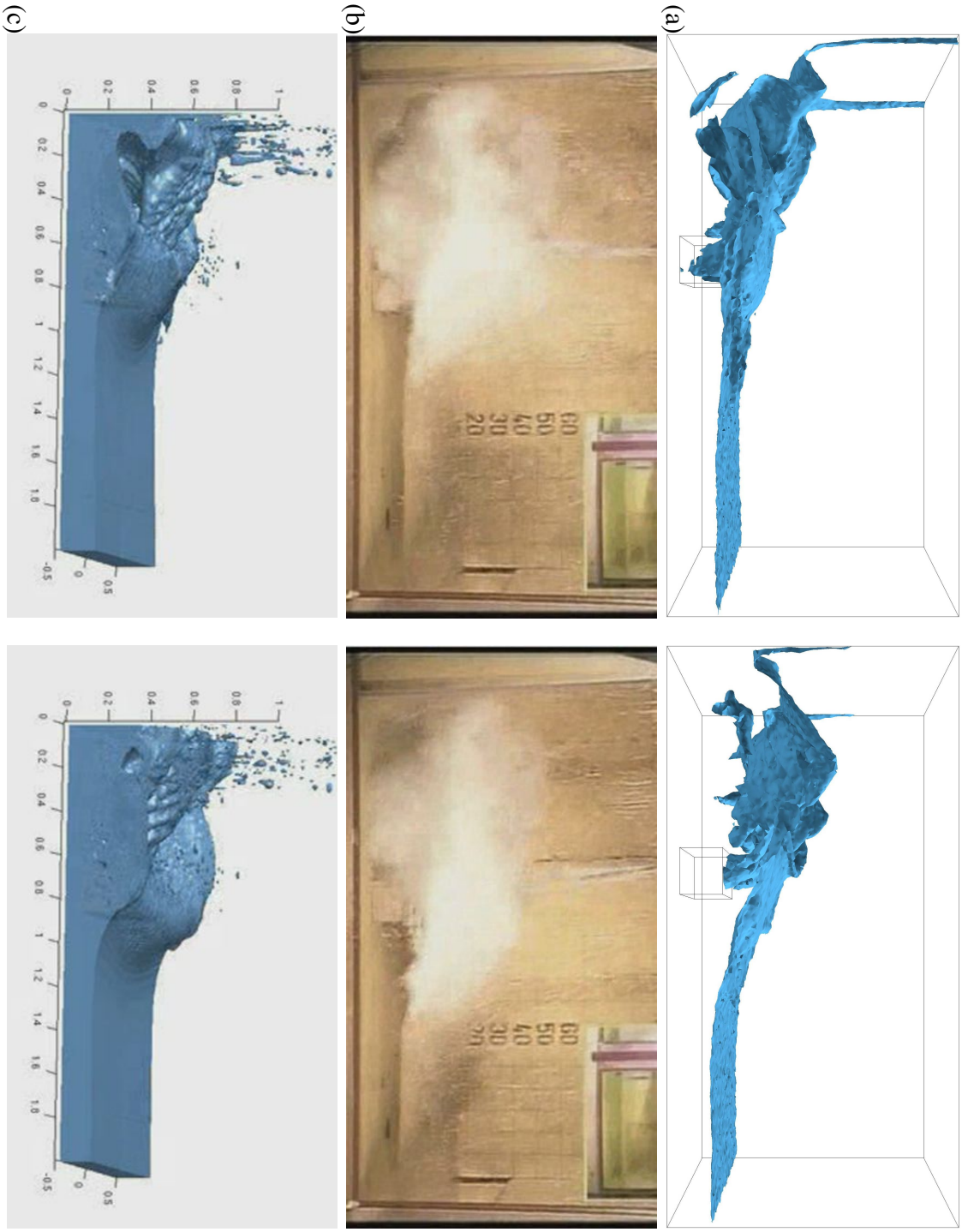


Figure 2.45: Continued from Figure 2.42, for time frames $t = 1.4$ and $t = 1.6$.

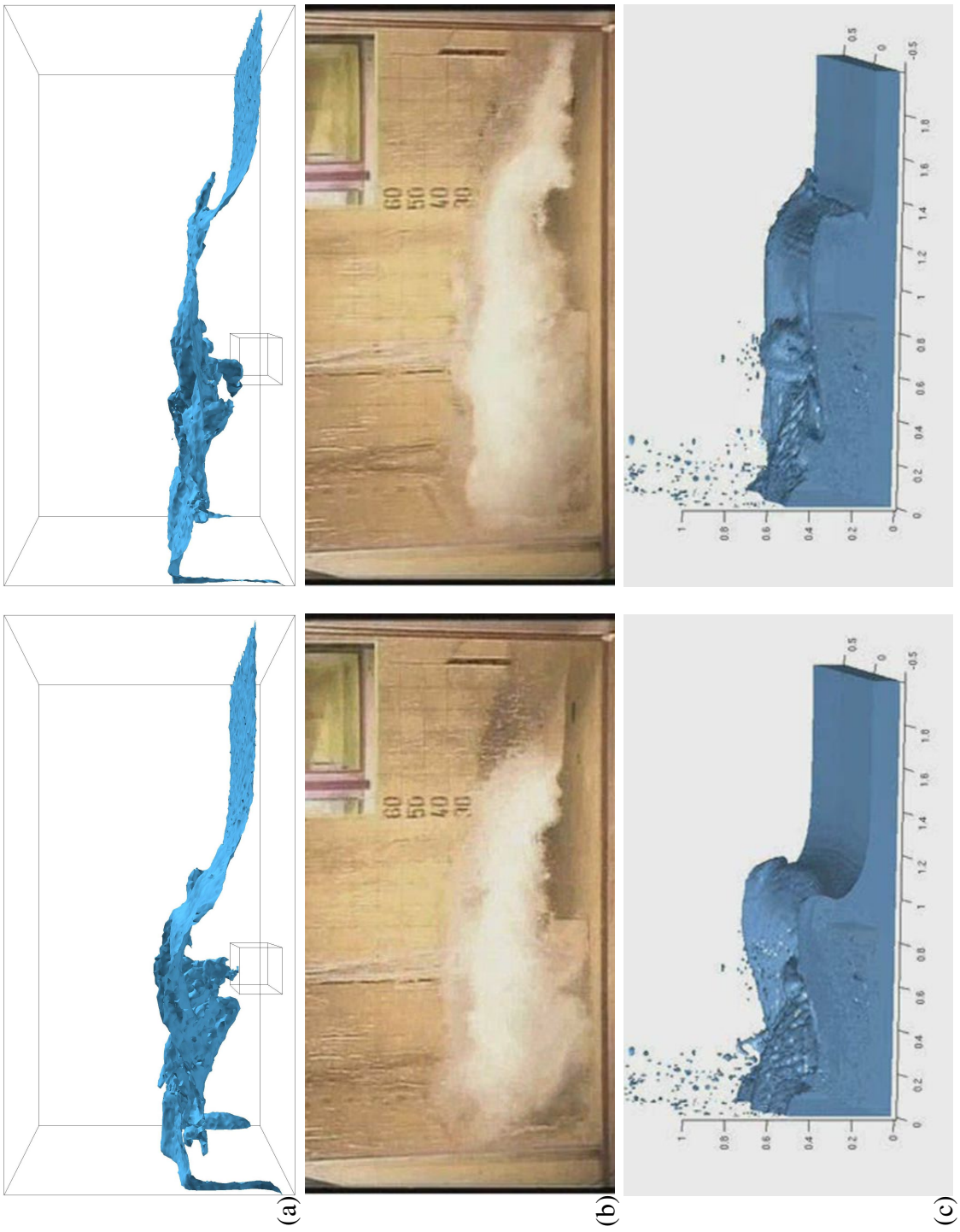


Figure 2.46: Continued from Figure 2.42, for time frames $t = 1.8$ and $t = 2.0$.

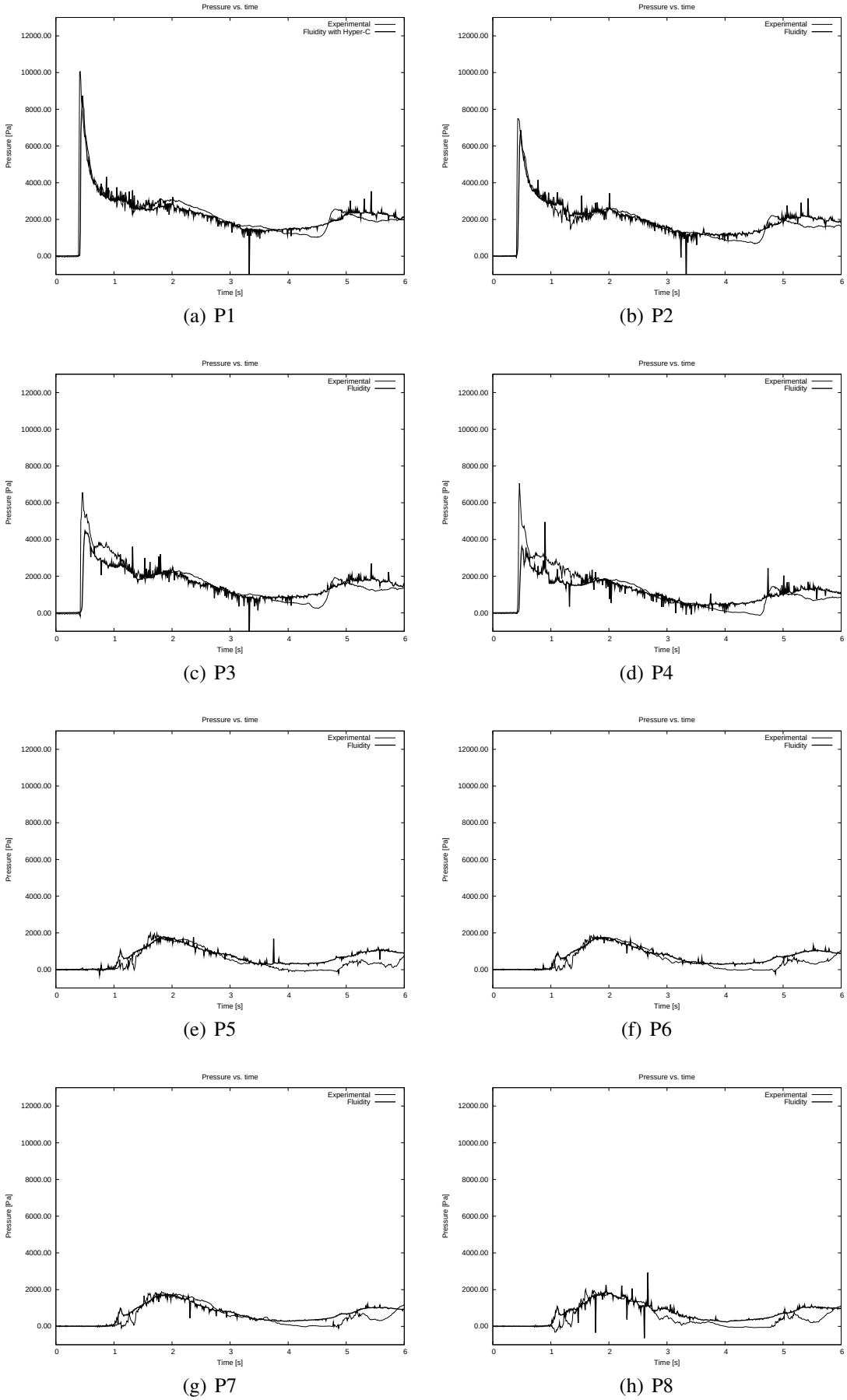


Figure 2.47: Pressure sensor comparison between results from this thesis's Hyper-C and experiments at MARIN facilities. Spikes that can be observed in the darker curve are attributed to solution remapping in adaptive steps.

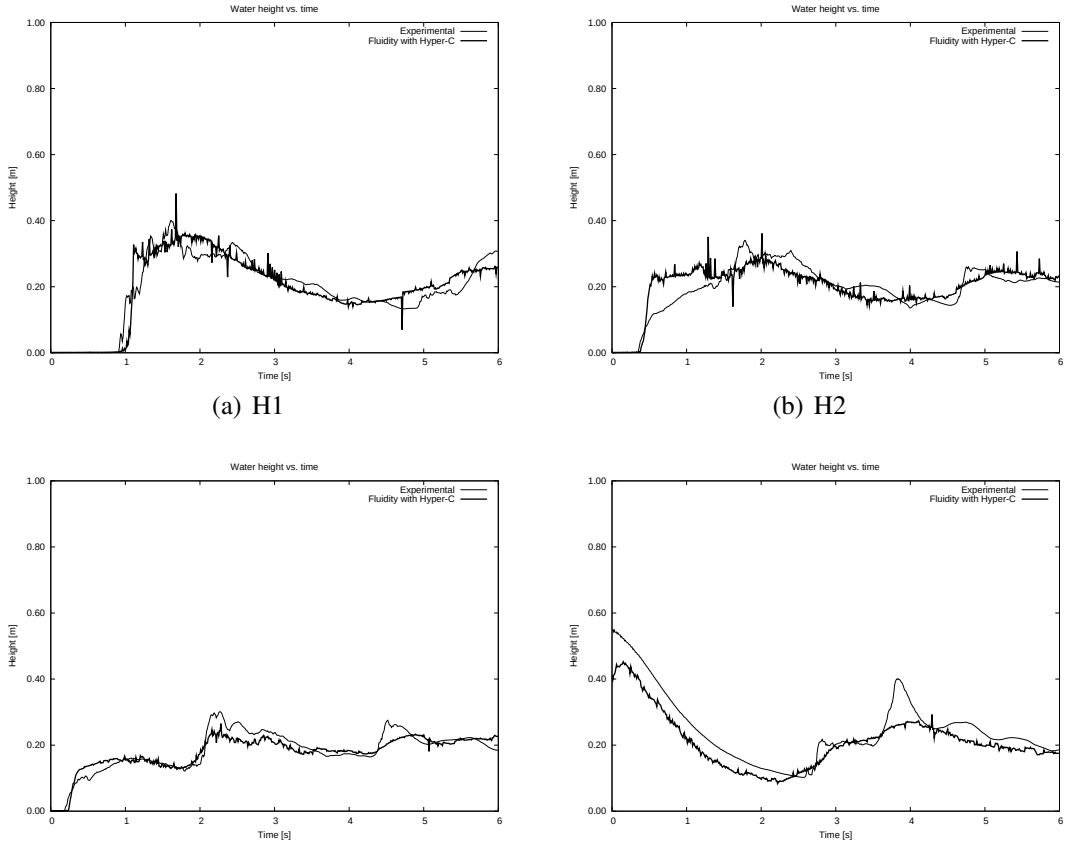


Figure 2.48: Height sensor comparison between results from this thesis's Hyper-C and experiments at MARIN facilities. Spikes that can be observed in the darker curve are attributed to solution remapping in adaptive steps.

2.7 Conclusions

In this Chapter, a new method has been successfully developed to simulate the interaction between immiscible fluids, allowing for the modelling of flows with breaking and merging interfaces while maintaining a sharp approximation. The technique involves the combination of the pseudo concentration scheme through a control volume finite element discretisation, and an incompressible flow solver. Fluid velocities and pressure are calculated through a finite element approximation of the Navier- Stokes equations, using implicit time-stepping to yield a stable formulation. The method is aimed at use with unstructured meshes, allowing for a high flexibility when representing complex domain boundaries and the usage of AMO.

The developed technique was tested against some of the most studied and commonly used techniques for interface tracking, achieving great performance. A special highlight of this are the cases where flow velocity is actually solved for, such as the two and three dimensional water collapses, in which the combination of this method with an adaptive mesh optimisation technique yields very accurate results when predicting the free surface shapes. Excellent agreement

is also achieved when comparing to the laboratory experiments of Martin and Moyce (1952), in which the leading edge and water height of the collapsing water columns are measured. Furthermore, in the three dimensional test case presented by Kleefsman et al. (2005), experimental pressure sensor data was compared to numerical probe data throughout the evolution of the simulation, yielding very good agreement.

Regarding future developments, the analytical tests presented in section 2.5 reveal the fact that the advection limiting process needs optimising. It must be noted, however, that even though the method developed herein slightly underperformed when compared to other VOF techniques for the analytical tests, most of these techniques require geometrically conditioned fluxing (see, for instance, the analysis by Rudman (1997)). While some interface reconstruction method might be relatively easily conceived in two dimensions, they are in general not a simple task in three dimensions. This fact is taken to an extreme when involving unstructured tetrahedral meshes, since the geometric reconstruction of the interface and calculation of fluxes can prove to be a complex and cost-ineffective task. Hyper-C does not suffer from this hindrance as it requires no complicated interface reconstruction and thus it easily fits into usage with an unstructured tetrahedral mesh. The latter statement not only allows for utilisation on complex domains, but also allows for the combination with adaptive mesh optimisation techniques such as the one by Pain et al. (2001a).

With respect to level set methods, which struggle to achieve mass conservation specially when resolution is insufficient, Hyper-C has proved to be more successful, which was shown throughout the different tests in sections 2.5 and 2.6.

Hyper-C pseudo concentration method promises to be one of the next generation of accurate and efficient techniques created to predict important natural phenomena involving interface capturing, having an impact on applications from many engineering areas. For the applied concerns of this thesis, this Chapter has provided considerable proof that CVFEM-Hyper-C is an excellent tool for the predictive modelling of coastal phenomena such as the breaking of waves and subsequent fluid behaviour. Chapter 5 provides more detail on the usage of this method in applied cases.

Chapter 3

Mesh Movement

Simulating phenomena involving two or more immiscible fluids requires the modelling of the evolving interface between them. Various interface tracking/capturing techniques have been developed to perform this task accurately, however the usefulness of each method is strongly dependent on the underlying geometry of the computational mesh.

Unstructured meshes have the outstanding capability to describe the most complex domains in two and three dimensions. However, the problem to overcome is that the accuracy of the numerical advection is limited by the mesh resolution, specially near boundaries and interfaces. In strongly advective problems such as those that involve the interaction of immiscible fluids, situations are bound to arise in which the mesh is not suitable and may not properly define sharp interfaces and/or the flow development. Often, a workaround is to generate a new mesh by means of "a posteriori" error measures (see, for instance Pain et al. (2001b)), hence the numerical values obtained on a given mesh need to be interpolated into the newly optimised mesh to be used in the next time step. These methods which require grid-to-grid interpolation introduce an error which is bound to, cumulatively, affect the simulation results. Interpolation error has a negative impact on the solution method's conservation properties, and even the benefit of using a compressive advection scheme like Hyper-C may be affected.

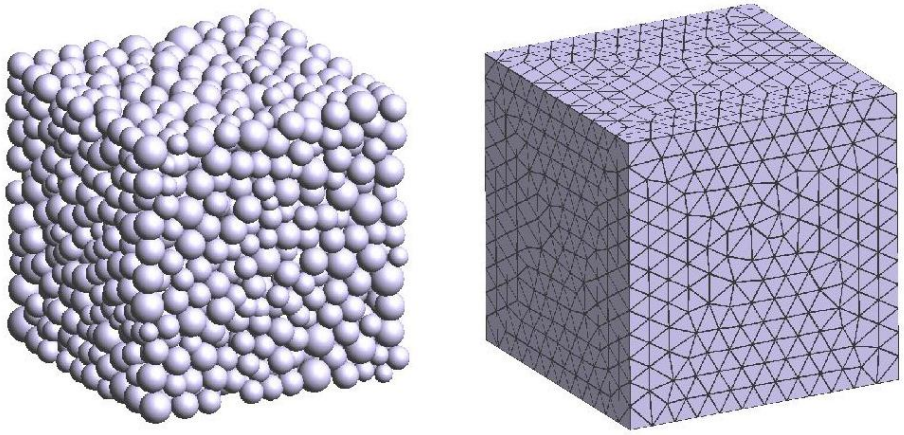
In this Chapter, a new mesh movement method is proposed, based on well-established arbitrary Lagrangian- Eulerian (ALE) methods, which provides a successful compromise between accurate Lagrangian interface tracking and infrequent mesh adaptation. In essence, the scheme optimises node movement to maximise element quality and minimise advective numerical diffusion of the fluid boundary. The new nodal positions are calculated through an iterative sequence of local node movements which minimise a functional embodying both element quality and interface accuracy. In broad terms, the functional value depends on the element size, element shape, and the convective term from the concentration scalar advection equation. This latter component ensures accuracy in the representation of the non-linear advective terms where steep material gradients are encountered. The result is a method that further extends the capabilities of the compressive advection scheme presented in Chapter 2, which can only sharpen the interface to the limit of the localised resolution. Compared to other ALE techniques, the

innovation of this scheme lies in the manner in which the functional is calculated and the mesh positions are updated. The method has been implemented in a general purpose unstructured mesh finite element model, perfectly fitting into the solution framework which was presented in Chapter 2.

This Chapter begins with a brief introduction to the concept of grid optimisation and node movement, followed by details of Lagrangian, Eulerian, and their combined descriptions of motion. The implementation of an arbitrary Lagrangian- Eulerian method into the solution framework of Chapter 2 is presented. Details of how the grid velocities are obtained are given, followed by performance tests.

3.1 Grids and Adaptation

The computational grid for a numerical model consists of a set of points or nodes where the variables will be calculated. These points generate a discrete representation of the domain geometry. When these nodes are interconnected to form non overlapping contiguous subdomains or "elements" the resulting geometric discretisation is referred to as the "mesh". When the nodes are not connected and their interaction in the calculation is dealt with through other means (see, for example Gingold and Monaghan (1977) on the formulation of smoothed particle hydrodynamics, SPH) then the discretisation is characterised as "meshless". Figure 3.1 illustrates the difference between these two geometric discretisation methods. Meshless methods are not the main interest of this thesis, however they are mentioned here for the sake of completeness.



(a) Meshless discretization of a cubic domain.

(b) Meshed discretization of a cubic domain.

Figure 3.1: Comparison between meshed and meshless geometric discretisations. Meshless domain discretisation can be viewed as spheres with a given radius that depends on the method. Meshed domain pictured here is a linear tetrahedral discretisation of the cube, of which the surface mesh can be seen.

A fixed computational grid, also called Eulerian grid, throughout the simulation is com-

monly used in most applications of computational fluid dynamics. However, due to the computational cost when discretising large domains and obvious inefficiencies when allowing certain regions of the domain to be unnecessarily resolved, it has become increasingly popular to utilise methods that optimise grid geometry.

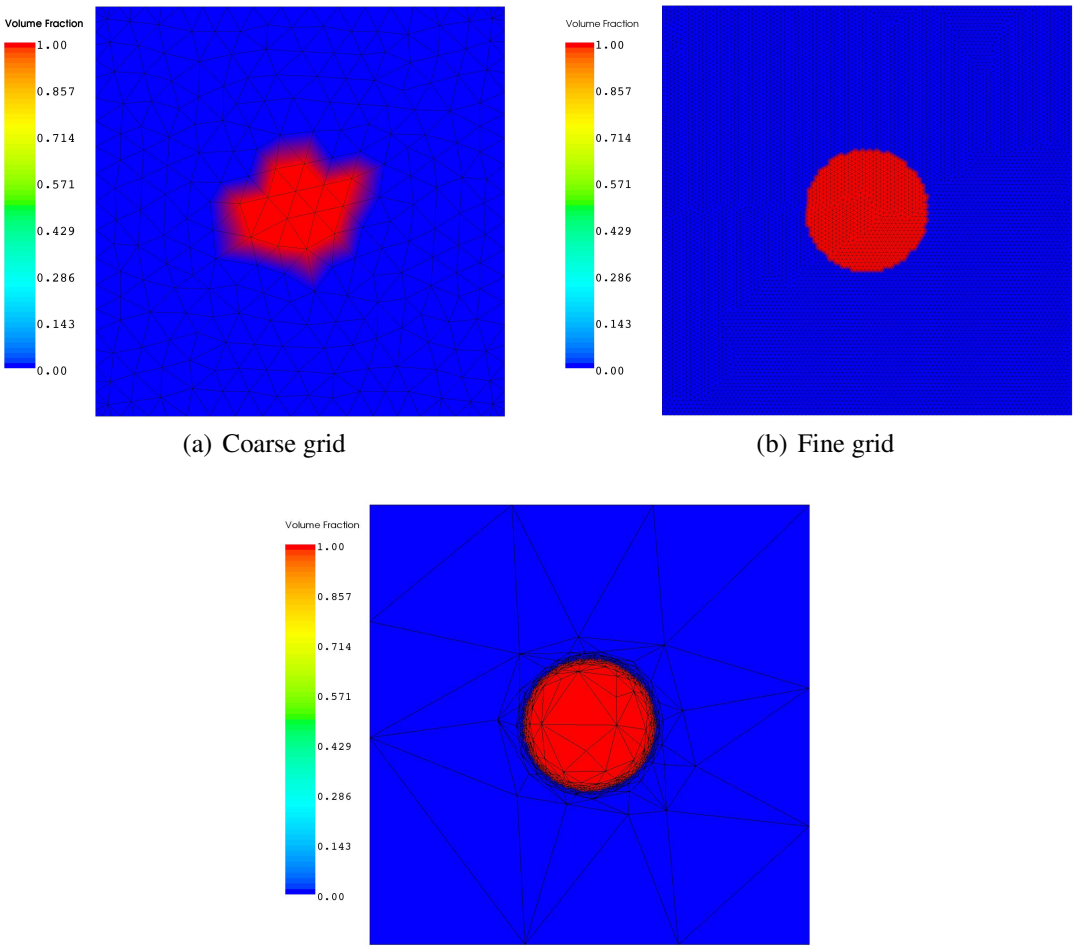
To reach an optimised state, the numerical grid may be deformed, subdivided, and/or coarsened in a different number of ways. Nodes from the grid may be relocated, added, subtracted, or any combination of the three, to allow for an optimised element size and shape distribution, subject to user defined defined parameters such as the desired interpolation error (see, for instance Piggott et al. (2005)). As an example, Figure 3.2 shows a scalar field and its associated computational mesh depicting a domain centred circular shape, which serves to illustrate the qualitative difference that exists when using an appropriate mesh for result representation. In the first case, shown in Figure 3.2(a), there is a coarse mesh which will lead to fast calculations while penalising accuracy. Secondly, Figure 3.2(b) shows a fine grid case, which leads to computationally costlier calculations while returning quite more accurate results than the coarse case. Thirdly, Figure 3.2(c) depicts the optimised mesh, which achieves a degree of precision at least as high as the fine grid, while keeping a much lower computational cost.

Judging from these simple observations, it is foreseeable that the future of computational methods lies in the optimisation of grid geometry and the associated computer memory and processor requirements. This vision, however, does not exist without its obstacles. Optimisation methods face three main hindrances: (a) optimisation algorithms are complex and their implementation is not straightforward, (b) simulations with very dynamic and strong discontinuities might require frequent adaptive steps, and (c) they are prone to interpolation error when passing results from one mesh to another on successive time iterations of a simulation.

Lagrangian meshed methods have the advantage that the constitutive laws for continuum mechanics are defined on the material points, and hence are naturally incorporated. They are typically used in solid mechanics, however it is well known that they suffer greatly when under considerable degrees of mesh deformation (see, for example, Gadala (2004)). This effect would be drastically increased if the method was used to model fluids in turbulent flow, or fluid interfaces breaking and merging. A typical observation is the "tangling" effect[1], shown in Figure 3.3 for a two dimensional triangular mesh. Tangling and other unwanted mesh deformation effects are usually overcome by a reduction of the timestep, a frequent series of mesh adaptive steps, or even both. Using mesh adaptivity in this case would solve the problem, but basically destroys the effort of using a Lagrangian method to eradicate numerical diffusion effects from advection. In addition, a decreased timestep may have dire impact on the simulation's computer run-time.

The above realisations about Eulerian and Lagrangian reference frame calculations have led to the development of many different techniques and algorithms for optimisation of meshes. These techniques, along with the goal of further enhancing the behaviour of Hyper-C, are the

[1]Geometrically speaking, for linear tetrahedra, this is described by having one of the nodes forced past the opposing face. In triangular elements it implies a node forced through an edge.



(a) Coarse grid

(b) Fine grid

(c) Optimized grid

Figure 3.2: Examples of 2D mesh suitability for the represented results.

core drive for the development of the material presented in this Chapter.

3.1.1 Grid optimisation techniques

Out of all the schemes that exist and are still being studied and developed, there are four basic methods that may be distinguished. These are the "h" method, the "r" method, the "p" method, and the re-meshing method.

The “h” method, used by Bank et al. (1983), Mavriplis (1997), Löhner (2001), and Plaza et al. (2000) among many other authors, is characterised by the mesh refinement and/or de-refinement via as series of operations of grid alterations ranging from element subdivision, as shown in Figure 3.4, to those more intricate such as edge collapsing, face to edge and edge to face swapping, and edge splitting (see Pain et al. (2001a)).

The “r” method, employed by Mavriplis (1995), Ait-Ali-Yahia et al. (2002), Habashi et al. (2000), Ivanenko and Azarenok (2002), and Pain et al. (2001a) among other authors, uses a re-location scheme to position existing nodes where resolution is necessary. The optimal positions are typically obtained through some form of error measure or functional that reflects the mesh’s

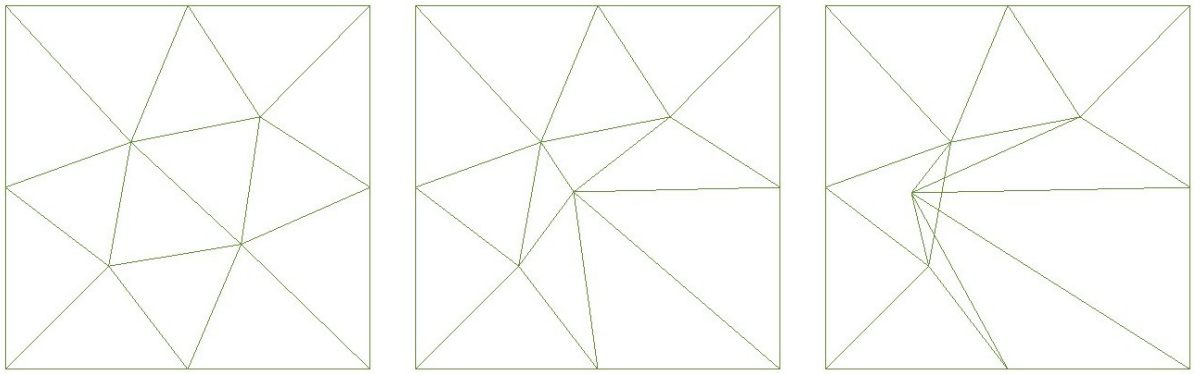


Figure 3.3: Frame sequence (a) to (c) describing a node in the mesh moving until it goes through an edge and turns the element “inside out”.

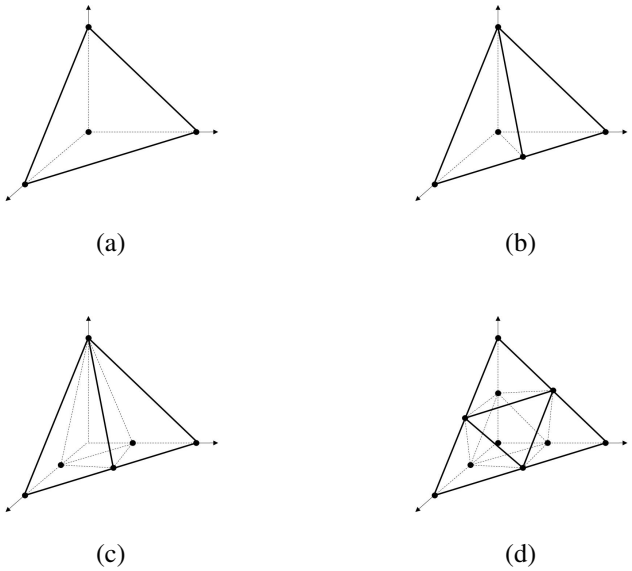


Figure 3.4: Subdivision of a tetrahedron (a), into (b) two, (c) four, and (d) eight sub-tetrahedra.

geometric suitability to the results. The nodal number and mesh connectivity remains intact as well as the elemental base interpolation functions.

The "p" method, also denominated p-enrichment, used by Demkowicz et al. (1991), Devloo et al. (1988), Barros et al. (2004), and Dong and Karniadakis (2003) among other authors, is based on shape function polynomial degree augmentation.

The re-meshing method is used by Peraire et al. (1987), Löhner (1989), Anderson et al. (2005), and Diez and Calderon (2007) among other authors. As the name states, it relies on the complete re-generation of the computational mesh starting from the geometric boundaries.

It is very important to understand the different terms that refer to mesh operations. In this thesis, "node relocation" or movement refers to the movement of the nodes without altering their number or the connectivity. The term "re-meshing" refers to a complete regeneration of the mesh from scratch. "Mesh optimisation", "mesh adaptivity", or "adaptive mesh optimisation"

(AMO) refers to the alteration of the current mesh, either by "h", "r", or "p" refinement or any combination of the three. Notably, Fluidity's mesh adaptivity, frequently mentioned throughout this thesis, combines "h" and "r" procedures. Adaptivity is sometimes incorrectly referred to as "re-meshing", which can be misleading and actually refers to a potentially more complex operation. The work presented herein is related and therefore will refer to many aspects of the adaptive process, however re-meshing and grid generation are out of the scope.

The method presented in this Chapter possesses the characteristics that resemble the "r" method of grid adaptivity, however results are not interpolated from an old mesh to a new mesh. Instead, the discretised governing equations are solved allowing the mesh to alter its nodal positions, and including the corresponding velocities as a change in the reference frame for the calculations at each node. Since each node in the computational mesh is free to move independently of the flow velocity, this type of method is often referred to as arbitrary Lagrangian-Eulerian. Details of obtaining a moving reference frame formulation of the governing equations are provided in section 3.2

3.2 Lagrangian and Eulerian descriptions of motion

When an observation point follows a material point in a continuum, it is said that the description of motion is Lagrangian. When the observation point remains in a fixed position in space, allowing material to flow through, the description of motion is referred to as Eulerian.

Let there be an inertial reference frame R_{xyz} with respect to which spatial coordinates are fixed, and $\mathbf{x}$ be a vector field of those spatial coordinates. Let $\mathbf{m}$ be an array containing the material "tags" or labels that distinguish each one of the infinite amount of particles that make up a given continuum, while $\mathbf{x}_{m_i}$ is a vector containing the coordinates of a given material particle i belonging to $\mathbf{m}$ at a given instant (i.e.: $t = 0$), then

$$\mathbf{x}_i = \varphi\left(\mathbf{x}_{m_i}, t\right) \tag{3.1}$$

where φ is the function that describes the motion of the material particles. For a material particle i or material coordinates $\mathbf{x}_{m_i}$, $\mathbf{x}_i$ contains the trajectorial line that describes the movement through function φ. The i sub-index is conveniently dropped at this point.

The function φ is described as a mapping function which allows the following statements:

1. $\varphi\left(\mathbf{x}_m, 0\right) = \mathbf{x}_m$, by definition.

2. φ is C^1

3. φ^{-1} is defined as the inverse mapping of φ and both functions are bijective[2].

[2]This guarantees that no particle may occupy two spatial points, and no two particles may occupy the same spatial point.

Material velocity is given by the change in spatial position with respect to time for a fixed material coordinate, hence

$$\mathbf{u}(\mathbf{x}_m, t) = \left.\frac{\partial \mathbf{x}}{\partial t}\right|_{\mathbf{x}_m} \tag{3.2}$$

where the expression $|_{\mathbf{x}_m}$ implies a fixed material coordinate[3]. The total derivative of a given property $\vartheta = \vartheta(\mathbf{x}, t)$ which may be scalar, vectorial, or tensorial, is calculated using the chain rule,

$$\left.\frac{\partial \vartheta(\varphi(\mathbf{x}_m, t), t)}{\partial t}\right|_{\mathbf{x}_m} = \left.\frac{\partial \vartheta(\mathbf{x}, t)}{\partial t}\right|_{\mathbf{x}} + \frac{\partial \vartheta(\mathbf{x}, t)}{\partial \mathbf{x}} \left.\frac{\partial \mathbf{x}}{\partial t}\right|_{\mathbf{x}_m} \tag{3.3}$$

where the local derivative term, denoted with $|_{\mathbf{x}}$, is the partial derivative while the spatial coordinates are fixed, and the convective term[4] is composed of the local gradient of the material property multiplied by the material velocity. Rewriting equation (3.3) in a more compact form yields,

$$\frac{d\vartheta}{dt} = \frac{\partial \vartheta}{\partial t} + \mathbf{u} \cdot \nabla \vartheta \tag{3.4}$$

where $\frac{d\vartheta}{dt}$ or $\frac{D\vartheta}{Dt}$ are common ways to refer to the total derivative, while a partial derivative is used for the local derivative.

The local derivative accounts for what can be measured by an observer at a fixed point in space with respect to R_{xyz}. The total derivative, however, takes into account what is measured by an observer that is in motion with a material particle and therefore involves the convective term $\nabla \vartheta \cdot \mathbf{u}$. As an example, let it be assumed that $\vartheta = \mathbf{u}$, where $\mathbf{u}$ is the flow velocity. The total derivative of this property implies the calculation of the total acceleration of the flow with respect to the Lagrangian reference frame. Following a material particle in steady incompressible flow through a narrowing nozzle proves that even though the velocity field is steady and the same velocity is measured at all times at a given point in the nozzle (i.e.: $\frac{\partial \vartheta}{\partial t} = \frac{\partial \mathbf{u}}{\partial t} = 0$), the fluid material particles in fact accelerate as they travel to satisfy continuity. This effect is calculated by the convective term and is due to the existence of a velocity gradient.

As mentioned earlier, the Lagrangian reference frame is largely used in solid mechanics. The nodes of the computational mesh are considered "material" points and hence move along with the deformation of the material. The advantage of this approach is that each material particle's movement can be tracked to describe the step by step deformation of a solid. However, for a material that can suffer large deformations the grid can become excessively distorted and cause problems, such as the exaggerated reduction of the time-step size[5]. Such a result can be appreciated when the Lagrangian reference frame is applied to fluid flow. Fluid particles do not

[3]Fixed material coordinate refers to the fact that an "observer" is positioned on the material particle measures the change of a property as it gets moved about..

[4]Many texts may also refer to the total derivative as the convective derivative. The latter name is omitted in this thesis to prevent confusion with the convective term.

[5]Normally, on all grid-based methods, the time step size is proportional to the smallest element in the mesh, to guarantee convergence.

remain close to each other, and their paths tend to be independent and/or to diverge. Having the computational grid follow such a movement would inevitably cause extreme distortion and possible geometric overlaps, which are both undesirable for calculations.

The Eulerian reference frame, on the other hand, is widely used to describe fluid dynamics. The grid points are fixed in space and the material is allowed to flow through them. Large distortion in the motion of the continuum can be dealt with straightforwardly, subjected to certain stability conditions given in combination with the time discretisation method (i.e.: Euler explicit, Euler implicit, Crank-Nicolson, etc.). However, the main draw back of the Eulerian description is that representation of flow details is highly mesh resolution dependant. For example, direct numerical simulations (DNS) of turbulent flow, can sometimes be quite computationally prohibitive due to the level of resolution required. If the mesh is fixed throughout the simulation, then the fine resolution needs to be applied throughout the domain, leading to large amounts of memory usage as well as CPU time. During the simulation, it may be noticed that resolution in some areas might not really be necessary, leading to inefficient use of computational resources. It is with this notion in mind that many researchers have devoted their time to developing grid optimisation methods, as mentioned in section 3.1.

From the main drawbacks of Eulerian and Lagrangian descriptions, it is then intuitive to enquire on the result of a combination of the two. The idea that the mesh can move in any direction in between Eulerian and Lagrangian reference frames yields the type of method denominated arbitrary Lagrangian- Eulerian (ALE). An ALE description of motion could be defined as a generalisation of the two classic viewpoints to kinematics. The essence of the methodology resides in the fact that the grid is neither fixed in space (Eulerian) or moving with the flow (Lagrangian). It combines the best aspects of both original reference frames allowing the mesh to follow the material, stay fixed, or move anywhere in between. The essence of the ALE approach resides in the choice of the amount of mesh deformation to allow at each timestep, which consists of the grid velocity.

The concept of moving grids has been studied for quite a few decades. A extensive review on different variants and applications of the ALE method was written by Benson (1992). The first papers describing a combination of Lagrangian and Eulerian reference frames for mesh motion were published by Noh (1964) , and Frank and Lazarus (1964). Both of them performed their implementations in a finite difference discretisation. Hirt was also among the first to describe an ALE methodology in 1972 (later republished in Hirt et al. (1997)), referred to the scheme as ALE due to the method's flexibility for grid movement calculation, and implemented it in a new finite difference method that resolved multi- dimensional problems employing an implicit solution technique that renders it usable for all flow speeds. Hughes et al. (1981) described the effective usage of ALE in a finite element incompressible flow solution method with application to fluid-structure interaction problems. However, using direct instead of iterative solvers limited its computational viability. Donea et al. (1982) described another finite element approach with ALE movement using an explicit time integration scheme, also applied to fluid-structure inter-

action problems. Benson (1989) developed a simple ALE method (SALE) coupled to non-linear finite elements. However, as Margolin states in his review (see Margolin (1997)), these simple ALE codes have trouble when remapping a material interface. This could appear to be solved through the implementation of a VOF scheme however this solution worked effectively only on incompressible flow solvers. Other related schemes include the GEL (Ghost-fluid Eulerian Lagrangian) described by Arienti et al. (2003), in which the whole meshed domain is divided into Lagrangian and Eulerian sub-regions using a level-set approach.

Most ALE schemes are composed of two basic steps. The first one involves a Lagrangian step, in which all the variables are calculated in a Lagrangian reference frame, where the mesh moves together with the material. After this step comes the "rezone" step, in which the Lagrangian mesh is relaxed and material is advected as the mesh relaxes. This re-mapping or rezoning step is implemented to prevent mesh tangling and to attempt to maintain overall grid quality. The overall strategy is often referred to as "operator splitting" because the set of governing equations are solved in sequence. Each step of the sequence involves the solution of one group of differential operators from the given equations.. Examples for rezoning methods can be found by Aymone (2004), Knupp et al. (2002), and Lipnikov and Shashkov (2006).

These schemes are very useful in most types of simulations, specially those that do not involve excessive distortion of the mesh. Non-linear turbulent flows or multi- fluid interface problems, however, may not be properly dealt with. For example, Morrell et al. (2007) described a scheme in which the mesh performs a Lagrangian step followed by a remap (or advection) of the state variables to a relaxed grid. This relaxed quadrilateral grid is improved by an Adaptive Mesh Refinement (AMR) scheme, introducing resolution where needed. This scheme would not be efficient in the event of simulating, for example, plunging and breaking of water waves. Complex velocity gradients would cause a Lagrangian step to be practically impossible since tangling would be imminent. This effect would be accentuated in the case where AMR had been used. The possible consequences of this are that either the time-step size is reduced to severely hindering levels, or the overall method is as good as a traditional Eulerian scheme coupled to an "a priori" (in this case) mesh refinement method.

The method described in this Chapter resembles the node relocation scheme (r-method) mentioned in section 3.1.1 and possesses the characteristics of arbitrary Lagrangian- Eulerian schemes, while allowing the simulation of highly complex flow and material interface evolution through the method presented in Chapter 2. It was developed with the purpose of modelling coastal engineering applications and nature related phenomena, although the description does not lose it's generality and may be applied to a wide span of subject areas. The moving mesh method was developed pertaining to the ultimate goals that can be enumerated as follows:

1. Further reducing numerical diffusion of the interface through aiding the compressive action of the Hyper-C scheme by improving local resolution.

2. Allow the mesh resolution to follow the material interfaces and therefore reducing the

adaptive mesh optimisation frequency. The latter also reduces the effects of grid to grid interpolation.

3. Prolong and possibly improve grid quality, protecting optimal resolution until AMO is eventually required.

3.2.1 ALE formulation

A good description of the development of ALE conservation laws and constitutive equations was given by Liu et al. (1986). Also, a detailed explanation of the ALE frame of reference was written by Donea et al. (2004), Sarrate et al. (2001), and Armero and Love (2003), among other authors. For the sake of completeness, a demonstration of the ALE description and the corresponding change in the typical Eulerian reference for the Navier-Stokes equations is provided in this section.

Let there be a mapping function ψ which describes the movement of an infinite array of material particles $\mathbf{m}$ with respect to a moving reference which follows its own independent path. This reference frame is "bijectively" linked to each material particle, and will be referred to as $\widetilde{R_{xyz}}$. The equation of motion for the continuum expressed in this configuration is given by,

$$\tilde{\mathbf{x}} = \psi(\mathbf{x}_m, t) \tag{3.5}$$

where $\tilde{\mathbf{x}} = \psi(\mathbf{x}_m, 0) = \mathbf{x}_m$ and its inverse,

$$\mathbf{x}_m = \psi^{-1}(\tilde{\mathbf{x}}, t) \tag{3.6}$$

exists and is also bijective. Coordinates $\tilde{\mathbf{x}}$ correspond to those measured for each material particle with respect to $\widetilde{R_{xyz}}$ following motion descriptor ψ at any given instant t. The total derivative of material property ϑ in terms of configuration $\widetilde{R_{xyz}}$ is given by,

$$\left.\frac{\partial \vartheta(\psi(\mathbf{x}_m, t), t)}{\partial t}\right|_{\mathbf{x}_m} = \left.\frac{\partial \vartheta(\tilde{\mathbf{x}}, t)}{\partial t}\right|_{\tilde{\mathbf{x}}} + \frac{\partial \vartheta(\tilde{\mathbf{x}}, t)}{\partial \tilde{\mathbf{x}}} \frac{\partial \tilde{\mathbf{x}}}{\partial t} \tag{3.7}$$

which may be written in compact form,

$$\frac{d\vartheta}{dt} = \widetilde{\frac{\partial \vartheta}{\partial t}} + \widetilde{\frac{\partial \vartheta}{\partial \tilde{\mathbf{x}}}} \cdot \tilde{\mathbf{u}} \tag{3.8}$$

where $\tilde{\mathbf{u}}$ is defined by rate of change of the coordinates $\tilde{\mathbf{x}}$ with respect to time, for a fixed material particle,

$$\tilde{\mathbf{u}}(\mathbf{x}_m, t) = \left.\frac{\partial \tilde{\mathbf{x}}}{\partial t}\right|_{\mathbf{x}_m} \tag{3.9}$$

It is useful to mention that $\tilde{\mathbf{u}}$ must not be mistaken with velocity of the $\widetilde{R_{xyz}}$ reference frame

with respect to R_{xyz}. The latter velocity is given by,

$$\hat{\mathbf{u}}(\mathbf{x}_m, t) = \left.\frac{\partial \mathbf{x}}{\partial t}\right|_{\tilde{\mathbf{x}}} \tag{3.10}$$

and is defined as the rate of change of the spatial coordinates with respect to the inertial reference frame R_{xyz} of a point that is fixed to the moving reference frame $\widetilde{R_{xyz}}$. Considering the equation,

$$\mathbf{x} = \varphi\left(\psi^{-1}(\tilde{\mathbf{x}}, t), t\right) \tag{3.11}$$

and after applying the chain rule and algebraic manipulation, velocities $\hat{\mathbf{u}}$, $\tilde{\mathbf{u}}$, and $\mathbf{u}$ are related by,

$$\mathbf{u} = \left.\frac{\partial \mathbf{x}}{\partial t}\right|_{\mathbf{x}_m} = \frac{\partial \varphi\left(\psi^{-1}(\tilde{\mathbf{x}}, t), t\right)}{\partial t} = \left.\frac{\partial \mathbf{x}}{\partial t}\right|_{\tilde{\mathbf{x}}} + \frac{\partial \mathbf{x}}{\partial \tilde{\mathbf{x}}}\frac{\partial \tilde{\mathbf{x}}}{\partial t} = \hat{\mathbf{u}} + \frac{\partial \mathbf{x}}{\partial \tilde{\mathbf{x}}} \cdot \tilde{\mathbf{u}} \tag{3.12}$$

This relationship can then be used in equation (3.8) to obtain,

$$\frac{d\vartheta}{dt} = \widetilde{\frac{\partial \vartheta}{\partial t}} + \frac{\partial \vartheta}{\partial \mathbf{x}} \cdot (\mathbf{u} - \hat{\mathbf{u}}) = \widetilde{\frac{\partial \vartheta}{\partial t}} + \nabla \vartheta \cdot (\mathbf{u} - \hat{\mathbf{u}}) \tag{3.13}$$

which describes the total derivative of property ϑ in terms of the local derivative with $\tilde{\mathbf{x}}$ fixed and the relative velocities between the material particles and the velocity of the reference frames belonging to $\widetilde{R_{xyz}}$.

Applying the relationship obtained for the total derivative in equation (3.13) to the two fluid incompressible flow governing equations from Chapter 2 yields,

$$\nabla \cdot \mathbf{u} = 0 \tag{3.14}$$

$$\rho\left(\frac{\partial \mathbf{u}}{\partial t} + ((\mathbf{u} - \hat{\mathbf{u}}) \cdot \nabla)\mathbf{u}\right) = \nabla \cdot \sigma + \mathbf{f} \tag{3.15}$$

$$\frac{\partial C}{\partial t} + (\mathbf{u} - \hat{\mathbf{u}}) \cdot \nabla C = 0 \tag{3.16}$$

which are the conservation equations written in the moving reference $\widetilde{R_{xyz}}$ configuration. The ~ sign has been dropped here for clarity.

The core of the ALE method resides in the calculation of the grid velocity $\hat{\mathbf{u}}$. Observing equations (3.14), (3.14), and (3.16), the difference between grid and flow velocities is directly proportional to the value of the advective terms. As a result the grid velocity not only has control of the reference frame for calculations, but also control of the advection, will influence the convergence of the solution method. If the velocity of the grid matches the velocity of the flow, purely Lagrangian movement of the grid is obtained, and thus convective derivatives vanish. If, on the other hand, the grid velocity is null an Eulerian description is obtained.

3.2.2 Implementation of grid velocity into the solution algorithm.

The solution algorithm presented in section 2.4.7 was performed considering an Eulerian reference frame for calculations. Introducing mesh movement implies some changes to be introduced into the solution algorithm. The discretised formulation of the momentum equations is now given by,

$$\frac{\mathbf{M}^{n+1}\mathbf{U}^{n+1}-\mathbf{M}^{n}\mathbf{U}^{n}}{\Delta t}+\mathbf{A}\mathbf{U}^{n+1}+\mathbf{K}\mathbf{U}^{n+1}+\mathbf{Q}\mathbf{p}^{n+1}=\mathbf{F}^{n+1} \tag{3.17}$$

where the components of matrix $\mathbf{A}$ are now calculated via,

$$A=\int_{\Omega_e}\rho\mathbf{N}\mathbf{N}^T\left(\mathbf{U}^{n+1}-\hat{\mathbf{U}}^{n+1}\right)\nabla\mathbf{N}^T\,d\Omega_e$$

and recalling the control volume finite element discretisation for the concentration scalar field, given by,

$$\left(\mathbf{M}_{CV}^{n+1}+\mathbf{B}|_k^{n+1}\right)\mathbf{C}^{n+1}\big|_{k+1}=R|_k \tag{3.18}$$

where the flux matrix matrix $\mathbf{B}|_k^{n+1}$ now contains the fluxes related to the convective velocity $\left(\mathbf{U}-\hat{\mathbf{U}}\right)$, and $\mathbf{M}_{CV}^{n+1}$ is now present to account for the changes in the control volume shape due to mesh movements.

The resulting method is described in Algorithm 2 The ALE grid velocities are calculated through non-linear Picard iterations, where at least two iterations are required to obtain a solution for $\hat{\mathbf{U}}^{n+1}$. Initial guess is calculated from the previous timestep solution, given by $\hat{\mathbf{U}}^{n}$. Notably, a mass matrix $\mathbf{M}^{n+1}$must be calculated at each non-linear iteration.

Algorithm 2 Solution algorithm for incompressible Navier Stokes equations, modified to include ALE reference frame.

loop {timestep}
  loop {Picard iterations (k)}
    Calculate $\hat{\mathbf{U}}_k^{n+1}$ based on values at the current non-linear iteration, starting with $\mathbf{C}^n$, and $\mathbf{U}^n$ for the first Picard iteration.
    Obtain solution for $\mathbf{C}_k^{n+1}$by solving the modified equation (3.18) in m_{iter} iterations
    Use $\mathbf{C}_k^{n+1}$to calculate f^{n+1}
    Solve equation (3.17) using $\mathbf{p}_*^{n+1}=\mathbf{p}^n$ or the solution of the PPE equation if on the very first timestep.
    Use the intermediate solution $\mathbf{U}_*^{n+1}$ to calculate the pressure correction through equation (2.72).
    Use the pressure correction obtained in previous step to obtain $\mathbf{U}_k^{n+1}$ from equation (2.73).
    Update nodal positions $\mathbf{x}_k^{n+1}=\mathbf{x}^n+\hat{\mathbf{U}}_k^{n+1}\Delta t$ and update mass matrices $\mathbf{M}_k^{n+1}$for next non-linear iteration.
  end loop
end loop

3.3 Calculating grid velocity

There are a variety of methods by which mesh velocity may be calculated. Donea et al. (1982) specified a grid velocity at each timestep via an averaging time integration formula derived somewhat by trial and error, using grid velocities from a previous timestep. This was found insufficient to prevent the squeezing of some elements, especially in the fluid-fluid interface, so a corrective term that prevented the edges of the elements from getting too short was added. Aymone (2004) calculated grid velocity through a combination of what was referred to as the Average Method, which controls geometric distortion of the mesh, and a Gradient Method, which concentrates resolution on important physical phenomena. The methods were combined through a weighed average of the coordinates resulting from applying each method independently, with this weight being problem dependent. An interesting method is described by Baines (2002), where new nodal positions are generated by minimising a least squares norm based on an equidistribution functional and in a steepest descent approach. Unfortunately, this method had not been developed for time dependent problems. Wells et al. (2005) described a method for generating grid velocities to be used in ALE solvers, namely through two monitor functions defined by a state variable and its gradient, respectively. In the first one, the monitor function is density, leading to Lagrangian movement of the mesh. The second monitor function is dependent on the gradient of the density, with the purpose of concentrating resolution where there are large variations of density. There was, however, no description of the possibility of combining of both monitor functions.

In the method presented by this Chapter, $\hat{\mathbf{u}}$ is to be calculated through an iterative functional minimisation process (see, for instance Daniel (1969)). The functional is built based on characteristics of the mesh suitability to the results, which have user-defined influence, and characteristics of the flow velocity. The resulting grid velocity regulates between Lagrangian and Eulerian reference frames, while attempting to concentrate or take away resolution where needed without altering element connectivity. The method's intent is to be robust and problem independent, and hence its formulation is entirely non-dimensional..

The functional is composed of three parts or sub-functionals, and maybe written as,

$$\mathscr{F} = w_1 F_L + w_2 F_{GQ} + w_3 F_{IP} \tag{3.19}$$

where w_1, w_2, and w_3 are individual weights corresponding to each functional's influence. Appropriate values for these weights may vary depending on user requirements, however satisfactory values found through trial and error procedures were $w_1 = 0.5$, $w_2 = 1.0$, and $w_3 = 5.0$. The respective sub-functional components can be briefly described as follows:

1. Lagrangian functional F_L. This functional is a measure of the difference between grid velocity and flow velocity. It helps prevent stability and/or convergence problems, while promoting Lagrangian motion which reduces advective numerical diffusion.

2. Grid Quality functional F_{GQ}. This functional is a measure of the element shape and size with respect to metric values as described in Pain et al. (2001a).

3. Interface Proximity functional F_{IP} . This functional enforces the need for the nodes to move along the interface rather than across it, while secondarily reinforcing the Lagrangian functional in the proximity of the interface.

A description of each of the components of functional $\mathscr{F}$ is provided in sections 3.3.1, 3.3.2, and 3.3.3.

3.3.1 The Lagrangian functional

The Lagrangian functional takes into account the magnitude and direction of the grid velocity with respect to the flow by calculating a localised value f_L for each node,

$$f_{L_i} = \left(\frac{\Delta t}{h}\right)^2 \mathbf{c}_i^2 \quad i \in \{1, 2, ...N_n\} \tag{3.20}$$

where $\mathbf{c} = \mathbf{u} - \hat{\mathbf{u}}^k$ denotes the relative velocity between the grid and the flow. It is useful to note that the relative velocity is calculated by,

$$\mathbf{c}_i = \mathbf{u}_i^k - \left(\frac{\mathbf{x}_i^k - \mathbf{x}_i^0}{\Delta t}\right) \tag{3.21}$$

where $\mathbf{x}_i^k$, and $\mathbf{x}_i^0$ are node i's current (k^{th} iterate) and original (at beginning of timestep) coordinates, respectively, $\mathbf{u}_i^k$ is the flow velocity vector associated with node i measured at $\mathbf{x}_i^k$ obtained by finite element interpolation, h is the user-defined minimum element size for the unstructured mesh, and N_n is the total number of nodes in the mesh. If anisotropic minimum element size is used, then $h = \sqrt{h_x^2 + h_y^2 + h_z^2}$, where h_x, h_y, and h_z are the minimum element sizes in each Cartesian direction. Summing the contributions from each node yields the functional,

$$F_L = \sum_{i=1}^{N_n} f_{L_i} \tag{3.22}$$

which, when minimised, creates the tendency for the mesh to move with the flow.

3.3.2 The Grid Quality functional

The grid quality functional takes into account the suitability of the mesh with respect to represented results. An adaptivity-based metric, as described in detail in Pain et al. (2001a), is generated for each node and an element quality measure is established for each element surrounding the current node i,

$$f_{GQ_i}^e = \frac{1}{2}\left[\sum_{l=1,L_e} (\delta_l)^2\right] + \mu\,(q_e)^2\,, \quad e \in \{1, 2, ...N_{ce}\} \tag{3.23}$$

where δ_l is the length of each side of the element, and q_e is a measure of the shape quality, both with respect to metric space. N_{ce} is the total number of elements connected to node i, L_e is the number of edges associated with the class of element (for this thesis, tetrahedra), and μ is a value which controls the balance between shape and size in measuring the quality. A value of $\mu = 1$ is used for calculating this functional.

The grid quality functional's localised value is calculated from each element surrounding node i,

$$f_{GQ_i} = \left\| f^e_{GQ_i} \right\|_\infty \quad e \in \{1, 2, ...N_{ce}\} \tag{3.24}$$

where $\left\| f^e_{GQ_i} \right\|_\infty$ implies the infinity norm over the set of $f^e_{GQ_i}$ values at the elements surrounding node i.

Finally the functional's value is obtained by calculating,

$$F_{GQ} = \sum_{i=1}^{N_n} f_{GQ_i} \tag{3.25}$$

Minimising this functional allows the nodes to fit shape and size distributions which are created according to user-defined error measures. These error measures, computed at the beginning of the timestep for each node, are given by a monitor function, typically called a "metric", consisting of the Hessian matrix of a given field (i.e. velocity, concentration, etc.). The metric is then used to obtain $f^e_{GQ_i}$ from equation (3.23), which is in turn used to obtain f_{GQ_i} and finally F_{GQ}. More in-depth details of how the metric is created and used to generate δ_l and q_e can be found in Pain et al. (2001a), and Piggott et al. (2005).

3.3.3 The Interface Proximity functional

This functional exists only in elements which contain the material interface, and therefore where $\nabla C \neq 0$. Its localised value for an element e connected to node i is given by,

$$f^e_{IP} = \left(\frac{1}{\Delta t} \right)^2 \frac{\int \left[\mathbf{c} \cdot \nabla C \right]^2 d\Omega_e}{\int d\Omega_e} \quad e \in \{1, 2, ...N_{ce}\} \tag{3.26}$$

where Ω_e is the corresponding volume of each element.

Evidently, this functional value will vanish if either $\mathbf{c} = \mathbf{u}$, or if the gradient $\nabla C = 0$. However it will also vanish if the dot product $\mathbf{c} \cdot \nabla C = 0$, which in other words means that the relative velocity is parallel to the interface. Enforcing this then implies that the nodes are not allowed to penetrate the interface.

As was the case with the grid quality functional, the localised nodal value is also calculated from local neighbour element values by taking the infinity norm, which yields

$$f_{IP_i} = \left\| f^e_{IP_i} \right\|_\infty \quad e \in \{1, 2, ...N_{ce}\} \tag{3.27}$$

and the functional can be constructed as,

$$F_{IP} = \sum_{i=1}^{N_n} f_{IP_i} \tag{3.28}$$

3.3.4 Minimisation process

The minimisation process is locally carried out for each node, iteratively, through the method of steepest descent (see, for instance Snyman (2005) for details on optimisation methods), also known in practise as the gradient method. Steepest descent was chosen among other options, such as Newton's method, due to its simplicity.

The procedure consists of a nodal loop nested inside a k iteration loop. The first condition to be satisfied by a given node i to be considered for movement is given by,

$$f_i^k \geqq f_{min} \tag{3.29}$$

where,

$$f_i^k = w_1 f_{L_i} + w_2 f_{GQ_i} + w_3 f_{IP_i} \tag{3.30}$$

which means that unless the local functional value is superior to $f_{min} = 0.15$, node i is not considered for the k^{th} iteration, and the next node is checked.

Assuming the condition from equation (3.29) is satisfied, then a gradient of the local functional value is calculated and used to determine a new position $\mathbf{x}_i^{k+1}$ by calculating,

$$\mathbf{x}_i^{k+1} = \mathbf{x}_i^k - \lambda \nabla f_i^k \tag{3.31}$$

where $\mathbf{x}_i^k$ and ∇f_i^k are the vector of nodal coordinates and the functional gradient of node i at iteration sweep k, respectively, and $\mathbf{x}_i^{k=0}$are the nodal coordinates with which the procedure began and correspond to those from the beginning of the current timestep. Scalar factor λ is given by,

$$\lambda = \frac{\beta}{\left|\nabla f_i^k\right|} \tag{3.32}$$

where β is the result of a line search in the direction of the gradient. To obtain a value for the gradient, f_i is evaluated at positions determined by small displacements in the x, y, and z direction independently. That is, using a finite difference approximation,

$$\nabla f_i^k = \begin{pmatrix} \frac{\partial f}{\partial x} & \frac{\partial f}{\partial y} & \frac{\partial f}{\partial z} \end{pmatrix} \approx \begin{pmatrix} \frac{\Delta_x f_i}{\Delta x} & \frac{\Delta_y f_i}{\Delta y} & \frac{\Delta_z f_i}{\Delta z} \end{pmatrix} \tag{3.33}$$

where Δx, Δy, and Δz are the small perturbations, and $\Delta_x f_i$, $\Delta_y f_i$, and $\Delta_z f_i$ are the corresponding changes in the functional values after displacements in each direction. The component $\frac{\partial f}{\partial x}$ of the gradient ∇f_i may be calculated by,

$$\frac{\partial f}{\partial x} = \frac{f\left(x_i^k + \Delta x\right) - f\left(x_i^k - \Delta x\right)}{2\Delta x} \tag{3.34}$$

where x_i^k is the current x coordinate of node i. Analogous calculation may be performed for y and z components of ∇f_i. The gradient calculation utilises a second order finite difference estimator in the work presented, which proved to provide acceptable results, however other approximation schemes may be used. The small displacements are calculated by,

$$\Delta x = \frac{h_x}{L_x}, \quad \Delta y = \frac{h_y}{L_y} \quad \Delta z = \frac{h_z}{L_z} \tag{3.35}$$

where h_x, h_y, and h_z are the user defined absolute minimum element sizes in each Cartesian direction, and L_x, L_y, and L_z are the associated reduction factors. Values of 1000 were used for these factors in this work.

The β factor is obtained through a line search technique in the direction opposite to the gradient ∇f. The functional is re-evaluated at N_β equidistant points away from the original $\mathbf{x}_i^k$ coordinates, as shown in Figure 3.5.

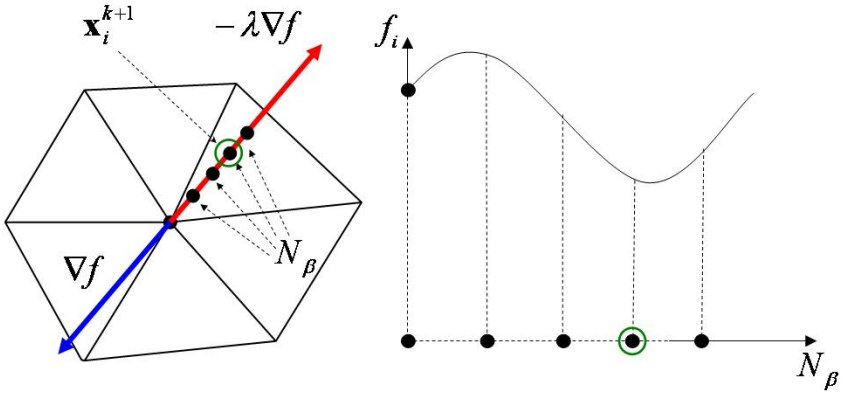


Figure 3.5: A two dimensional schematic of the minimisation process for node i in the computational mesh. Blue arrow shows the calculated functional gradient direction, while the red arrow depicts the line search direction. A graph is also shown indicating the choice of the new location for the node (circled in green).

A typical value of $N_\beta = 10$ was used in this thesis, unless otherwise stated, and a total search length equal to the distance to the closest node neighbour was used. The point that produces the minimum f_i value from these N_β is chosen. The new coordinates for node i may be accepted for the next iteration, subject to the condition that,

$$f_i^k - f_i^{k+1} \geqq \kappa \tag{3.36}$$

where $\kappa = 0.2$ was used in this work. Once the coordinates are, or are not, accepted and updated, the procedure continues on to the next node until all nodes are covered moment at which k^{th}

iteration ends. In order to optimise the procedure for the next iteration, functional values are calculated for all the nodes at their current (i.e. k^{th} update) position, and a bubble sort[6] algorithm is used to establish the order of the next nodal sweep. This is done in an attempt to attack first those nodes which are in the most compromised positions with respect to functional values.

Parameters f_{min} and κ are important in allowing the mesh to converge on a given position. If these parameters were set to zero, the mesh would oscillate indefinitely or until a maximum number of iterations was reached.

Iterations through equation (3.31) are carried on until convergence or a limit of iterations is reached. Once the iterative sweeps have finalised, the grid velocity vector field $\hat{\mathbf{u}}$ can be calculated as,

$$\hat{\mathbf{u}} = \frac{\mathbf{x}^k - \mathbf{x}^0}{\Delta t} \tag{3.37}$$

where $\mathbf{x}^k$, and $\mathbf{x}^0$ are the vector fields containing the coordinates of all the nodes in the new and in the original node configuration, respectively.

A summary of the complete method for obtaining grid velocity $\hat{\mathbf{u}}$ is provided in Algorithm 3.

Algorithm 3 Procedure for calculating grid velocity.

Require: Initialise $\hat{\mathbf{u}}$ to zero on all nodes.
Require: Calculate functional f_i^k for all nodes before iterations.
Require: Perform bubble sort on node list, highest functional value first.
loop {iterations of nodal sweeps (k)}
  loop {nodes i of the mesh}
    Calculate element volume Ω_e.
    Calculate local functional f_i^k through equation (3.30).
    if ($f_i^k \leq f_{min}$) **then**
      Skip to next node in the loop.
    else
      Obtain β factor by performing a line search.
      Choose β that minimises functional value.
      Calculate f_i^{k+1}.
      if ($f_i^k - f_i^{k+1} \geqq \kappa$) **then** {coordinate acceptance criteria}
        Update coordinates for node i using equation (3.31).
      end if
    end if
  end loop {nodes of the mesh}
  Perform bubble sort of the list of nodes for the next iteration based on new functional values.
end loop {iterations of nodal sweeps}

[6]Bubble sort is a typical comparison algorithm that operates by repeatedly checking though an array, comparing two elements at a time, and swapping their order if needed be.

3.4 Test problems

The following tests are meant to illustrate the robustness, flexibility, and accuracy of the mesh movement method developed in this Chapter. Even though some issues remain to be polished, such as optimal functional weights, line search points, and minimum thresholds f_{min} and κ, it will be shown in the following tests that the method performs well in conjunction with the interface modelling method presented in Chapter 2. It should be noted that even though the following tests are two dimensional, the algorithm presented at the end of section (3.3.4) is capable of functioning in three dimensions.

3.4.1 GCL test

An important issue to take into account when dealing with mobile meshes is to satisfy the geometric conservation law (GCL, see Thomas and Lombard (1979)), which states that for a uniform flow there should be no disturbances introduced by the arbitrary mesh motion. This seemingly trivial law, when violated, causes the apparition of artificial mass sources which lead to an increase or decrease in mass. Fluctuations in mass, in turn, lead to inaccurate results which typically have a magnitude that is proportional to the timestep and mesh size. This law was studied by Trépanier et al. (1991), in applications of a finite volume discretisation of the Euler inviscid flow equations. Mavriplis and Yang (2006) extended the idea to apply it to schemes of higher order of accuracy in time.

A test for the GCL law was developed by Trépanier et al. (1991), who proposed a meshed rectangular domain with fixed boundaries and constant state variables. Flow velocity was set to zero in the whole domain, and the mesh velocity was randomised. A similar approach is portrayed here, in which the mesh of the rectangular domain is extremely distorted, as shown in Figure 3.6.

As a result of this initial distorted mesh, and also providing proof that the developed method is functional, the grid velocity generated should move the nodes to produce a uniform mesh of higher quality elements. In this case, given the fact that the solution is uniform flow and state variables, a “good” element is the one possessing a shape very close to an equilateral triangular element. The resulting mesh after a few iterations is provided in Figure 3.7 on the following page.

Quantitative results can be observed in the graphs provided in Figure 3.8. The graph presents evidence that no perturbation is produced on the velocity field or other state variables such as density.

This test, as explained by Mavriplis and Yang (2006), is a necessary and sufficient condition that guarantees that the moving grid method does not affect the order of accuracy of the underlying solution method for momentum and scalar advection. It can be observed that the method developed in this Chapter passes this test.

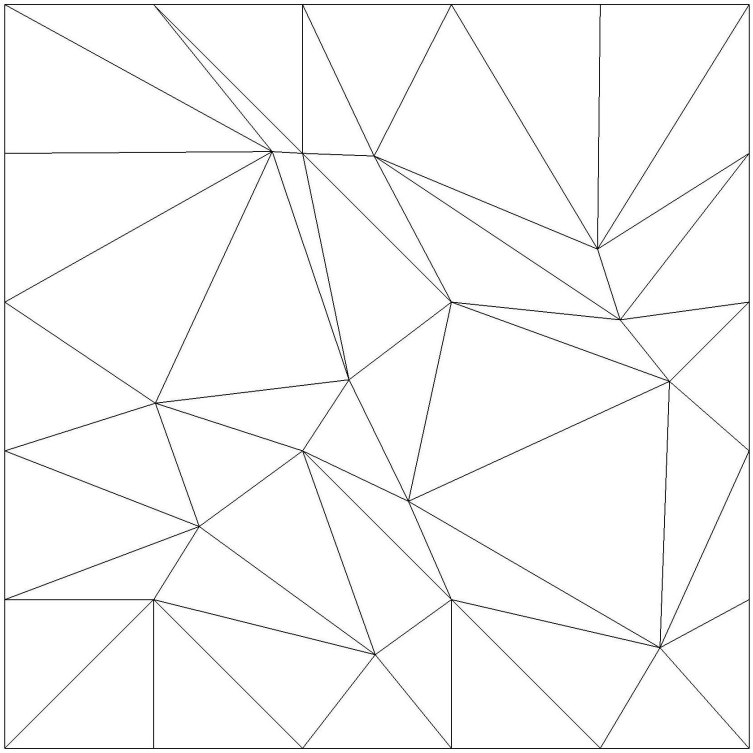


Figure 3.6: Graph of initial mesh for the GCL test.

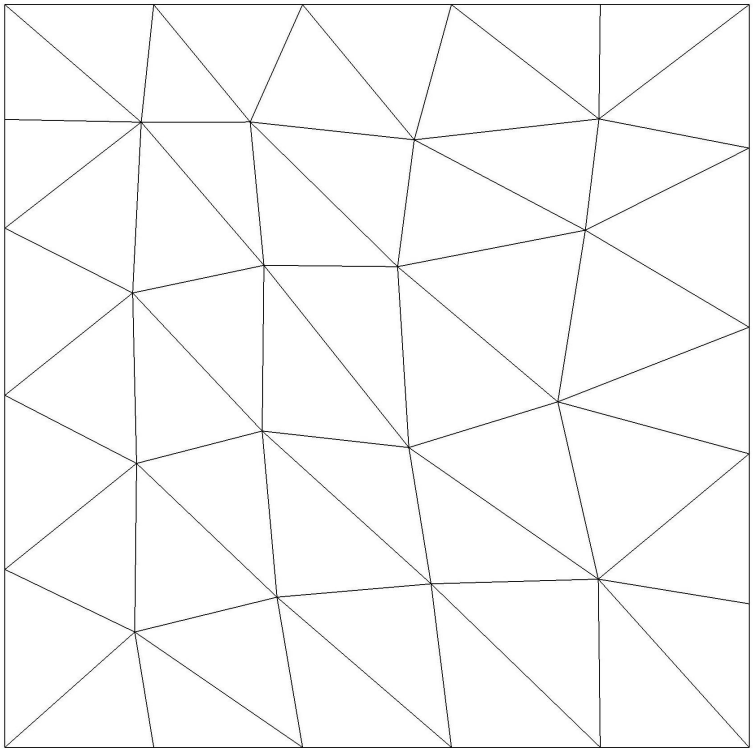
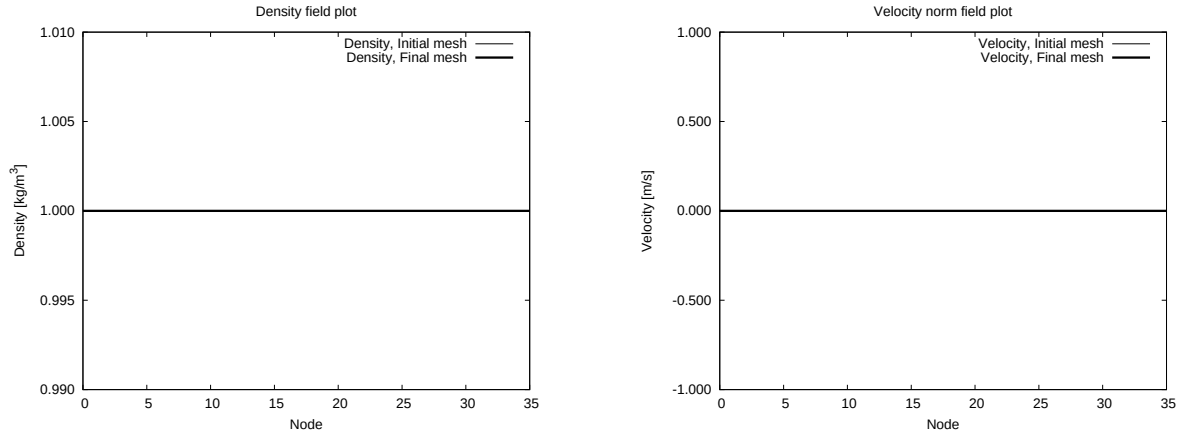


Figure 3.7: Graph of final mesh for the GCL test.



(a) (b)

Figure 3.8: Scalar field plot of density (a) and vector field plot of the velocity norm (b) for the initial mesh and final mesh.

3.4.2 Static interface capture

This two dimensional test was developed in this thesis to demonstrate the advantages of using ALE instead of, or in conjunction with, purely adaptive meshes. It is aimed to show how the sharpness of the interface is improved due to the fact that the moving mesh works together with the compressive advection scheme developed in Chapter 2. Results are compared against a purely adaptive mesh, and a static scalar field of concentration is used to establish a comparison case. Figure 3.9 shows the initial conditions for the simulation, constructed from the condition that any node to the left of the line $x = 0.5$ has a concentration value of 1, and the rest have a value of 0. This test is meant to evidence the fact that although both AMO and ALE schemes are meant to optimise the mesh, AMO introduces a certain amount of "smearing" of the interface, while ALE achieves a nearly identical level of resolution of the interface while keeping it sharper.

Results are obtained for an ALE case after several timesteps to allow the mesh to move without affecting convergence, and for an AMO case. The solutions for both cases can be observed in Figure 3.10. The result using ALE is sharper than the AMO result.

Further evidence of the method's high compatibility with interface tracking is shown in Figure 3.11, where a cross sectional line $y = 0.5$ is used to plot concentration values. In accordance to results from Figure 3.10(b), ALE allows for a sharper resolution of the interface, even though the resolution obtained through AMO is slightly finer.

The result from this test brings out a special feature of the method when coupled to HyperC. In contrast with mesh adaptivity, which simply interpolates results from one mesh to another, ALE allows the limited advection scheme to compress the interface further as the mesh moves closer to the interface in order to refine it.

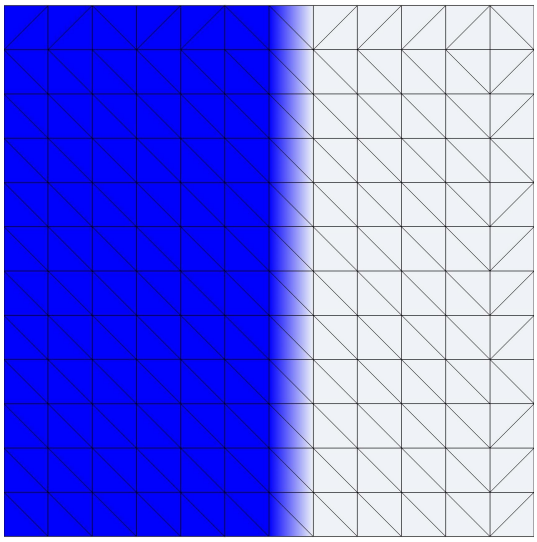
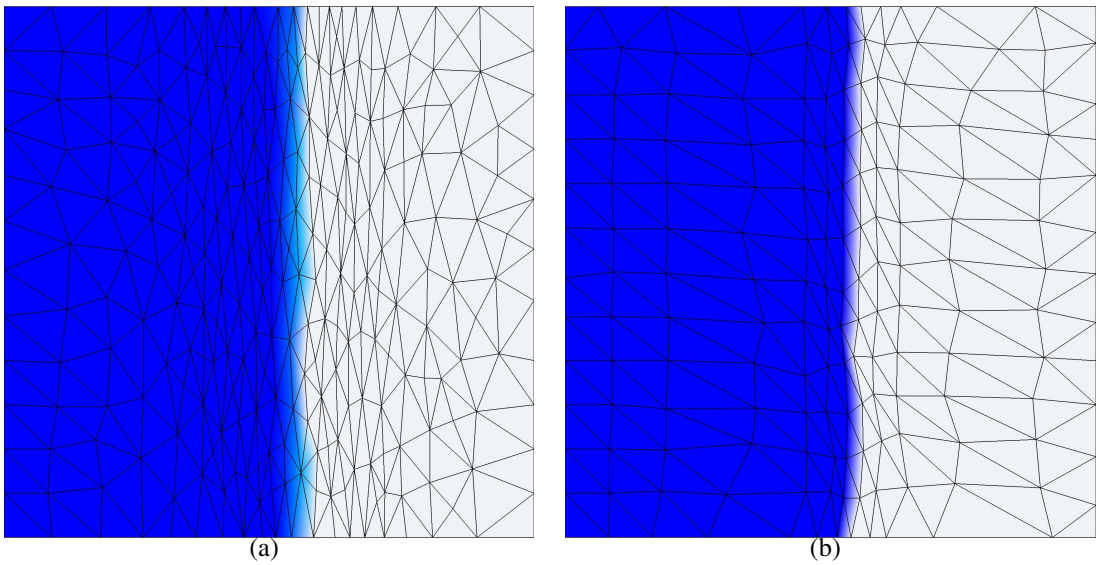


Figure 3.9: Initial mesh and concentration scalar field.



(a) (b)

Figure 3.10: Solution after several timesteps for (a) a mesh adaptive case, and (b) an ALE case.

3.4.3 Sloshing tank

A problem which is often used as a reference for testing free surface modelling is the sloshing tank. Nasar et al. (2008) performed experimental measurements of sloshing oscillations during regular wave induced motion of the tank. Liu and Lin (2008) performed two dimensional and three dimensional tests using a second order accurate VOF method. Bodard et al. (2008) used a spectral element discretisation coupled to an ALE formulation of the Navier- Stokes equations to study sloshing phenomena of high amplitude using moving grids. The latter was done by modelling the free surface, and hence the top or "lid" of the domain acts as a moving boundary.

The test proposed in this case is similar to the one used by Bodard et al. (2008) with the difference that the domain is larger. This is due to the fact that space needs to be included to model the air through the pseudo concentration method. A free surface configuration is

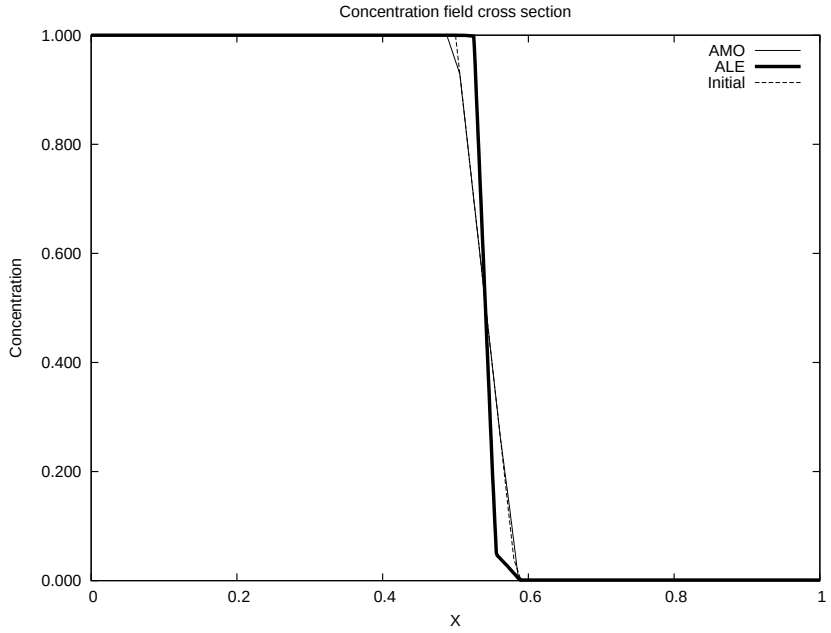


Figure 3.11: Cross section graph of the concentration scalar field across a line given by equation $y = 0.5$.

considered to described the initial water profile, and is obtained from nonlinear wave theory for finite amplitude standing waves, given by,

$$\eta(x,t) = A(t) \cdot cos(kx) \cdot cos(\omega t) - \frac{k \cdot A^2(t) \cdot cos(2kx)}{2 \cdot tanh(kh)} \cdot \left(cos^2(\omega t) + \frac{3 \cdot cos(2\omega t) - tanh^2(kh)}{4 sinh^2(kh)} \right) \tag{3.38}$$

where η is the wave height measured from the equilibrium state $y = h$, $A(t)$ is the amplitude, k is the wave number given by $k = 2\pi/\lambda$, and the dispersion relationship ω is given by $\sqrt{g \cdot k \cdot tanh(kh)}$. For simplicity, gravity is chosen as $g = 2\pi\lambda \cdot tanh^{-1}(kh)$, yielding a wave period of $T = 1$. Boundary conditions applied were free slip on the top and side walls, and no-slip at the bottom of the domain. Initial condition for velocity was set to zero in the whole domain. Initial concentration scalar field was set according to equation (3.38), where $A(0) = 0.2$ and $h = 1$. Details of the initial concentration field can be seen in Figure (3.12), while a summary of parameters for the simulation case are provided in Table 3.1. A sequence of results can be seen in Figure 3.15, while a mass conservation graph is shown in Figure 3.14.

From these results it can be seen that the moving mesh is able to follow the standing wave shape without the need for any adaptive mesh optimisation. No adaptive steps have been used for the duration of the simulation. This shows the excellent capabilities of the method, especially in cases where splitting and merging of volume fractions is not present. Conservation of mass remains within 1% , however these variations are attributed to slight local non-convergences (i.e. exceeding Courant number of 1) associated with the choice of weight factors and starting mesh configuration. A graph of conservation from $Re = 250$ is shown in Figure

3.14.

As further validation of the presented moving mesh method, Figure 3.15 shows a sequence of results for a low amplitude ($a \ll h$) sloshing simulation. In this sequence, the line plotted in red represents the evolution of the free surface as given by equation (3.38). A higher Reynolds number was used in this case to prevent wave dissipation due to viscous effects. A lower amplitude was used to prevent the onset of non-linear effects, which would invalidate equation (3.38). Good agreement is observed between the simulation and the analytical prediction. Slight deviations can be attributed to the fact that the flow is not strictly irrotational nor inviscid as it is assumed in the theoretical solution. The initial mesh is the same as the one shown in Figure 3.12, and Table 3.2 provides a summary of the rest of the parameters used for this simulation.

Figure 3.16 shows a graph of the computation time taken to produce the low amplitude sloshing simulation, while Figure 3.17 depicts the computational cost per timestep. A comparison was performed against the same simulation while using AMO and a fixed mesh. All three cases were run on a single processor workstation, and the mesh adaptive process was setup to maintain roughly the same amount of nodes as the moving mesh case. Figure 3.17 shows the overall computational cost per timestep. It can be observed that the ALE simulation's computational cost is higher than in the other two cases, however the fixed mesh case has a limited interface resolution given that the mesh does not start in an optimised state.

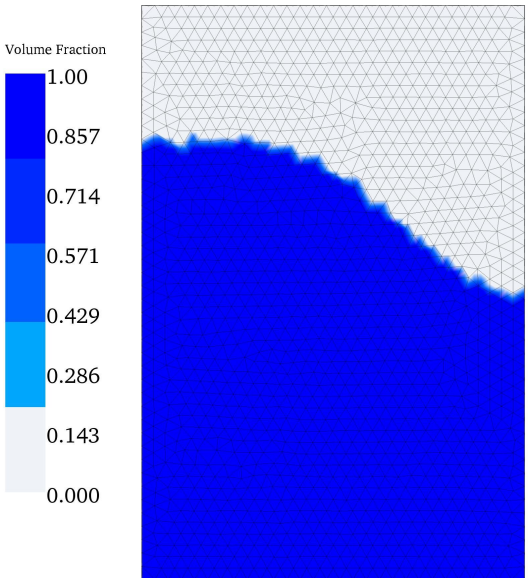


Figure 3.12: Initial concentration field for the sloshing tank problem

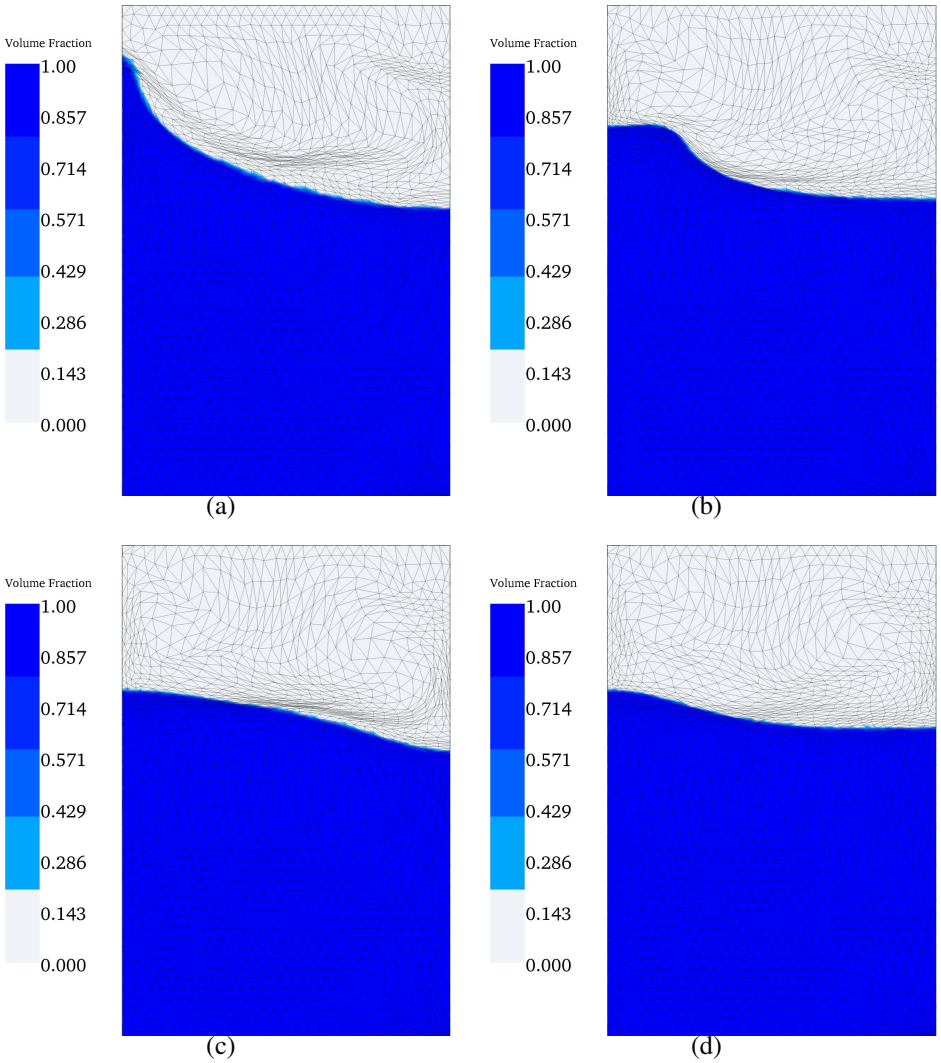


Figure 3.13: Sloshing sequence of results showing frames at (a) $t = 1$, (b) $t = 2$, (c) $t = 3$, and $t = 4$ seconds.

Parameter	Value
Reynolds N. $\left[\frac{\rho h\sqrt{gh}}{\mu}\right]$	250
ρ_1(light fluid) $\left[\frac{kg}{m^3}\right]$	1.0
ρ_2(heavy fluid) $\left[\frac{kg}{m^3}\right]$	1000.0
μ_1 $\left[\frac{N\cdot s}{m^2}\right]$	1.78×10^{-5}
μ_2 $\left[\frac{N\cdot s}{m^2}\right]$	14.153
Domain height $[m]$	1.5
Domain width $[m]$	1.0
Still water level $[m]$	1.0
Gravity (y-dir) $\left[\frac{m}{s^2}\right]$	-12.61
Initial Amplitude $[m]$	0.2
Mesh Nodes	1459

Table 3.1: Table of parameters for the sloshing test case.

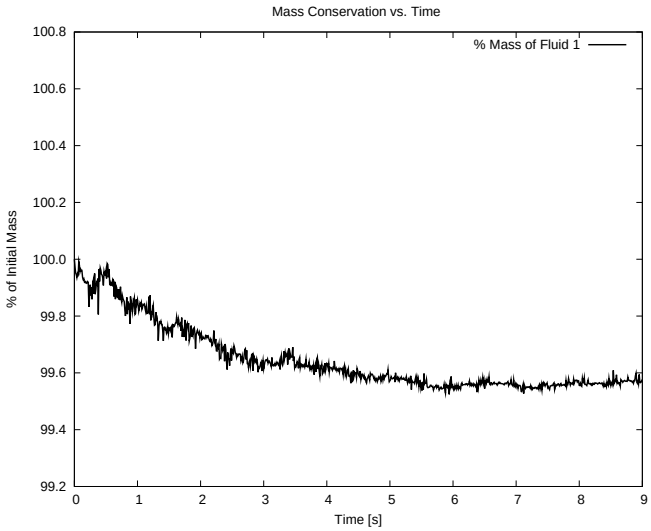


Figure 3.14: Mass conservation graph obtained for the sloshing simulation with $Re = 250$.

Parameter	Value
Reynolds N. $\left[\frac{\rho h\sqrt{gh}}{\mu}\right]$	3.5×10^6
ρ_1(light fluid) $[\frac{kg}{m^3}]$	1.0
ρ_2(heavy fluid) $[\frac{kg}{m^3}]$	1000.0
μ_1 $\left[\frac{N \cdot s}{m^2}\right]$	1.78×10^{-5}
μ_2 $\left[\frac{N \cdot s}{m^2}\right]$	0.001
Domain height $[m]$	1.5
Domain width $[m]$	1.0
Still water level $[m]$	1.0
Gravity (y-dir) $[\frac{m}{s^2}]$	-12.61
Initial Amplitude $[m]$	0.05
Mesh Nodes	1459

Table 3.2: Table of parameters for the sloshing test case.

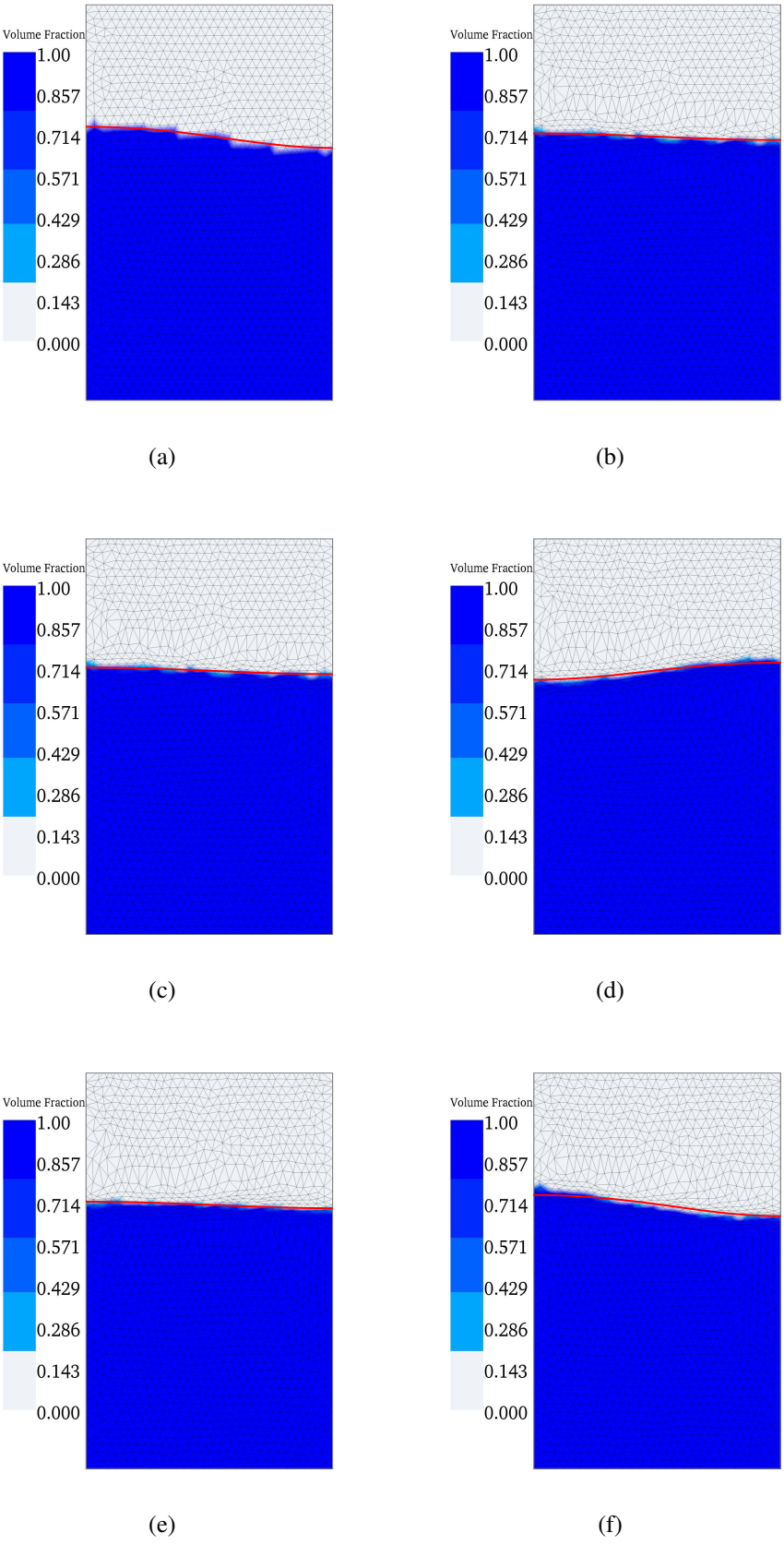


Figure 3.15: Low amplitude sloshing sequence of results showing frames at (a) $t = 0.0$, (b) $t = 0.2$, (c) $t = 0.4$, (d) $t = 0.6$, (e) $t = 0.8$, and (f) $t = 1.0$ seconds.

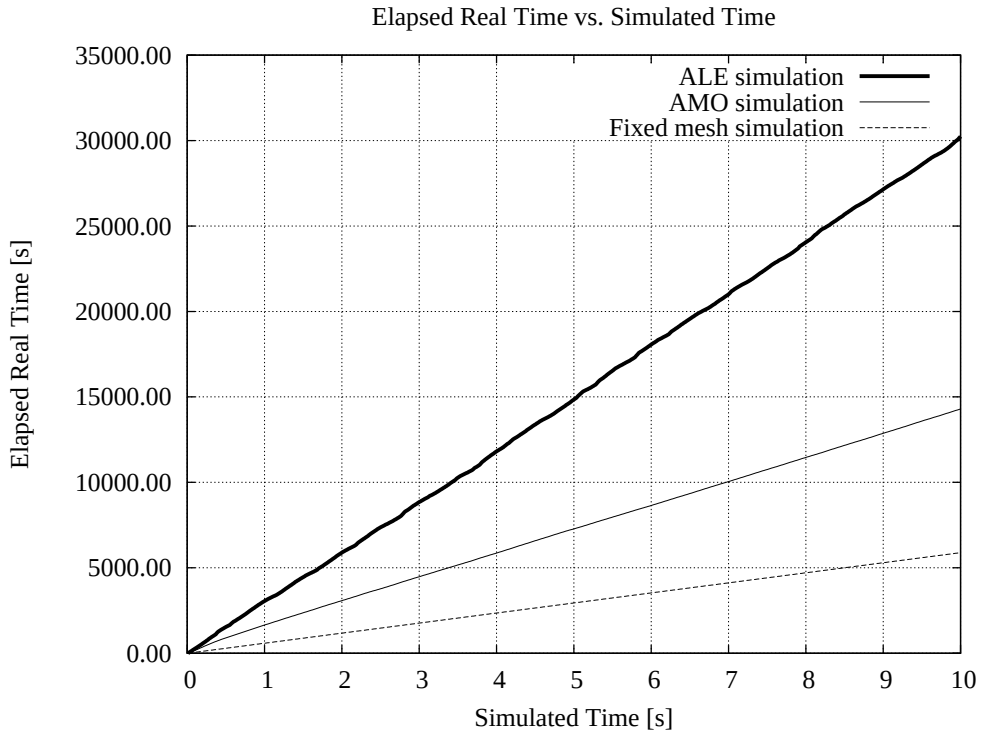


Figure 3.16: Graph of accumulated real time versus simulated time, for the sloshing simulation with $Re = 3.5 \times 10^6$.

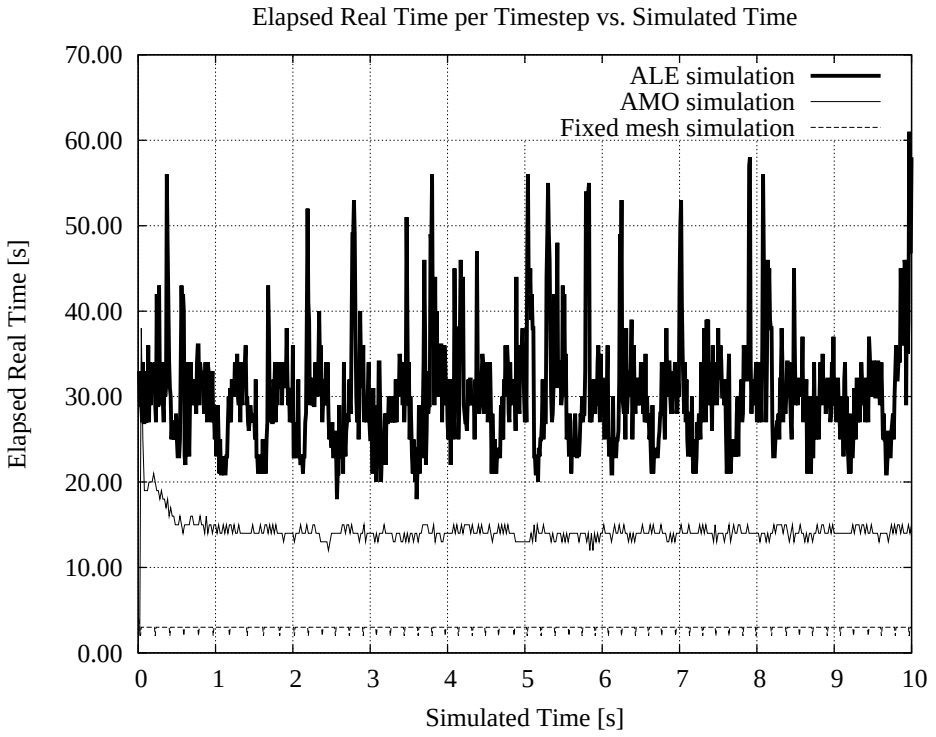


Figure 3.17: Graph of computational real time spent per timestep, for the sloshing simulation with $Re = 3.5 \times 10^6$.

3.5 Conclusions

A moving mesh method to complement the compressive advection scheme presented in Chapter 2 has been developed in this Chapter. The method brings together arbitrary Lagrangian-Eulerian reference frame characteristics and error measures typical from adaptive mesh optimisation techniques. These two aspects are merged to obtain a method that optimises nodal positions based on reducing advective terms while increasing or maintaining grid quality.

The functional combination is conceived from the idea that the method automatically decides which direction is best for moving each individual node, finally reaching an equilibrium. The method was still in its developmental stages when this thesis was presented, considering the innumerable amount of functional weight combinations and other parameters to choose from. However, it is the idea for this Chapter to provide another starting point for future research in the area of mesh movement with special focus on sharp interfaces of immiscible fluids.

Results presented depict a method that is able to predict the movement of interfaces and allow the computational grid to follow it accordingly. From Figure 3.11, it can be seen that the method is able to make further use of the compressive advection thus increasing sharpness of the interface.

It is important to note that the presence in the ALE conservation equations of transport terms accounting for arbitrary grid velocities of the fluid nodes should automatically guarantee conservation of mass, momentum, and energy during the continuous mesh movement process. This represents a significant advantage over the techniques which remap the mesh at fixed times in Lagrangian programs. Furthermore, the non-dimensional formulation of the functionals allow for a problem independent framework for optimisation, thus reducing the amount of fine tuning needed for usage parameters.

An important conclusion obtained from this Chapter is the fact that grid velocity calculation algorithms need careful attention and are not usually obtained in a straightforward manner. Factors such as the line search and minimisation process need careful analysis and as this was not done to completion due to the broad subject coverage of this thesis. It is logical to conclude that movable reference frame methods are an optimal way to deal with problems of moving interfaces. Moving meshes are highly convenient when dealing with problems involving unwanted numerical diffusion and grid to grid interpolation error and further development of the method presented should be a matter of future research.

Chapter 4

Fluid structure interaction

Fluid interaction with structures is of high importance within many engineering applications. Some of these applications include metal forging and extrusion, fatigue testing on aircraft wings, fuel tank sloshing, drop testing of containers, inkjet printers, wind force analysis on tall buildings, and most importantly for the subject of this thesis, water wave interaction with structures. In all of these, interaction with fluid forces can cause complex dynamic behaviour in the structure. For this reason, solid-fluid coupling modelling is of crucial importance when performing design tasks.

Arguably, and although methods and fundamental physical phenomena have much wider applications, fluid structure interaction (FSI) stems from aerospace engineering. Hence, for an introduction to the field, see Bisplinghoff et al. (1996), Fung (1955), and Dowell et al. (1995). As for coastal engineering studies of FSI, Karadeniz (1993) investigated wave current effects on deep water offshore structures, and Lara et al. (2008) studied wave interaction with low-mound breakwaters applying the VOF method (see Chapter 2). An extensive plethora of FSI applications can also be found in the conference proceedings by Chakrabarti et al. (2005).

With the advancement of computer technology, numerical simulation of fluid structure interaction (FSI) problems have become increasingly possible. However, due to the fact that fluids and solids behave differently (e.g. have different constitutive properties), the creation of a multi-physical modelling technique which describes both is no easy task.

It is the main intent of this Chapter to portray a numerical fluid structure interaction model which is flexible and sufficiently robust to handle most types of problems with particular emphasis to coastal engineering applications. A multiphase approach was followed when modelling both the fluid and the solid, and thus it perfectly fits with the material presented in Chapter 2.

The main characteristics of the method presented here when compared to others in the literature lies in the following points

1. Particles may be modelled at any relative size with respect to the computational grid of the fluid phase. The focus of this thesis is mainly on particles larger than grid size scale, however the theory is presented for the future development of a subgrid scale model which

will allow co-existence of multi- scale size of particles.

2. Solid behaviour (e.g. deformation, translation, rotation, contact detection, contact forces, internal stresses, etc.) is modelled through the discrete element method (DEM) or the finite discrete element method (FEMDEM). See Chapter 1.

3. In cases where the solid behaviour models allow (e.g. fracture-capable FEMDEM), solids may fragment. Coupling fragmentation of solids within fluids is difficult task (see, for instance, Mohammadi and Pooladi (2007); Munjiza et al. (2000)), specially in three dimensions.

4. Many solid particle models such as DEM or FEMDEM require structure penetration to calculate contact forces and interaction between the solids. This poses a problem again for the typically used two mesh approach, however it is not a problem for the multiphase approach followed here.

5. The method is combined with an elaborate adaptive mesh optimisation technique, which allows the resolution of complex flow details and structure shape while keeping memory usage at a minimum.

This Chapter begins with an introduction to fluid structure interaction followed by multiphase flow and its application to modelling fluids and solids. Next, the governing equations are presented followed by techniques used to model particles which are larger than the grid scale size and others which are smaller than the grid scale size. Drag calculation is presented, followed by the method's implementation into the computational framework of Fluidity. Finally, steady and unsteady flow test cases are described. Coupling with DEM and FEMDEM codes, and comparison to other numerical methods and experimental work found in the literature are also presented.

4.1 Background

Numerical simulation of FSI faces several considerations when determining the modelling approach. One of these is the fact that, considering that structures and fluids behave differently, time and space discretisation formulations are relatively different for each and hence finding a suitable coupled solution method might be difficult. This issue is of particular importance when using two independent solution codes to solve the interaction. Another important obstacle is the geometric discretisation of the domain. The geometric subdivision of the fluids and FEMDEM domains may be quite different which is particulary important at the solid-fluid interfaces. This leads to methods normally named non-matching mesh methods. See, for instance, Hansbo and Hermansson (2003) on penalty techniques for treating non-matching mesh methods, and Dohrmann et al. (2000) on the master-slave concept. Finally, the performance of the method as

a whole might be hindered by effects from the interaction between the two solution techniques. While independently each code might be accurate and stable, it may not be the case for the coupled scheme. It is clear then that a compromise must be obtained to ensure not only that each solution method is working well independently, but that they are working well together.

As with other areas of application of numerical modelling, there are many approaches for simulating FSI. There are monolithic approaches (see, for instance Blom (1998)) in which the solution algorithm is fully coupled and unified and hence some of the above mentioned obstacles are overcome. Unfortunately, this type of approach seems not to be efficient when solving systems with a large number of degrees of freedom (see Ross (2006)). Wandinger (1998), for example, used the Craig Bamptom method (see Craig and Bampton (1968)) to reduce the resulting coupled system of equations.

It is, however, often advantageous to subdivide the complete method into two separate operator components and to perform a staggered evolution. Felippa and Deruntz (1984) developed a staggered partitioned method which handled the modelling of hull cavitation in underwater shock problems. Explicit time stepping was used for the fluid equations while implicit time stepping was used for the structure. This method was later improved by Sprague and Geers (2004) who introduced a spectral element formulation among other features, and developed a procedure which incorporates predictor corrector method where the predictor step should be carefully parameterised to avoid stability degradation. Young (2008) combined the boundary element method (BEM) and the finite element method to model fluid interaction with composite marine propellers. Mitra and Sinhamahapatra (2008) used a two dimensional finite element model combining quadrilateral elements for the compressible inviscid fluids and line elements for the structures, solving a coupled system iteratively. Longatte et al. (2008) investigated the staggered time advancement of code coupling methods. Borazjani et al. (2008) extended the interface capturing method developed by Ge and Sotiropoulos (2007) to model large displacements in three dimensional structures by introducing a Lagrangian grid and using a ray tracing algorithm to keep track of its position with respect to the background mesh. Ross (2006) presented a Lagrange multiplier method which facilitates the combination of spatial and time discretisations for staggered approaches. All of the mentioned approaches prove that staggered solution methods, although not ideal, are a very popular technique and provide a successful compromise.

The method proposed in this Chapter could be thought of as a step between staggered and monolithic methods. A set of governing equations are solved for the fluid, while another set is solved for the solid structures. The two solutions are coupled through the introduction of multiphase terms in the fluid governing equations. The coupled system is then solved through non-linear Picard iterations. Mesh dependence can therefore be fully focused on the fluid domain, where adaptive mesh optimisation is used to resolve flow details and structural shapes. However, if the particles are too small to be resolved by the mesh, the technique applies param-

eterised drag forces[1] on the solids and an equal and opposite force on the fluids. The resulting method allows for flexibility of using a DEM code to define rigid sphere movement where geometries are described through point and radius data, or through the FEMDEM method, which allows for irregular shapes defined by a mesh independent from the fluid. FEMDEM particles in general have the capability to deform and/or break. Furthermore, due to the fact that contact detection must be calculated for both DEM and FEMDEM approaches, timestep sizes are much smaller for the DEM or FEMDEM models than for the fluid model. This problem is solved by sub-cycling the solution for the solid movement within a fluid timestep.

4.2 Multiphase approach and coupled solvers

In an effort to better introduce the theory behind the solid-fluid interaction model presented in this Chapter, a brief background on multiphase flow is provided. It must be noted, however, that multiphase flow is a very broad field of study with innumerable amount of application areas and approaches, and hence an attempt at a rigid generalisation is somehow difficult, if not futile. Excellent background reading on multiphase flow can be obtained from books by Clift et al. (1978), Gidaspow (1994), and Brennen (2005). A good review can also be found in Drew (1983).

It is typically found in the literature that multiphase models are divided into two main areas: disperse flows and separated flows. The first kind deals with flows that contain discrete particles, while the second models continuous fluids delimited by an interface. As an excellent example, the pseudo concentration method presented is Chapter 2 can be considered a multiphase separated flow model. Disperse flow modelling, in turn, is divided into two methods. One is the trajectory method, in which the finite particles (disperse phase) are moved about within the continuous fluid phase using parameterised forces. The other one is the two-fluid model, in which the discrete particles are treated as a continuous phase able to mix and interact with the fluid phase.

Having multiple phases interacting with each other gives rise to modifications to the usual governing conservation laws. In a general case, this involves the introduction of mass interaction, force interaction, and energy interaction terms. In this work, mass is not transmitted between the phases which means that there are no chemical reactions or state changes (e.g. sublimation, vaporisation, melting), and hence mass interaction terms vanish. Furthermore, there is no additional energy to be exchanged between the phases since the whole system is considered isothermal, hence the energy interaction term is also null. The force interaction term, arising in the conservation of momentum law, is the critical term which will define the interaction between the phases. Important contributions regarding coupling techniques have been presented by Wallis (1991); Arnold et al. (1989); Drew (1991); Sangani and Didwania (2006);

[1] parameterised drag refers to the fact that drag may be calculated from drag curves obtained either analytically (i.e. very low Re) or experimentally.

Zhang and Prosperetti (1994); Shirvanian et al. (2006); Zhao et al. (2002b); Kajishima and Takiguchi (2002); Takeuchi et al. (2008), however, the problem of providing a consistent link or generalised model for multiphase flow at any flow regime remains a very difficult task.

For the material presented in this Chapter, the existence and interaction with fluids of solid particulates was done following a combination of the two disperse multiphase flow models. Considering the scope of applications of this thesis, it should be assumed that in general there are three materials involved in most of the simulations. These materials are air, water, and solid structure. For the sake of simplicity, both air and water are considered together as the fluid phase while the second is the particulate or solid phase. Coupling was achieved through the implementation of two Lagrangian particle solvers into the Fluidity framework. The first solver, a discrete element method (DEM) solver (see Figure 4.1), treats each individual particle as a sphere or clusters of analytically primitive shapes, while allowing them to collide and interact with each other and domain boundaries. The second solver is the FEMDEM solver which allows the modelling of particle interaction where shapes are defined by a mesh of, for example, tetrahedra. Internal particle stress and deformation are determined through a finite element model associated to the grid which defines the shape. See Figure 4.2 for frames depicting a two dimensional version of the method, while Figure 4.3 shows the recently developed three dimensional version. Particle interactions are modelled through contact detection and force calculation techniques (see Munjiza (2004)) , while eventual fracture[2] may also be modelled.

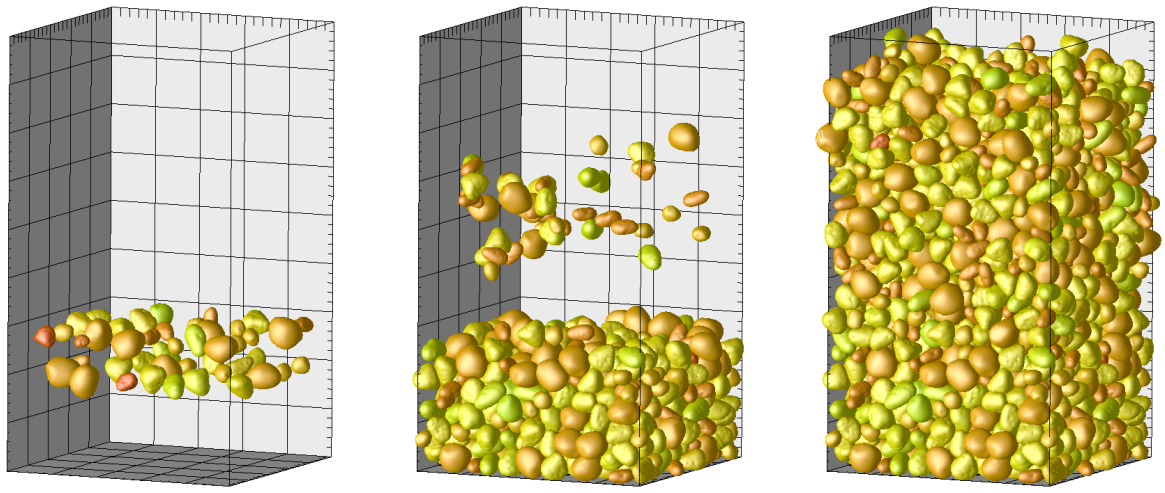


Figure 4.1: Dynamically deposited pack of DEM particles. Each particle is represented as a cluster of simpler shapes, such as spheres or other more sophisticated shapes (i.e. parametric superquadrics). Frames are courtesy of Xavier Garcia.

[2]At the submission date of this thesis, fracture modelling through FEMDEM solvers was limited to two dimensions.

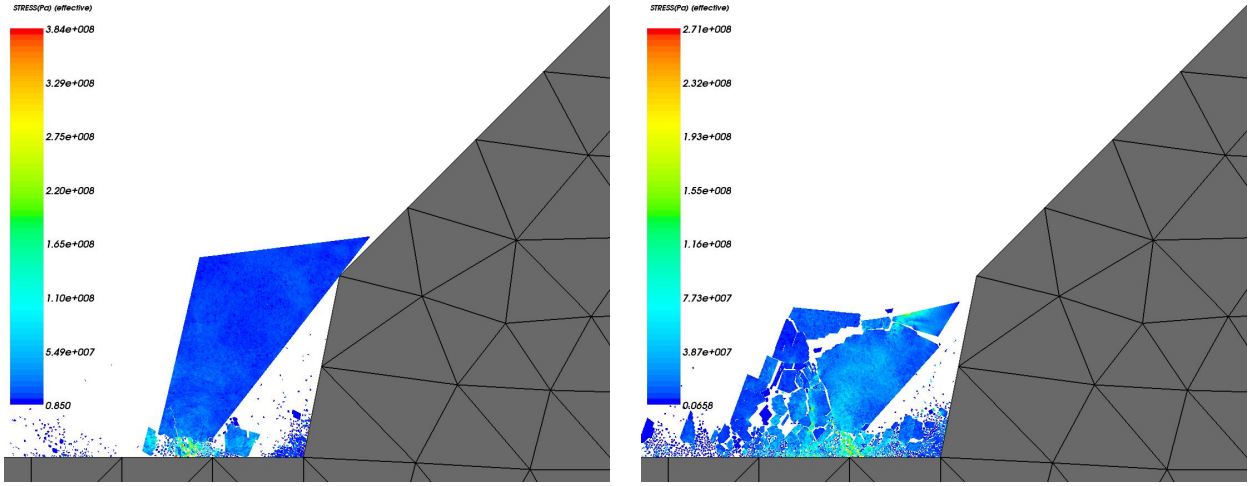


Figure 4.2: Frames depicting a two dimensional FEMDEM simulation of an avalanche process. (courtesy of Dr. Jiansheng Xiang using y2d code originally developed by Professor Antonio Munjiza)

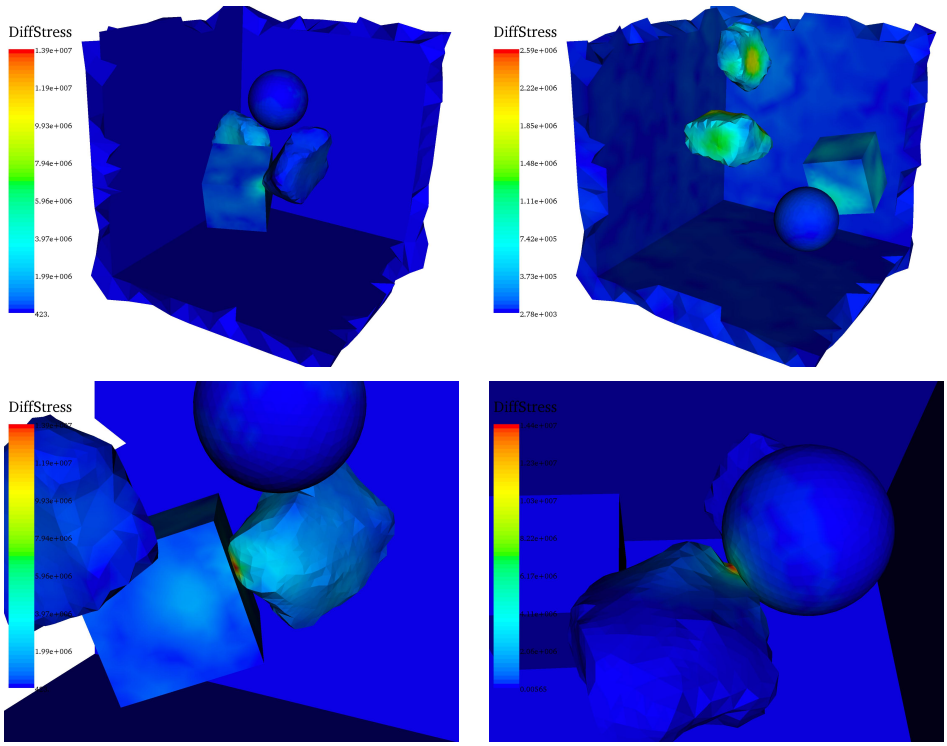


Figure 4.3: Three dimensional FEMDEM simulation of random interaction of different particle shapes. (courtesy of Dr. Jiansheng Xiang)

The following sub sections set the scene, as far as governing equations are concerned, for the solid- fluid interaction. Sections 4.2.1 and 4.2.2 describe the fundamental governing equations used in this Chapter to derive the discretisation formulation ultimately yielding the solid-fluid interaction modelling technique. Further details on the algorithmic coupling of DEM and FEM-DEM with the Fluidity framework is presented later in section 4.3.

4.2.1 Mass conservation law

The typical formulation of the mass conservation law for a multiphase problem is given by,

$$\frac{\partial}{\partial t}\sum_k \alpha_k \rho_k + \sum_k \nabla \cdot (\alpha_k \rho_k \mathbf{u}_k) = 0 \quad \forall k \in \{f, s\} \tag{4.1}$$

where k can either refer to the solid (s) or the fluid (f) phases, α_k refers to the concentration or volume fraction of each phase, and ρ_k and $\mathbf{u}_k$ are the density and velocity, respectively, corresponding to each phase. Considering incompressibility, and assuming that there can be no changes in mass in each of the phases due to chemical reactions or changes of state, equation (4.1) becomes,

$$\nabla \cdot \left(\alpha_f \mathbf{u}_f\right) = -\nabla \cdot \left[(1-\alpha_f)\mathbf{u}_s\right] = -\nabla \cdot (\alpha_s \mathbf{u}_s) \tag{4.2}$$

which, considering that $\alpha_f + \alpha_s = 1$ can be re-written as,

$$\nabla \cdot \left(\alpha_f \mathbf{u}_f\right) = -\nabla \cdot \left[(1-\alpha_f)\mathbf{u}_s\right] \tag{4.3}$$

where it may be noted that the term $-\nabla \cdot (\alpha_s \mathbf{u}_s)$ acts like a mass source or sink for the fluid, depending if the solid particle is entering or leaving a region of the domain. When $\alpha_f = 1$, then the region considered consists only of fluid and the right hand side of equation (4.3) vanishes leaving the typical continuity equation for a single phase fluid.

4.2.2 Momentum conservation law

The momentum equation for the incompressible fluid taking into account the presence of the disperse phase, s, is expressed by,

$$\alpha_f \rho_f \left[\frac{\partial \mathbf{u}_f}{\partial t} + \left(\mathbf{u}_f \cdot \nabla\right)\mathbf{u}_f\right] = \nabla \cdot \sigma + f_{D_f} + f_b \tag{4.4}$$

where f_D is the interactive force per unit volume (see, for instance, Brennen (2005)), also viewed as a momentum source, which links the fluid phase to the solid phase equation. Viscous and pressure stresses are included in the stress tensor σ. Notably equation (4.4) does not take into account a moving reference frame. For notation simplicity, formulation derived in this Chapter was done in an Eulerian reference frame. However, the material presented in Chapter 3 may easily and analogously be applied here.

In accordance with Newton's third law of motion, the interactive force f_D essentially establishes a balance of forces together with the force acting on the solid phase,

$$f_{D_f} + f_{D_s} = 0 \tag{4.5}$$

and is composed of terms to account for the relative motion between the phases. It can be

expressed as,

$$\mathbf{f}_{D_f} = \kappa_D \left(\mathbf{u}_f - \mathbf{u}_s \right) \tag{4.6}$$

where κ_D is referred to as the fluid-particle friction coefficient (see, for instance, Gidaspow (1994)). This coefficient represents an absorption or inverse time scale relaxation factor for modelling the transition of velocities between the fluid and the particle phase submerged in the fluid domain. It should be noted here that there are two philosophies when analysing the inter-phase forces. Gidaspow (1994) defines two different models, namely A and B, in which the fraction of inter-phase forces due to the pressure gradient of the continuous fluid phase are included in the disperse phase equation rather than the fluid equation, and vice versa. Including this term, as also suggested by Brennen (2005), would involve adding a term $-\alpha_s \nabla p$ to equation (4.6). However, it is omitted for numerical convenience and in order to yield a symmetric positive definite pressure matrix. The total force calculation process is explained in section 4.4.

Assuming the only common body force acting on both phases is gravity, the force term f_b is given by,

$$\mathbf{f}_b = \rho_f \mathbf{g} \tag{4.7}$$

where $\mathbf{g}$ is the gravity acceleration vector.

The equation for the conservation of momentum for the fluid phase can be re-written as,

$$\alpha_f \rho_f \left[\frac{\partial \mathbf{u}_f}{\partial t} + \left(\mathbf{u}_f \cdot \nabla \right) \mathbf{u}_f \right] = \nabla \cdot \sigma + \kappa_D \left(\mathbf{u}_f - \mathbf{u}_s \right) + \rho_f g \tag{4.8}$$

Equation (4.8) is the basis for the solution method presented in the following section, and corresponds to Model B suggested by Gidaspow (1994), pp 151-152. For convenience, subscripts will be dropped from α at this point, and α will correspond to the solid volume fraction (i.e. $(1-\alpha)$ will be used for fluid phase volume fraction). Furthermore, subscript f will be dropped from fluid velocity.

4.3 Geometry mapping and friction coefficients

As was mentioned earlier in this Chapter, FSI is a vast area of research and its computational simulation methods typically rely on a staggered solution technique which couples solid and fluid momentum equations. In order to apply a multiphase approach to the numerical modelling of these problems, and contemplating equation (4.8), it is clear that a method must be devised to allow successful communication of forces between the phases. Along with this, though, comes the fact that individual solid particles which compose the disperse phase may differ in size and shape. This last detail poses a particular problem for the computational mesh, since, for example, particles might be too small for their size to be resolved, or their shape irregular enough that their effects on the fluid may not be approximated by typical analytical shapes such as spheres or ellipses. Also, there might be modelling cases involving diverse particle sizes

which could differ by several orders of magnitude.

The method presented herein relies on the flexible calculation of the κ_D particle friction coefficient and a complex geometric mapping method. The particle friction coefficient calculation allows for the solution of flows involving different particle sizes, differentiating between those that are and are not resolved by the local mesh resolution. The geometric mapping method creates the solid concentration scalar field α, by projecting respective particle shapes onto the computational mesh. Obtaining values for κ_D and α are the main objectives which allow the method to function along with the definition of the velocity vector field of the solid phase $\mathbf{u}_s$ which is mapped in a similar way to α.

From equation (4.8), calculation of total drag force F_D exerted on a given particle p is carried out through the integration of the drag term,

$$\mathbf{F}_{D_p} = \int_{\Omega} \kappa_D(\mathbf{u} - \mathbf{u}_{s_p}) d\Omega \tag{4.9}$$

and it is with this term that the main communicative action takes place between the fluids and solids solution. Detail of the algorithm used are discussed later in section 4.4. The following subsections define the procedure carried out to capture the effects of particle size and shape on the interactive forces.

4.3.1 Sub-grid scale particles

Simulation cases arise where, either due to larger particles breaking down or particles being naturally small, the computational mesh lacks enough resolution to define shape or size properly. A schematic of this idea is shown in Figure 4.4. Unless the particle number density[3] is considerable, a typical approach is to simply solve the single phase fluid equations and apply the resulting flow velocities on the particles directly. This approach is known as “one-way” coupling.

However, the matter becomes more complex as the number density increases along with solid phase concentration, which is given by,

$$\alpha = \sum_p n_p \Omega_p \tag{4.10}$$

where n_p is the number of particles per unit volume, and Ω_p corresponds to the individual particle volume. The sum is performed over a size distribution and the result is valid for a representative control volume which could be, for example, an element of the mesh. For the sake of simplicity it is assumed that only one particle size and shape is present, and hence the sum is dropped. Higher concentrations values lead to the disperse phase having a greater effect on the continuous phase, and hence the coupling cannot be neglected and the one way approximation becomes inaccurate.

[3]Particle number density is known as the number of particles per unit volume.

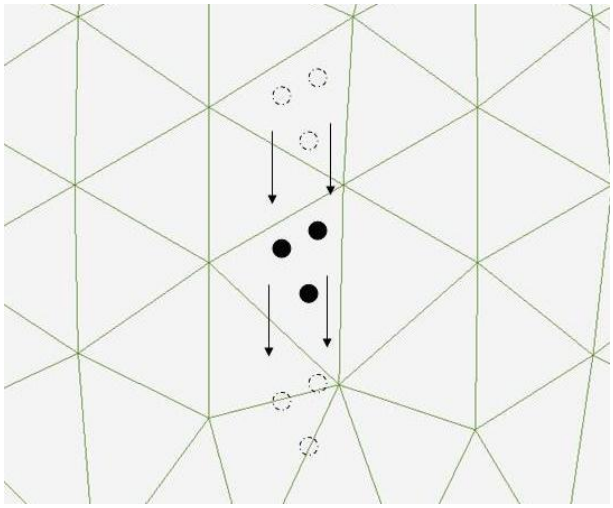


Figure 4.4: Schematic of a sub-grid scale particle case

The method proposed here involves the approximation of the small particle effects on the fluid phase through the means of drag correlations. These drag correlations are used to model the effects of the particles on the fluid phase by establishing values for κ_D (see Gidaspow (1994)). A number of different drag models have been proposed in the literature, originally intended to model fluidised beds. Ergun (1952) developed a drag model which was usable for high concentrations of solid phase. Gidaspow (1994) enhanced this model with drag equations developed by Rowe (1961) and Wen and Yu (1966) obtaining a model that was valid for both high and low concentrations of solid phase. Other drag models have been proposed by Syamlal and O'Brien (1987), Di Felice (1994), and Zhang and Reese (2003). Halvorsen (2005) also compared many of these models to experiments.

As a result, and as also suggested by Gidaspow (1994), the friction coefficient model for multiphase flow chosen for a low range of α is based on drag of a single sphere, and can be expressed by,

$$\kappa_D = \frac{3}{4} C_D \frac{\alpha \rho_f \left| \mathbf{u} - \mathbf{u}_s \right|}{d_p} \mathscr{F}(\alpha) \;\; \forall \, \alpha < 0.2 \tag{4.11}$$

where C_D is the Reynolds number dependant drag coefficient, d_p is the particle diameter in the dispersed phase, and $\mathscr{F}(\alpha_p)$ is a function that brings into account the presence of surrounding particles, given by,

$$\mathscr{F}(\alpha) = (1-\alpha)^{-0.265} \tag{4.12}$$

The equation used here for the drag coefficient was obtained from Brown and Lawler (2003), who studied a vast number of different sets of experimental data of drag forces on spheres and combined them into one, along with wall effect corrections. This drag correlation is given by,

$$C_D = \frac{24}{Re_p} \left(1 + 0.15 Re_p^{0.681} \right) + \frac{0.407}{1 + \frac{8710}{Re_p}} \tag{4.13}$$

where the particle Reynolds number is given by,

$$Re_p = \frac{(1-\alpha)\rho_f \left|\mathbf{u}-\mathbf{u}_p\right| d_p}{\mu} \tag{4.14}$$

and $\mathbf{u}_p$ is the particle velocity. The fluid viscosity is given by μ.

For higher values of α, the friction coefficient is derived from the Ergun equation, and is given by,

$$\kappa_D = 150\frac{\alpha^2\mu}{((1-\alpha)\,d_p\phi)^2} + 1.75\frac{\rho_f\left|\mathbf{u}-\mathbf{u}_s\right|\alpha}{(1-\alpha)\phi d_p}, \quad \alpha_p \geq 0.2 \tag{4.15}$$

where ϕ is the particle sphericity[4].

4.3.2 Resolved particles

As mesh resolution increases near a particle, or if particle size is large enough, there are regions of the computational mesh where the elemental volumes are completely filled with the solid phase. This technically means that there are regions of the domain where $\alpha = 1$. This idea is graphically expressed in Figure 4.5.

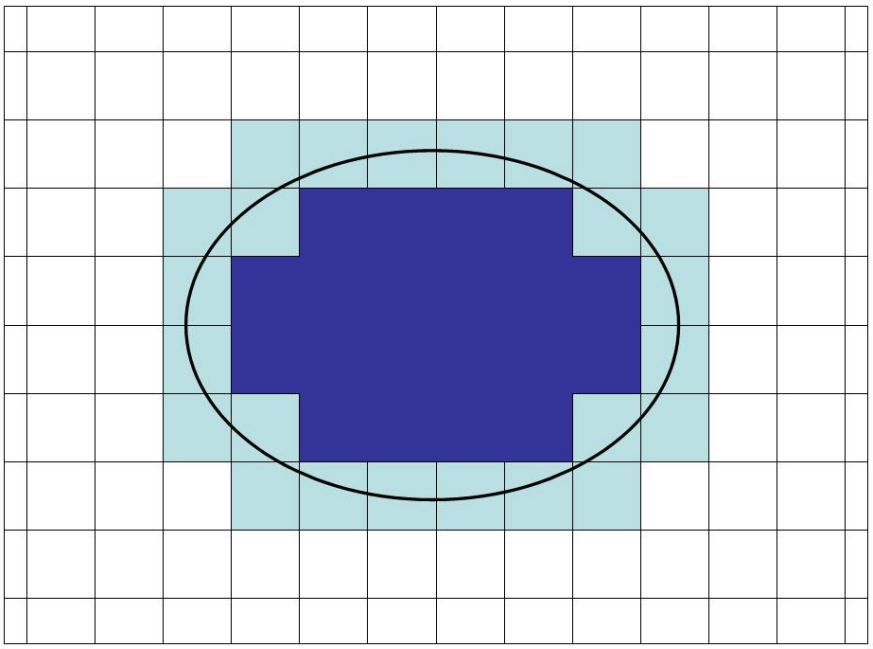


Figure 4.5: Particle shape (i.e. thick black line) being resolved by the computational mesh. Blue region corresponds to $\alpha = 1$, while lighter coloured region has intermediate has $0 < \alpha < 1$. Outside of these two regions, $\alpha = 0$.

In contrast with the sub-grid particle case, the friction coefficient κ_D must become large

[4]Sphericity is a measure of the roundness of an object. It is calculated by dividing the surface area of a sphere of the same volume as the object by the actual surface area of the particle.

enough to account for the no-slip condition imposed on the solid phase wherever $\alpha = 1$. In essence, κ_D becomes a relaxation factor between the fluid and the solid velocities through the momentum source in equation (4.8). To give a vague idea of its effects, suppose that fluid was passing around the elliptic shape in Figure 4.5 which is fixed in space. An unrealistically low value of κ_D would yield a velocity field result depicting flow through the particle when there should be none. This flow would decay with time, depending on the chosen value of κ_D, however this might take an extremely long time and hence it is not practical for unsteady flow cases. However, although there may be a non-zero value of slip velocity (i.e. relative velocity between the fluid and the solid), this velocity would not be transporting any momentum or mass.

In contrast, for high values of κ_D it was observed that the velocity field solution in regions containing the solid phase was zero and this result was relatively insensitive to variations of the friction coefficient. In conclusion, and as was also proposed by Takeuchi et al. (2008), the friction coefficient for particles which can be resolved by the computational mesh is given by,

$$\kappa_D = \frac{\alpha \rho_f}{\Delta t} \tag{4.16}$$

which allows the coefficient to vary its value according to the timestep chosen. This is done to ensure that the flow velocity relaxed quick enough to the solid velocity within a timestep.

4.3.2.1 Spheres and cylinders

If the resolved solid is a sphere or a cylinder, then the simplest way to map the shapes is given in Algorithm 4, where the required parameters are a reference centre point (x_p, y_p, z_p) and a radius R_p associated with each particle. Assuming that the particles involved are inside the fluid domain boundaries, mapping of the solid shape is performed by a loop over the domain nodes having a sub loop that checks if a mesh node is inside a given particle. If the node is inside the shape, its associated scalar field value α is flagged to 1. Solid velocity is mapped in a similar way, taking into account the angular velocity ω in the case the particle is spinning. Total solid velocity is then given, for a node i inside a particle p, by

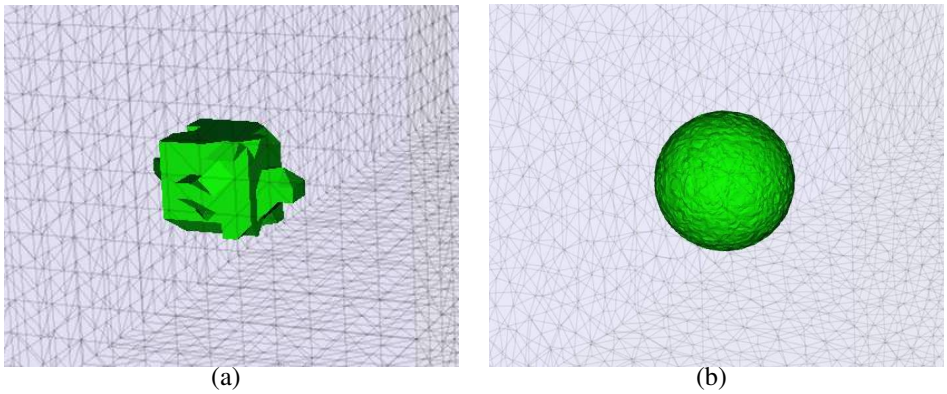
$$\mathbf{u}_{s_i} = \mathbf{u}_p + \omega_p \times \mathbf{r}_i \tag{4.17}$$

where ω_p is the angular velocity vector for the particle, and $\mathbf{r}_i = (x_i - x_p, y_i - y_p, z_i - z_p)$ is the distance vector between the node and the particle centre. In the two dimensional case of cylinders, the $\mathbf{r}_i$ vector has simply two components, as do all the others in equation (4.17).

Figure 4.6, shows a sphere mapped to a three dimensional domain, while Figure 4.7 shows the outcome of solid velocity mapping for a spinning cylinder under flow conditions.

Algorithm 4 Sphere and cylinder shape mapping to the computational fluid domain.

loop {nodes in fluid domain mesh i}
  loop {particles p}
    if $(\mathbf{r}_i^2 \leqq R_p^2)$ **then**
      $\alpha_i = 1$ and $\mathbf{u}_{s_i} = \mathbf{u}_p + \omega \times \mathbf{r}$
      store node to particle index $n2p_i = p$
    else
      $\alpha_i = 0$ and $\mathbf{u}_{s_i} = 0$
    end if
  end loop {particles}
end loop {nodes in fluid domain mesh}



(a) (b)

Figure 4.6: Particle shape being resolved by the computational mesh. In (a) sphere shape is resolved by a coarse mesh, while in (b) sphere shape is resolved by an optimised mesh.

4.3.2.2 External meshes

Perhaps one of the most common and simple ways to approximate the shape of an irregular object is through the use of an unstructured linear tetrahedral mesh. This is particularly useful if the same mesh is used to perform calculations, for example, of internal stresses (i.e. when using FEMDEM). Many tools may be used to generate a meshed domain, including Gmsh[5], GiD(TM)[6], and many other commercial, and non-commercial open source codes. The method used to map the solid geometry shape of this kind into the computational mesh of the fluid domain is based on the fastest neighbour to neighbour search (FNN), as presented in Löhner (2001). This algorithm performs efficiently for mesh to mesh interpolation, which is needed in particular to map velocities, forces, and volume fraction values from one mesh to the other. Algorithm 5 depicts a basic schematic of the method used here, which was combined with a bin sort technique (see, for instance Corwin and Logar (2004)) . This was performed to provide a better starting element for the search on the originating mesh, which is performed to calculate field values of a point in space corresponding to a node in the destination mesh. Algorithm 6

[5]Gmsh is distributed under the terms of the GNU General Public License (GPL), and is available from http://www.geuz.org/gmsh/

[6]GiD is a commercial software, with free evaluation features. For more information, see the official website: http://www.gidhome.com.

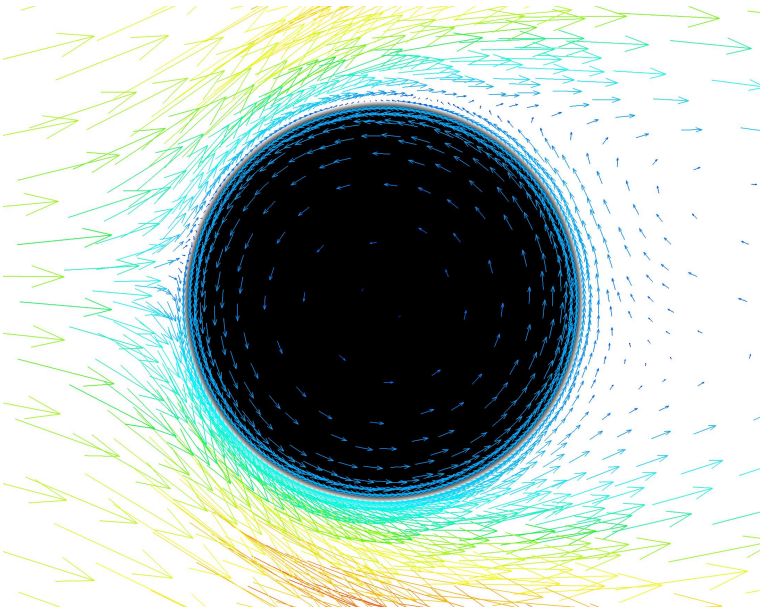


Figure 4.7: Rotating cylinder subjected to fluid flow.

shows the algorithm used to generate the bin.

The fastest neighbour to neighbour search is not flawless, however, and it does struggle over certain domain discontinuities. For this reason a fall back search layer is used consisting of a brute force[7] type approach. See Algorithm 7 on page 150 for details on the brute force search. Figure 4.8 on page 151 shows the comparison between the original solid geometry, the corresponding mesh, and the mapping outcome.

4.4 Interacting forces and solution algorithm

Accurate interaction between Fluidity and DEM or FEMDEM codes requires the correct communication of forces to obtain realistic results. The total instantaneous force exerted by the fluid on a solid particle p due to relative motion is given by equation (4.9), however a term must be added to account for buoyancy forces, and hence the total interactive body force on a solid particle is given by,

$$F_{I_p} = F_{D_p} + \int \alpha \left(\rho_{s_p} - \rho_f\right) \mathbf{g} \, d\Omega_p \tag{4.18}$$

where Ω_p is the volume of the particle. Algorithm 8 on page 150 depicts the procedure to calculate individual particle forces based on friction coefficient and velocity vector fields using the lumped mass matrix (see, for instance, Zienkiewicz and Taylor (1991) on details of mass matrix lumping). Equation (2.71) is modified to take into account the momentum source, yielding,

[7] Brute force refers to a search technique in which all possibilities (in this case, elements on destination mesh) are scanned until a solution is found.

Algorithm 5 Generic shape field value mapping algorithm with fall back on brute force technique

Require: element to element list of originating mesh and node to element list of both meshes.
loop {nodes (i_{node}) in destination mesh}
if (i_{node} coordinates are inside bin region) **then**
calculate bin array indexes i, j, k to obtain starting element e_0 of the originating mesh for FNN search
else
cycle to next node in destination mesh
end if
loop {begin search i_{try} from $e = e_0$ for a total of m_{try} attempts}
calculate shape function values for element e at coordinates of node i_{node}.
store local node i_{min} corresponding to minimum shape function value
if (sum of form function values =1) **then**
exit loop and flag e as found
else
jump to element sharing face opposite to node i_{min}
end if
end loop
if (e has not been found or $i_{try} = m_{try}$) **then**
perform brute force search, as described in Algorithm 7.
else
interpolate field values from element e
store node to particle index $n2p_i = p$
end if
end loop {nodes in destination mesh}

$$\left(\mathbf{M} + \Delta t\left(\mathbf{A} + \mathbf{K}\right) + \mathbf{I}\,\Upsilon^{n+1}\right)\mathbf{U}^{n+1} + \Delta t\mathbf{Q}\mathbf{p}^{n+1} = \mathbf{M}\mathbf{U}^{n} + \Delta t\mathbf{F}^{n+1} - \Upsilon^{n+1}\mathbf{U}_{s}^{n+1} \tag{4.19}$$

where Υ^{n+1} is the vector field of friction coefficients κ_D in each Cartesian direction, and the block matrices components of $\mathbf{M}$ and $\mathbf{A}$ defined in equation (2.54) must also be modified to obtain,

$$M = \int_{\Omega_e} (1-\alpha)\rho_f \mathbf{N}\mathbf{N}^T \,\mathbf{d}\Omega_e \quad A = \int_{\Omega_e} (1-\alpha)\rho_f \mathbf{N}\mathbf{N}^T \mathbf{U}^{n+1} \nabla \mathbf{N}^T \, d\Omega_e \tag{4.20}$$

noting that solid concentration should be capped by $\alpha_{max} = 1 - \varepsilon_\alpha$, where ε_α is a tolerance and a nominal value of 10^{-4} was used. This tolerance prevents matrix singularities when solving equations (2.72) and (2.73).

The solution Algorithm 1, presented in Chapter 2, section 2.4.7, must also be modified to take into account the multiphase approach. Modifications can be observed in Algorithm 9 on page 152.

Algorithm 6 Construction of sorting bins for elements in the originating mesh.

Require: Bin region maximum and minimum sizes (x_{min}, y_{min}, z_{min}, x_{max}, y_{max}, z_{max})
Require: Bin spacing Δx, Δy, Δz.
Ensure: Bin integer matrix **bin** is initialised with -1.
  loop {elements (e) in the originating mesh}
    Obtain volumetric centre $(x_{avg}, y_{avg}, z_{avg})$
    calculate $i = int(\frac{|x_{avg}-x_{min}|}{\Delta x}+1)$
    calculate $j = int(\frac{|y_{avg}-y_{min}|}{\Delta y}+1)$
    calculate $k = int(\frac{|z_{avg}-z_{min}|}{\Delta z}+1)$
    if $(\mathbf{bin}_{i,j,k} = -1)$ **then** {if the bin hasn't been marked yet}
      $\mathbf{bin}_{i,j,k} = e$
    end if
  end loop {elements in the originating mesh}

Algorithm 7 Brute force search algorithm used as a fall back in FNN search.

Require: node to element list
  loop {nodes i_{node} on the destination mesh}
    loop {elements e on the originating mesh}
      Calculate element volume Ω_e
      loop {local nodes i_{loc}}
        Exchange coordinates of i_{loc} with those of i_{node}.
        Calculate partial element volume Ω_{eloc}
        Restore coordinates
      end loop {local nodes}
      if $(\sum \Omega_{eloc} - \Omega_e \leq \varepsilon_{tol})$ **then** {ε_{tol} is a tolerance to account for round off errors}
        interpolate field values from e to i_{node}.
        store node to particle index $n2p_i = p$
        exit element loop
      end if
    end loop {elements on the originating mesh}
  end loop {nodes on destination mesh}

Algorithm 8 Calculation of particle force and friction coefficient terms.

Require: Mapping of the solid shape to produce scalar field α.
  loop {nodes in fluid mesh (i)}
    Calculate $\Upsilon_i^{n+1} = \frac{\alpha_i \rho_{f_i}}{\Delta t}$ for resolved particles, or using equations (4.11) and (4.15) for under resolved ones.
  end loop
Require: Initialise $F_{D_p} = 0$
  loop {nodes in fluid mesh (i)}
    Calculate (p) particle tag for node i using index $n2p_i$. {from Algorithms 4, 5, and 7}
    $F_{D_p}^{n+1} = F_{D_p}^{n+1} + \Upsilon_i^{n+1}\left(\mathbf{U}_i^{n+1} - \mathbf{U}_{s_i}^{n+1}\right) \cdot M_{L_i}$
  end loop

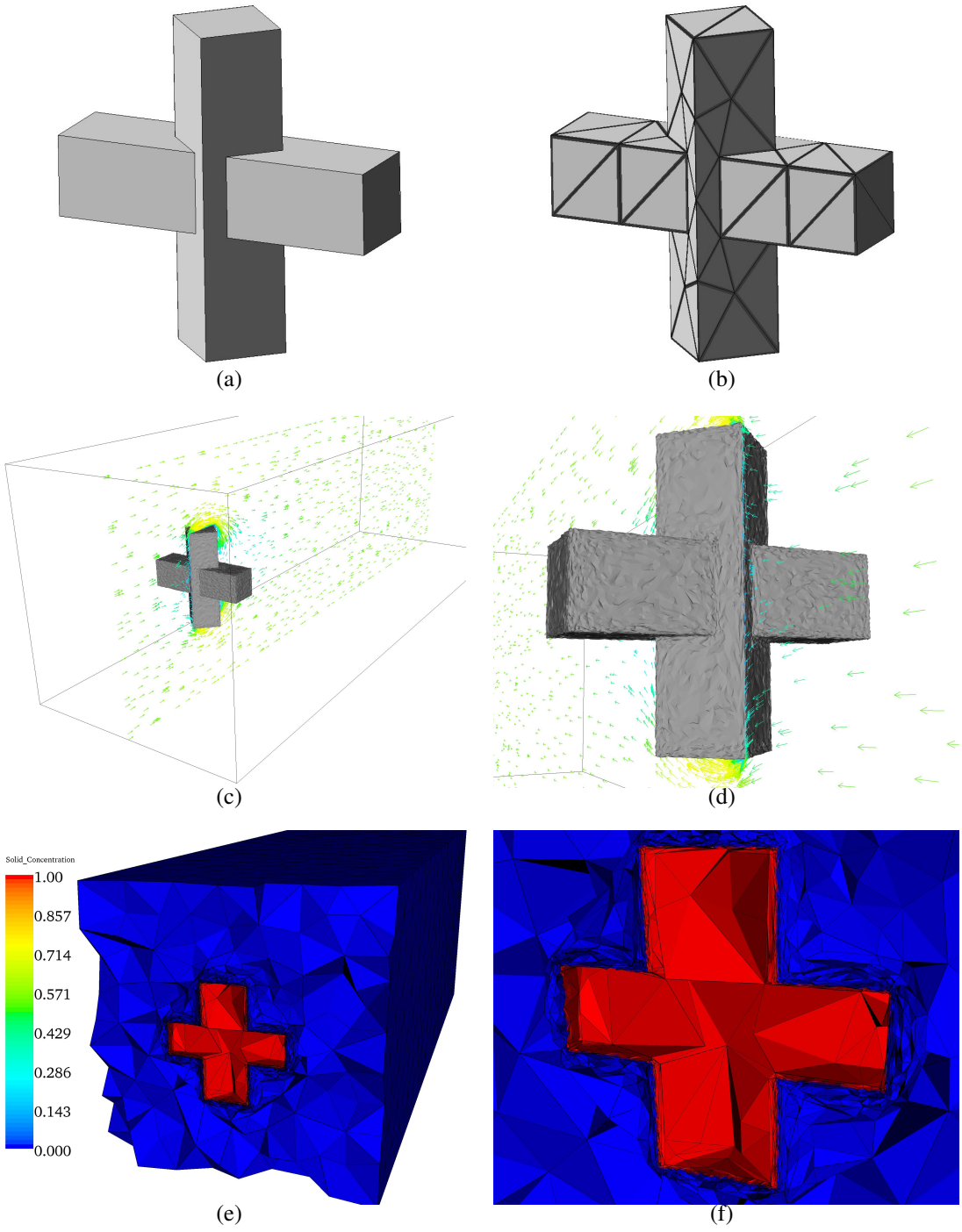


Figure 4.8: External mesh mapping example. Sub- figure (a) shows the original shape of the particle while (b) shows the corresponding particle mesh, (c) and (d) provide different views on the 0.5 value isosurface produced by the solid volume fraction scalar field, and finally (d) and (e) provide a volumetric cutaway of the unstructured grid, showing the scalar solid concentration values along with the optimised resolution of the mesh.

Algorithm 9 Solution algorithm for incompressible Navier Stokes equations, taking into account the presence of solids.

loop {timestep}
  loop {Picard iterations (k)}
    Obtain solution for $\mathbf{C}_k^{n+1}$by solving equation (2.69) in m_{iter} iterations
    Use $\mathbf{C}_k^{n+1}$to calculate buoyancy forces acting on the solid
    Calculate $f_{D_k}^{n+1}$, for which the initial guess ($k = 0$) are values from previous timestep (n)
    Solve modified equation (4.19) in the same manner as was done in Algorithm (1).
    Use the solution $\mathbf{U}_*^{n+1}$ to calculate the pressure correction through equation (2.72) using matrices defined in equation (4.20).
    Use the pressure correction obtained in previous step to obtain $\mathbf{U}_k^{n+1}$ from equation (2.73), also using matrices from equation (4.20).
  end loop
end loop

4.5 Test cases

The following test cases were brought together as further proof that the method developed is highly useful and contains excellent features when modelling solid fluid interaction. Great effort was made in providing as many tests as possible to show its capabilities. Both unsteady and steady flows are modelled, providing evidence of the possibility of two-way coupling between solid particles and the fluids. The tests presented in sections 4.5.1, 4.5.2, 4.5.3, and 4.5.4 involve static shapes subjected to fluid flow. Forces were measured and compared to experimental results, and in some cases to an extensive list of numerical results by other authors. Results are also presented for unsteady flow simulations where the capabilities of two-way coupling are shown.

4.5.1 Flow past a cylinder

Flow past a cylinder is a classic test in fluid dynamics, both computational and experimental. Parameters of this test were obtained from Schäfer and Turek (1996) and a schematic of the domain dimensions used is shown in Figure 4.9.

Boundary conditions used were no slip at the upper and lower walls, while the right boundary was left open. Inflow conditions at the left boundary were set according to the equation given by,

$$\mathbf{u} = \begin{pmatrix} 4 \cdot u_{max} \cdot y \frac{(H-y)}{H^2} \\ 0 \end{pmatrix} \tag{4.21}$$

where H is the height of the domain ($H = 0.41$), and $u_{max} = 0.3\,[m \cdot s^{-1}]$. The Reynolds number

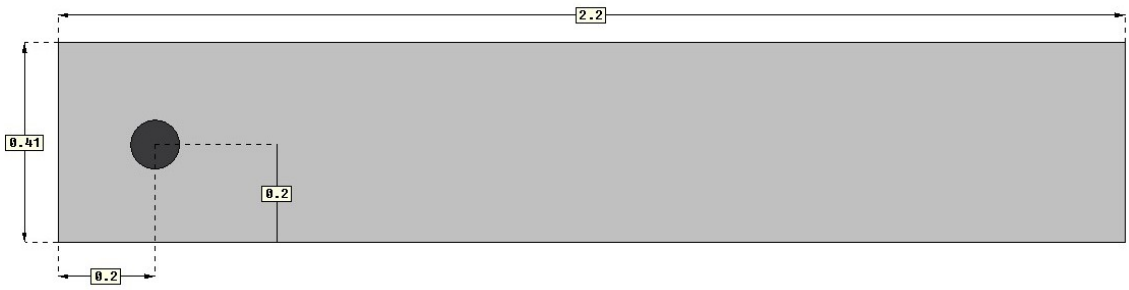


Figure 4.9: Schematic of the the two dimensional cylinder in flow test case.

is given by,

$$Re = \frac{\rho_f |\mathbf{u}| 2R}{\mu} \tag{4.22}$$

where the radius of the cylinder is $R = 0.05\,[m]$. Reference velocity used was $|\mathbf{u}| = \frac{2}{3}\,\mathbf{u}|_{y=\frac{H}{2}}$, with a fluid dynamic viscosity of $\mu = 0.001\,[Pa \cdot s]$, yielding a Reynolds number value of 20 for this test. Figures 4.10 and 4.11 depict far and close views of a group of flow streamlines past the cylinder. Table 4.1 on page 155 shows quantitative comparison of the forces calculated through the method presented in this thesis to others in the literature. Figures 4.12 and 4.13 illustrate the variation of drag and lift forces during the initial transient response. Force coefficients, assuming that Cartesian coordinates are aligned with the incident flow, were calculated using,

$$c_L = \frac{F_{R_y}}{\rho_f |\mathbf{u}|^2 R},\; c_D = \frac{F_{R_x}}{\rho_f |\mathbf{u}|^2 R} \tag{4.23}$$

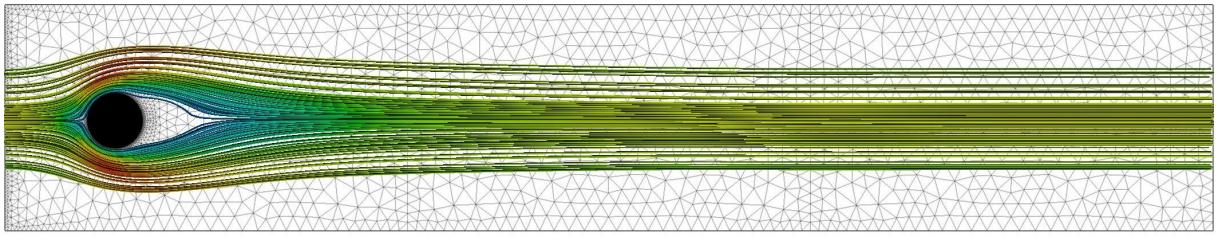


Figure 4.10: Streamlines obtained from results of flow past a cylinder.

Results for this test case show how the method is able to perform extremely well when compared to many others in the literature. Calculation of forces through mapping and friction coefficient calculations from Algorithm 8 shown in Figures 4.12 and 4.13 evidence these remarkable results.

4.5.2 Flow past a 3D cylinder

This test case is essentially the three dimensional version of the test in section 4.5.1. Details of the domain geometry is provided in Figure 4.14.

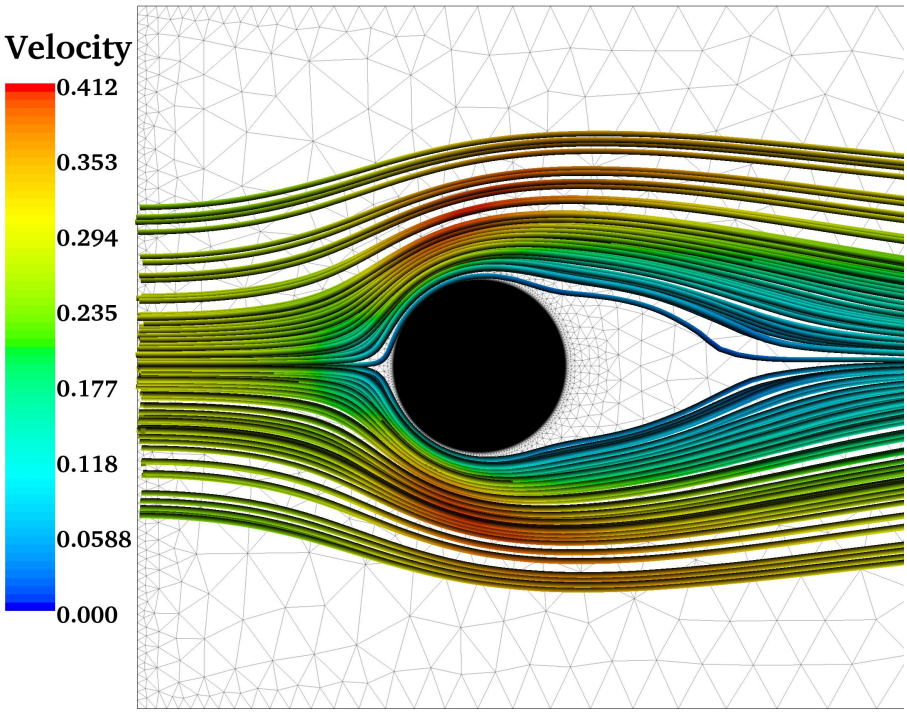


Figure 4.11: Streamlines obtained from results of flow past a cylinder (close view).

Boundary conditions used were no slip below and above the cylinder, while the side walls were set to free slip. Inflow conditions were set according to the equation given by,

$$\mathbf{u} = \begin{pmatrix} 16 \cdot u_{max} \cdot y \cdot z \cdot \frac{(H-y)(H-z)}{H^4} \\ 0 \\ 0 \end{pmatrix} \tag{4.24}$$

where H is the height of the domain ($H = 0.41$), and $u_{max} = 0.45\,[m \cdot s^{-1}]$. The Reynolds number is given by,

$$Re = \frac{\rho_f\,|\mathbf{u}|\,2R}{\mu_f} \tag{4.25}$$

where the radius of the cylinder is $R = 0.05\,[m]$. Reference velocity used was $|\mathbf{u}| = \frac{4}{9}\,\mathbf{u}|_{y=\frac{H}{2},z=\frac{H}{2}}$, together with fluid dynamic viscosity $\mu = 0.001\,[Pa \cdot s]$, yielding a Reynolds number value of 20 for this test. Figure 4.15 on page 157 depicts a group of flow streamlines, next to a cut plane mesh and scalar field plots of the solid volume fraction. Figure 4.16 shows a close up on the streamlines. Table 4.2 shows quantitative comparison of the forces with data from Schäfer and Turek (1996). Figures 4.17 on page 159 and 4.18 on page 159 illustrate the evolution of drag and lift coefficients which were calculated using,

$$c_L = \frac{F_{R_y}}{\rho_f\,|\mathbf{u}|^2\,R \cdot H},\; c_D = \frac{F_{R_x}}{\rho_f\,|\mathbf{u}|^2\,R \cdot H} \tag{4.26}$$

This test case provides evidence that the method works as well in three dimensions. Result

Method	# of unknowns	c_D	c_L	Method	# of unknowns	c_D	c_L
This thesis	4547	5.6501	0.0095	8a	20487	5.576	0.011
	28794	5.7937	-0.0005		6297	5.571	0.013
	75831	5.6057	0.031		2298	5.445	0.02
1	200607	5.5567	0.0106	9a	240000	5.5803	0.0106
	51159	5.5567	0.0106		60000	5.5786	0.0106
	13299	5.5661	0.0105		15000	5.5612	0.0109
3a	10800	5.6	0.012	10	2665728	5.5755	0.0106
4	297472	5.5678	0.0105		667264	5.5718	0.0105
	75008	5.5606	0.0107		167232	5.5657	0.0102
	19008	5.5528	0.0118		42016	5.5608	0.0091
6	1314720	5.819	0.011	12	32592	5.5069	0.0132
	332640	5.774	0.003		26970	5.5125	0.0056
	85140	5.789	-0.006		22212	5.6026	-0.0031
7a	294912	5.5846	0.0106				
	73728	5.5852	0.0105				
	18432	5.5755	0.0102				

Table 4.1: Comparison table for the 2D cylinder test case using different methods (See Appendix A for details of each the methods).

comparison from Table 4.2 shows that the drag coefficient is being slightly underestimated. This however is very likely due to the fact that other methods used higher resolutions for their tests, which can also be seen in Table 4.2.

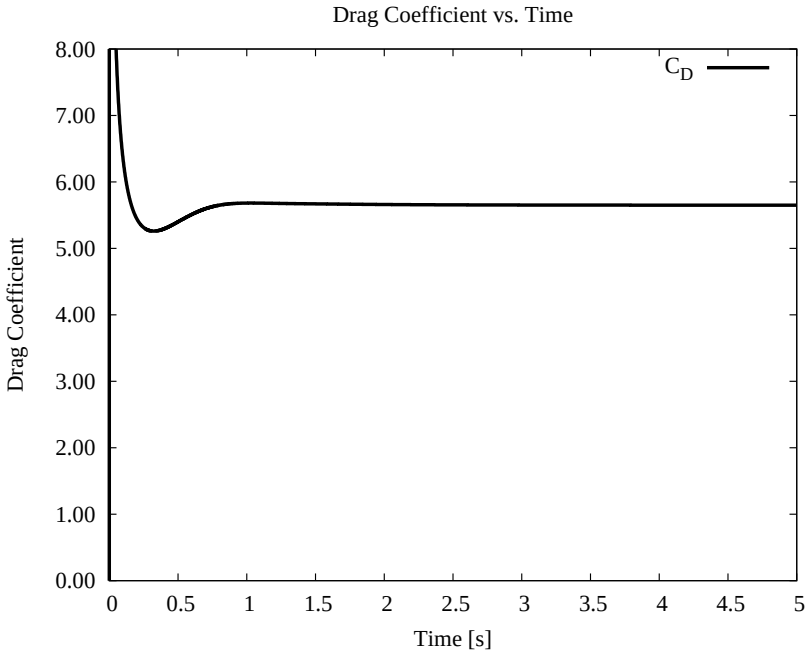


Figure 4.12: Drag coefficient evolution for the two dimensional cylinder test case.

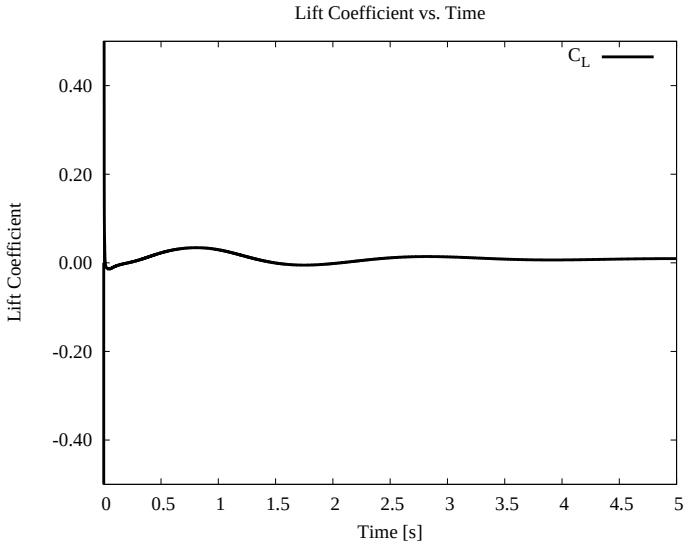


Figure 4.13: Lift coefficient evolution for the two dimensional cylinder test case.

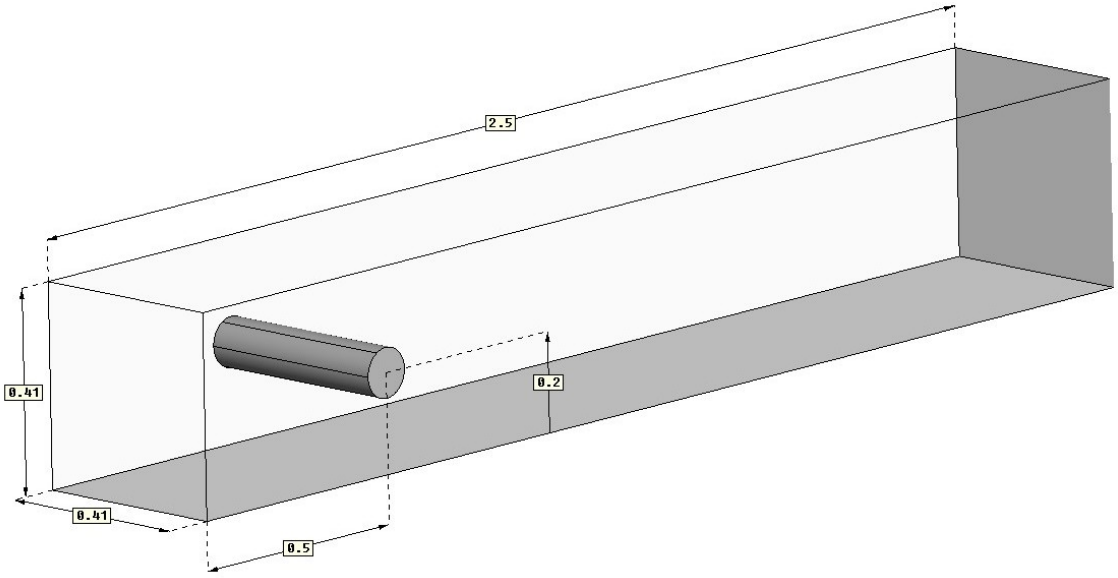


Figure 4.14: Schematic of the the three dimensional cylinder in flow test case.

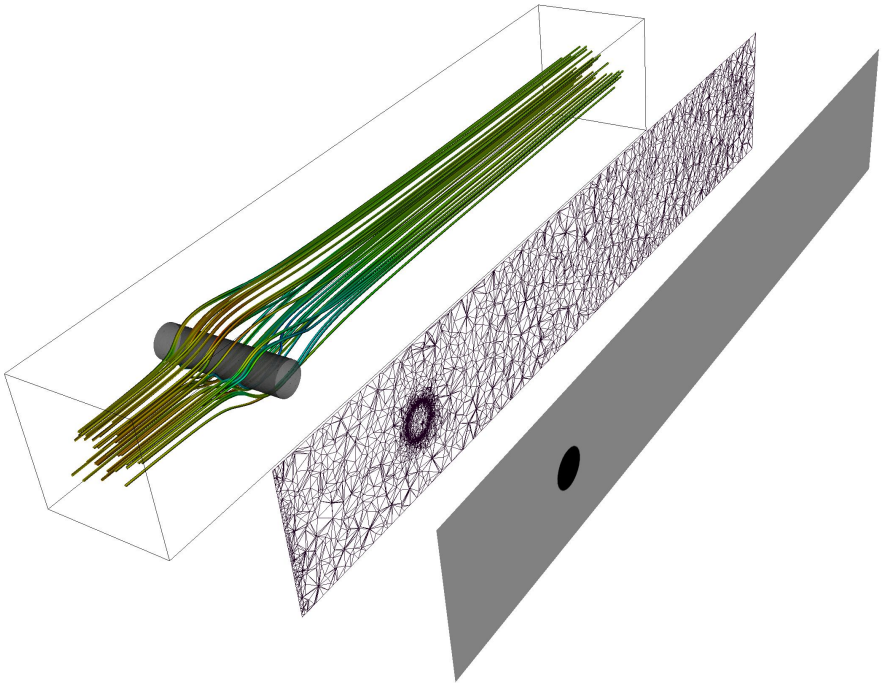


Figure 4.15: Streamlines obtained from results of flow past a cylinder. Middle cut planes are presented parallel to the 3D plot of a group of streamlines. Left slice gives detail of the mesh, while the right slice shows a scalar field plot of solid concentration.

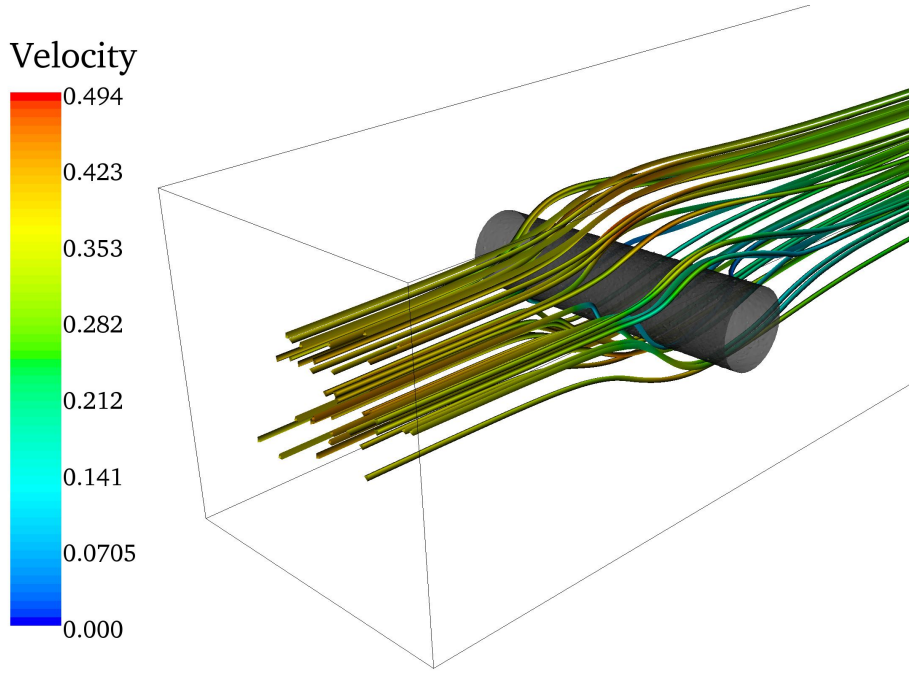


Figure 4.16: Streamlines past the cylinder in close view. Grey isosurface is shown corresponding to a solid volume fraction value of 0.5.

Method	# of unknowns	c_D	c_L
This thesis	57468	5.7302	0.0133
	180771	5.986	0.011
1	2426292	6.1295	0.0093
	630564	6.123	0.0095
2	555000	6.144	0.0074
	276800	5.86	0.0042
3	608496	6.16	0.0095
6	6303750	6.233	-0.004
7b	12582912	6.1932	0.0093
	1572864	6.1868	0.0092
	196608	6.1366	0.0098

Method	# of unknowns	c_D	c_L
8a	362613	6.143	0.0084
	73262	6.099	0.0067
9	2355712	6.18	-0.001
	753664	6.172	0.009
	94208	6.131	0.01
10	6116608	6.1043	0.0079
	771392	5.9731	0.0059
	98128	5.8431	0.0061

Table 4.2: Comparison table for the 3D cylinder test case using different methods (See Appendix A for details of each the methods).

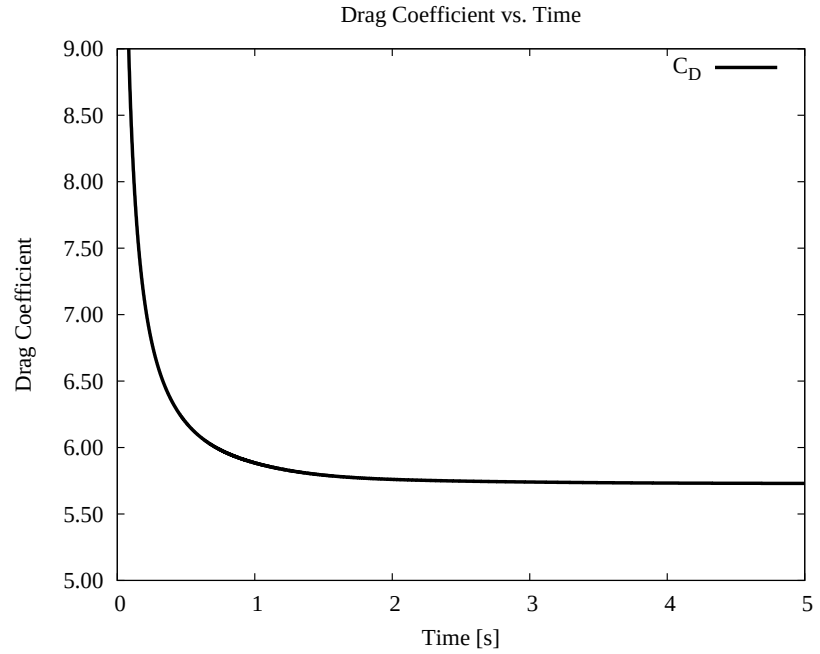


Figure 4.17: Drag coefficient evolution for the three dimensional cylinder test case.

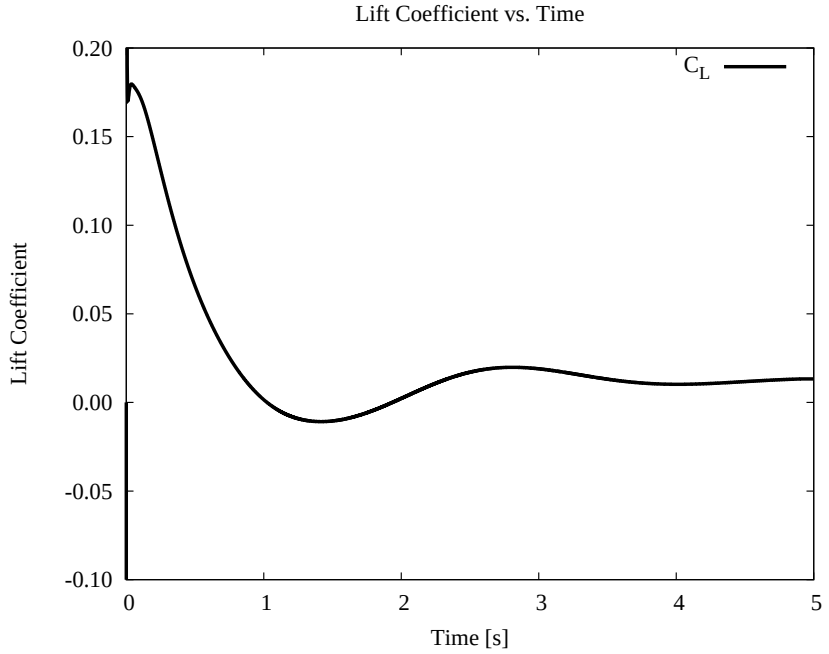


Figure 4.18: Lift coefficient evolution for the three dimensional cylinder test case.

4.5.3 Flow past a 3D square section tube

This test was also originally presented by Schäfer and Turek (1996). Details of the domain geometry is provided in Figure 4.19. Boundary conditions are identical as the ones for section 4.5.2, and inflow conditions are given by equation 4.24. The Reynolds number is given by,

$$Re = \frac{\rho_f |\mathbf{u}| L}{\mu_f} \tag{4.27}$$

where the square side length $L = 0.1\,[m]$. A Reynolds number value of 20 was used as well for this test.

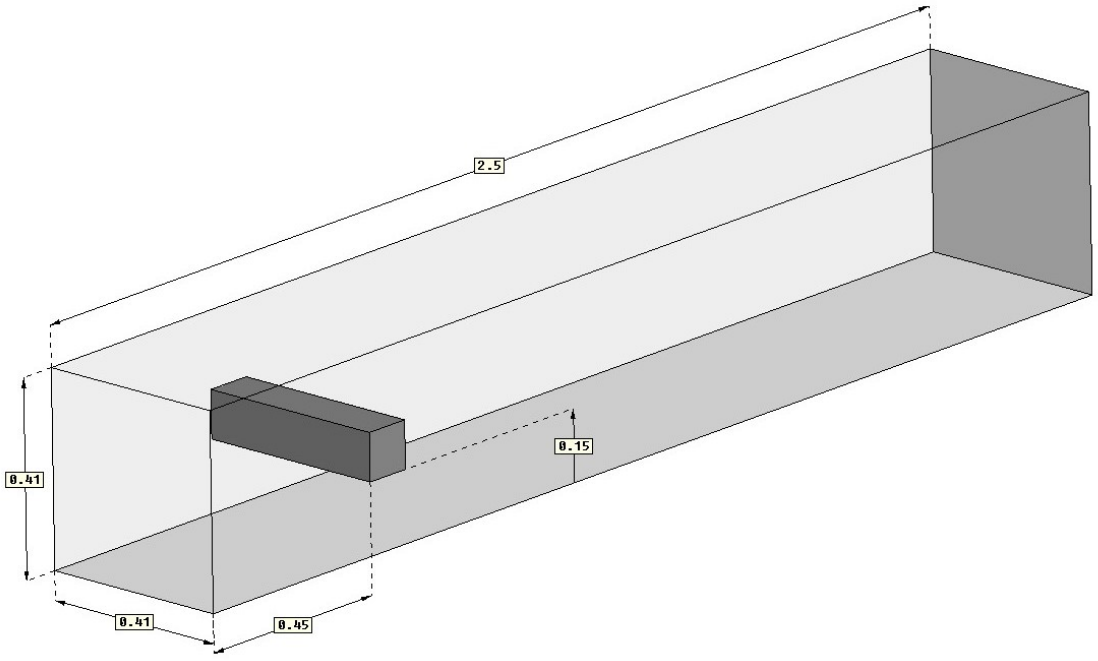


Figure 4.19: Schematic of the the three dimensional square section tube in flow test case.

Figure 4.20 depicts a group of flow streamlines past the square section, next to mesh and scalar field plots of the solid volume fraction. Figure 4.21 shows a close up on the streamlines while 4.22 shows velocity contour plots on horizontal and vertical cut planes. Figure 4.23 reveals the flow detail captured behind the solid shape. Table 4.3 shows quantitative comparison of the forces with data from Schäfer and Turek (1996). Figures 4.24 and 4.25 illustrate the evolution of drag and lift coefficients which were calculated using,

$$c_L = \frac{2F_{R_y}}{\rho_f |\mathbf{u}|^2 L \cdot H},\ c_D = \frac{2F_{R_x}}{\rho_f |\mathbf{u}|^2 L \cdot H} \tag{4.28}$$

This set of results not only adds to the credibility of the method's suitability for FSI simulations, but also shows the FNN external mesh mapping method in working order.

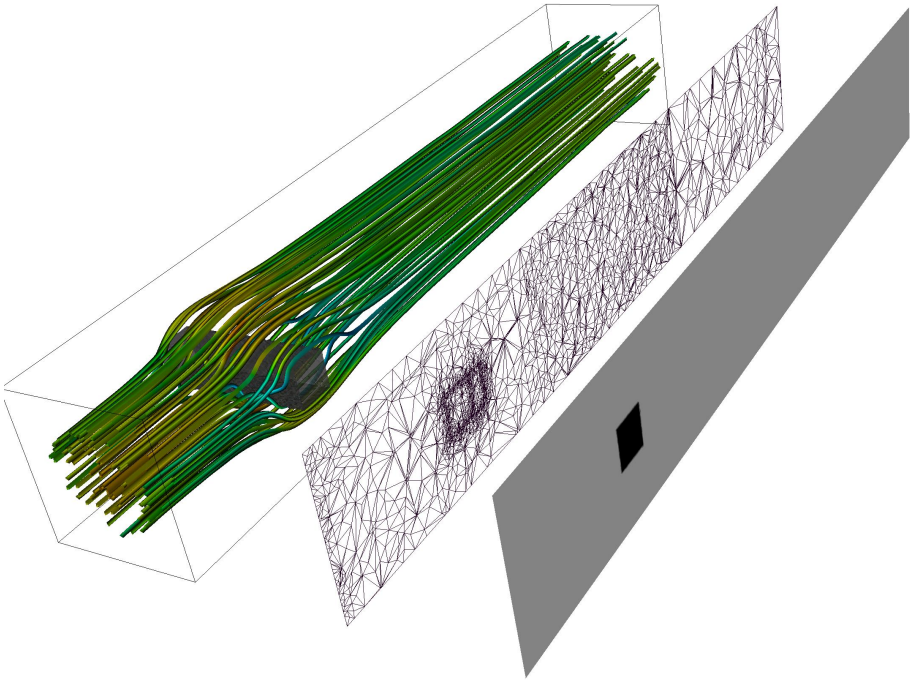


Figure 4.20: Streamlines obtained from results of flow past a square section tube.

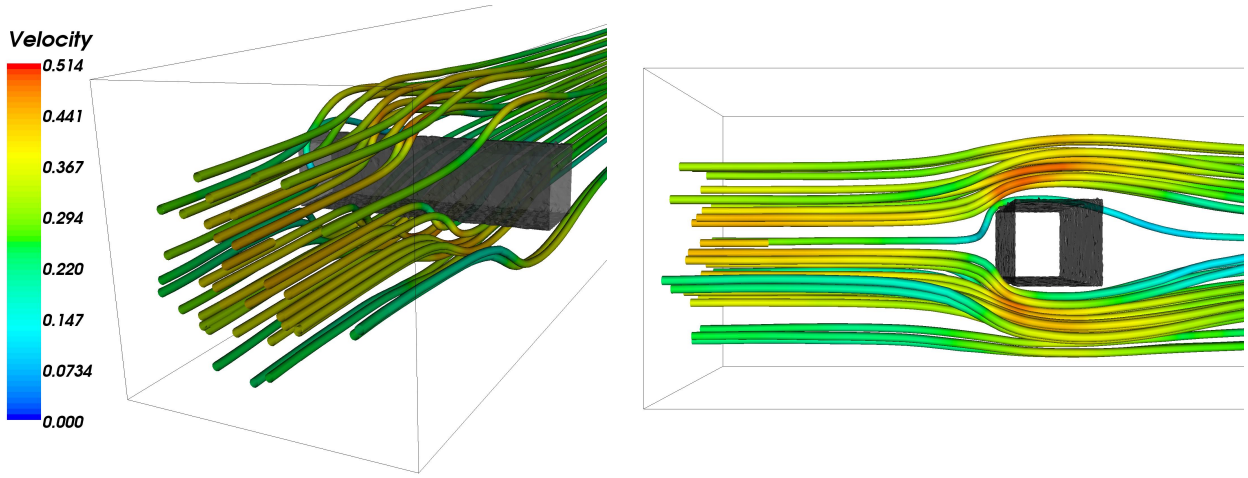


Figure 4.21: Streamlines obtained from results of flow past a square section tube. (close view)

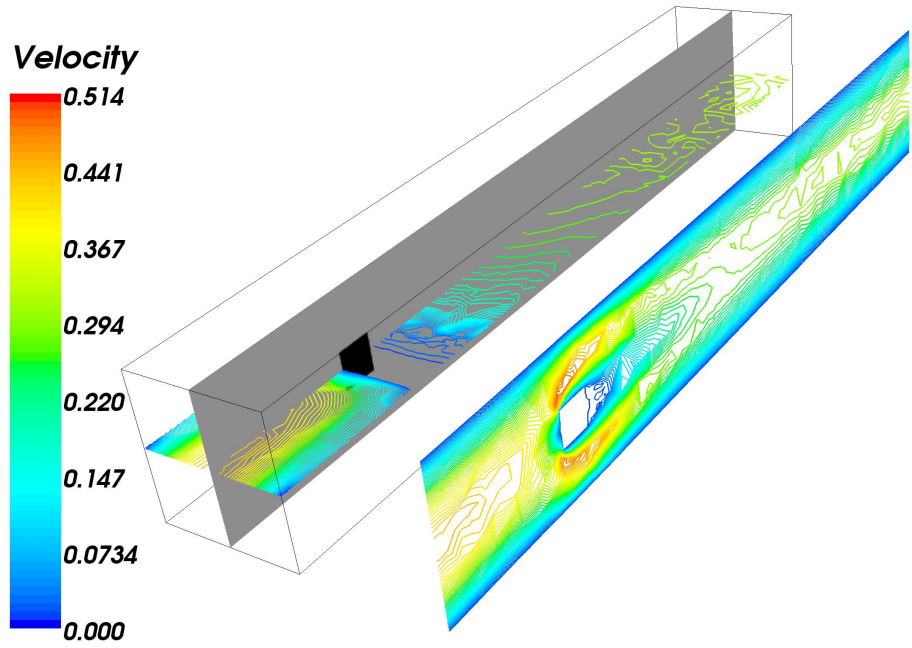


Figure 4.22: Velocity isocontours on vertical and horizontal cut planes.

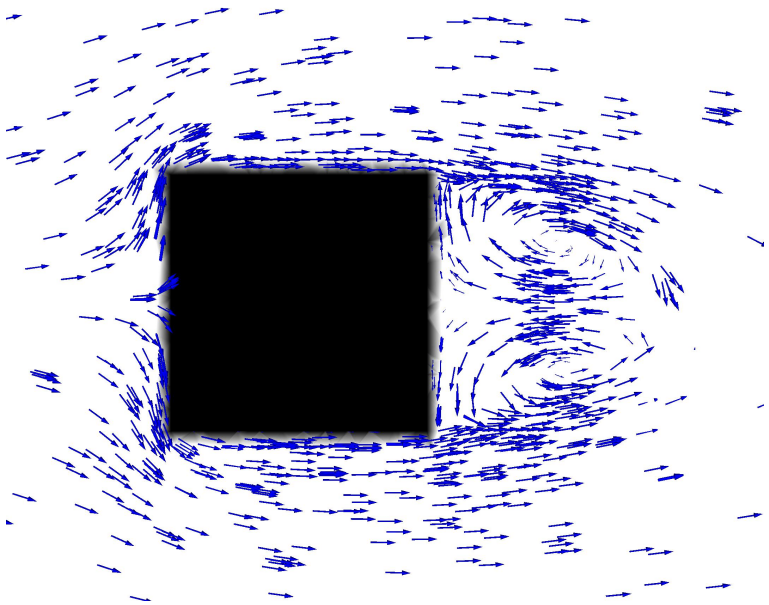


Figure 4.23: Close up view of velocity field vector plot (vectors not in magnitude scale) showing the detail captured in the wake behind the square section.

Method	# of unknowns	c_D	c_L
This thesis	25055	7.3174	0.1048
	215847	7.425	0.0236
1	2530836	7.6415	0.0673
	657492	7.6029	0.0665
3	634872	7.61	0.0642
5a	1472000	7.92	0.0645
	184000	8.04	0.0642
	23000	7.66	0.072
5b	1472000	7.44	0.0615
	184000	7.28	0.0582
	23000	6.74	0.0615
6	6303750	8.093	0.07
7a	454656	7.5395	0.0797
	56832	7.128	0.0861
	7104	6.459	0.0988

Method	# of unknowns	c_D	c_L
8a	362613	7.648	0.067
	73262	7.653	0.059
8b	97822	7.634	0.066
10	6094976	7.6148	0.06
	768544	7.5622	0.0503
	97736	7.3069	0.0348
11	1425600	7.7583	0.0511
	460800	7.7673	0.0406
	128000	7.2372	0.0602
15b	6724000	6.077	0.0859
	1681000	5.506	0.142
16	2007040	7.37	0.0619
	405503	7.25	0.0549

Table 4.3: Comparison table for the 3D square section test case using different methods (See Appendix A for details of each the methods).

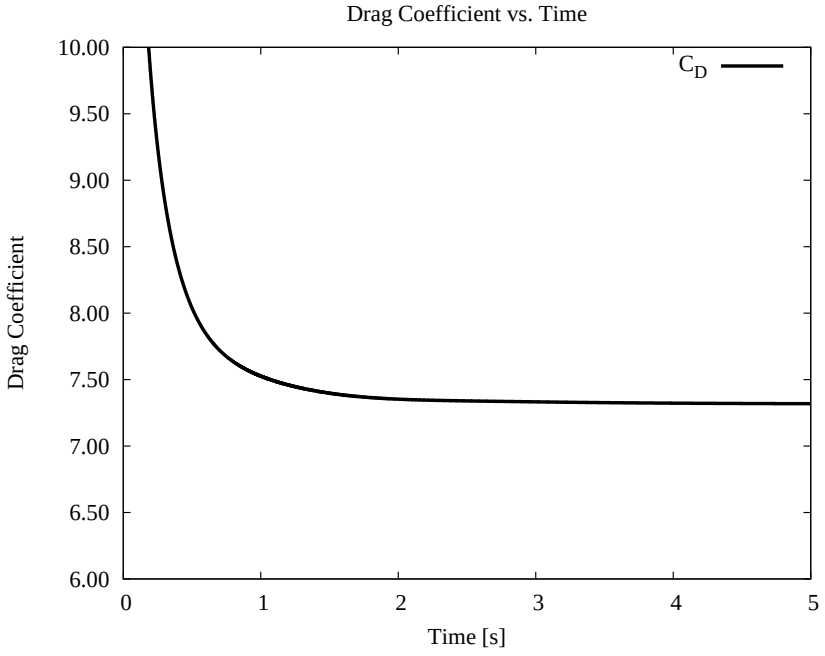


Figure 4.24: Drag coefficient evolution for the three dimensional square section test case.

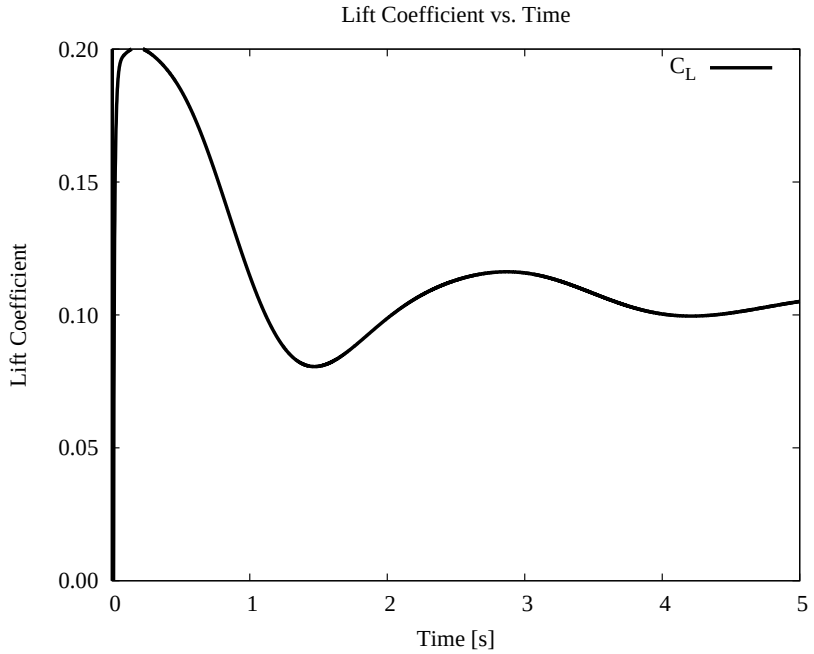


Figure 4.25: Lift coefficient evolution for the three dimensional square section test case.

4.5.4 Flow past a sphere

Another test case which is well documented is that of fluid flow past a static rigid sphere. It is a problem which has been studied for decades and many authors have published experimental data on the case. See, for instance, Bailey and Hiatt (1971); Clift et al. (1978); Khan and Richardson (1987); Dennis and Walker (1971); Turton and Clark (1987); Zigrang and Sylvester (1981). Magnaudet et al. (1995) performed a numerical simulation study on the Reynolds range between 0.1 and 300 to later study the effects of acceleration. A good compilation of a number of data sets for a variety of Reynolds ranges can also be found in Brown and Lawler (2003), who also presented a drag correlation combining all the sets in one equation.

A sphere of radius 0.125 m was submerged in an otherwise uniform fluid flow of 1 m/s, in a prismatic domain of $10 \times 10 \times 30$ m dimensions. The problem's schematic is shown in Figure 4.26. The sphere is located in a centred position 7.5 m away from the inflow boundary. Boundary conditions used were free slip on all walls and open boundary in the outflow surface. Figure 4.27 shows results for Reynolds number of 5000. In Figure 4.27(a) an isosurface of $\alpha = 0.5$ is shown together with a cut plane of velocity isocontours, while Figure 4.27(b) shows the streamlines.

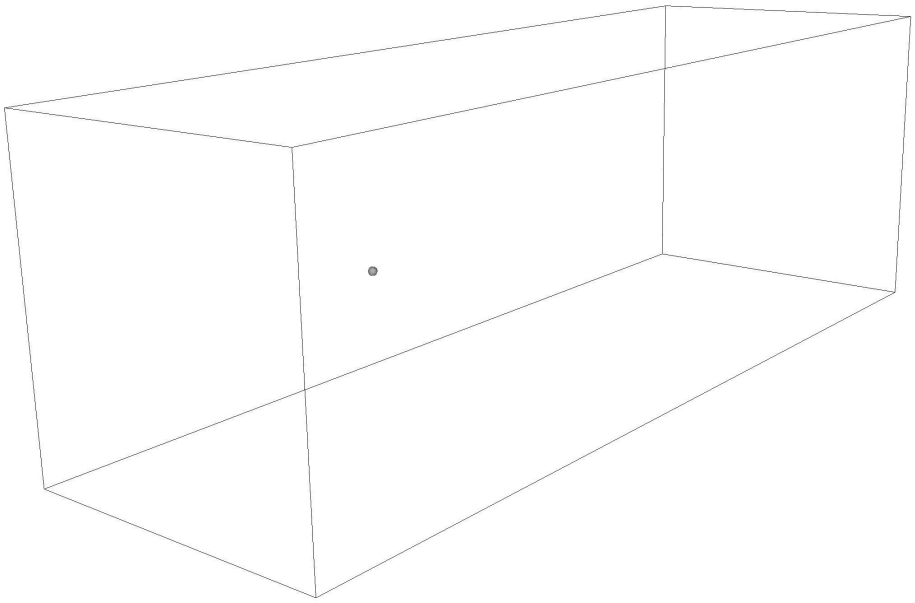


Figure 4.26: Domain schematic of the sphere drag test case. The sphere is placed closer to the inflow surface boundary. A large domain was used to suppress any possibility of wall effects.

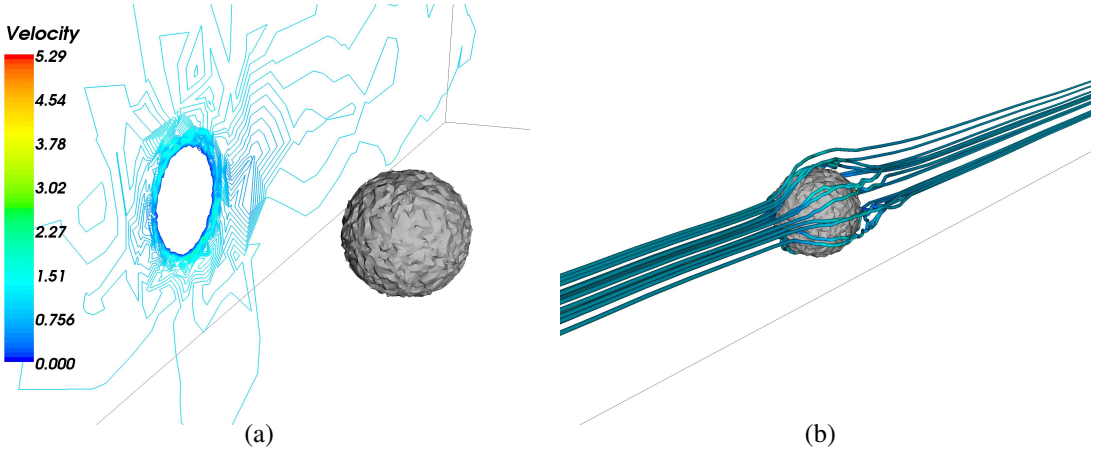


Figure 4.27: Result plots for flow past a rigid sphere, showing (a) the solid isosurface, and (b) the corresponding flow streamlines.

Series of simulations were run for Reynolds numbers ranging from 2.5 to 5000 and in each case the drag force value was transformed to a drag coefficient by calculating,

$$c_D = \frac{2 \cdot F_D}{\rho_f u_0^2 \cdot \pi \cdot R^2} \tag{4.29}$$

where values used for density ρ_f, and reference velocity u_0 were $1.0 kg \cdot m^{-3}$ and $1.0 m \cdot s^{-1}$, respectively. Table 4.4 contains the values recorded for the different simulation results using the multiphase approach presented in this Chapter. Figure 4.28 shows a graph of the values from Table 4.4, compared against the experimental curve from equation 4.13. Excellent agreement

is observed between the simulation and the experiments, for the chosen range of Reynolds numbers.

Re	c_D	Re	c_D
2.5	15.809	400	0.754
5	8.874	500	0.734
7.5	6.450	750	0.721
10	5.193	1000	0.713
12.5	4.413	1500	0.661
15	3.878	2000	0.652
22.5	2.944	2500	0.587
25	2.746	3000	0.546
50	1.801	4000	0.531
100	1.247	5000	0.528
150	1.035		
200	0.921		
250	0.858		

Table 4.4: Drag coefficient values at different Reynolds numbers for the flow submerged sphere test case.

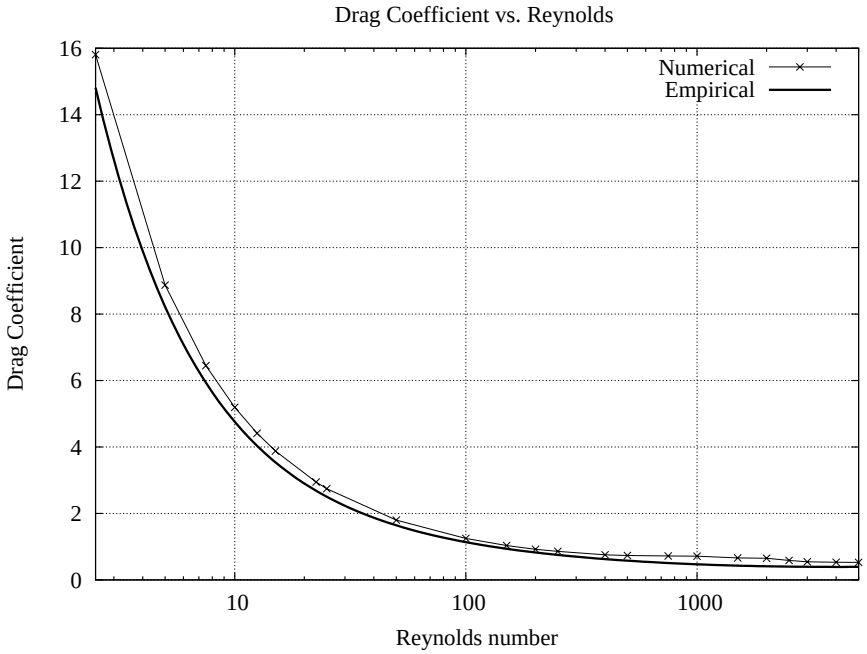


Figure 4.28: Sphere drag coefficient vs. Reynolds number plot of numerical results obtained through the coupling method presented in this thesis. The experimental curve used for comparison was obtained from Brown and Lawler (2003).

4.5.5 Sphere drop in a viscous fluid

To further validate the solid-fluid coupling method, a test case involving an accelerated particle is presented here. Clift et al. (1978) performed a review on studies of accelerated flows for

spheres, while also addressing free fall. Hartman et al. (1989) devised an explicit method for predicting sphere particle terminal velocity based on drag force correlations, seemingly predicting with greater accuracy than the method by Turton and Clark (1987).

The validation test in this case consists of allowing the sphere to reach its terminal velocity, and comparing the results against empirical data. When the particle reaches terminal velocity, added mass[8] effect, and Basset force [9] effects are considered no longer present. Figure 4.29 shows a schematic of the domain, which is of size $4.5 \times 4.5 \times 12$ m. The sphere radius is of $R = 0.125$ m, and it was dropped from coordinates $(2.25, 2.25, 10)$.

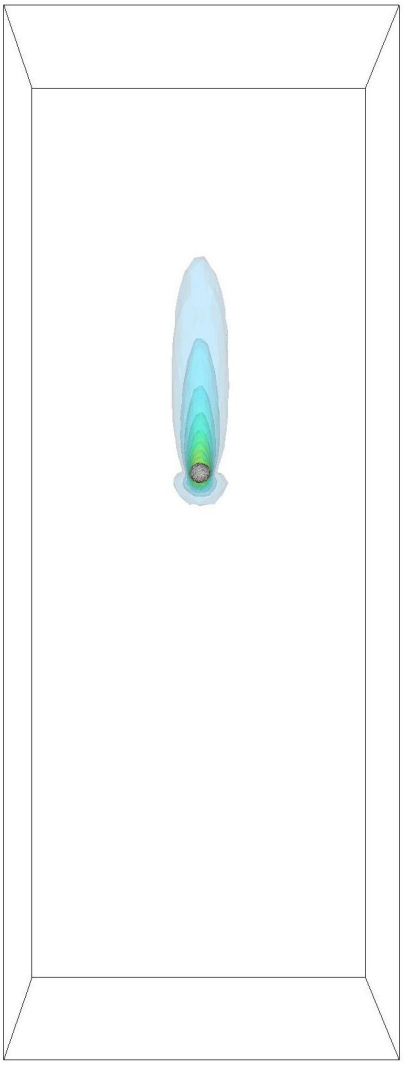


Figure 4.29: Schematic of the domain used in the sphere drop simulation.

Based on theory from Clift et al. (1978, page 287), the one dimensional calculation of forces

[8]Added mass, also termed "virtual mass", accounts for the fact that acceleration of the sphere involves also acceleration of the fluid.

[9]The Basset force, also known as the "history force" , takes into account effects from flow development, such as vorticity, during the acceleration.

acting on a sphere during accelerated free fall is given by,

$$F_D = c_D \frac{1}{2}\rho_f u_p^2 \pi R^2 + \frac{V_p \rho_f}{2}\frac{du_p}{dt} + 6R^2\sqrt{\pi\rho_f\mu_f}\int_{-\infty}^{t}\left(\frac{du_p}{dt}\right)_{t=s}\frac{1}{\sqrt{t-s}}ds \tag{4.30}$$

where V_p is the volume of the sphere, u_p is the velocity of the sphere, s is a dummy variable used for integration, and c_D is calculated from drag curve correlations (see equation (4.13)). The first term is the typical steady state drag term, which was analysed in the test cases presented up to this point. The second term is the mentioned added or virtual mass, while the last term is the history or Basset force (see, for instance, Abbad and Souhar (2004)). Equation (4.30) was solved to obtain the terminal velocity, and compared against the results from using the solid-fluid coupling method of this thesis, producing the plot shown in 4.30. Two other lines were plotted in this graph, showing the effects of the added mass and Basset force. Results agree quite well with the prediction, although a certain discrepancy can be observed during the initial acceleration. It is difficult to assess whether this discrepancy is due to a flaw in the empirical model, hence the matter will not be further addressed in this thesis. It does remain, however, a subject of future research. Figures 4.31 and 4.32 depict a sequence of frames from the sphere acceleration, showing velocity contours.

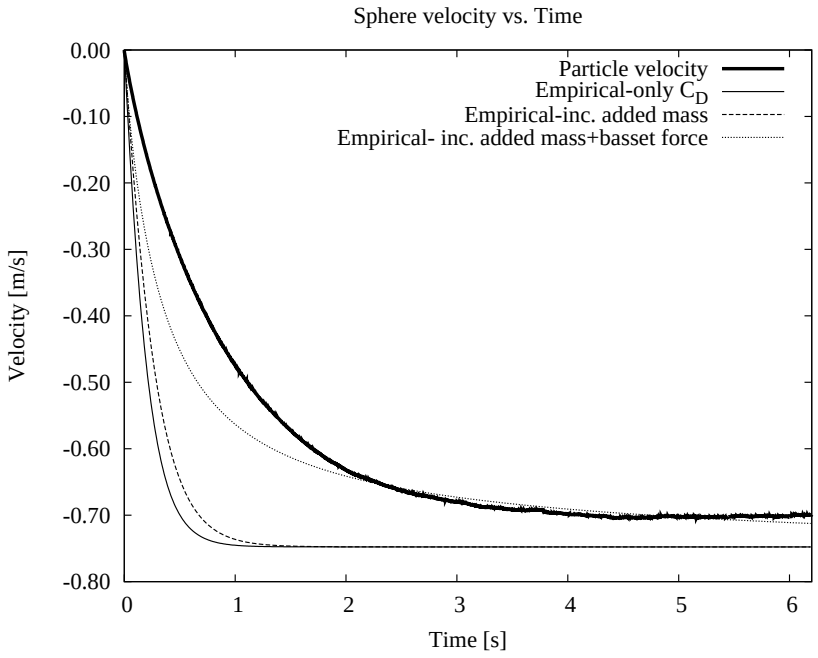


Figure 4.30: Graph of dropped sphere velocity vs. time elapsed. It can be observed that the sphere reaches terminal velocity.

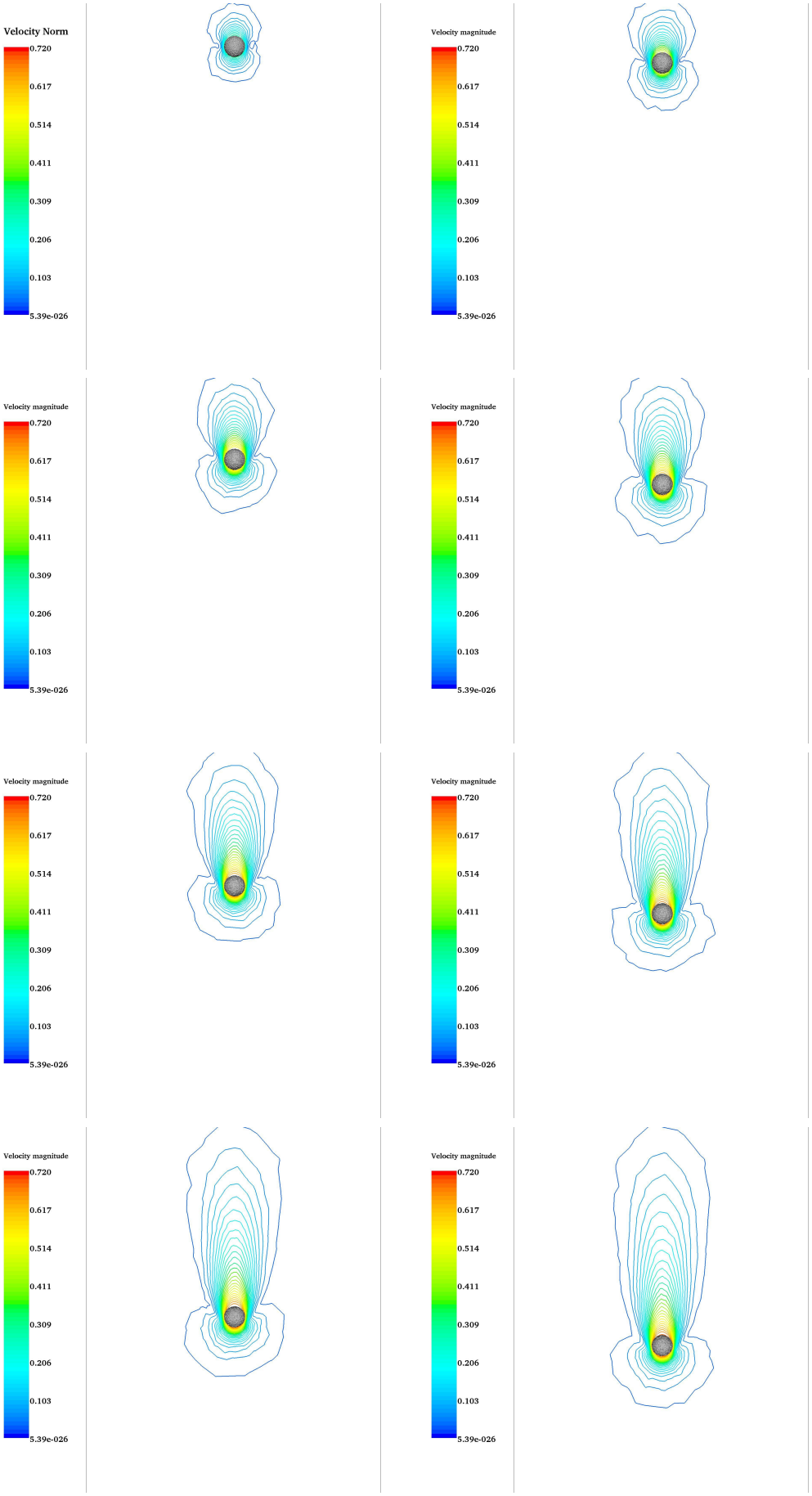


Figure 4.31: Sequence of frames for the sphere drop simulation. The time frames included are for t =0.5, 1.0, 1.5, 2.0, 2.5, 3.0, 3.5, and 4.0 . Data shown are velocity magnitude contours for a middle cut-plane across the sphere.

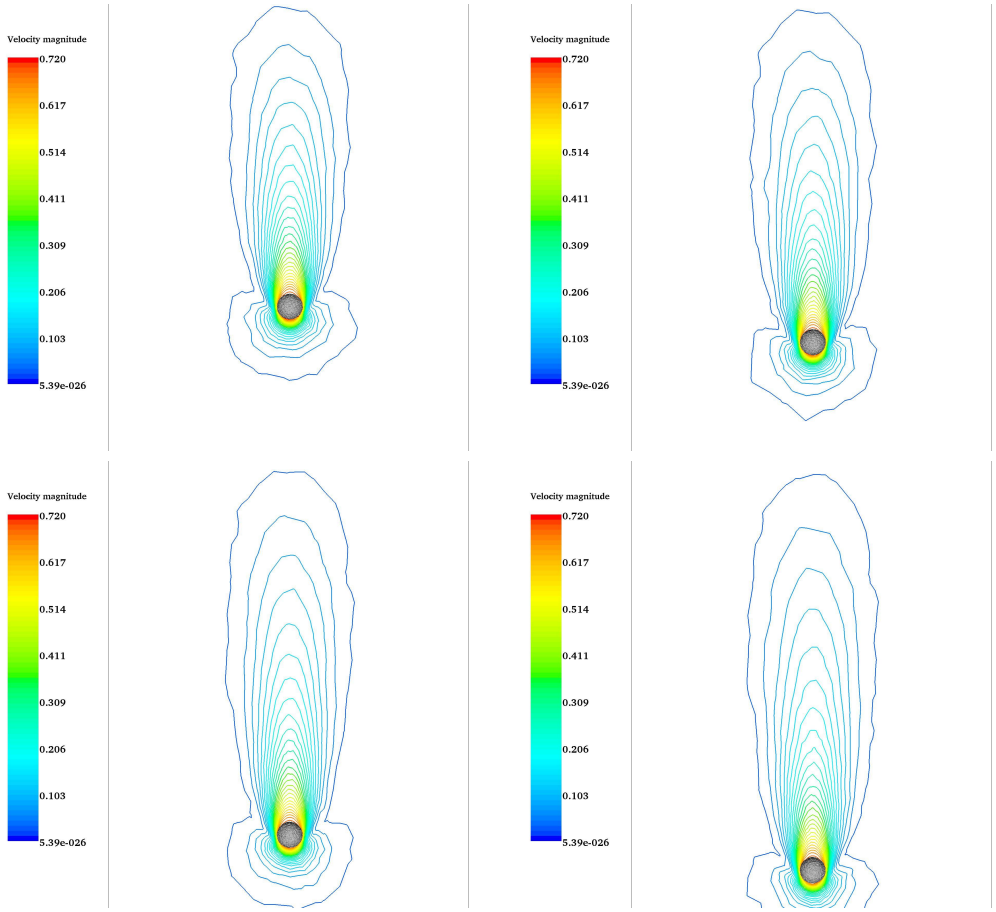


Figure 4.32: (Continued from Figure 4.31) Sequence of frames for the sphere drop simulation. The time frames included are for t =4.5, 5.0, 5.5, and 6.0.

4.5.6 Two spheres in viscous fluid

Wacholder and Weihs (1972) developed exact solutions for the presence of another fluid particle (in tandem), however the solutions were for Reynolds numbers in the creeping flow range (e.g. $Re \rightarrow 0$). Glowinski et al. (2000) performed sedimentation simulations with a considerable amount of cylinders, and also studied the case of two spheres positioned in tandem in free fall.

For this particular test case, two simulations are shown, for which the domain size is $20 \times 20 \times 100$mm. The first one involves two spheres of radius $R = 1.5\,mm$ and $\rho_p = 7800\,kg \cdot m^3$ positioned initially at coordinates $(10, 85, 10)$ [mm] and $(10, 75, 10)$, respectively, and allowed to free fall in a quiescent fluid of density $\rho_f = 935.0\,kg \cdot m^3$ and viscosity $\mu = 0.01\,Pa \cdot s$. It can be observed from the sequence of frames in Figure 4.33 how the top sphere enters the wake of the lower sphere thus reducing its drag and having a greater acceleration.

The second simulation also depicts two spheres of the same size. They are, however, positioned at $(11.2, 4.5, 11.2)$ and $(10, 5, 10)$ and aimed at each other for impact with velocities of $(0, 200, 0)$ and $(0, -200, 0)$ [mm/s], respectively. Impact between the particles is handled via coupling with a DEM code. Figure 4.34 shows a sequence of frames of the impact followed by separation of the spheres. These results provide proof that the multiphase approach presented in this Chapter can be successfully coupled to a DEM[10] code.

[10]The DEM code used for these two simulations in particular was originally written by Dr. Jiangsheng Xiang. (e-mail : j.xiang@imperial.ac.uk)

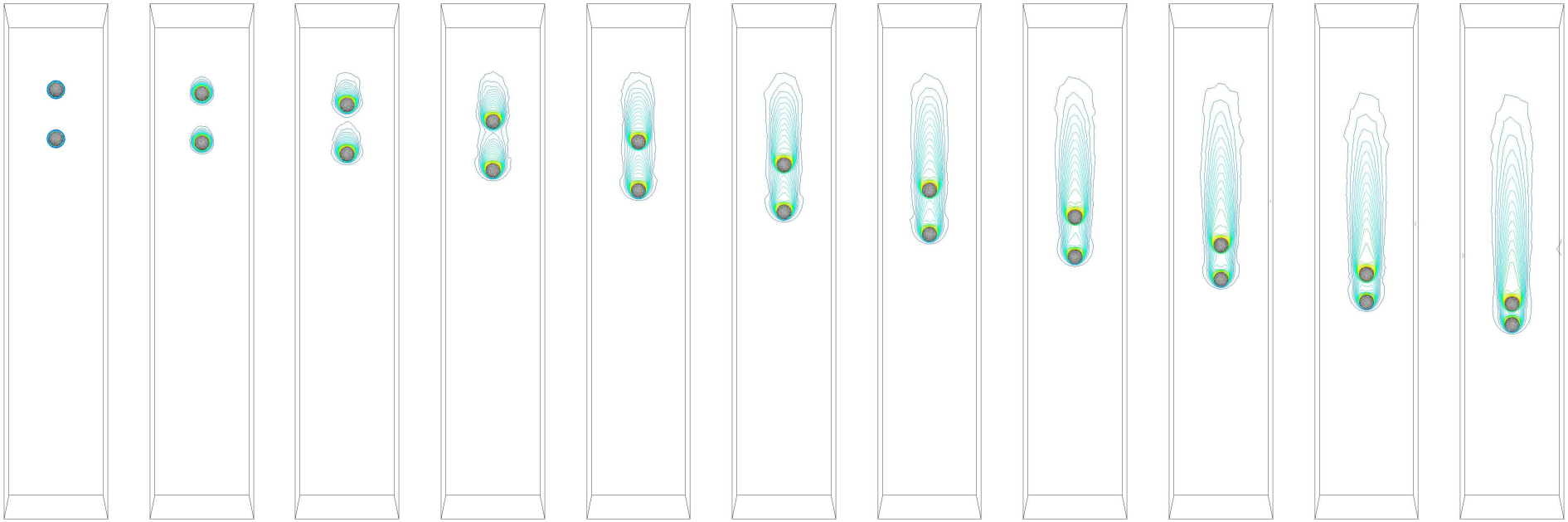


Figure 4.33: Sequence of frames for the two spheres in tandem drop simulation. The time frames included are for t =0.5, 1.0, 1.5, 2.0, 2.5, 3.0, 3.5, 4.0, 4.5, 5.0, 5.5, and 6.0. Data shown are velocity magnitude contours for a middle cut-plane across the spheres.

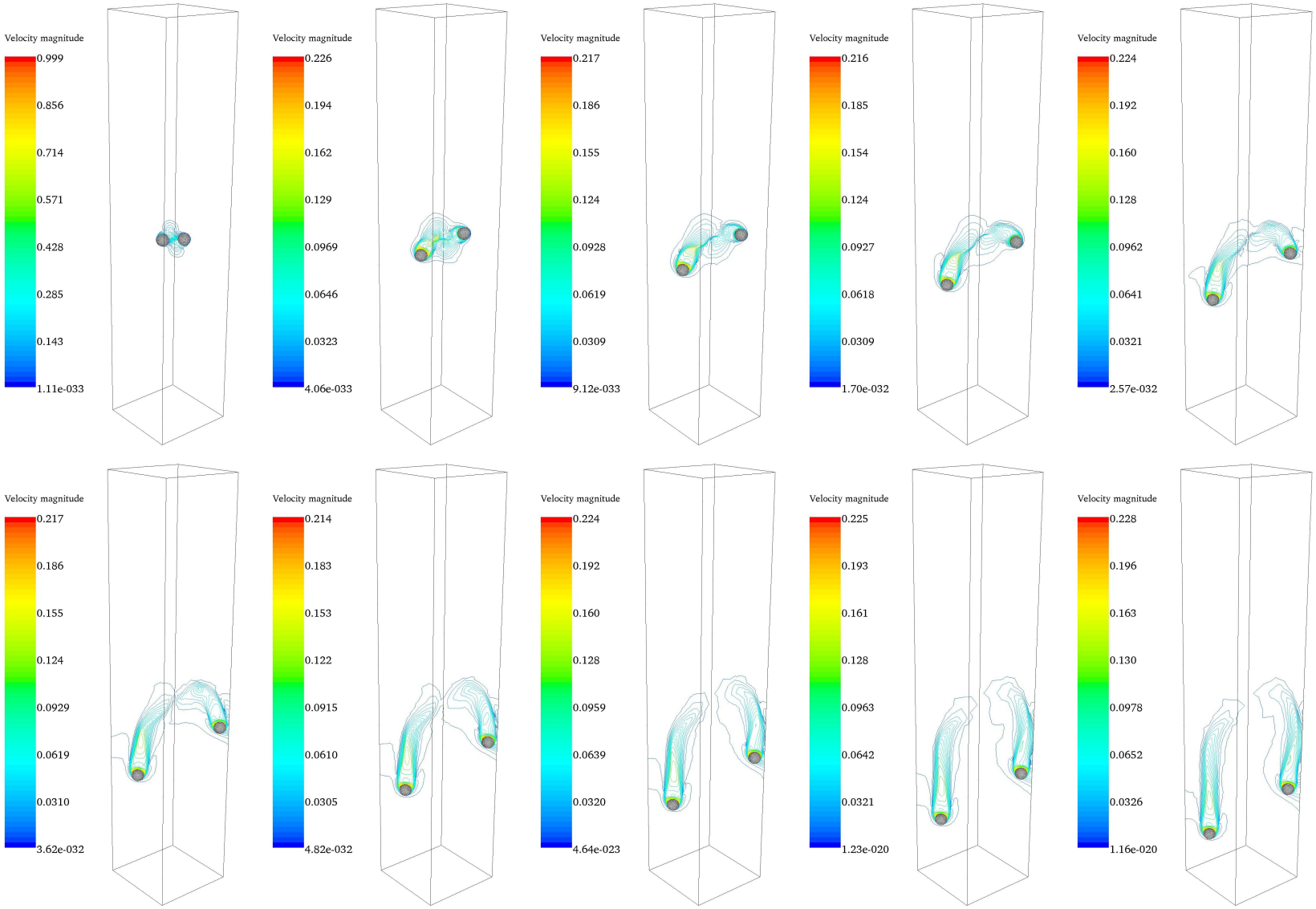


Figure 4.34: Sequence of frames showing the impact of two spheres initially set off against each other in an otherwise quiescent fluid. The time frames included are for t =0.02, 0.04, 0.06, 0.08, 0.1, 0.12, 0.14, and 0.16 seconds. Data shown are velocity magnitude contours for a middle cut-plane across the spheres.

4.5.7 2D FEMDEM one way coupling

This particular test case was constructed as a proof of concept for the coupling between Fluidity's new solid coupling method presented here and a FEMDEM[11] solver. The problem consists of a triangular shape conformed of a group of triangles weakly linked together. The triangle particle is placed extremely close to the domain bottom and with an initial downward velocity, to allow a forceful impact. Due to the collision, the bigger triangle breaks into pieces, revealing spaces which the fluid must fill in. A sequence of frames providing evidence of the successful coupling between FEMDEM and the multiphase solid-coupling approach presented in this Chapter is shown in figures 4.35, 4.36, 4.37, 4.38, and 4.39. Figures 4.35 and 4.36 show simulation time frames of a triangle of base 1.0 and height 0.5 composed of 16 smaller triangles impacting on the bottom of a domain, and splitting apart due to the collision. Initial downward velocity of -3 $m \cdot s^{-1}$and fluid properties used were $\rho_f = 1000 kg \cdot m^3$ and $\mu = 100.0 Pa \cdot s$. Figures 4.37, 4.38, and 4.39 depict a similar case, however the subdivision of the solid was much greater.

[11]The solver "y2d" used for this simulation was originally coded by Professor Antonio Munjiza.

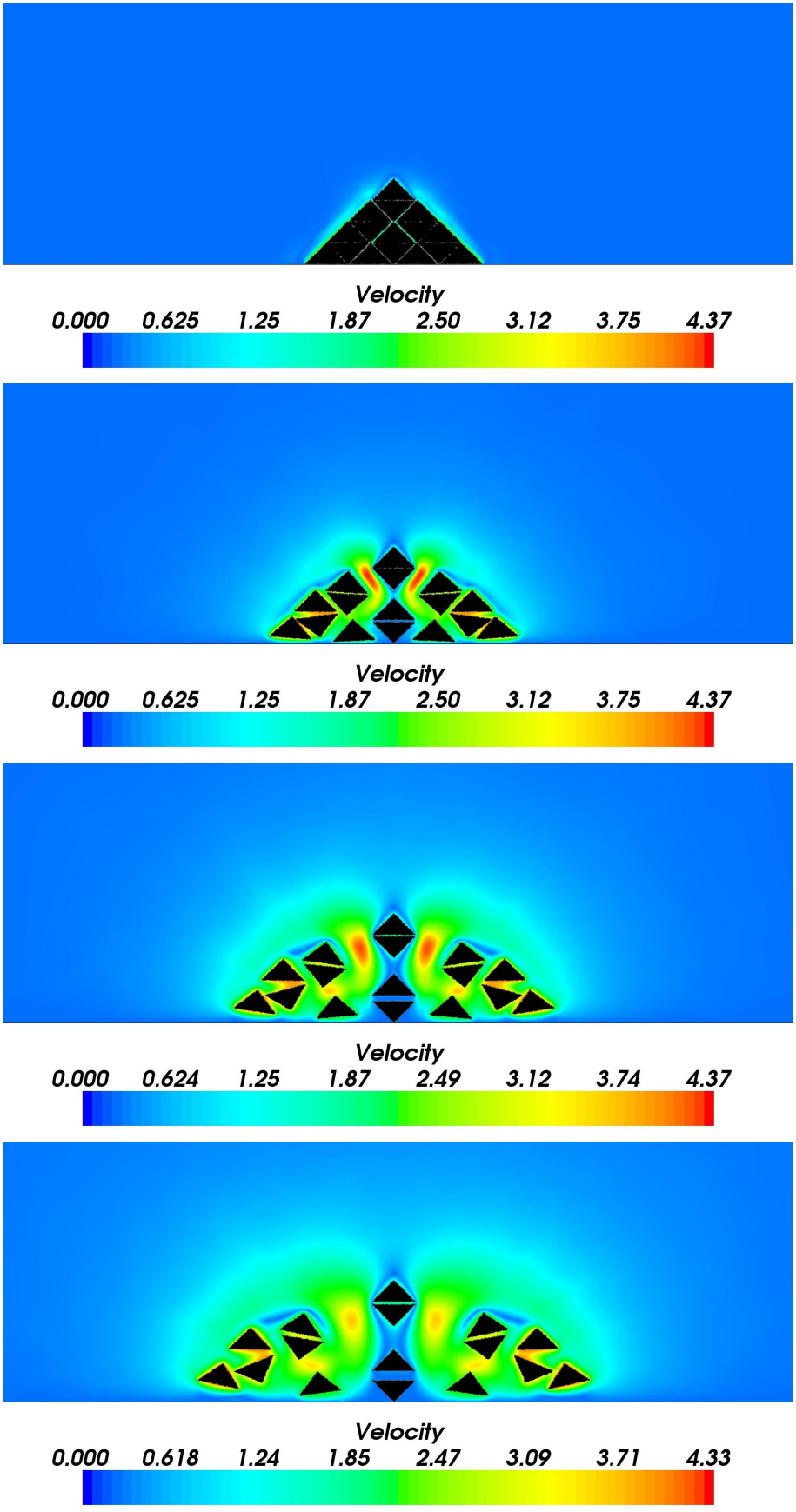


Figure 4.35: Sequence of frames showing the two dimensional triangle impacting a horizontal base and splitting into its smallest components. Time frames, in order from top to bottom, are for t =5, 55, 105, and 155 milliseconds

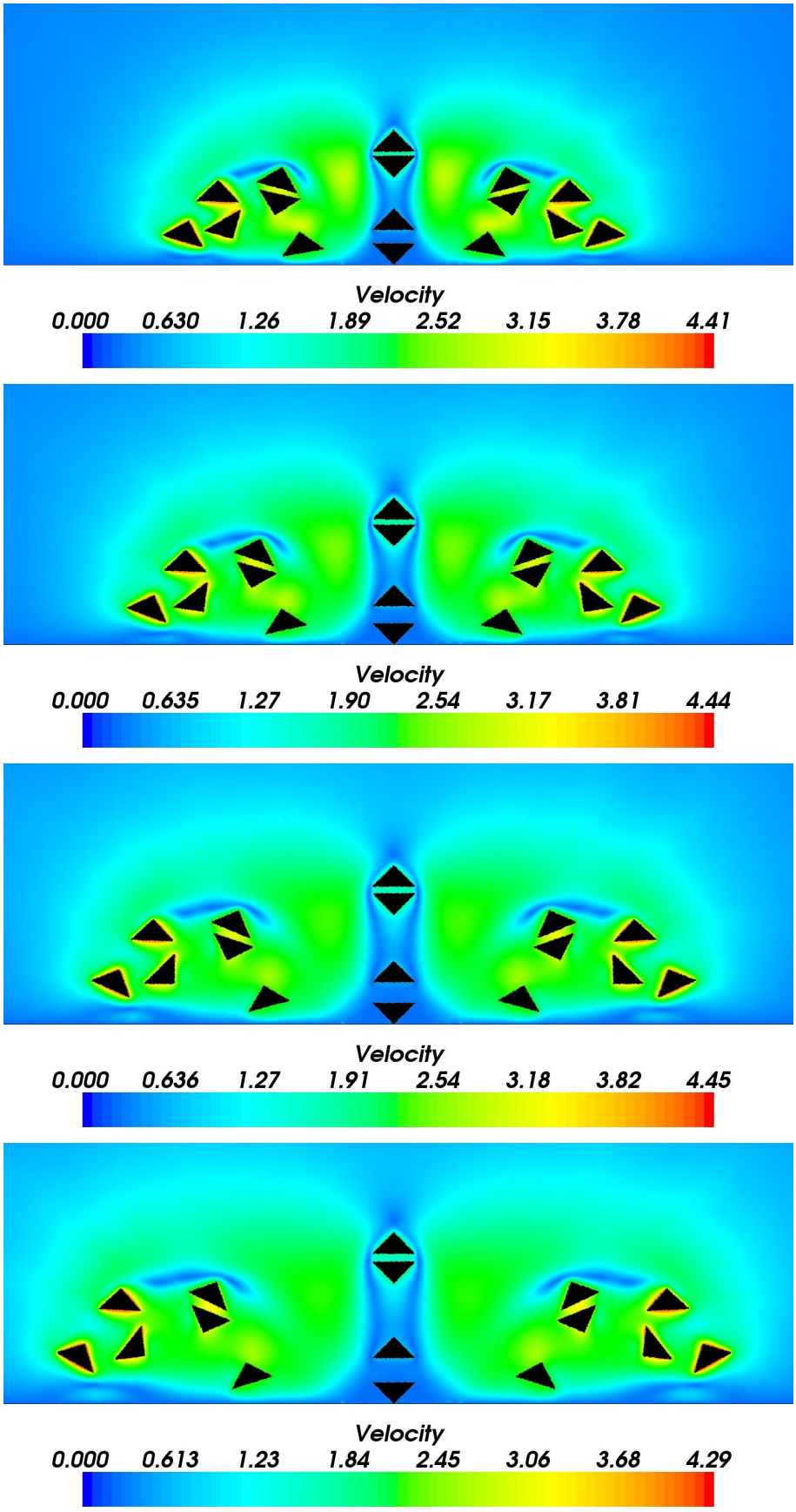


Figure 4.36: (Continued from Figure 4.35) Sequence of frames showing the two dimensional triangle impacting a horizontal base and splitting into its smallest components. Time frames, in order from top to bottom, are for t =205, 255, 305, and 355 milliseconds.

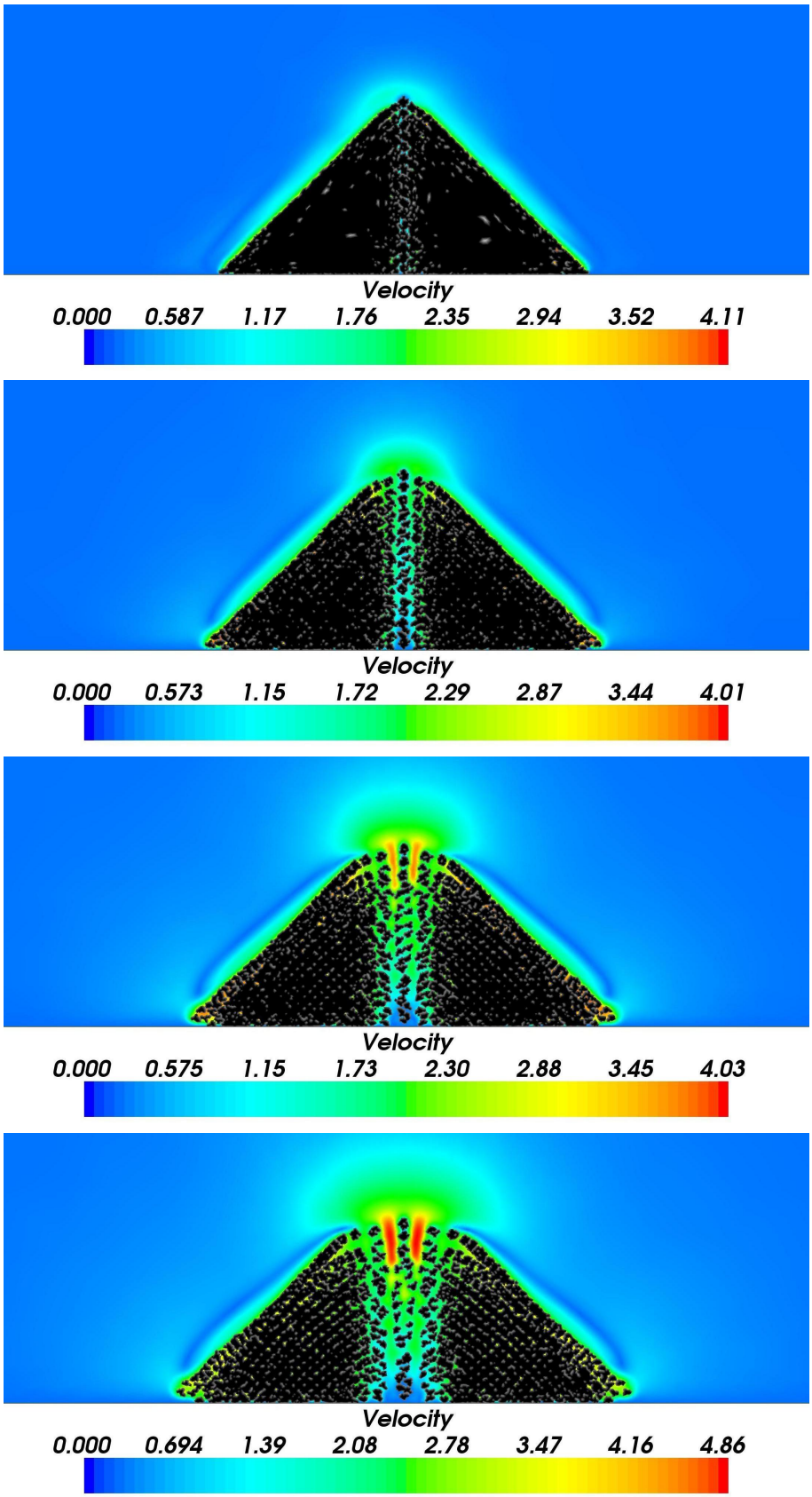


Figure 4.37: Sequence of frames showing the two dimensional triangle impacting a horizontal base and splitting into its smallest components. Time frames, in order from top to bottom, are for t =5, 15, 25, and 35 milliseconds

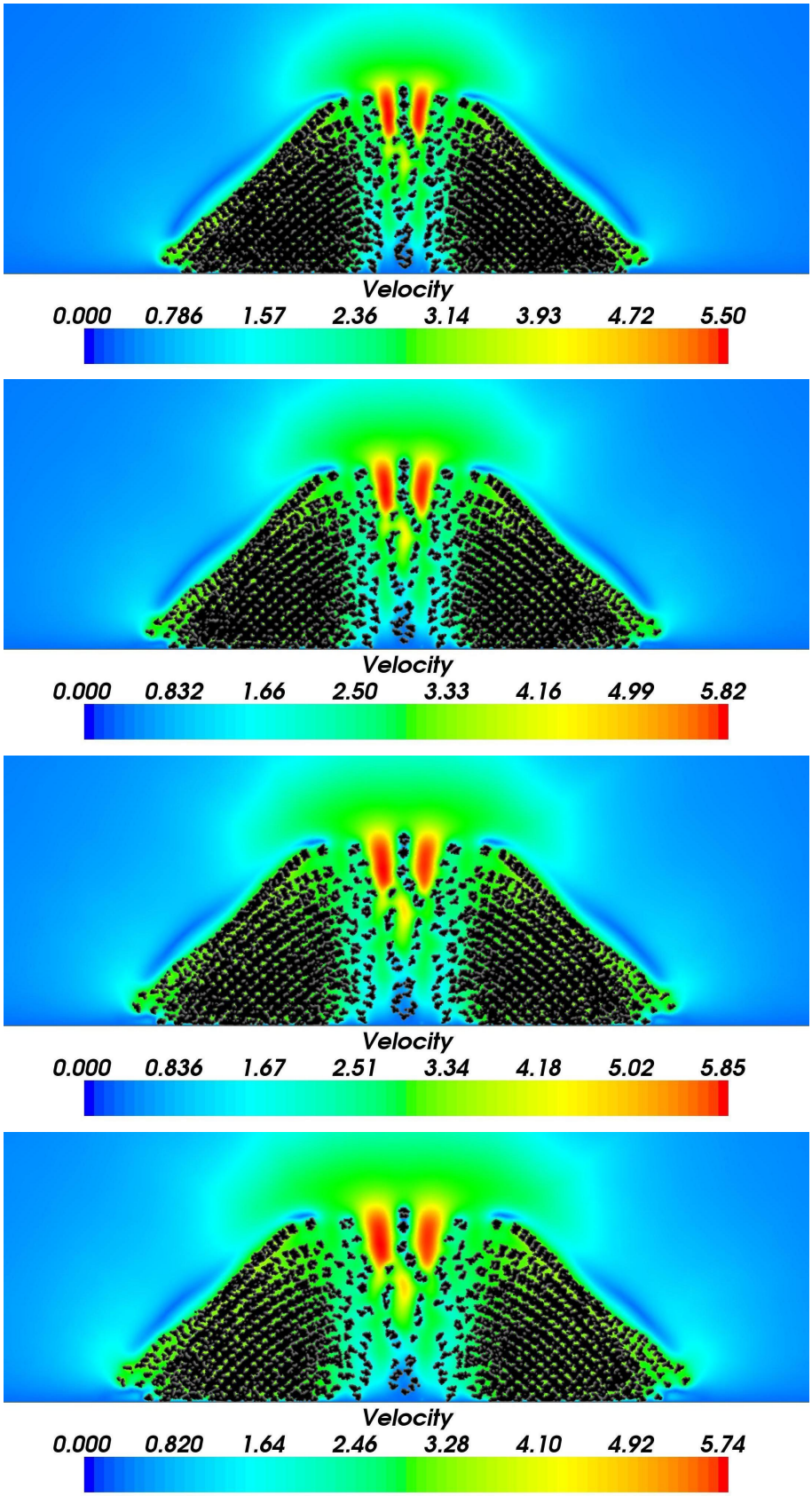


Figure 4.38: (Continued from Figure 4.37) Sequence of frames showing the two dimensional triangle impacting a horizontal base and splitting into its smallest components. Time frames, in order from top to bottom, are for t =45, 55, 65, and 75 milliseconds.

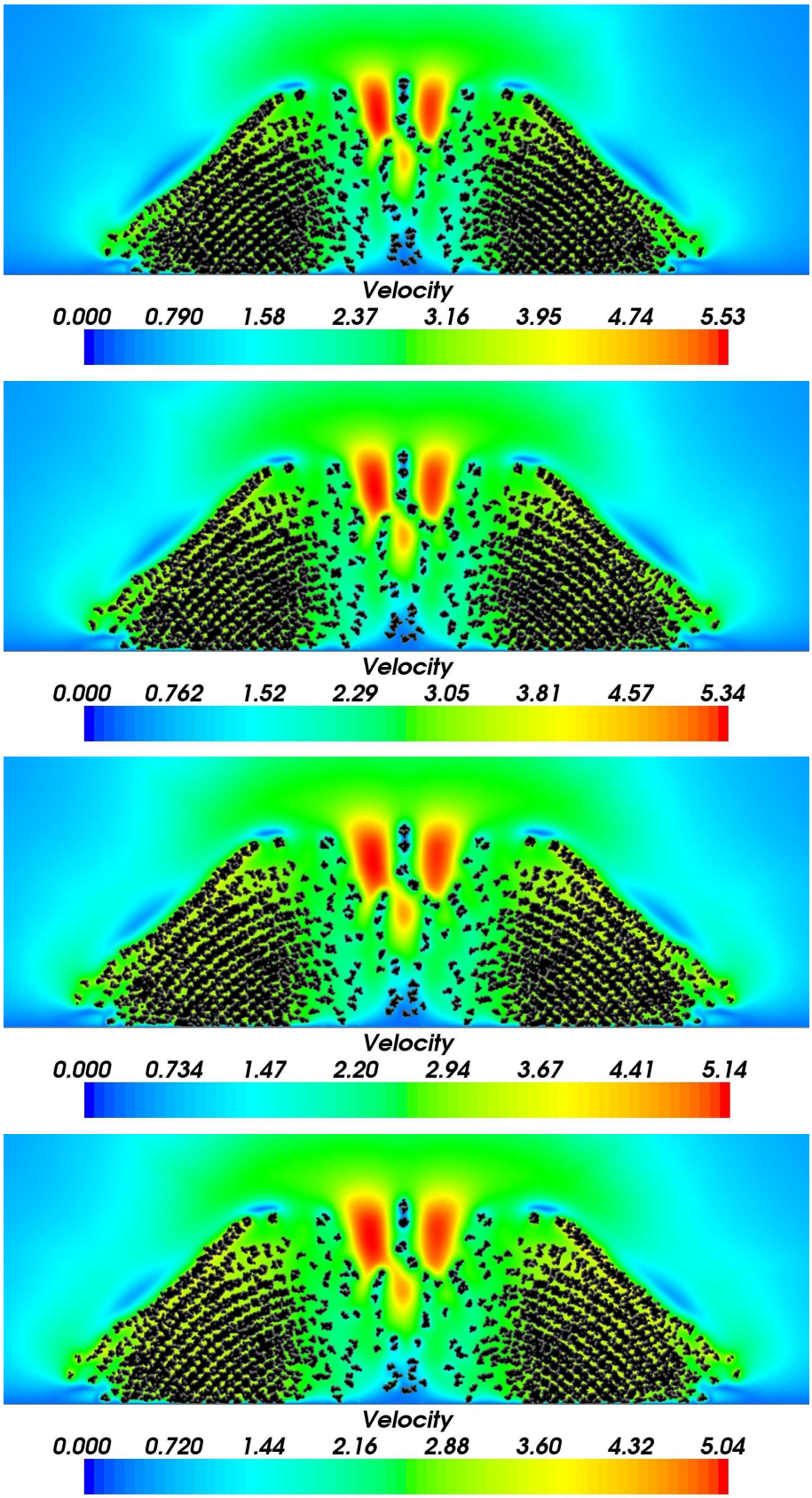


Figure 4.39: (Continued from Figure 4.38) Sequence of frames showing the two dimensional triangle impacting a horizontal base and splitting into its smallest components. Time frames, in order from top to bottom, are for t =85, 95, 105, and 110 milliseconds.

4.6 Conclusions

A method was presented in this Chapter to model fluid interaction with solids. The technique involves the coupling of a CFD solver to Lagrangian particle solvers involving DEM and/or FEMDEM methods. Using unstructured meshes to resolve solid shapes and fluid flow details, the method is also able to capture particles which are too small to be resolved and are yet able to affect the flow. Drag parameterisation found in the literature and used in multiphase flow models were used to provide a smooth transition between the highly resolved and the under resolved particles. Mapping techniques were implemented to permit the modelling of solid particulates ranging from the simplest of shapes to the most complex that may be defined by a tetrahedral mesh.

A multiphase approach was chosen to allow flexibility of meshes when solid particles eventually break or suffer large deformations. Furthermore, the technique developed easily integrates with the method presented in Chapter 2 for the modelling of fluid interfaces. Results from test cases in section 4.5 show the method's capability to capture forces accurately for a wide range of Reynolds numbers, as well as steady and unsteady flows.

Chapter 5

Applications to coastal defence design

The effective design and construction of coastal defence and harbour breakwaters remains a huge challenge to coastal engineers world wide (i.e. as discussed in Chapter 1 in relation to concerns with climate change). Their stability has mostly been studied by means of physical modelling (e.g. see Palmer and Christian (1998); Hudson and Jackson (1953); Van der Meer (1988)). Some advantages of physical modelling are the ease by which forces on the structures can be obtained through small scale models based on Froude's law (see, for instance, White (1999, pp 294, 303-304)), and that due to the large number of variables involved, theoretical descriptions are complex. The main disadvantage of physical models is the scale effect, which occurs when ratios of forces or physical properties, such as the Reynolds number, cannot be scaled correctly (see Allsop (2005, page 106)). Other unwanted effects in laboratory experiments such as the parasitic reflection from model boundaries have been resolved by applying modern wave generation techniques, of which a good review can be found by Naito (2006). Large scale modelling is possible and would avoid scale effects, however facilities and infrastructure might be prohibitively expensive or unavailable for many design projects.

On the other hand, numerical modelling has been developed over the past decades. It has complemented, or even replaced to some extent, physical modelling as it tends to be less economically expensive. This fact is emphasised by advancing technology and faster processing capabilities involving higher CPU clocks and/or parallel processing. It can be used to simulate coastal processes at any scale, for example, to examine water particle velocities, force transmission and structure response as the wave energy dissipates. This approach will be needed in the future to better understand, for example, whether a beach, structure, or a combination of both is likely to be effective or not. The approach will also be instrumental in understanding how the energy of sea waves can be best harnessed for clean energy production, and to investigate the multi- purpose utilisation of the sea as a resource. To contribute to this now well established research field that seeks to create numerical models for wave structure study (e.g. numerical wave flumes, tanks, etc. Hsu et al. (2002); Lara et al. (2006); Lin and Liu (1998)) important steps have been taken in this thesis towards a numerical flume to generate 3D waves that surge, break or spill on a sloping foreshore or structure.

The main objective of this chapter is to show the advances in the usage of the tools developed in previous chapters, by bringing them into the coastal engineering context. Results are shown in which waves are generated through the collapse of a water column, thus achieving a qualitatively similar behaviour to that observed, for example, for a solitary wave in a wave basin. The chapter begins by providing a brief background on physical modelling techniques underlying design parameters of coastal defence structures, followed by numerical modelling approaches found in the literature. Two dimensional and three dimensional results are finally presented, depicting wave interaction with a slope, as well as structures which represent breakwater armour units.

5.1 Design techniques

Assuming the use of rubble mound breakwaters as a coastal defence measure, the design objective is to determine the stability, size, and layout of the composition of the breakwater. As also explained in **?**, Chapter 5 there are four groups of parameters to be kept in mind while attempting to achieve structure stability, and they are wave attack, rubble mound composition, cross-section, and structure response parameters. Most of the information regarding these parameters was made available over decades of experience and physical modelling, resulting in guidelines such as USACE (2002); **?**.

The primary function of a rubble mound coastal defence structure (i.e. granular structure protected by loosely placed rock or concrete blocks) such as a breakwater or revetment is to prevent water waves from excessively damaging the coastline. It works by dissipating the wave energy through wave run-up and turbulent flows in the voids between blocks, however if the slope of the structure is sufficiently flat, further energy is dissipated through overturning and plunging of the waves. Left over energy is transformed into potential energy and together with outflows from the porous mound within the core of the structure, is returned to the sea as reflected waves. Some water might flow over the structure (i.e. overtopping discharge) if the wave height is sufficiently high and the breakwater crest is sufficiently low compared to it during storm conditions of tide and swell. Consequently, a breakwater's effectiveness (i.e. cost is another factor in design optimisation) is measured in terms of how much energy it can dissipate from incoming waves without unacceptable displacement and damage of armour units.

The typical design procedure of a coastal defence structure involves determining a number of threshold conditions. Given that these sloped structures are typically made up of layered rubble mounds reinforced with heavy armour units (see Figure 5.1), a common design criteria that stems from physical modelling is the establishment of a damage tolerance. A certain amount of material, for example sand, gravel, stones, or even armour units is allowed to shift around under wave loading. The amount of material allowed to shift under a given wave load condition sets the threshold level. If the material shifting exceeds this level, the design is deemed unstable and must be altered or improved. In order to test for stability, a physical model is constructed,

normally based on a scaled version of the structure using Froude's law (see Figure 5.2), and parameters such as maximum run-up, wave reflection, and wave transmission are measured using a wave flume or basin. Scale ratios normally depend on flume tank water depth, wave heights reproducible by the flume's wave generator, and the available sizes of the model armour units.

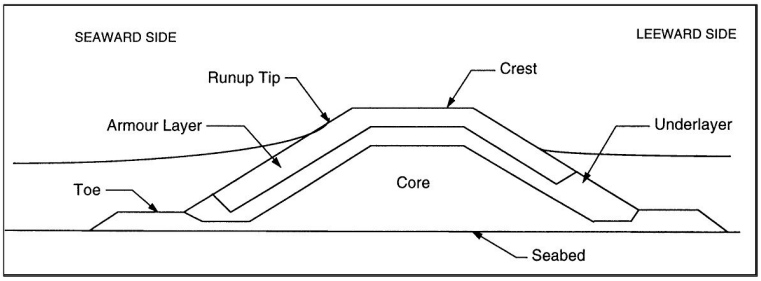


Figure 5.1: Typical breakwater cross section (diagram obtained from Palmer and Christian (1998, page 21))



Figure 5.2: Setup of a multi- layer breakwater for a scale model test in a wave basin; colouring is designed for ease of interpretation of damage patterns. (picture obtained from **?**)

Physical modelling is currently the methodology used to address most coastal engineering design questions. The basic physical model consists of a scaled version of the structure subjected to waves generated in a wave flume or wave basin. Given that the real structure is of a large magnitude, such experiments become economically justified. Larger scale models are built in some cases, however the increase in cost is exponential.

In a physical model test, the wave representation is of the utmost importance. Wave generators, also called wave makers, are used to create waves. The manner in which these wave makers impel water is what turns them into a tool for engineers to input most wave shapes into the wave flume or basin. Many state of the art wave makers are able to input parameters like the

spectral distribution for irregular waves, such as the JONSWAP[1] spectrum (see Hasselmann et al. (1973) for details on this spectrum). Furthermore, wave makers must be able to acceptably absorb reflected waves by correcting the impeller position instantly. These reflected waves are an unwanted effect, since in reality they travel towards the open sea.

5.1.1 Wave attack parameters

The stability number N_s gives the relationship between the design wave conditions and the primary armour structure element. This parameter is calculated from,

$$N_s = \frac{H_s}{\Delta D_r} \tag{5.1}$$

where H_s defines a wave height, typically the significant wave height[2], D_r is the characteristic size of the armour stone, and Δ is the ratio between densities, given by,

$$\Delta = \frac{\rho_r - \rho_w}{\rho_w} \tag{5.2}$$

where ρ_w is the water density and ρ_r is the rubble mound apparent mass density. The latter is equal to the concrete density in the case of concrete armour units. Van der Meer (1988) gives the following guidelines for stability analysis based on the stability number:

- In the case of caissons or sea walls, $N_s < 1$ and no damage is allowed since the structures are fixed. D_r is given by a reference dimension such as the height or width of the structure.
- In the case of statically stable breakwaters, $1 < N_s < 4$. This is the typical and most used case that relies on the mass and interlock of loosely placed outer layer of reinforcing with heavy rocks or armour units. Very little damage is allowed in this case, and units are typically replaced as they are damaged. D_r in this case is given by a characteristic dimension of the armour unit or rock.
- In the case of dynamically stable structures, such as berm breakwaters, $3 < N_s < 6$ as changes from the initial shape are expected.

A second important parameter is the surf similarity parameter (see Iribarren and Nogales (1949)), also known as the surf parameter or Irribarren number, which is a dimensionless parameter used to describe the characteristics of wave phenomena. This parameter is given by,

$$\xi = \frac{tan(\theta)}{\sqrt{\frac{H_i}{L_o}}} = \frac{tan(\theta)}{\sqrt{\frac{2\pi H_i}{gT^2}}} \tag{5.3}$$

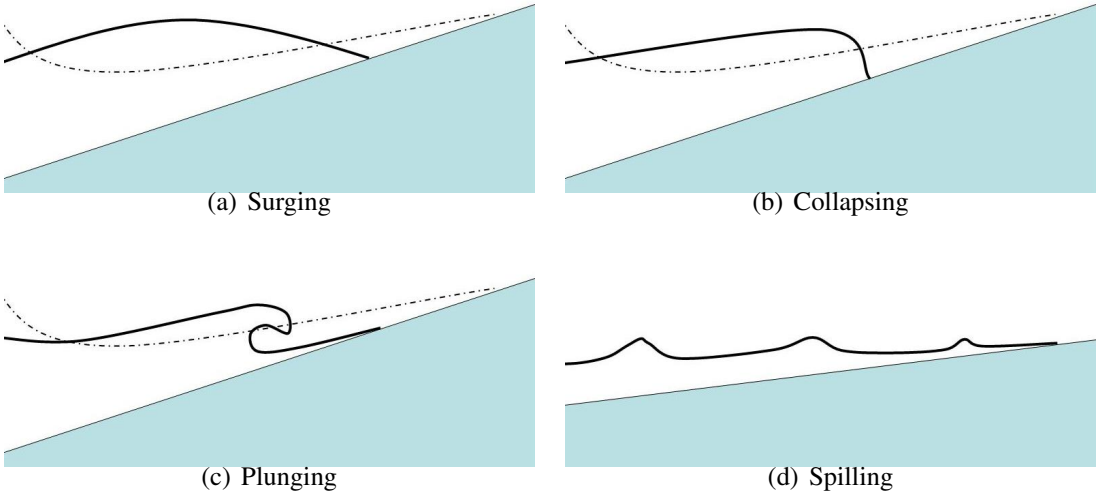
[1]The JONSWAP (Joint North Sea Wave Project) spectrum is an empirical relationship that defines the distribution of energy with respect to frequency in the ocean.

[2]The significant wave height is the average height of the highest third of the waves in a given sea state

where H_i is the incident wave height, $tan(\theta)$ is the slope of the sea bottom or structure, T is the wave period, g is the gravitational acceleration, and L_0 is the deep water wavelength (i.e. the wave length the wave would have if it would be in deep rather than shallow water). Using linear wave theory, the surf similarity parameter is normally used to determine whether or not waves are breaking, and if they are, which type of breaker to expect. This allows for design options based on how much energy is being dissipated during the shoaling, breaking, and run-up of the waves. Battjes (1974) established that for a slope range between $1/20 \leqq tan(\theta) \leqq 1/5$, the expected wave type is,

1. Surging and collapsing if $\xi > 3.3$
2. Plunging if $0.5 < \xi < 3.3$
3. Spilling if $\xi < 0.5$

Examples of the types of wave breaking are shown in Figure 5.3.



(a) Surging (b) Collapsing

(c) Plunging (d) Spilling

Figure 5.3: Diagram of breaker types (Battjes, 1974).

On a further note, structural response to wave attack depends on the approach followed during the design. If the structure design is aimed to be statically stable not much damage is allowed, which is measured through the number of displaced units or the amount of profile deformation suffered by the structure. In the case of concrete armour units, it is difficult to determine the new profile of the cross section, and hence the damage is assessed by calculating the number of units that are displaced out of a given section width of the breakwater, and/or by the ratio of units displaced over the total number of units originally within a given area. If the structure is being designed as dynamically stable, the damage allowed is considerably larger, and stones and/or gravel will shift position from the original design profile until a constant profile is obtained. It is assumed that for a wider consideration of dynamic structures including

gravel and sand beaches, a dynamically stable rubble mound profile is reached when the net transport of material (e.g. rick armour, gravel, sand, etc) is practically zero over a period of years (i.e. profile fluctuations are naturally expected during storm events).

5.2 Numerical modelling

As stated by Peregrine (2003), obtaining reproducible results in experiments is difficult, particularly due to the effects of turbulence and pressure measurement issues. Additionally, the effects of compressibility of air which is normally mixed in with the breaking wave are not easily scaled to prototype size, and the typical usage of Froude scaling normally results in over prediction of forces due to violent wave impacts. Even today the prediction of wave characteristics that reach the structures, given the offshore wave climate data (e.g. from a wave rider buoy historic data), and bathymetry is a challenging and important task. In contrast with numerical modelling, not all variables in a physical model are easily measurable, and this might incur into problems when determining design variables for empirical formulae. A study was presented by Chan and Melville (1988) where deep-water wave plunging on vertical walls was analysed. It concluded that the violent outcome of wave breaking was not only caused by the inertia forces, but also by the trapping of air. These impacts lead to high impulse pressures which are augmented by the compressed air. Bullock et al. (2007) performed experiments with violent wave impacts, with particular attention to entrained and entrapped air, concluding that aeration increases both the force and the impulse on the structure.

On another note, as reviewed by Hamm et al. (1993), there are several aspects to the modelling of coastal processes which also deserve particular attention: wave randomness, refraction due to structural obstacles, energy dissipation, and non-linearities. The latter aspect is of high importance in the nearshore zone.

A possible solution for modelling is to use a mathematical description of the governing processes, such as Airy theory or Stokes theory. See, for instance, Airy (1845); Le Roux (2008) for a description of Airy waves, and Chang and Liou (2006); Stokes (1847) on Stokes waves. A good review on wave theory may also be found in Stoker (1957). Weggel and Maxwell (1970) utilised a small amplitude pressure disturbance model to study the evolution of impacts against vertical walls. Walkden et al. (2001) studied the seaward impulse loads on caisson breakwaters caused by overtopping, by comparing physical model tests to a theoretical model for the pressure impulse. Tedesco et al. (2003) studied the response of a particular type of armour unit by using a semi-empirical wave force model.

Unfortunately, analytical solutions are often limited in application due to their high complexity. The above facts are accountable for numerical modelling being increasingly seen as a viable cost-effective solution, and thus a considerable amount of research has been done over the last few decades.

Most existing numerical models to simulate waves are based on the depth-integration of the

Navier Stokes equations. Shallow water equations, or the Boussinesq approximation are typical choices. However, since these models inherently disallow the proper breaking of the waves (i.e. waves cannot overturn), parameterisations must be introduced to emulate the energy dissipation due to shoaling and breaking. Zelt and Raichlen (1990) developed a Lagrangian model for the Boussinesq equations and used a finite element model to research wave run-up. An artificial viscosity term was added to the momentum equations to take into account energy dissipation of breaking waves. Li and Raichlen (2002) solved the non linear shallow water equations using a shock capturing scheme and parametrised energy dissipation, although neglecting bottom friction. The work of Kobayashi (1989); Kobayashi and Wurjanto (1990) should also be mentioned, where a shallow water model for wave run up on a slope was developed and the water velocities were coupled with stability criteria for the case of slopes of rubble. In this way stability design curves for monochromatic waves were derived directly through the numerical model.

Depth integrated methods, in general, suffer from their inherent restrictions through the introduction of semi-empirical factors to account for breaking and porous media flow, as well as some difficulty when simulating free surfaces. It must be said, however, that they are very efficient and are well suited to analyse wave evolution over long periods of time (e.g. modelling long wave trains).

Tanaka et al. (1987) extended a method developed originally by Dold and Peregrine (1986), where fluid is modelled as incompressible and irrotational, and a boundary integral method is used for the time integration. The method was used in particular to study the impact of violent solitary waves. Peregrine and Cooker (1990) found that the wave face can achieve accelerations exceeding 1000 times gravity. Cooker and Peregrine (1990) compared the results of this boundary integral method to experiments by Arami and Hattori (1989), who examined 10 cm high waves.

Methods which have recently increased in popularity in coastal engineering applications, mostly due to advances in computer technology, are those that involve the solution of the Navier Stokes equations. These methods allow the solution of the whole velocity field taking the least amount of simplifying assumptions (e.g. flows may be rotational or irrotational, viscous or inviscid, etc.). Reynolds averaged Navier Stokes (RANS) type methods are very popular in modelling coastal processes, as they can account for turbulent interactions (i.e. generation and dissipation) within flows. Hsu et al. (2002) developed a Volume Averaged Reynolds Averaged Navier Stokes (VARANS) method to describe surface waves interacting with permeable or impermeable structures, which also included turbulent effects within the porous material. A sharp interface between air and water was kept by using the volume of fluid method (VOF, see Chapter 2 in this thesis for more details). Lara et al. (2008) compared results when using a the code COBRAS-UC, based on the VARANS method, to laboratory tests in a 1:20 wave flume using both regular and irregular waves interacting with permeable structures. VOF was also used for fluid interface capturing in this case. The COBRAS code was based on free surface modelling code RIPPLE, developed by Kothe et al. (1991b). Losada et al. (2005) introduced a 3D Navier

Stokes equations solver using the a finite volume discretisation. The method used a Large Eddy Simulation model for turbulence, and used a modified version of VOF by Hirt and Nichols (1981) to capture interfaces.

Another group of relatively recent and popular approaches to numerical modelling are the mesh-less approaches. These are based on a Lagrangian motion of nodes or particles of a fluid or other material, and are able to model high deformations such as the evolution of a free surface. Koshizuka and Oka (1996) presented a moving particle semi-implicit method (MPS)[3] which used the solution of the pressure Poisson equation (PPE, see, for instance Gresho and Sani (2000, page 457)) with the assumption of incompressibility. Another mesh-less scheme is the smoothed particle hydrodynamics method (SPH), first introduced by Gingold and Monaghan (1977) for modelling astrophysics. Shao (2006) also developed an incompressible SPH modelling method using a turbulence model for wave breaking. Dalrymple and Rogers (2006) developed a similar SPH model, however it was not completely incompressible, and hence a sound speed was present in the simulations. This difficulty, coupled to the fact that, in general, SPH methods require a large amount of resolution (i.e. number of particles) yields extremely small time steps in the simulation, leading to lengthy run times and overall time inefficiency.

Compared to the alternative approaches mentioned here for fluid-fluid and fluid-structure interaction, the approach presented in this thesis has some distinct advantages in that it can employ the excellent flexibility of tetrahedral meshes to model geometries, while being able to optimise them through an adaptive mesh optimisation procedures (AMO). It must also be mentioned that unstructured meshes may be used while modelling sharp interfaces mainly due to the existence of the compressive advection method developed in Chapter 2, which is conservative and efficient. Furthermore, although under development at the time of submission of this thesis, the method may be coupled to a FEMDEM model which can simulate solid structures of any shape and resolve the internal stresses associated with its interaction with the fluids and/or other structures as well. Indication of this was given in the FEMDEM test case in Chapter 4. As can be seen in the following sections of this chapter, 3D results produced are very promising and encourage further developments. The combined Fluidity/FEMDEM approach presented here is complementary to others that may be less computationally costly, such as Lattice-Boltzmann methods which have also been coupled with DEM particle methods, (e.g. see Feng et al. (2007)). However, the building of a numerical wave flume with realistic random waves requires detailed capture of the fluid interface, either with another fluid or a structure, which allows engineers to, for example, obtain a detailed distribution of stresses on each particle. This is of particular interest for single layer armour unit design. While Reynolds Average Navier-Stokes solvers in combination with volume of fluid (VOF) methods have proved effective (e.g. see Lara et al. (2006)), the results achieved with Fluidity using adaptive unstructured meshes, shown below, also look to be a very promising basis for wave modelling. The advantages of adaptive meshes come into play again in their potential for greater accuracy in the drag

[3]Note that this particular method was used in the 2D collapsing water column validation case in Chapter 2.

prediction for coupled problems as the solid geometry is better defined in the fluid domain.

5.3 Results

The subsections below present a series of results from 2D and 3D simulations. These may not qualify as a proper validation of coastal applications for the techniques presented in this thesis, however they provide a very promising qualitative assessment of the performance of the developed method.

5.3.1 Two dimensional wave breaking at different slope angles

Carrier and Greenspan (1958) performed an analytical study of wave run up on a sloping beach using non-linear inviscid theory, moreover their focus was on non-breaking waves. Lee and Heo (2005) studied wave motions in shallow waters and on slopes using a two dimensional finite difference discretisation of the Navier Stokes equations. Stokes wave theory was employed to generate waves at the inflow boundary. The essential aim in these cases was to compare breaker types with predictions based on Irribaren's surf-similarity parameter (e.g. see Iribarren and Nogales (1949); Battjes (1974)).

In a similar attempt, and as a demonstration of wave-simulation capability, the numerical scheme presented in this thesis was applied in a two dimensional domain as described in Figure 5.4. An initial disturbance was generated by a collapsing water column on the left hand side of the wave tank, thus generating a sequence of waves that travel to the right hand side and interact with the slope. Several simulations were run varying the angle of the slope, achieving characteristic dynamics for each case. Fluid properties used (i.e. density, and viscosity) correspond to those of air and water at standard temperature, and free slip boundary conditions were used on all surfaces.

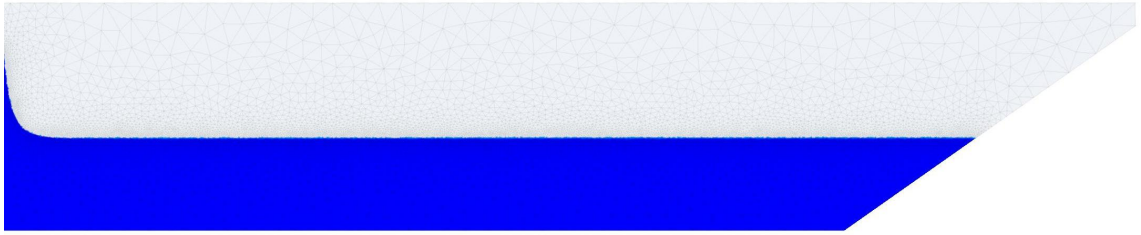


Figure 5.4: Sketch of the setup for the 2D simulations of wave run-up Height of the domain is 0.5 m while length is 2.5 m. Initial still water depth was set to 0.2 m, while the slope pictured here is of 0.7.

Whereas it is recognisable the waves created are not a regular wave train and hence a wave period cannot be established to use Iribarren's formula (see equation (5.3)), it is reassuring to observe plunging and surging waves in expected regimes of slope angle (i.e steep slopes tend to produce surging breakers, while mild slopes tend to produce plunging or spilling breakers).

See Figure 5.7 on page 193 for frames corresponding to these simulations. Figure 5.5 on the facing page and Figure 5.6 show a close up view on a plunging wave breaker, where refined details of the creation of the break due to backflow below combined with forward flow above can be observed. The way that the air is incorporated into the turbulent field generated by the overturning wave is particularly (and exceptionally) detailed with the help of mesh adaptivity. This level of detail has not been observed among any other methods, namely VOF and SPH.

Results were also obtained for a similar three dimensional simulation where shoaling, breaking, and run up can be observed at different angles in Figure 5.8 on page 194.

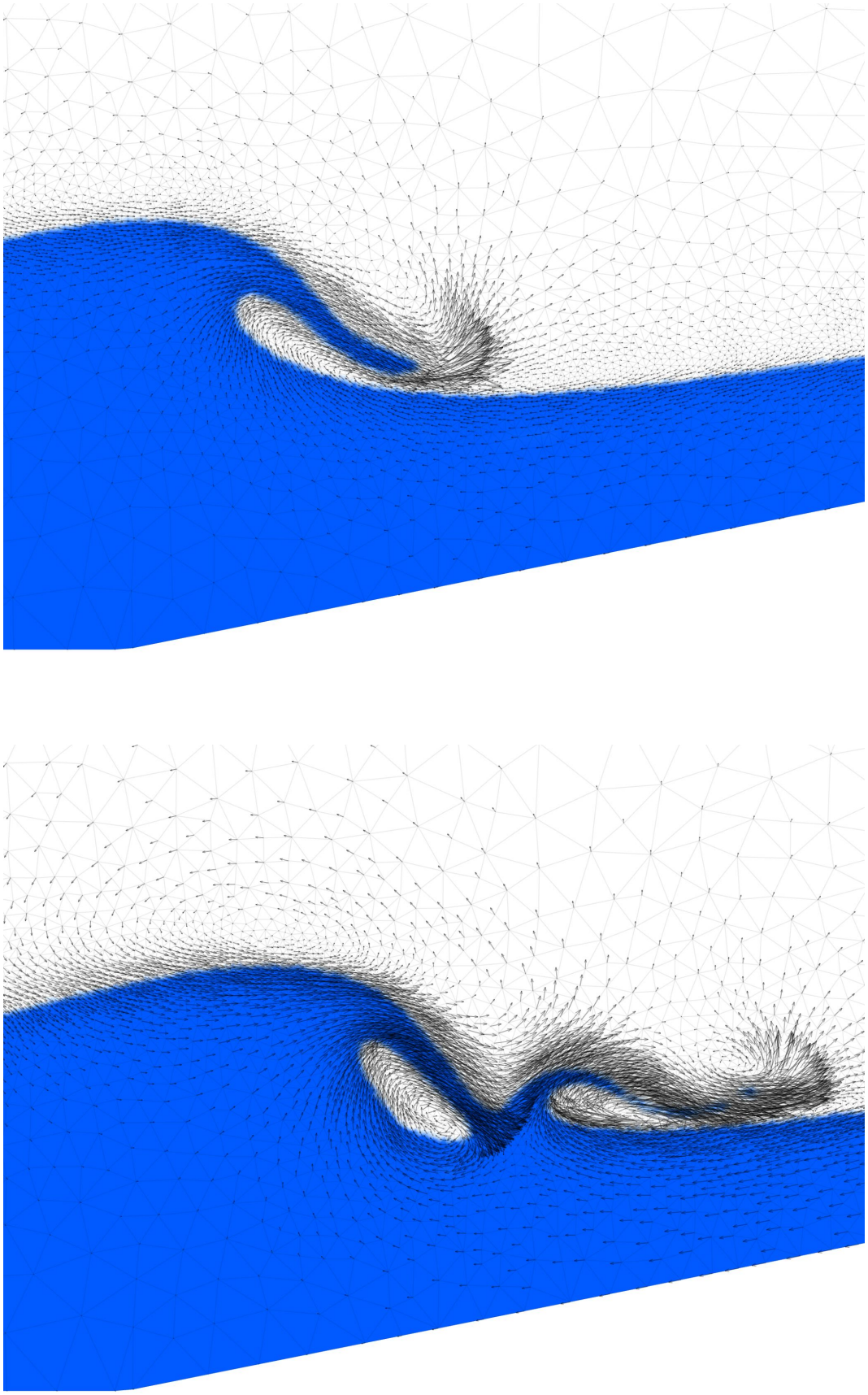
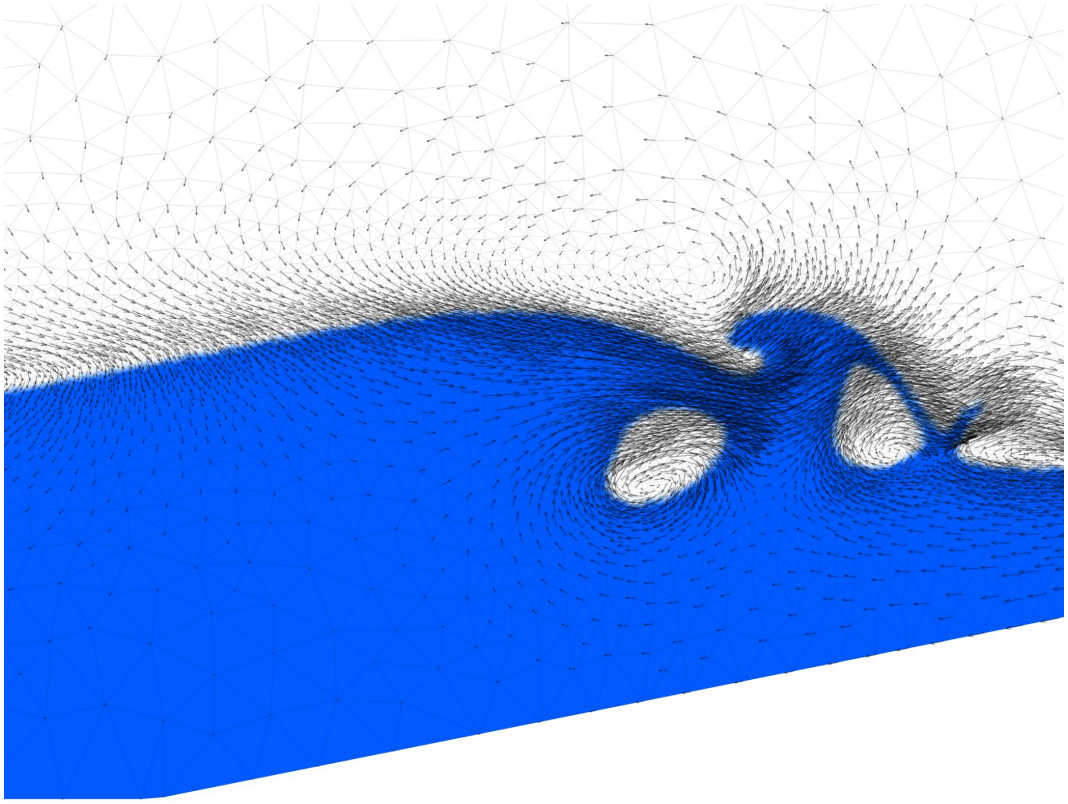


Figure 5.5: Velocity vectors within a plunging breaker. Slope angle used here is 1:5.



(a)

Figure 5.6: (Continued from Figure 5.5)

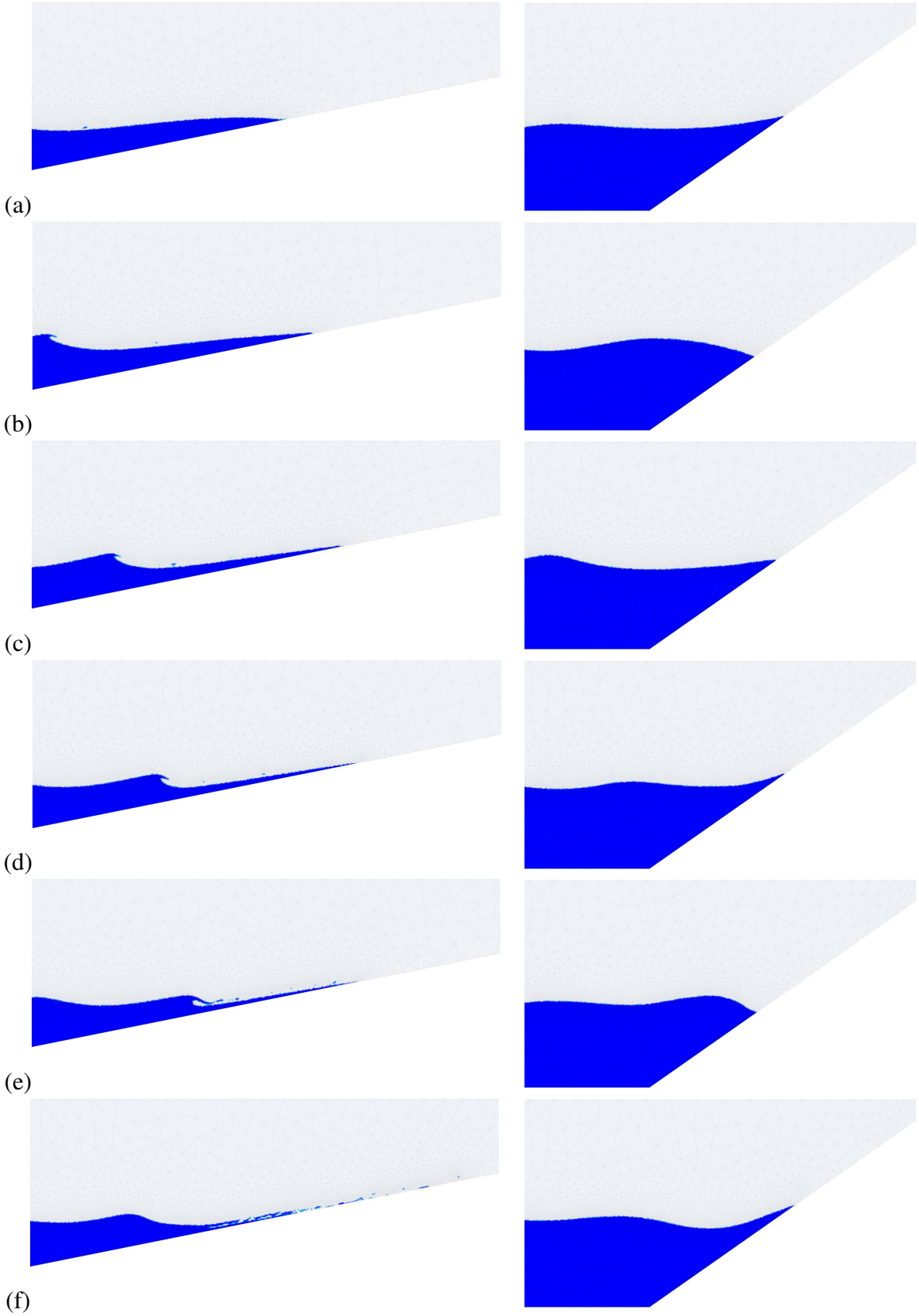


Figure 5.7: Sequence of frames corresponding to wave run up simulations at two different slope angles. For the gentle slope case of 1:5 (i.e. left column), (a) $t = 1.3\,s$, (b) $t = 1.45\,s$, (c) $t = 1.65\,s$, (d) $t = 1.80\,s$, (e) $t = 1.95\,s$, and (f) $t = 2.3\,s$. For the steep slope case of (1:1.4), (a) $t = 3.05\,s$, (b) $t = 3.30\,s$, (c) $t = 3.55\,s$, (d) $t = 3.80\,s$, (e) $t = 4.05\,s$, and (f) $t = 4.30\,s$.

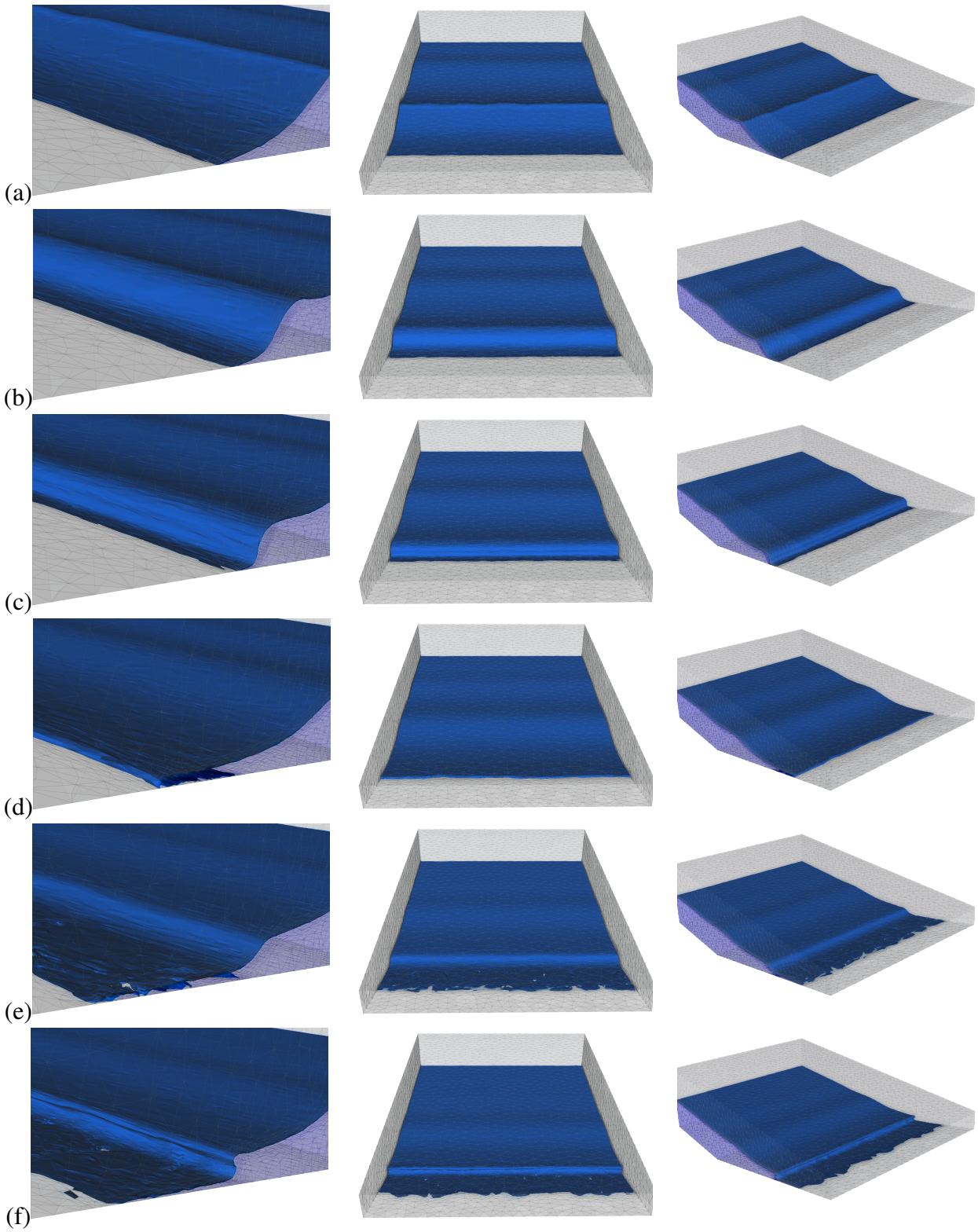


Figure 5.8: 3D version of the gravity generated solitary wave breaking on a slope. Dimensions of the tank in this case were : height=0.2 m, width 0.5 m, length=0.75 m. The slope used here was of 0.3. Still frames were taken at (a) $t = 0.375$, (b) $t = 0.435$, (c) $t = 0.459$, (d) $t = 0.555$, (e) $t = 0.615$, and (f) $t = 0.663$ seconds.

5.3.2 Three dimensional wave impact on breakwater amour units

This simulation involves an applied usage of the solid fluid coupling technique and the modelling of interface evolution between two immiscible fluids. The problem consists of a tank of dimensions $1.0 \times 0.3 \times 0.34$ m with an inclined slope of 1:2.8 on one end filled with water up to 0.12 m depth. A water column is placed on top of this water level against the side opposite to the slope. The instantaneous release of this water column causes a wave to be generated. Two armour-layer-type solid units[4] are placed on the slope to represent the existence of an armoured breakwater. These units are then impacted by the wave, generating turbulence in the flow. Free slip boundary conditions were used on all walls in this simulation, with exception of no-slip which is naturally applied on the units through the method developed for resolved particles in Chapter 4. At this developmental stage the units are unable to react to the fluid forces and hence the coupling is one sided, given that the armour units do influence the flow although they are not disturbed by it. Nevertheless, result frames presented in Figure 5.9 are able to depict the behaviour of the free surface of the incoming wave when confronted with the armour units. Additionally, Figures 5.10, 5.11, 5.12, and 5.13 are able to show the flow details related to the presence of the armour units as an obstruction to the wave. This level of detail was obtained through the use of adaptive mesh optimisation, and it is one of the greatest advantages of this approach with respect to other popular methods such as the RANS approaches (see Section 5.2). Furthermore, Figures 5.14, 5.15, 5.16, and 5.17 show a sequence of frames depicting associated flow vorticity, which is a key parameter in analysing turbulence effects.

Figure 5.18 and Figure 5.19 show a plot of the horizontal and vertical forces, respectively. The evolution of the forces observed in the plots evidence an initial, secondary, and less powerful third impact on the modelled armour units. Force values obtained do not appear unrealistic, however future experimental tests and validation of the impact forces need to be carried out before this is proved to be true. Small variations, or "spikes", are also observed in the graphs. As was seen in the 3D water collapse with obstruction simulation results of Chapter 2, these spikes are the results of spurious errors introduced by the adaptive process.

It is useful to mention the importance and significance of the results presented in this section. Like in the 2D case presented in section 5.3.1, details of a very realistically shaped wave breaker can be observed before it impacts the armour units. It should be noted that much resolution has been spent in resolving the air flow patterns, however it is a matter of future development to implement a method to concentrate more resolution on water flow patterns. Levels of detail achieved in these simulations, using a variety of element scales to improve efficiency, has not been observed elsewhere in the literature. The velocity and vorticity magnitude plots presented show that not only flow patterns can be described with accuracy, but also the associated energy distribution and dissipation. Additionally, and although it has not been performed for these results, the pressure distributions can be measured by inserting sensors in the manner that was

[4]The shape of these units was based on the Core-loc (TM) armour unit shape.

achieved in Chapter 2. The latter two features are just samples of the strong potential in the developments of this thesis for coastal engineering applications.

5.4 Conclusions

A numerical model that describes wave phenomena in two and three dimensions has been developed and described in this thesis. Two-fluid (air and water) interface modelling on 3D unstructured finite element meshes has been achieved using a compressive advection scheme to ensure interface sharpness to the limit of the local resolution. This is of particular interest when modelling breaking waves, as can be seen in Figures 5.7, 5.8, and 5.9. Qualitative results were obtained for both 2D and 3D simulations. Breaker type variation with slope angle was observed in the 2D simulations of section 5.3.1, although many more tests remain to be performed with the future implementation of a proper wave-making technique. Advantages of this method with respect to others in the literature lie in its relatively simpler implementation in three dimensions, and the usage of unstructured meshes, which are the key for a more flexible description of domain geometries. Results portrayed in this chapter further encourage the testing and enhancement (e.g. development of a wave maker technique) of the produced methodology from each chapter. Although still in a developmental stage, it is well on its way to be proven as a potential powerful modelling tool for coastal engineers.

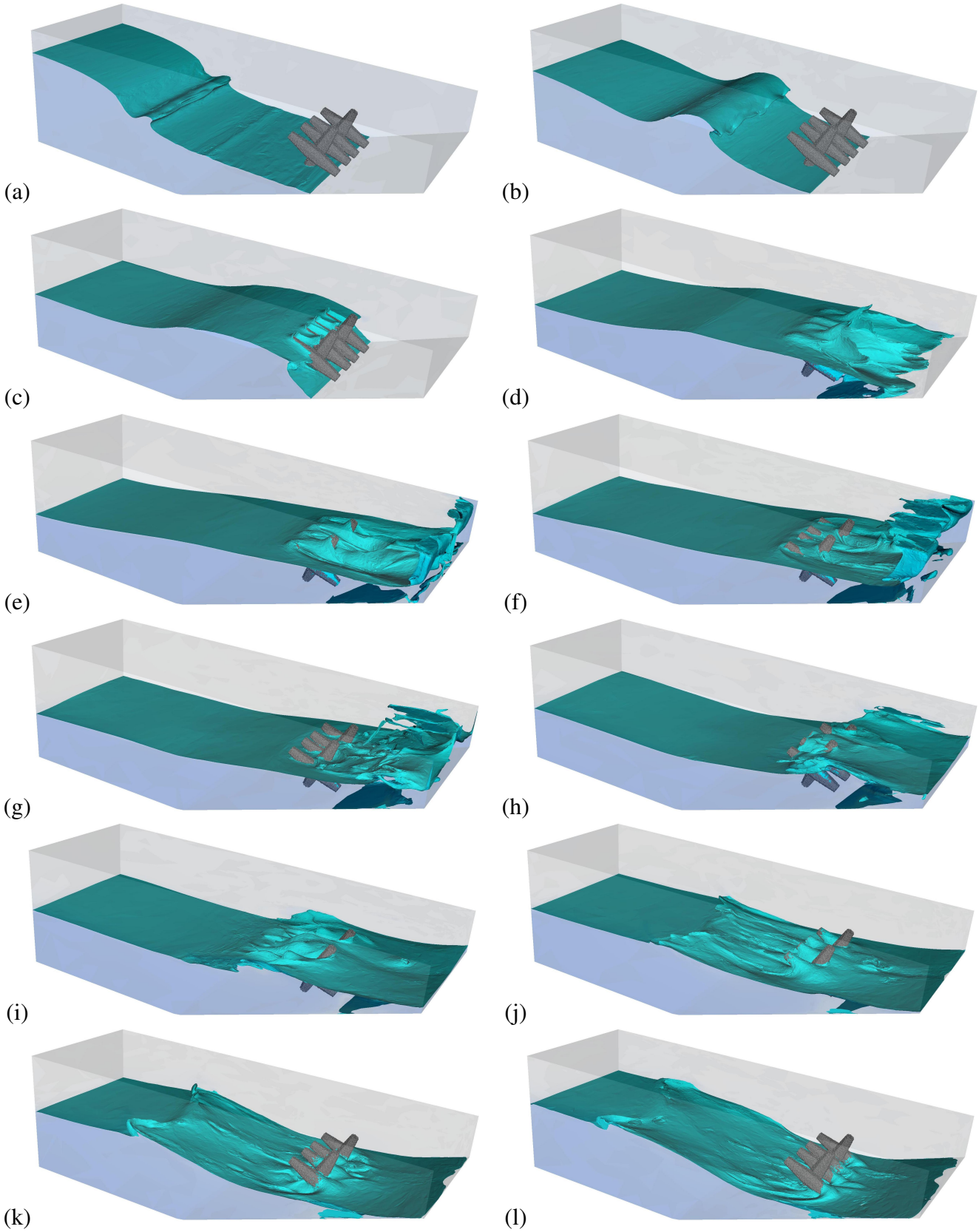


Figure 5.9: Frames depicting free surface evolution and one way interaction with the solid armour units. Still frames were taken at (a) $t = 0.1$, (b) $t = 0.2$, (c) $t = 0.3$, (d) $t = 0.4$, (e) $t = 0.5$, (f) $t = 0.6$, (g) $t = 0.7$, (h) $t = 0.8$, (i) $t = 0.9$, (j) $t = 1.0$, (k) $t = 1.1$, and (l) $t = 1.2$.

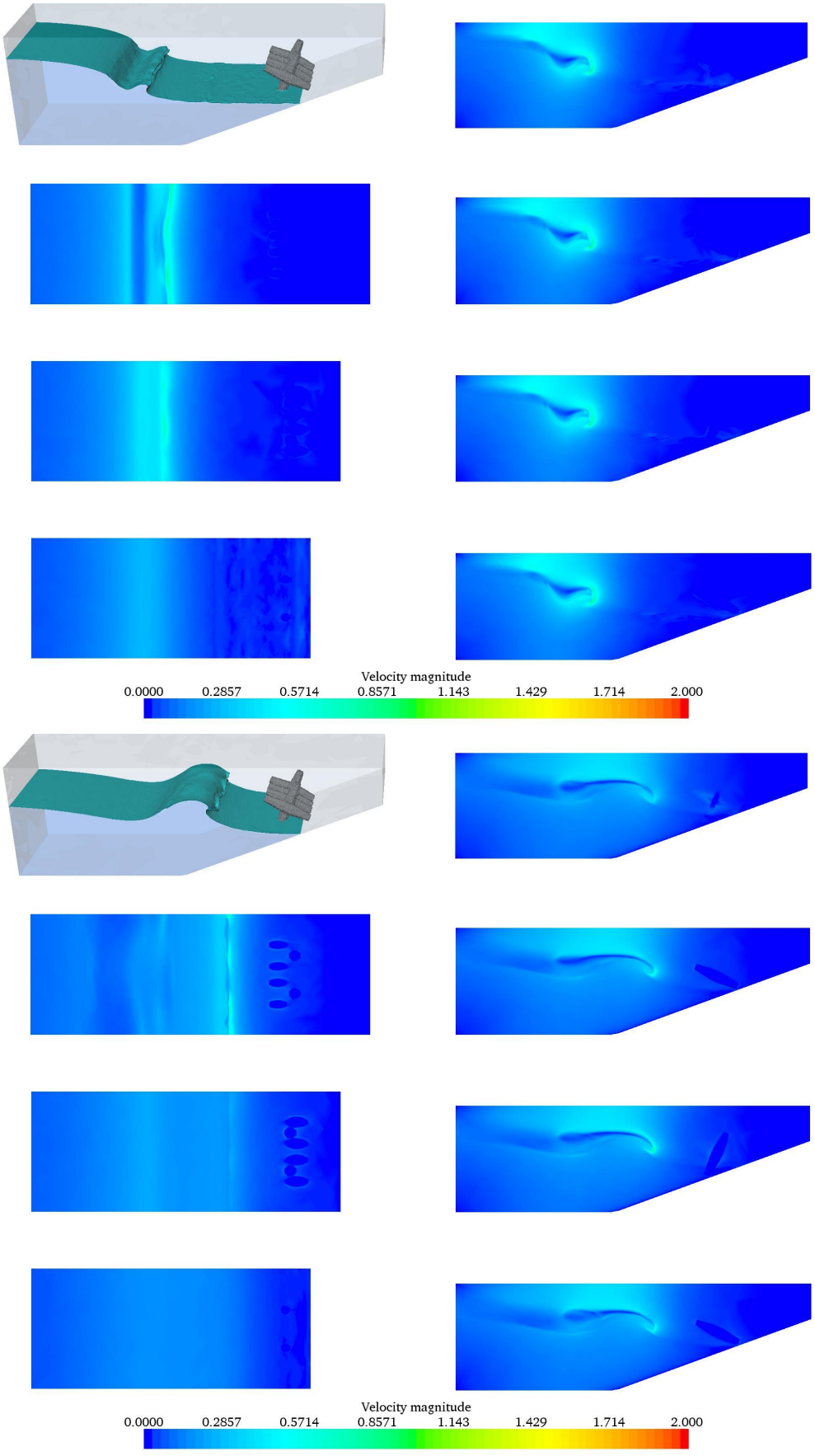
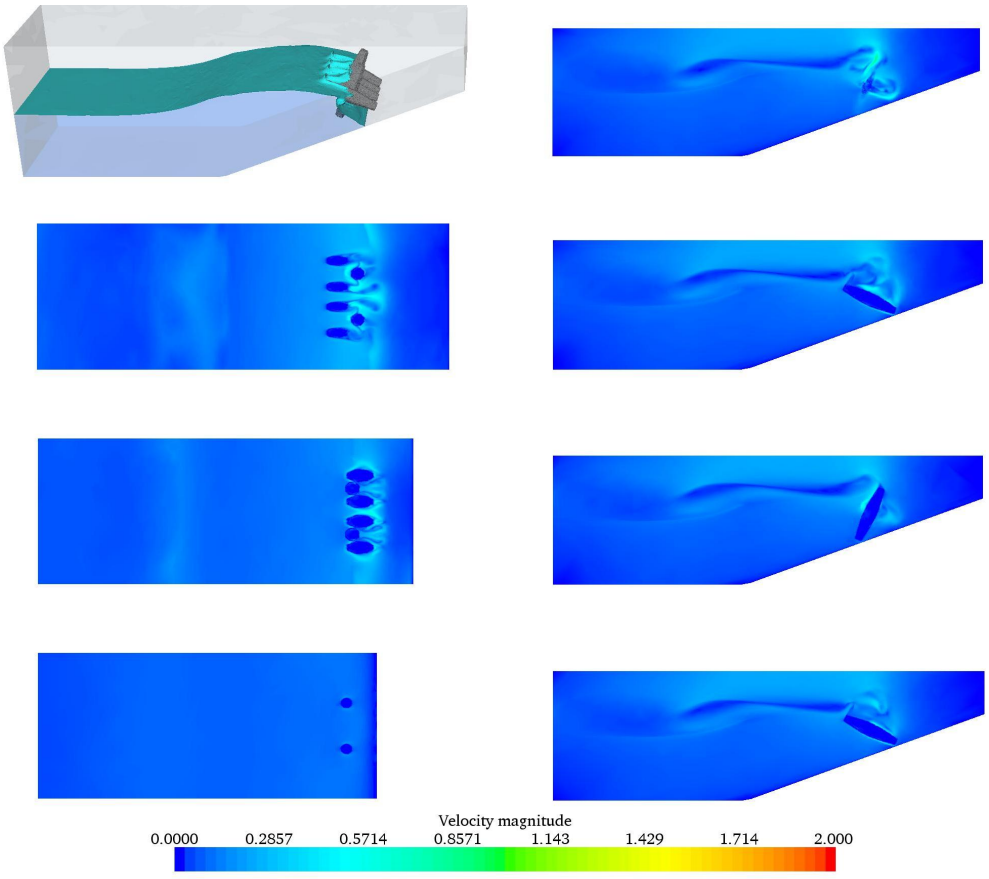


Figure 5.10: Result frames depicting a group of scalar cut planes across different sections of the fluid domain. The three left hand side ones correspond to horizontal cuts, while the four right hand side ones are vertical cuts. Time frames depicted here are $t = 0.1$ (top), and $t = 0.2$ (bottom). Plots are of velocity magnitude.



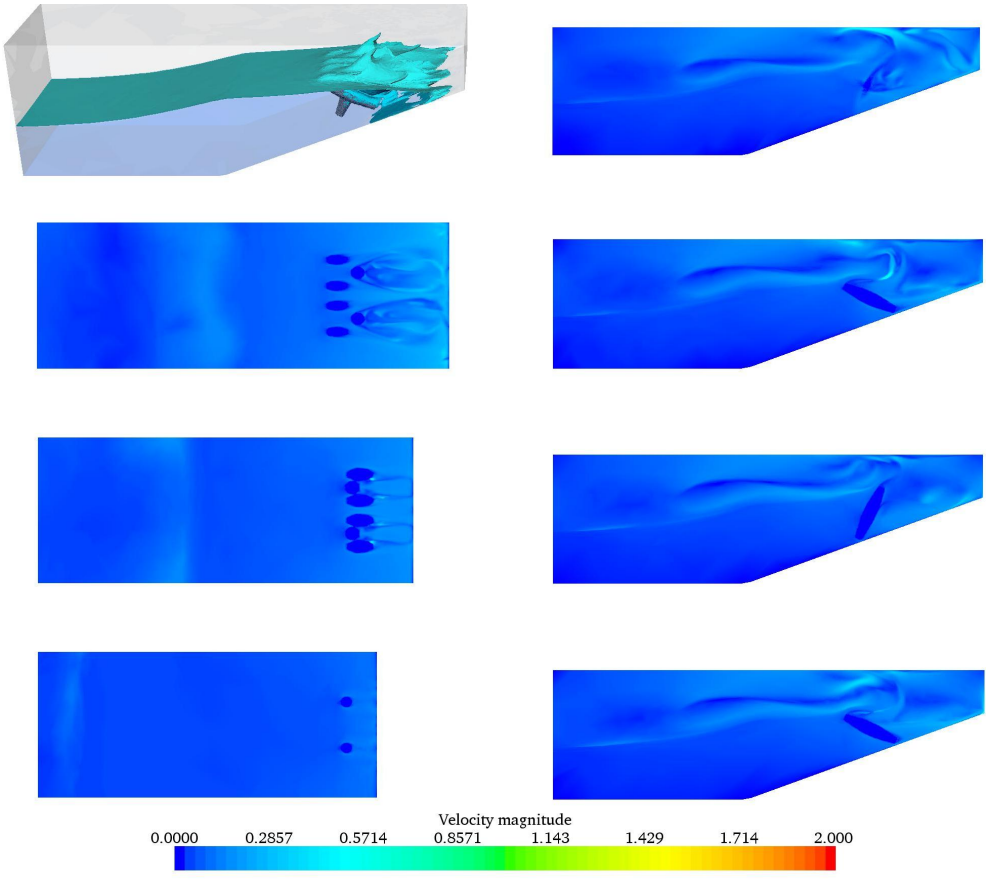
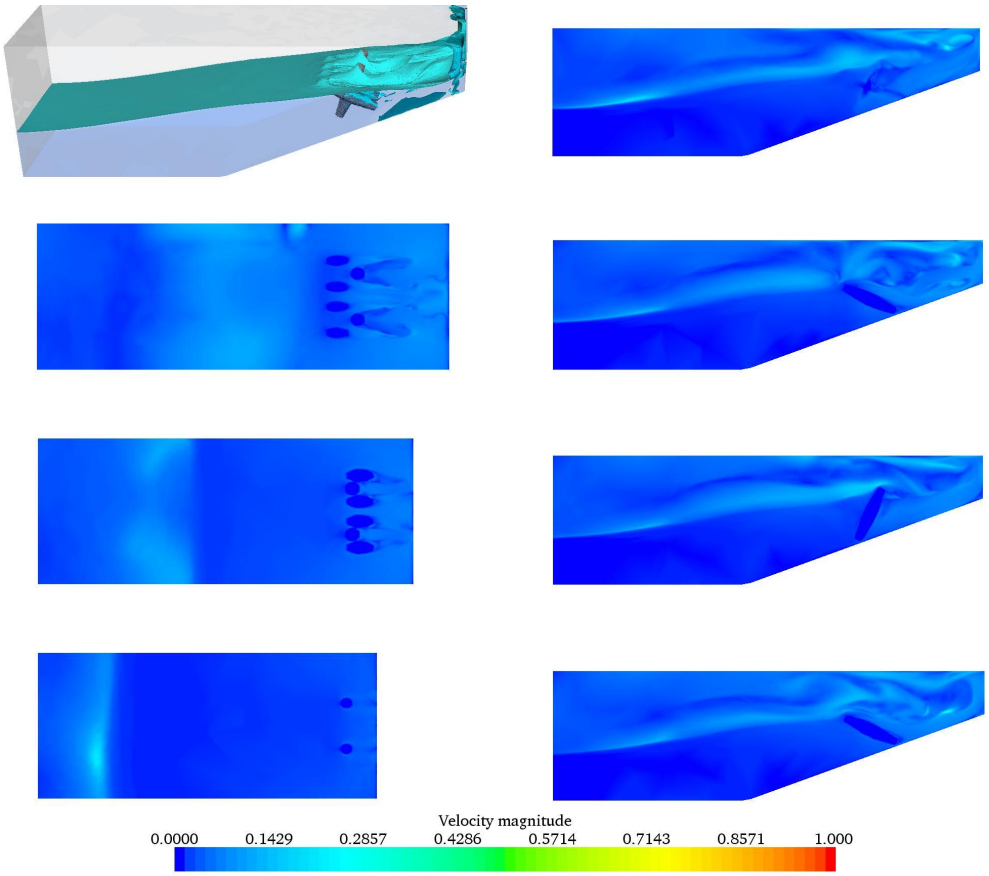


Figure 5.11: Continued from Figure 5.10. Time frames depicted here are $t = 0.3$ (top), and $t = 0.4$ (bottom).



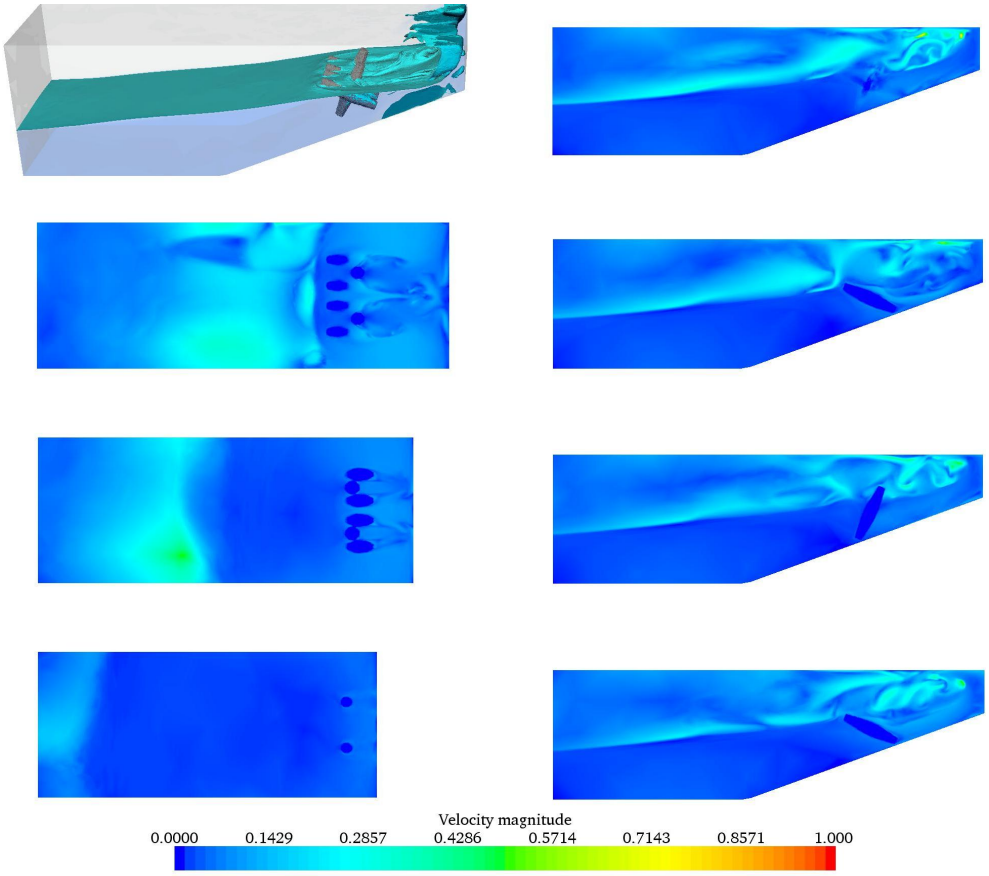


Figure 5.12: Continued from Figure 5.11. Time frames depicted here are $t = 0.5$ (top), and $t = 0.6$ (bottom).

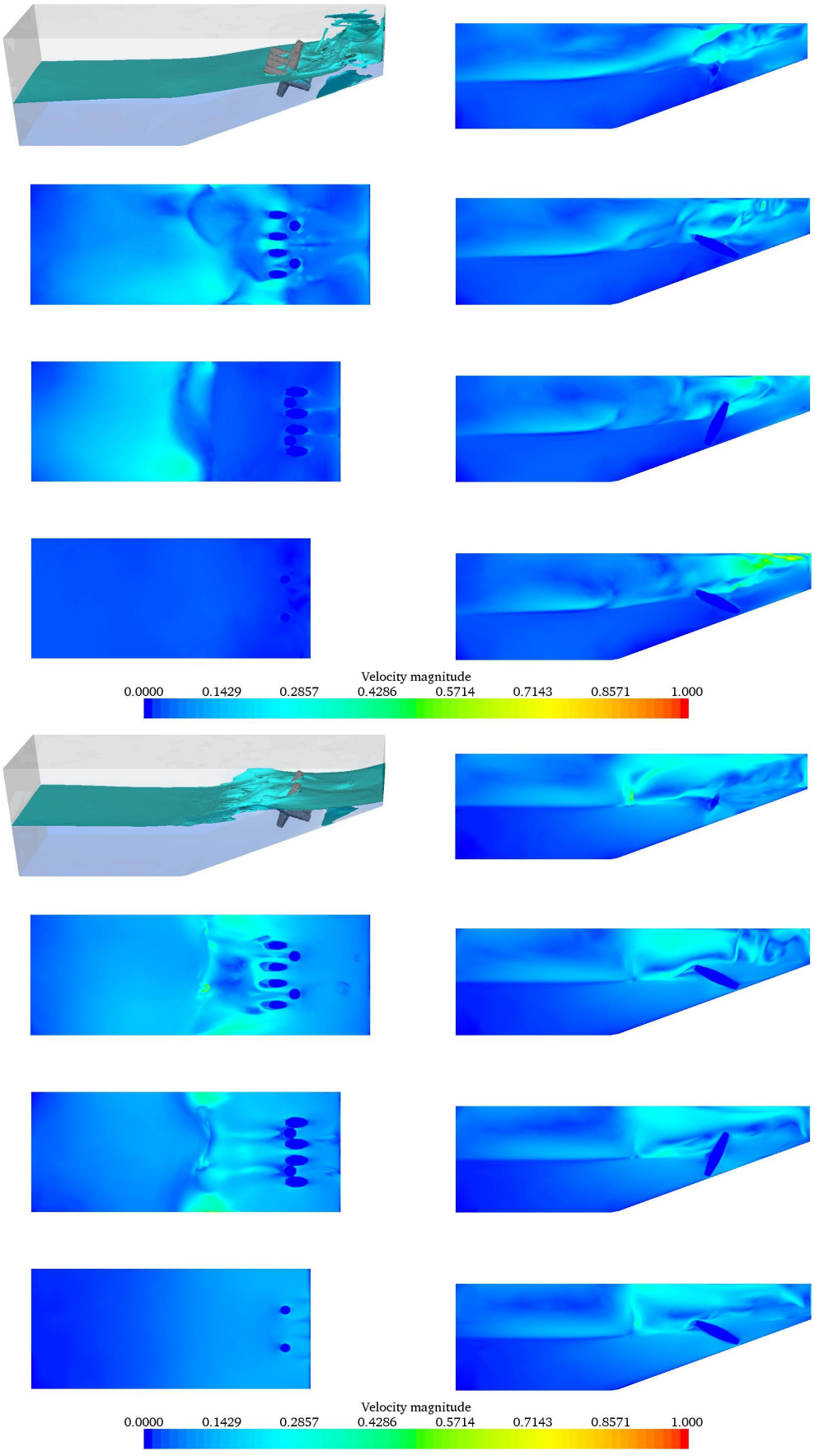


Figure 5.13: Continued from Figure 5.12. Time frames depicted here are $t = 0.7$ (top), and $t = 0.8$ (bottom).

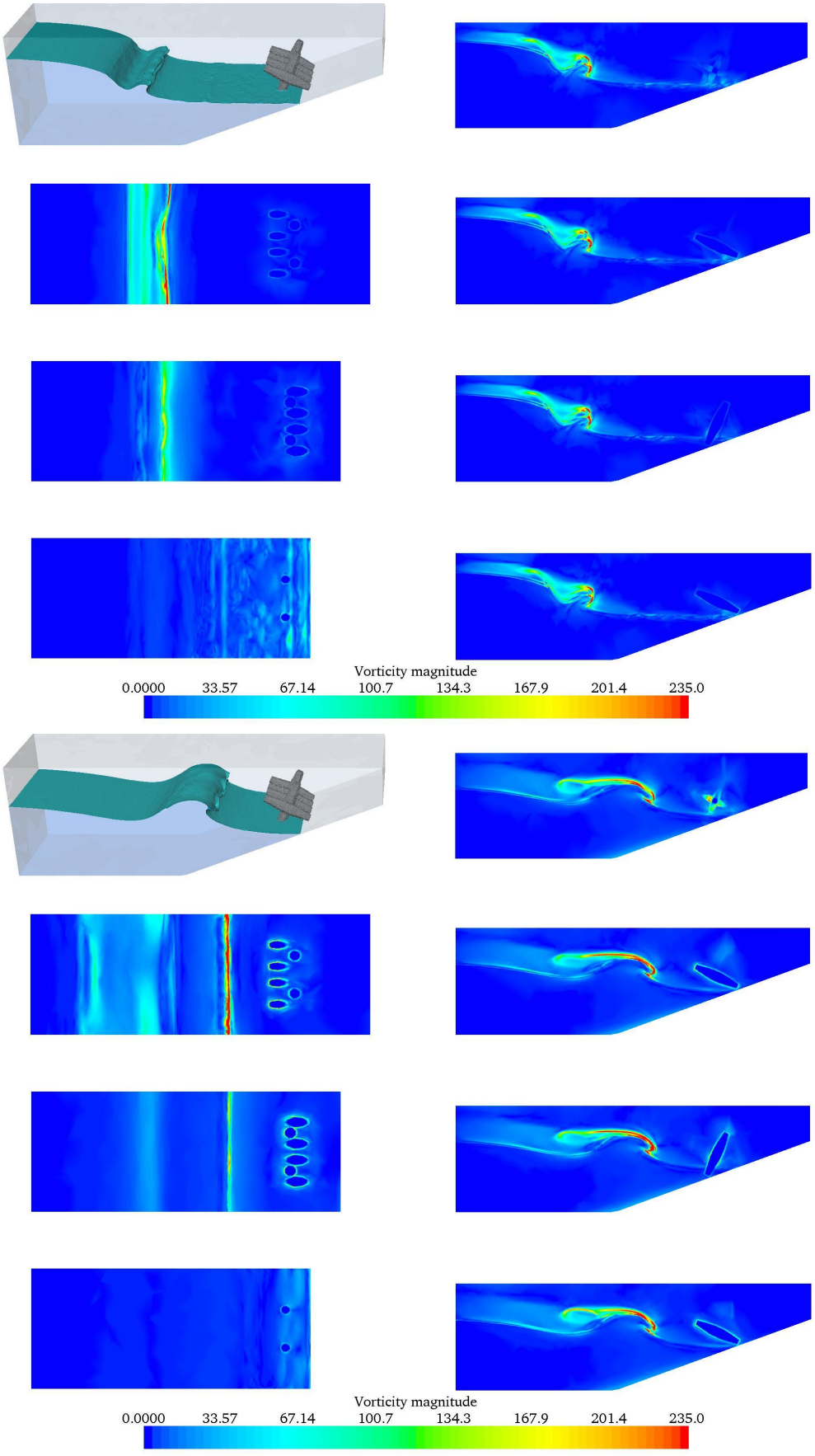
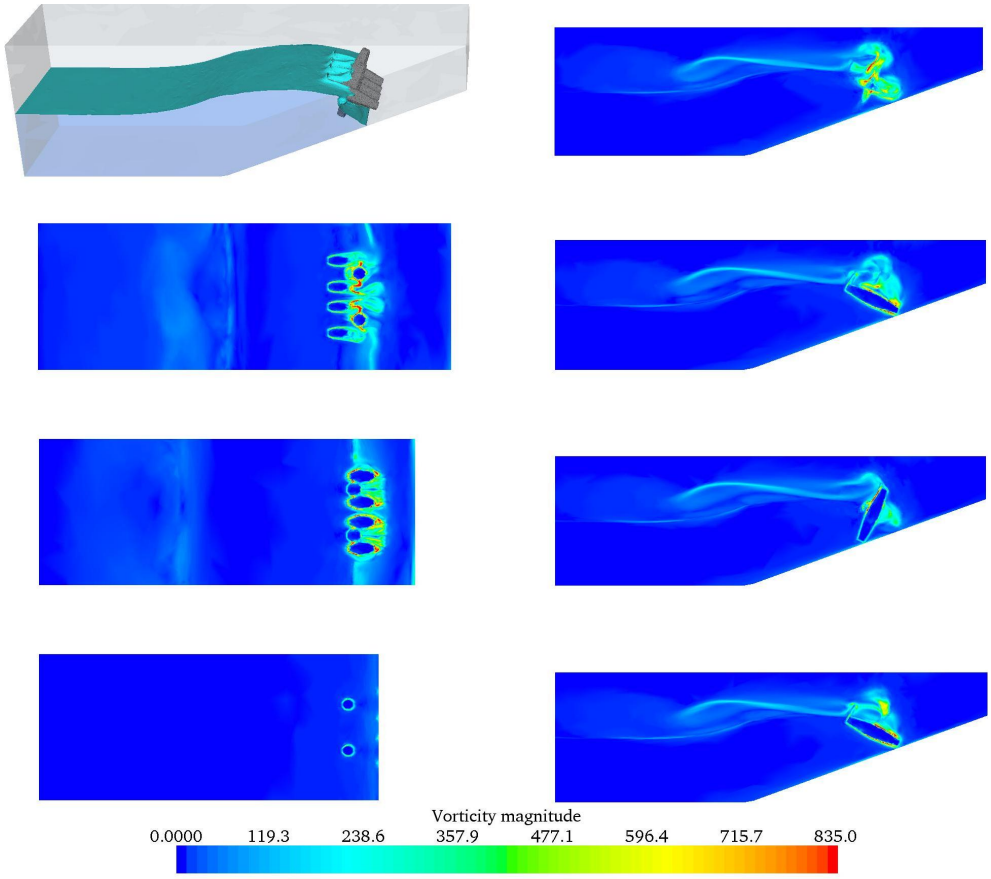


Figure 5.14: Result frames depicting a group of scalar cut planes across different sections of the fluid domain, analogous to Figure 5.10. The three left hand side ones correspond to horizontal cuts, while the four right hand side ones are vertical cuts. Time frames depicted here are $t = 0.1$ (top), and $t = 0.2$ (bottom). Plots are of vorticity magnitude.



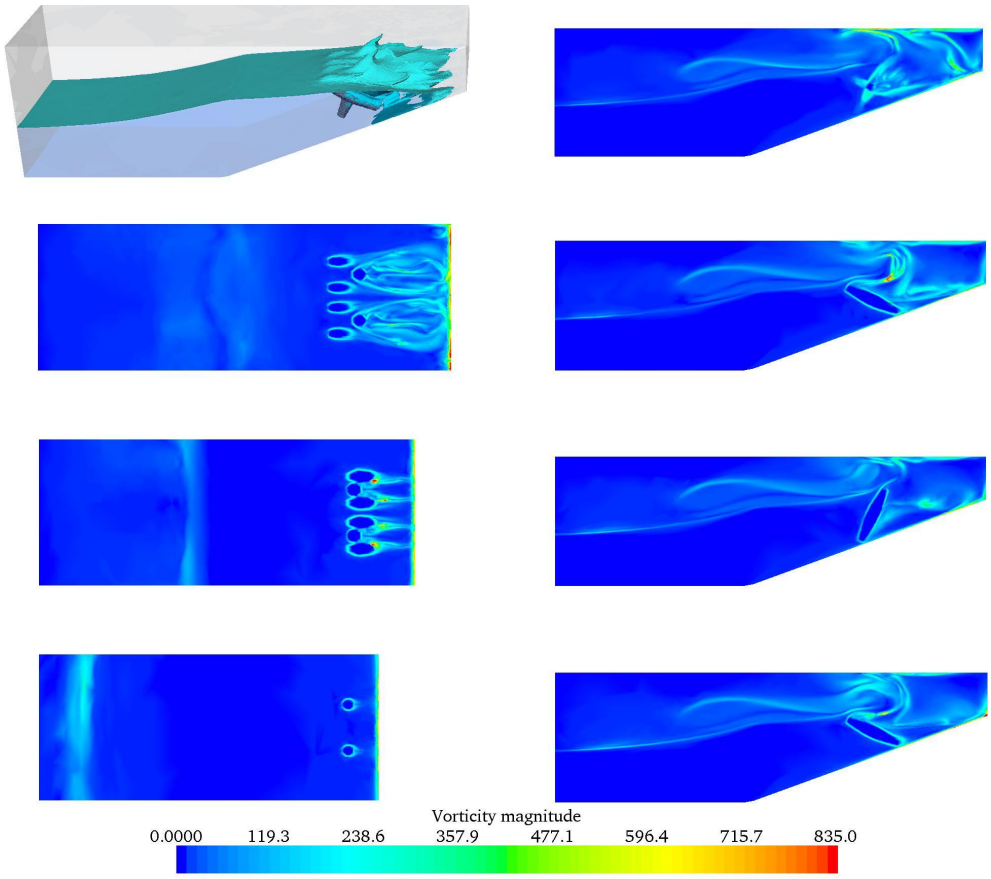
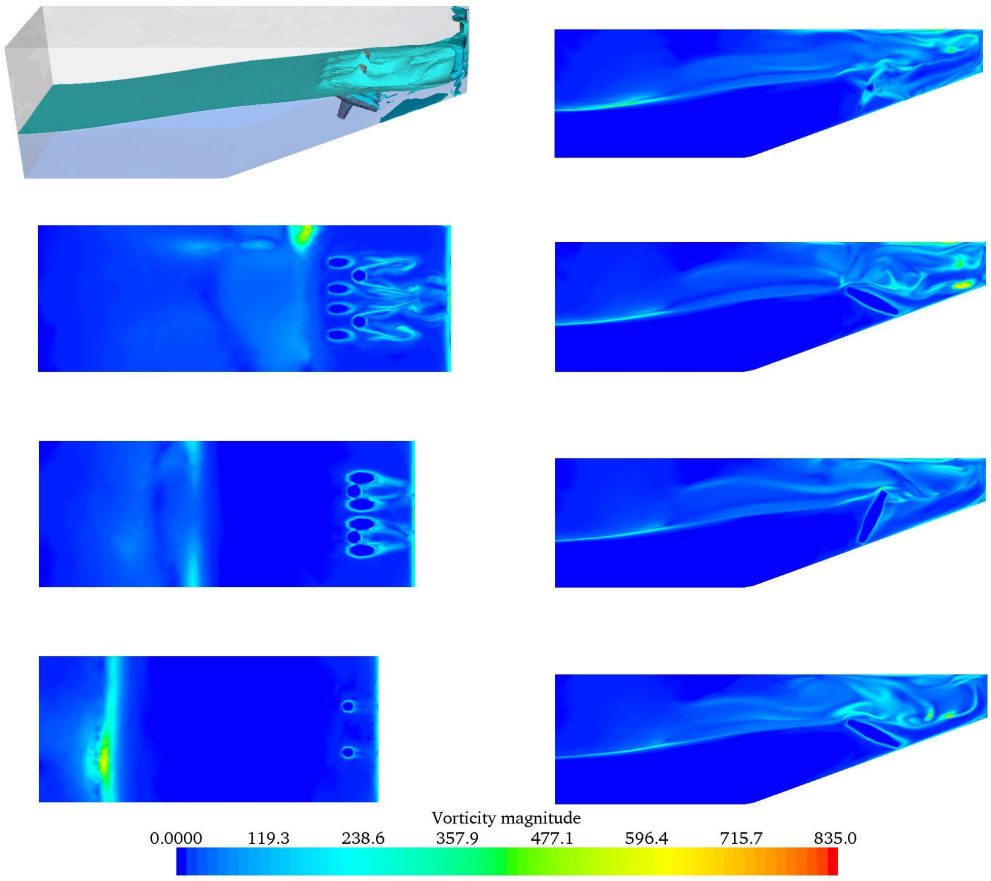


Figure 5.15: Continued from Figure 5.14. Time frames depicted here are $t = 0.3$ (top), and $t = 0.4$ (bottom).



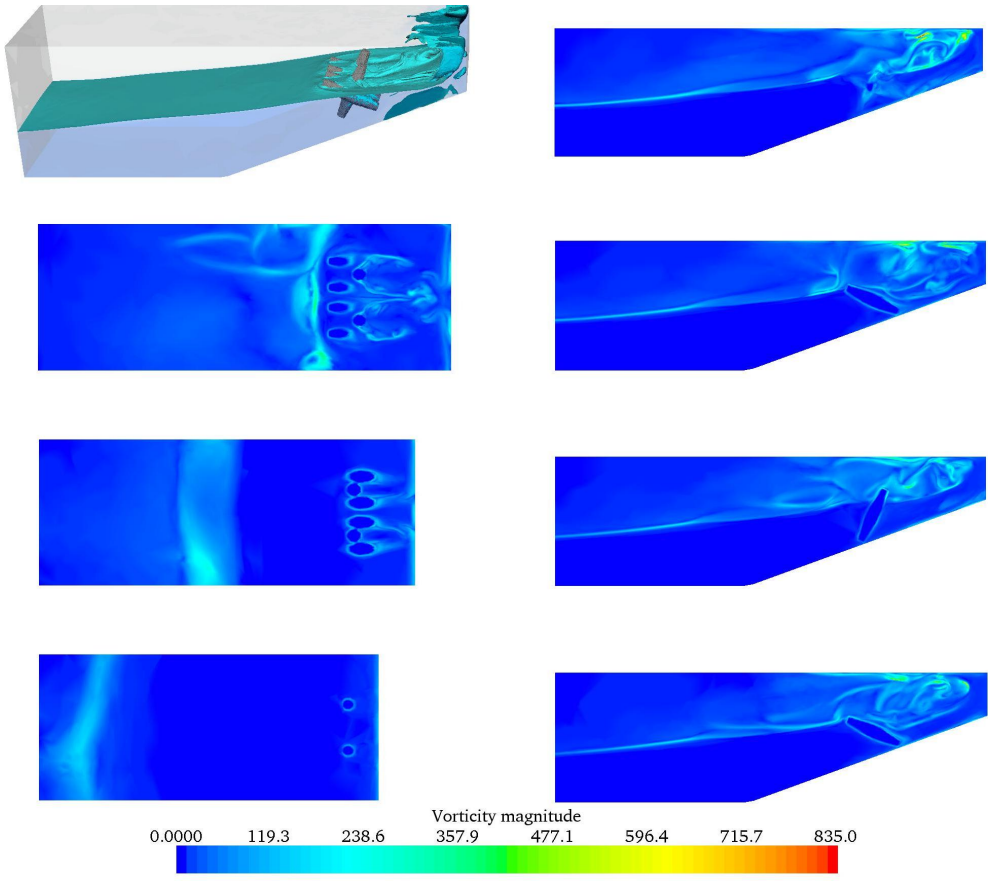


Figure 5.16: Continued from Figure 5.15. Time frames depicted here are $t = 0.5$ (top), and $t = 0.6$ (bottom).

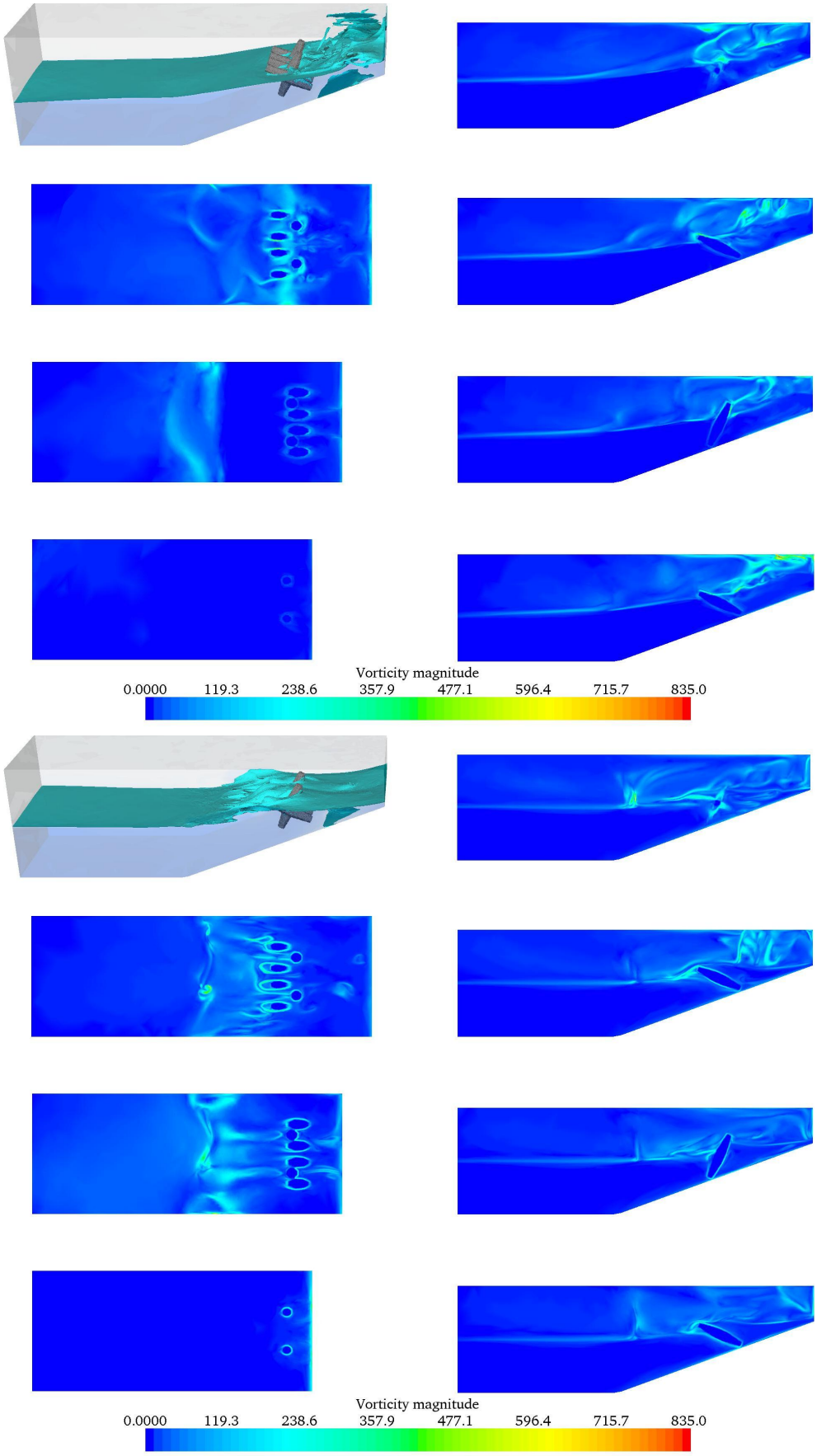


Figure 5.17: Continued from Figure 5.16. Time frames depicted here are $t = 0.7$ (top), and $t = 0.8$ (bottom).

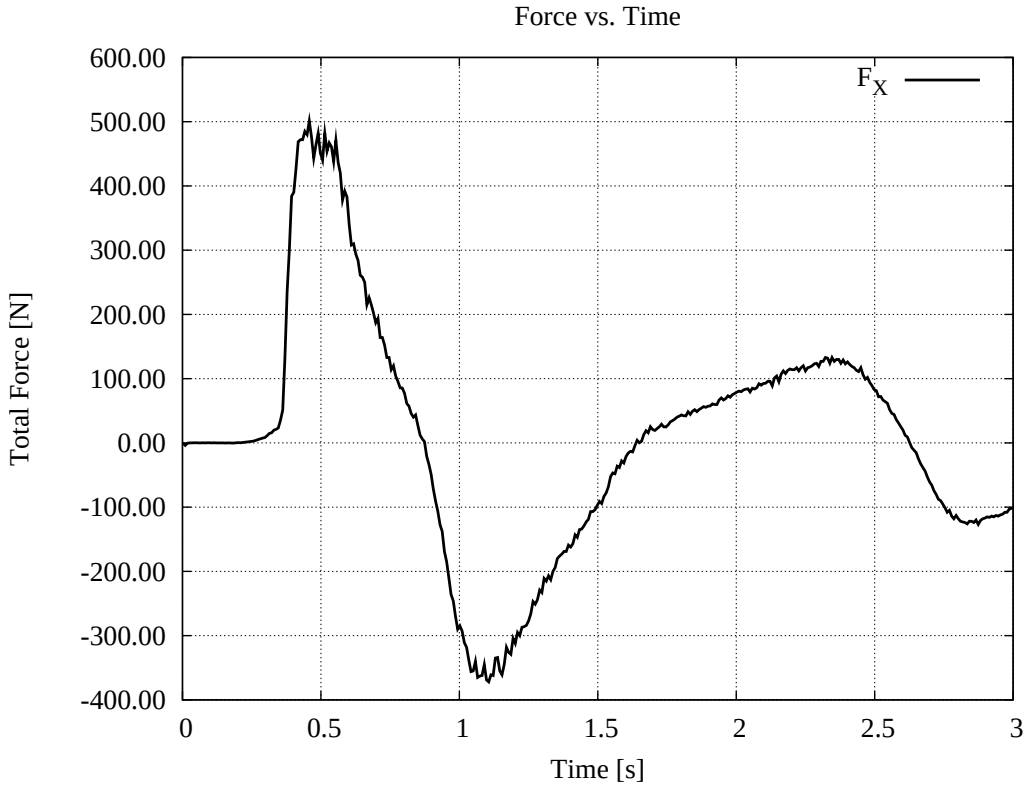


Figure 5.18: Graph of the total horizontal force acting on both units due to the wave impact.

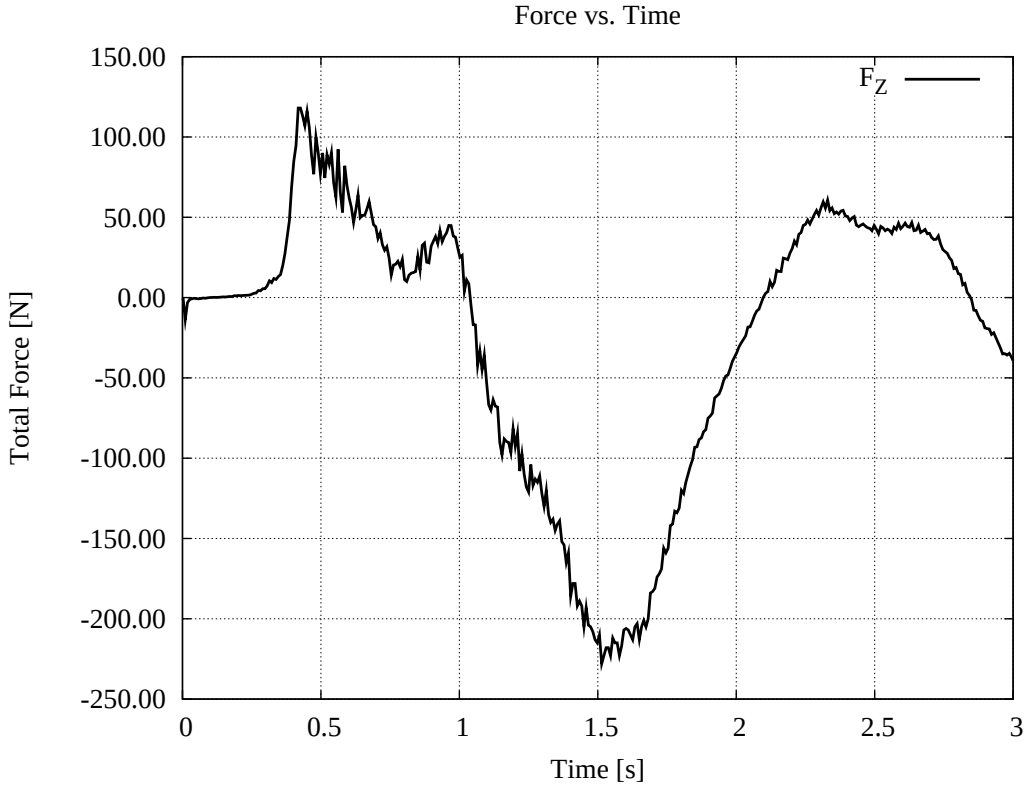


Figure 5.19: Graph of the total vertical force acting on both units due to the wave impact.

Chapter 6

Conclusion

6.1 Concluding remarks

The work presented herein was highly motivated by coastal engineering applications. The original aim was to develop leading software technology for numerical modelling of complex solid-fluid interaction processes. According to some of the world's most renowned designers of coastal structures, engineers have struggled to make significant progress in recent decades. This is thought to be because physical modelling and prototype observations have been the only methodologies available to investigate the complex coupled processes arising when waves break on rubble mound coastal structures. There is now, however, a general consensus that numerical tools will be required to provide the level of understanding that will bring significant improvements in design. Coupled numerical models that handle the key hydraulic, structural and geotechnical interactions are required to reduce reliance on physical tests and provide a more detailed understanding of hydraulic loads, unit stresses and underlying coastal processes. This is an active research area with groups in Netherlands, Norway, Spain, USA, Denmark, France, Japan, and UK. However, many provide only 2D simulations whereas 3D simulations are required to capture the full physics of these processes. As discussed in the introduction, the motions, forces, and stresses induced in armour units and under design storm conditions are extremely difficult to deduce with confidence.

The numerical modelling of a system containing not only two immiscible fluids with a large density and viscosity contrast, but also the interaction these fluids have with randomly shaped and sized structures, poses a complex problem to researchers worldwide. A good strategy to focus resolution and maximise computational power is also vital.

The set of tools developed in this thesis goes a significant way towards solving this conundrum, and consists of three main contributions or components.

- The first component is the implementation of a compressive advection algorithm into the Fluidity framework. The objective was to produce a conservative and efficient scheme which can be easily used on unstructured three dimensional meshes. Achieving the latter

characteristic is either an impossible task, or highly complex for many interface capturing methods such as the volume of fluid methods. In Chapter 2 a series of validation cases were shown in which the method developed here proved to be extremely successful. This first component in the overall modelling approach has proved to be efficient, conservative, and applicable to unstructured 3D meshes.

- The second component is the addition of a moving reference frame formulation for the nodes of the computational grid. Based on other arbitrary Lagrangian-Eulerian schemes in the literature, this method aims to further enhance the compressive behaviour of the advection scheme (i.e. first component) by improving local resolution. This process is also achieved in a manner that reduces numerical diffusion, while preserving the quality of the elements.

- Lastly, a solid-fluid interaction modelling technique was developed which is based on the geometrical mapping of solid shapes. An important consideration in the coupling strategy was to allow for the overlapping of solid bodies which is an essential feature of the discrete element formulations (DEM and FEMDEM) used to model the solid-solid force interactions. A multiphase approach was used in conjunction with the geometrical mapping to allow the presence of solid structures within the fluid phase. Coupling the solids and the fluids is achieved through inter- phase force calculation, which allows for a modular construction and interfacing of originally independent solid modelling and fluid modelling software. Steady and unsteady test cases were provided in Chapter 4. These showed the excellent progress that has been achieved during this PhD project.

The above mentioned components are finally brought together, as seen in Chapter 5, to show the potential for applications in the coastal engineering field. Excellent qualities are shown by the method when reproducing wave breaking phenomena, as well as modelling the details of turbulent flows in the interstitial spaces between armour units. This is further proof of the ability that is now available to study local energy absorption in time and space (i.e. an energy absorption rate contoured simulation), which will show how well distributed the energy is for different structure designs. Sites of instability and armour movement are expected to be correlated with such energy distributions. Sites of likely armour unit breakage can then be modelled directly with the complementary methods embodied in the solids modelling part of the coupled model (i.e. the FEMDEM part).

The following section provides summary of work in progress, or lines of future work.

6.2 Future work

It is evident from the results and conclusions presented on each individual Chapter of this thesis, that a number of aspects of each one of the tools presented are open to further research and

development. This section attempts to provide some brief detail on these aspects in the hope that they will covered by future development.

- In the case of the material presented in Chapter 2, work remains to be done towards fully understanding the compressive behaviour of the advection scheme in unstructured meshes. Judging from results obtained mainly for the Zalesak problem (e.g. see Section 2.5.1) which, although not presenting diffusion shows some spurious deformation of the interface, it is clear that work remains to be done on understanding the effects of key components in the limiting process. One of these is the calculation of the upwind value (e.g. see section 2.3), which was presented here as being obtained through extrapolation from a point in a element which neighbours the central node (i.e. in the direction opposite to the downwind node). Another important factor which will likely have effects on the quality of the interface is the normalised face value calculation (i.e. obtained from NVD diagrams; see section 2.3), since varying the slope of the curve which defines the Hyper-C method (e.g. see Figure 2.10) will also affect the limiting process. Details of the extent of these effects and further developments concerning the compressive advection using unstructured meshes are to be included in forthcoming publications (e.g. Wilson et al. (2008)).

- Concerning moving meshes, the main future development is the optimisation (i.e. speed up) of the associated code, which would involve alterations of the algorithms presented in Chapter 3 (e.g. see Algorithms 2 and 3) . Also, an in-depth study on the effects of individual weight factors for each functional should provide insight on an optimal, and quite possibly adaptive, method which would allow the automatic calculation of more parameters and less usage of the "trial and error" techniques.

- Furthermore, an important future development to further exploit the moving mesh method is parallel implementation. Recently, multi- core processor computers have become increasingly popular in the markets, and hence the coding tendency nowadays in numerical modelling lies in parallel processing of tasks. Implementing parallelism in the method presented in Chapter 3 is far from trivial, with specific emphasis on the fact that each node calculates its functional value based on local neighbour element shapes (e.g. see section 3.3.2), and communicating nodal coordinate changes across a group of processors can prove relatively inefficient.

- With respect to the solid modelling, a primary concern is the optimisation of the solid mapping technique (i.e. Fastest Neighbour to Neighbour search). A bin sort method is currently used for determining the starting point (i.e. element) of this search technique. However this can prove relatively memory inefficient and problem dependent, as the individual bin sizes are related to the size of the mapped domain and size of the elements on

the mesh. Usage of mesh adaptivity can produce large elements combined with significantly smaller elements, thus producing problems when determining the starting element [1]. The optimal development which is foreseen for this problem is the implementation of an improved spatial indexing method such as the R-tree or Oc-tree methods (e.g. see, for instance Arge et al. (2004)). These techniques should bring about a significant overall algorithm speed up.

- Another concern within solid modelling is the full coupling (i.e. two-way) between FEMDEM models and Fluidity. In Chapter 4 fully coupled simulations were shown for DEM models, however FEMDEM simulations were coupled only one-way. In contrast with modelling rigid spheres, where the resulting inter- phase force is applied at the centre of mass, the calculation of inter- phase forces between the fluids and FEMDEM structures requires the mapping back of volumetric forces rather than calculating a resulting force (e.g. see section 4.4). This procedure is currently work in progress, as it also depends on the previously mentioned search technique optimisation.

The quality of coastal engineering applications can be improved through the development of a wave maker model, which is a key necessary component for generating waves. A possible hybrid of the CFD approach developed in this thesis with a simpler model such as a Boussinesq model, to provide boundary conditions compliant to offshore wave behaviour would greatly improve the application possibilities. Alternatively, schemes for the setting of error metrics in the adaptive meshing can be used to retain lower resolutions until higher resolution is required near and on the structure. Massively parallel methods will undoubtedly be necessary to run sequences of waves that come close to representing a random wave spectrum.

As a concluding remark, the overall approach shown in this thesis serves to show that greater realism can be achieved when simulating wave breaking phenomena and associated interaction with structures. The progress of computer technology, and the possibility of parallelisation suggests a number of problems previously considered impossible to solve numerically, can now be realistically contemplated in the near future. As computer simulation continues to cement its role as the "third leg of science" (i.e. joining theory and experiment) researchers rely more and more on the power offered by modern systems.

[1] In essence, and as explained in Chapter 5.1.1, the method has a fall back on brute- force type search. However the latter is unwanted as it can lead to significantly higher run- times, particularly for large problems.

References

Abbad, M. and Souhar, M. (2004). Effects of the history force ona an oscillating rigid sphere at low reynolds number, *Experiments in Fluids* **36**: 775–782.

Airoldi, L., Abbiati, M., Beck, M., Hawkins, S., Jonsson, P., Martin, D., Moschella, P., Sundelöf, A., Thompson, R. and Åberg, P. (2005). An ecological perspective on the deployment and design of low-crested and other hard coastal defence structures, *Coastal Engineering* **52**(10-11): 1073–1087.

Airy, G. (1845). Tides and waves:, *Encyc. Metrop.* **192**: 241–396.

Ait-Ali-Yahia, D., Baruzzi, G., Habashi, W. G., Fortin, M., Dompierre, J. and Vallet, M.-G. (2002). Anisotropic mesh adaptation: towards user-independent, mesh-independent and solver-independent cfd. part ii. structured grids, *International Journal for Numerical Methods in Fluids* **39**(8): 657–673.

Allsop, N. W. H. (2005). *Coastlines, Structures and Breakwaters*, Institution of Civil Engineers.

Anderson, A., Zheng, X. and Cristini, V. (2005). Adaptive unstructured volume remeshing - i: The method, *Journal of Computational Physics* **208**(2): 616–625.

Andrillon, Y. and Alessandrini, B. (2004). A 2d+t vof fully coupled formulation for the calculation of breaking free-surface flow, *Journal of Marine Science and Technology* **8**: 159–168.

Arami, A. and Hattori, M. (1989). Experimental study on shock wave pressures, *Technical report*, Dept. Civil Eng., Faculty of Science, Chuo University, Japan.

Arge, L., de Berg, M., Haverkort, H. J. and Yi, K. (2004). The priority r-tree: A practically efficient and worst-case optimal r-tree, *SIGMOD 2004 June 13-18, Paris, France*.

Arienti, M., Hung, P., Morano, E. and Shepherd, J. (2003). A level set approach to eulerian-lagrangian coupling, *Journal Computational Physics* **185**: 213–251.

Armero, F. and Love, E. (2003). An arbitrary lagrangian-eulerian finite element method for finite strain plasticity, *International Journal for Numerical Methods in Engineering* **57**(4): 471–508.

Arnold, G. S., Drew, D. A. and Lahey, R. T. (1989). Derivation of constitutive equations for interfacial force and reynolds stress for a suspension of spheres using ensemble cell averaging, *Chemical Engineering Communications* **86**(1): 43–54.

Astarita, T., Ocone, R. and Astarita, G. (1997). The rayleigh approach to the rheology of compressible granular flow, *Journal of Rheology*.

Aymone, J. L. F. (2004). Mesh motion techniques for the ale formulation in 3d large deformation problems, *International Journal for Numerical Methods in Engineering* **59**(14): 1879–1908.

Bailey, A. B. and Hiatt, J. (1971). Sphere drag coefficients for a broad range of mach and reynolds numbers, *AIAA Journal* **10**: 1436–1440.

Baines, M. J. (2002). Moving meshes, conservation laws and least squares equidistribution, *International Journal for Numerical Methods in Fluids* **40**(1-2): 3–19.

Baliga, B. R. and Patankar, S. (1980). A new finite-element formulation for convection-diffusion problems, *Numerical Heat Transfer* **3**: 393–409.

Bank, R., Sherman, A. and Weiser, A. (1983). Some refinement algorithms and data structures for regular local mesh refinement, *Scientific Computing* pp. 3–17.

Baquerizo, A. and Losada, M. A. (2008). Human interaction with large scale coastal morphological evolution. an assessment of the uncertainty, *Coastal Engineering* **55**(7-8): 569–580.

Barros, F. B., Proença, S. P. B. and de Barcellos, C. S. (2004). On error estimator and p-adaptivity in the generalized finite element method, *International Journal for Numerical Methods in Engineering* **60**(14): 2373–2398.

Batchelor, G. K. (2000). *An Introduction to Fluid Dynamics*, Cambridge University Press.

Bates, B. C., Kundzewicz, Z. W., Wu, S. and Palutikof, J. P. (2008). Climate change and water, *Technical Paper of the Intergovernmental Panel on Climate Change, IPCC Secretariat* p. 210 pp.

Battjes, J. (1974). Computation of set-up, longshore currents, run-up and overtopping due to wind-generated waves, *Technical Report 74-2*, Dept. Civ. Eng. Delft Tech Univ.

Benson, D. J. (1989). An efficient, accurate, simple ale method for nonlinear finite element programs, *Computer Methods in Applied Mechanics and Engineering* **72**(3): 305–350.

Benson, D. J. (1992). Computational methods in lagrangian and eulerian hydrocodes, *Computer Methods in Applied Mechanics and Engineering* **99**: 235–394.

Bisplinghoff, R. L., Ashley, H. and Halfman, R. L. (1996). *Aeroelasticity*, Dover Publications.

Blom, F. J. (1998). A monolithical fluid-structure interaction algorithm applied to the piston problem, *Computer Methods in Applied Mechanics and Engineering* **167**(3-4): 369–391.

Bodard, N., Bouffanais, R. and Deville, M. O. (2008). Solution of moving-boundary problems by the spectral element method, *Applied Numerical Mathematics* **58**(7): 968–984.

Borazjani, I., Ge, L. and Sotiropoulos, F. (2008). Curvilinear immersed boundary method for simulating fluid structure interaction with complex 3d rigid bodies, *Journal of Computational Physics* **227**(16): 7587–7620.

Brackbill, J. U., Kothe, D. B. and Zemach, C. (1992). A continuum method for modeling surface tension, *Journal of Computational Physics* **100**(2): 335–354.

Brennen, C. E. (2005). *Fundamentals of Multiphase Flow*, Cambridge University Press.

Brown, P. P. and Lawler, D. F. (2003). Sphere drag and settling velocity revisited, *Journal of Environmental Engineering*.

Bullock, G., Obhrai, C., Peregrine, D. and Bredmose, H. (2007). Violent breaking wave impacts. part 1: Results from large-scale regular wave tests on vertical and sloping walls, *Coastal Engineering* **54**(8): 602 – 617.

Carnell, R. E., Senior, C. A. and Mitchell, J. F. B. (1996). An assessment of measures of storminess: simulated changes in northern hemisphere winter due to increasing co2, *Climate Dynamics* **12**: 467–476.

Carrier, G. F. and Greenspan, H. P. (1958). Water waves of finite amplitude on a sloping beach, *Journal of Fluid Mechanics Digital Archive* **4**: 97–109.

Chakrabarti, S., Hernandez, S. and Brebbia, C. (eds) (2005). *Fluid Structure Interaction and Moving Boundary Problems*, WIT Press.

Chan, E. S. and Melville, W. K. (1988). Deep-water plunging wave pressures on a vertical plane wall, *Proceedings of the Royal Society of London. Series A, Mathematical and Physical Sciences*, Vol. 417, pp. 95–131.

Chang, H.-K. and Liou, J.-C. (2006). Fixed-frequency stokes wave expansion, *Ocean Engineering* **33**(3-4): 417–424.

Cheng, M. and Liu, G. R. (2002). A novel finite point method for flow simulation, *International Journal for Numerical Methods in Fluids* **39**(12): 1161–1178.

Chorin, A. J. (1980). Flame advection and propagation algorithms, *Journal of Computational Physics* **35**: 1–+.

Clarke, M. L. and Rendell, H. M. (2008). The impact of north atlantic storminess on western european coasts: A review, *Quaternary International* **In Press, Accepted Manuscript**: –.

Cleary, P. and Prakash, M. (2004). Discrete-element modeling and smoothed particle hydrodynamics: potential in the environmental sciences, *Philos Transact A Math Phys Eng Sci.* **362**: 2003–2030.

Clift, R., Grace, J. R. and Weber, M. E. (1978). *Bubbles, Drops, and Particles*, Academic Press.

Colella, P., Henderson, L. and Puckett, E. (1989). A numerical study of shock wave refraction at a gas interface, *Proceedings of the AIAA 9th Computational Fluid Dynamics Conference, Buffalo, New York*.

Cooker, M. J. and Peregrine, D. H. (1990). Water wave impact: Computations. theory and a comparison with measurements, *6th Intl. Workshop on Water Waves & Floating Bodies, Wood's Hole, Ma. USA*.

Corwin, E. and Logar, A. (2004). Sorting in linear time - variations on the bucket sort, *J. Comput. Small Coll.* **20**(1): 197–202.

Courant, R., Friedrichs, K. and Lewy, H. (1928). Ueber die partiellen differenzengleichungen der mathematischen physik, *Mathematische Annalen* **100**: 32–74.

Craig, R. and Bampton, M. (1968). Coupling of substructures for dynamic analysis, *AIAA Journal* **6**: 1313–1319.

Dalrymple, R. and Rogers, B. (2006). Numerical modeling of water waves with the sph method, *Coastal Engineering* **53**(2-3): 141–147.

Daniel, J. W. (1969). On the approximate minimization of functionals, *Mathematics of Computation* **23**(107): 573–581.

Dean, R. G. and Dalrymple, R. A. (1991). *Water Wave Mechanics for Engineers and Scientists*, World Scientific Publishing.

DeBar, R. (1974). A method in two-d eulerian hydrodynamics, *Technical Report UCID-19683*, Lawrence Livermore National Laboratory.

Demkowicz, L., Oden, J., Rachowicz, W. and Hardy, O. (1991). An h-p taylor-galerkin finite element method for compressible euler equations, *Computer Methods in Applied Mechanics and Engineering* **88**: 363–396.

Dennis, S. C. R. and Walker, J. D. A. (1971). Calculation of the steady flow past a sphere at low and moderate reynolds numbers, *J. Fluid Mech.* **48**: 771–789.

Depres, B. and Lagoutiere, F. (2001). Contact disconntinuity capturing schemes for linear advection and compressible gas dynamcs, *Journal of Scientific Computing* **16**(4): 479–524.

Devals, C., Heniche, M., Bertrand, F., Hayes, R. E. and Tanguy, P. A. (2005). A finite element strategy for the solution of interface tracking problems, *International Journal for Numerical Methods in Fluids* **49**(12): 1305–1327.

Devloo, P., Oden, J. T. and Pattani, P. (1988). The h-p adaptive finite element method for the numerical simulation of compressible flow, *Comput. Methods Appl. Mech. Eng.* **70**(2): 203–235.

Di Felice, R. (1994). The voidage function for fluid-particle interaction systems, *Int. Journal Multiphase Flow* **20**: 153–159.

Diez, P. and Calderon, G. (2007). Remeshing criteria and proper error representations for goal oriented h-adaptivity, *Computer Methods in Applied Mechanics and Engineering* **196**(4-6): 719–733.

Dohrmann, C. R., Key, S. W. and Heinstein, M. W. (2000). A method for connecting dissimilar finite element meshes in two dimensions, *International Journal for Numerical Methods in Engineering* **48**(5): 655–678.

Dold, J. W. and Peregrine, D. H. (1986). An efficient boundary-integral method for steep unsteady water waves, *in* K. W. Morton and M. J. Baines (eds), *Numerical Methods for Fluid Dynamics II*, Oxford University Press, pp. 671–679.

Donea, J., Giuliani, S. and Iialleux, J. (1982). An arbitrary lagrangian-eulerian finite element method for transient dynamic fluid-structure interactions, *Computer Methods in Applied Mechanics and Engineering* **33**: 689–723.

Donea, J., Huerta, A., Ponthot, J.-P. and Rodriguez-Ferran, A. (2004). *Arbitrary Lagrangian-Eulerian Methods*, Vol. 1, John Wiley & Sons, Ltd, chapter 14, pp. 1–25.

Dong, S. and Karniadakis, G. E. (2003). P-refinement and p-threads, *Computer Methods in Applied Mechanics and Engineering* **192**(19): 2191–2201.

Dowell, E., Crawley, E. F., Jr., H. C. C., Peters, D. A., Scanlan, R. H. and Sisto, F. (eds) (1995). *A modern course in Aeroelasticity*, Springer.

Drew, D. A. (1983). Mathematical modeling of two-phase flow, *Annual Review of Fluid Mechanics* **15**(1): 261–291.

Drew, D. A. (1991). Effect of particle velocity fluctuations in particle-fluid flows, *Physica A: Statistical and Theoretical Physics* **179**(1): 69–80.

Drikakis, D. (2003). Advances in turbulent flow computations using high-resolution methods, *Progress in Aerospace Sciences* **39**: 405–424.

Dufour, S. and Pelletier, D. (1998). An adaptive finite element method for incompressible flows with interfaces, *Proceedings of the 29th AIAA Fluid Dynamics Conference,*, number AIAA AIAA Paper 98-2969, American Institute of Aeronautics and Astronautics (AIAA).

Dufour, S. and Pelletier, D. (2001). Computations of multiphase flows with surface tension using an adaptive finite element method, *Numerical Heat Transfer Part A: Applications* **40**: 335–362(28).

Ergun (1952). Fluid flow through packed columns, *Chemical Engineering Progress* **48**: 9.

Fang, J. and Parriaux, A. (2008). A regularized lagrangian finite point method for the simulation of incompressible viscous flows, *J. Comput. Phys.* **227**(20): 8894–8908.

Felippa, C. A. and Deruntz, J. A. (1984). Finite element analysis of shock-induced hull cavitation, *Computer Methods in Applied Mechanics and Engineering* **44**(3): 297–337.

Feng, Y. T., Han, K. and Owen, D. R. J. (2007). Coupled lattice boltzmann method and discrete element modelling of particle transport in turbulent fluid flows: Computational issues, *International Journal for Numerical Methods in Engineering* **72**(9): 1111–1134.

Ford, R., Pain, C. C., Piggott, M. D., Goddard, A. J. H., de Oliveira, C. R. E. and Umpleby, A. P. (2004). A nonhydrostatic finite-element model for three-dimensional stratified oceanic flows. part i: Model formulation, *Monthly Weather Review* **132**(12): 2816–2831.

Foster, N. and Fedkiw, R. (2001). Practical animation of liquids, *Siggraph 2001 Annual Conference*.

Fox, R. W., McDonald, A. T. and Pritchard, P. J. (2003). *Introduction to Fluid Mechanics*, Wiley.

Frank, R. and Lazarus, R. (1964). Mixed eulerian-lagrangian method, *Fundamental Methods in Hydrodynamics* **3**: 47–63.

Fries, T.-P. and Matthies, H.-G. (2004). Clasification and overview of meshfree methods, *Technical report*, Institute of Scientific Computing, Technical University Branunschweig, Brunswick, Germany.

Fung, Y. C. (1955). *An Introduction to the Theory of Aeroelasticity*, John wiley.

Gadala, M. S. (2004). Recent trends in ale formulation and its applications in solid mechanics, *Computer Methods in Applied Mechanics and Engineering* **193**(39-41): 4247–4275.

Ge, L. and Sotiropoulos, F. (2007). A numerical method for solving the 3d unsteady incompressible navier-stokes equations in curvilinear domains with complex immersed boundaries, *Journal of Computational Physics* **225**: 1782–1809.

Gidaspow, D. (1994). *Multiphase Flow and Fluidization: Continuum and Kinetic Theory Descriptions*, Academic Press.

Gingold, R. and Monaghan, J. (1977). Smoothed particle hydrodynamics: theory and application to non-spherical stars, *Royal Astronomical Society, Monthly Notices* **181**: 375–389.

Glowinski, R., Pan, T.-W., Hesla, T. I., Joseph, D. D. and Periaux, J. (2000). A distributed lagrange multiplier/fictitious domain method for the simulation of flow around moving rigid bodies: application to particulate flow, *Computer Methods in Applied Mechanics and Engineering* **184**(2-4): 241–267.

Gopala, V. R. and van Wachem, B. G. (2008). Volume of fluid methods for immiscible-fluid and free-surface flows, *Chemical Engineering Journal* **141**(1-3): 204–221.

Gorman, G. J., Pain, C. C., Umpleby, A. P. and Piggott, M. D. (2007). Combined parallel tetrahedral mesh optimisation and dynamic load-balancing, *Comp. Methods Appl. Mech. Engrg.*

Greaves, D. M. (2006). Simulation of viscous water column collapse using adapting hierarchical grids, *International journal for numerical methods in fluids* **50**: 693–711.

Gresho, P. M. and Sani, R. L. (2000). *Incompressible Flow and the Finite Element Method*, John Wiley & Sons, Ltd.

Gresho, P. M., Sani, R. L. and Engelman, M. S. (1998). *Incompressible flow and the Finite Element Method, Volume 2: Isothermal Laminar Flow*, John Wiley & Sons Ltd.

Habashi, W. G., Dompierre, J., Bourgault, Y., Ait-Ali-Yahia, D., Fortin, M. and Vallet, M.-G. (2000). Anisotropic mesh adaptation: towards user-independent, mesh-independent and solver-independent cfd. part i: general principles, *International Journal for Numerical Methods in Fluids* **32**(6): 725–744.

Halvorsen, B. (2005). *An Experimental and Computational Study of Flow Behaviour in Bubbling Fluidized Beds*, PhD thesis, Telemark University College, Norway.

Hamm, L., Madsen, P. A. and Peregrine, D. (1993). Wave transformation in the nearshore zone: A review, *Coastal Engineering* **21**(1-3): 5 – 39. Special Issue Coastal Morphodynamics: Processes and Modelling.

Hansbo, P. and Hermansson, J. (2003). Nitsche's method for coupling non-matching meshes in fluid-structure vibration problems, *Computational Mechanics* **32**(1): 134–139.

Harten, A. (1997). High resolution schemes for hyperbolic conservation laws, *J. Comput. Phys.* **135**(2): 260–278.

Hartman, M., Havlín, V., Trnka, O. and , C. M. (1989). Predicting the free-fall velocities of spheres, *Chemical Engineering Science* **44**(8): 1743–1745.

Hasselmann, K., Barnett, T., Bouws, E., Carlson, H., Cartwright, D., Enke, K., Ewing, J., Gienapp, H., Hasselmann, D., Kruseman, P., Meerburg, A., Mller, P., Olbers, D., Richter, K., Sell, W. and Walden, H. (1973). Measurements of wind-wave growth and swell decay during the joint north sea wave project (jonswap), *Ergnzungsheft zur Deutschen Hydrographischen Zeitschrift Reihe* **12**: 95.

Hayes, M. O. (1978). Impact of hurricanes on sedimentation in estuaries, bays, and lagoons, *Estuarine Interactions* pp. 323–346.

Heinze, H. W., Dickson, S. M., Belknap, D. F. and Kelley, J. T. (2001). The effects of storm-generated currents on the sand beaches in southern maine, *AGU Spring Meeting Abstracts* pp. 51–+.

Hirsch, C. (2002). *Numerical Computation of Internal and External Flows, Volume 2, Computational Methods for Inviscid and Viscous Flows*, John Wiley & Sons Ltd.

Hirt, C. (1988). *Flow-3D Users Manual*, Flow Sciences, Inc.

Hirt, C., Amsden, A. and Cook, J. (1997). An arbitrary lagrangian-eulerian computing method for all flow speeds, *Journal of Computational Physics* **135**: 203–216.

Hirt, C. W. and Nichols, B. (1981). Volume of fluid (vof) method for the dynamics of free boundaries, *Journal of Computational Physics* **39**: 201–225.

Hsu, T.-J., Sakakiyama, T. and Liu, P. L. F. (2002). A numerical model for wave motions and turbulence flows in front of a composite breakwater, *Coastal Engineering* **46**(1): 25–50.

Hubbard, P. (1995). *Collision detection for interactive graphics applications, PhD Thesis*, PhD thesis.

Hubbard, P. (1996). Approximating polyhedra with spheres for time-critical collision detection, *ACM Trans. Graph.* **15**: 179–210.

Hudson, R. Y. and Jackson, R. A. (1953). Wave forces on breakwaters, *Trans. Am. Soc. Civil. Eng.* **118**: 653–85.

Hughes, T. J., Liu, W. J. and Zimmermann, T. K. (1981). Lagrangian-eulerian finite element formulation for incompressible viscous flows, *Compter Methods in Applied Mechanics and Engineering* **29**: 329–349.

Hughes, T. J. R. (2000). *The Finite Element Method: Linear Static and Dynamic Finite Element Analysis*, Dover Publications.

Iribarren, C. and Nogales, C. (1949). Protection des ports, *Section II, Comm. 4, XVIIth Int. Nav. Congress, Lisbon*.

Ivanenko, S. A. and Azarenok, B. N. (2002). Application of moving adaptive grids for numerical solution of 2d nonstationary problems in gas dynamics, *International Journal for Numerical Methods in Fluids* **39**(1): 1–22.

Jenkins, J. T. and Savage, S. B. (1983). A theory for the rapid flow of identical, smooth, nearly elastic, spherical particles, *J. Fluid Mech.* **130**: 187–202.

Jones, J. E., Cai, Z., Mccormick, S. F. and Russell, T. F. (1997). Control-volume mixed finite element methods, *Comput. Geosci* **1**: 289–315.

Kaceniauskas, A. (2005). Dam break flow simulation by the pseudo-concentration method, *MECHANIKA*.

Kajishima, T. and Takiguchi, S. (2002). Interaction between particle clusters and particle-induced turbulence, *International Journal of Heat and Fluid Flow* **23**(5): 639–646.

Karadeniz, H. (1993). Wave-current and fluid-structure interaction effects on thestochastic analysis of offshore structures, *International Journal of Offshore and Polar Engineering*.

Khan, A. R. and Richardson, J. F. (1987). The resistance to motion of a solid sphere in a fluid, *Chemical Engineering Communications* **62**: 135–150.

Kleefsman, K., Fekken, G., Veldman, A., Iwanowski, B. and Buchner, B. (2005). A volume-of-fluid based simulation method for wave impact problems, *Journal of Computational Physics* **206**(1): 363–393.

Kölke, A., Walhorn, E., Hübner, B. and Dinkler, D. (2003). Numerical modeling of fluid-structure interaction with free surface flows, *PAMM* **3**(1): 412–413.

Knupp, P., Margolin, L. G. and Shashkov, M. (2002). Reference jacobian optimization-based rezone strategies for arbitrary lagrangian eulerian methods, *Journal of computational Physics* **176**: 93–128.

Kobayashi, N. (1989). Wave transmission over submerged breakwaters, *Journal of Waterway, Port, Coastal and Ocean Engineering* **115**: 662–680.

Kobayashi, N. and Wurjanto, A. (1990). Numerical model for waves on rough permeable slopes, *Journal of Coastal Research* **Special Issue, No.7**: 149–166.

Kohno, H. and Tanahashi, T. (2004). Numerical analysis of moving interfaces using a level set method coupled with adaptive mesh refinement, *International Journal for Numerical Methods in Fluids* **45**(9): 921–944.

Koshizuka, S. and Oka, Y. (1996). Moving-particle semi-implicit method for fragmentation of incompressible fluid, *Nuclear Science Engineering* **123**: 421–434.

Koshizuka, S., Oka, Y. and Tamako, H. (1995a). A particle method for calculating splashing of incompressible viscous fluid, *Proc. Int. Conf. Mathematics and Computations, Reactor Physics, and Environmental Analyses*.

Koshizuka, S., Tamako, H. and Oka, Y. (1995b). A particle method for incompressible viscous flow with fluid fragmentation, *Comput. Fluid Dynamics J.* **4**: 29.

Kothe, D. B., Mjolsness, R. C., Upton, H. A., Protsik, R., Gamble, R. E., Otwell, R. L. and Lunsford, D. W. (eds) (1991a). *RIPPLE - A new model for incompressible flows with free surfaces*.

Kothe, D., Mjolsness, R. and Torrey, M. (1991b). Ripple: a computer program for incompressible flows with free surfaces, *Technical Report Technical Report LA-12007-MS*, Los Alamos National Laboratory.

Lara, J., Garcia, N. and Losada, I. (2006). Rans modelling applied to random wave interaction with submerged permeable structures, *Coastal Engineering* **53**(5-6): 395–417.

Lara, J., Losada, I. and Guanche, R. (2008). Wave interaction with low-mound breakwaters using a rans model, *Ocean Engineering* **35**(13): 1388–1400.

Latham, J. P., Mindel, J., Guises, R., Garcia, X., Xiang, J., C., P. C. and A., M. (2007). Coupled dem and cfd for coastal structures: application to armour stability and breakage, *Fifth International Conference on Coastal Structures, Venice, Italy, July 2-4, 2007*, Published by COPRI on behalf of ASCE.

Le Roux, J. (2008). An extension of the airy theory for linear waves into shallow water, *Coastal Engineering* **55**(4): 295–301.

Lee, Y.-G. and Heo, J.-K. (2005). Simulating nonlinear waves on the free surface in surf zones with two-dimensional sloping beach, *Ocean Engineering* **32**(1): 57–84.

Leonard, B. (1991). The ultimate conservative difference scheme applied to unsteady one-dimensional advection, *Computer Methods in Applied Mechanics and Engineering* **88**: 17–74.

Löhner, R. (1989). Adaptive remeshing for transient problems, *Computer Methods in Applied Mechanics and Engineering* **75**(1-3): 195–214.

Löhner, R. (2001). *Applied CFD Techniques: An Introduction Based on Finite Element Methods*, John Wiley & Sons Ltd.

Li, Y. and Raichlen, F. (2002). Non-breaking and breaking solitary wave run-up, *Journal of Fluid Mechanics* **456**: 295–318.

Lin, P. and Liu, P. F. (1998). A numerical study of breaking waves in the surf zone, *Journal of Fluid Mechanics* **359**: 239–264.

Lipnikov, K. and Shashkov, M. (2006). The error-minimization-based rezone strategy for arbitrary lagrangian-eulerian methods, *Numerical Methods for Partial Differential Equations* **22**(3): 617–637.

Liu, D. and Lin, P. (2008). A numerical study of three-dimensional liquid sloshing in tanks, *Journal of Computational Physics* **227**(8): 3921–3939.

Liu, G. R. and Liu, M. B. (2003). *Smoothed Particle Hydrodynamics. A Meshfree Particle Method*, World Scientific Publishing.

Liu, G. R. and Yang, Y. T. (1999). A new meshless method for stress analysis for solids and structures, *Proceedings of Fourth Asia-Pacific Conference on Computational Mechanics, Singapore*.

Liu, W. K., Belytschko, T. and Chang, H. (1986). An arbitrary lagrangian-eulerian finite element method for path-dependent materials, *Comput. Methods Appl. Mech. Eng.* **58**(2): 227–245.

Longatte, E., Verreman, V. and Souli, M. (2008). Time marching for simulation of fluid-structure interaction problems, *Journal of Fluids and Structures* **In Press, Corrected Proof**: –.

Losada, I. J., Lara, J. L., Christensen, E. D. and Garcia, N. (2005). Modelling of velocity and turbulence fields around and within low-crested rubble-mound breakwaters, *Coastal Engineering* **52**(10-11): 887–913.

Losada, M. and Gimenez-Curto, L. (1981). Flow characteristics on rough, permeable slopes under wave action, *Coastal Engineering* **4**: 187–206.

López, J., Hernández, J., Gómez, P. and Faura, F. (2004). A volume of fluid method based on multidimensional advection and spline interface reconstruction, *Journal of Computational Physics* **195**(2): 718–742.

López, J., Hernández, J., Gómez, P. and Faura, F. (2005). An improved plic-vof method for tracking thin fluid structures in incompressible two-phase flows, *Journal of Computational Physics* **208**(1): 51–74.

Maddrell, R. J. (2005). Lessons re-learnt from the failure of marine structures, *ICE Conference, Coastlines, Structures and Breakwaters*.

Magnaudet, J., Rivero, M. and Fabre, J. (1995). Accelerated flows past a rigid sphere or a spherical bubble. part 1. steady straining flow., *J. Fluid Mech.* **284**: 97–135.

Margolin, L. (1997). Introduction to "an arbitrary lagrangian-eulerian computing method for all flow speeds", *Journal of Computational Physics* **135**: 198–202.

Martin, J. C. and Moyce, W. J. (1952). An experimental study of the collapse of liquid columns on a rigid horizontal plane, *Philos. Trans. R. Soc. London Ser. A* **244**: 312.

Masalu, D. C. (2008). Coastal data and information management for integrated coastal management: The role of iode, *Marine Policy* **32**(4): 544–550.

Mavriplis, D. J. (1995). Unstructured mesh generation and adaptivity, *Technical report*, NASA - ICASE.

Mavriplis, D. J. (1997). Adaptive meshing techniques for viscous flow calculations on mixed element unstructured meshes, *Technical report*.

Mavriplis, D. J. and Yang, Z. (2006). Construction of the discrete geometric conservation law for high-order time-accurate simulations on dynamic meshes, *Journal of Computational Physics* **213**(2): 557–573.

McKee, S., Tomé, M., Ferreira, V., Cuminato, J., Castelo, A., Sousa, F. and Mangiavacchi, N. (2008). The mac method, *Computers & Fluids* **37**(8): 907–930.

Melby, J. A. and Turk, G. F. (1997). Core-loc concrete armor units, *Technical report*, US Army Corps of Engineers, Waterways Experiment Station.

Mitra, S. and Sinhamahapatra, K. (2008). 2d simulation of fluid-structure interaction using finite element method, *Finite Elements in Analysis and Design* **In Press, Corrected Proof**: –.

Mohammadi, S. and Pooladi, A. (2007). Non-uniform isentropic gas flow analysis of explosion in fractured solid media, *Finite Elements in Analysis and Design* **43**(6-7): 478–493.

Mohan, R. K., Magoon, O. and Pirrello, M. (eds) (2003). *Advances in Coastal Structure Design*, American Society of Civil Engineers.

Morrell, J., Sweby, P. and Barlow, A. (2007). A cell by cell anisotropic adaptive mesh ale scheme for the numerical solution of the euler equations, *Journal of Computational Physics* **226**(1): 1152–1180.

Munjiza, A. (2004). *The Combined Finite-Discrete Element Method*, John Wiley & Sons.

Munjiza, A. and Latham, J. (2004). Virtual geoscience workbench detailed design, *Internal report*, Department of Engineering, Queen Mary, University of London.

Munjiza, A., Latham, J. P. and Andrews, K. R. F. (2000). Detonation gas model for combined finite-discrete element simulation of fracture and fragmentation, *International Journal for Numerical Methods in Engineering* **49**(12): 1495–1520.

Naito, S. (2006). Generation and absorption in wave basins: Theory and application, *International Journal of Offshore and Polar Engineering* **16**: 81–89.

Nasar, T., Sannasiraj, S. and Sundar, V. (2008). Experimental study of liquid sloshing dynamics in a barge carrying tank, *Fluid Dynamics Research* **40**(6): 427–458.

Nezami, E. G., Hashash, Y. M. A., Zhao, D. and Ghaboussi, J. (2006). Shortest link method for contact detection in discrete element method, *International Journal for Numerical and Analytical Methods in Geomechanics* **30**: 783–801.

Nezami, E. G., Hashash, Y. M., Zhao, D. and Ghaboussi, J. (2004). A fast contact detection algorithm for 3-d discrete element method, *Computers and Geotechnics* **31**(7): 575–587.

Noh, W. (1964). A time dependent two space dimensional coupled eulerian lagrangian code, *Fundamental Methods in Hydrodynamics* **3**: 117–179.

Noh, W. F. and Woodward, P. (1976). Slic /simple line interface calculation, *in* A. I. van de Vooren and P. J. Zandbergen (eds), *Some Methods of Resolution of Free Surface Problems*, Vol. 59 of *Lecture Notes in Physics, Berlin Springer Verlag*, pp. 330–340.

Oñate, E., Idelsohn, S. and Zienkiewicz, O. C. amd Taylor, R. (1996). A finite point method in computational mechanics. applications to convective transport and fluid flow, *International Journal for Numerical Methods in Engineering* **39**(22): 3839–3866.

Omri, A. and Nasrallah, S. (1999). Control volume finite element numerical simulation of mixed convection in an air-cooled cavity control volume finite element numerical simulation of mixed convection in an air-cooled vavity, *Numerical Heat Transfer: Applications* **36**: 615–637(23).

Osher, S. and Fedkiw, R. (2003). *Level Set Methods and Dynamic Implicit Surfaces*, Springer.

Osher, S. and Sethian, J. (1988). Fronts propagating with curvature dependent speed: Algorithms based on hamilton-jacobi formulations, *J. Comput. Physics* **79**: 12–49.

Pain, C. C., Eaton, M. D., Gomes, J. L. M. A., de Oliveira, C. R. E., Goddard, A. J. H., Umpleby, A. P., Ackroyd, R. T., van Dam, H., van der Hagen, T. H. J. J. and Lathouwers, D. (2003). The fission power of a conceptual fluidised bed thermal nuclear reactor, *Nuclear Energy* **42**: 113–123.

Pain, C. C., Umpleby, A. P., de Oliveira, C. and Goddard, A. J. (2001a). Tetrahedral mesh optimization and adaptivity for steady-state and transient finite element calculations, *Computer Methods in Applied Mechanics and Engineering* **190**: 3771–96.

Pain, C., Mansoorzadeh, S., de Oliveira, C. and Goddard, A. (2001b). Numerical modeling of gas-solid fluidised beds using the two fluids approach, *International Journal for Numerical Methods in Engineering* **36**: 91–124.

Palmer, G. N. and Christian, C. D. (1998). Design and construction of rubble mound breakwaters, *IPENZ Transactions* **25**: 19–33.

Peraire, J., Vahdati, M., Morgan, K. and Zienkiewicz, O. C. (1987). Adaptive remeshing for compressible flow computations, *J. Comput. Phys.* **72**(2): 449–466.

Peregrine, D. H. (2003). Water-wave impact on walls, *Annual Review of Fluid Mechanics* **35**(1): 23–43.

Peregrine, D. H. and Cooker, M. J. (1990). Computations of waves breaking against a vertical wall, *5th Intl. Workshop on Water Waves & Floating bodies*.

Piggott, M. D., Gorman, G. J., Pain, C. C., Allison, P. A., Candy, A. S., Martin, B. T. and Wells, M. R. (2008). A new computational framework for multi-scale ocean modelling based on adapting unstructured meshes, *International Journal for Numerical Methods in Fluids* **56**(8): 1003–1015.

Piggott, M., Pain, C., Gorman, G., Power, P. and Goddard, A. (2005). h, r, and hr adaptivity with applications in numerical ocean modelling, *Ocean Modelling* **10**(1-2): 95–113.

Pilliod, J. (1992). *An analysis of piecewise linear interface reconstruction algorithms for volume-of-fluid methods*, Master's thesis, U. C. Davis.

Pilliod, J. E. and Pucket, E. G. (2004). Second-order accurate volume-of-fluid algorithms for tracking material interfaces, *Journal of Computational Physics* **199**: 465–502.

Plaza, A., Padrón, M. A. and Carey, G. F. (2000). A 3d refinement/derefinement algorithm for solving evolution problems, *Appl. Numer. Math.* **32**(4): 401–418.

Poshel T., B. V. (1993). Static friction phenomena in granular materials: Coulomb law versus particle geometry, *Physical Review Letters* **71**: 3963–3966.

Preece, D. S., Jensen, R. P., Perkins, E. D. and Williams, J. R. (1999). Sand production modelling using superquadric discrete elements and coupling of fluid flow and particle motion, *Technical report*, Sandia National Laboratories, Albuquerque, NM.

Prosperetti, A. and Tryggvason, G. (eds) (2007). *Computational Methods for Multiphase Flow*, Cambridge University Press.

Puckett, E. (1991). A volume-of-fluid interface tracking algorithm with applications to computing shock wave refraction, *Proceedings of the Fourth International Symposium on Computational Fluid Dynamics, Davis, CA*, pp. 933–938.

Robichaud, M. P., Tanguy, P. A. and Fortin, M. (1990). An iterative implementation of the uzawa algorithm for 3-d fluid flow problems, *International Journal for Numerical Methods in Fluids* **10**: 429–442.

Ross, M. R. (2006). *Coupling and Simulation of Acoustic Fluid-Structure Interaction Systems Using Localized Lagrange Multipliers*, PhD thesis, University of Colorado.

Rowe, P. N. (1961). Drag forces in a hydraulic model of fluidized bed - part ii, *Trans. Instn. Chem.* **39**: 175.

Rudman, M. (1997). Volume tracking methods for interfacial flow calculations, *International Journal for Numerical Methods in Fluids* **24**: 671–691.

Saad, Y. (2003). *Iterative Methods for Sparse Linear Systems, Second edition*, SIAM.

Sangani, A. S. and Didwania, A. K. (2006). Dispersed-phase stress tensor in flows of bubbly liquids at large reynolds numbers, *Journal of Fluid Mechanics Digital Archive* **248**(-1): 27–54.

Sarrate, J., Huerta, A. and Donea, J. (2001). Arbitrary lagrangian-eulerian formulation for fluid-rigid body interaction, *Computer Methods in Applied Mechanics and Engineering* **190**: 3171–3188.

Sawaragi, T. (1995). *Coastal Engineering, Waves, Beaches, Wave Structure Interaction*, Elsevier.

Scardovelli, R. and Zaleski, S. (1999). Direct numerical simulation of free-surface and interfacial flow, *Annual Review of Fluid Mechanics* **31**: 567–603.

Schäfer, J., Dippel, S. and Wolf, D. E. (1996). Force schemes in simulations of granular materials, *J. Phys. I (France)* **6**: 5.

Schäfer, M. and Turek, S. (1996). Benchmark computations of laminar flow around cylinder, *in* E. Hirschel (ed.), *Flow Simulation with High-Performance Computers II. Notes on Numerical Fluid Mechanics*, Vol. 52, Vieweg, p. 547.

Shao, S. (2006). Incompressible sph simulation of wave breaking and overtopping with turbulence modelling, *International Journal for Numerical Methods in Fluids* **50**(5): 597–621.

Shirvanian, P., Calo, J. and Hradil, G. (2006). Numerical simulation of fluid-particle hydrodynamics in a rectangular spouted vessel, *International Journal of Multiphase Flow* **32**(6): 739–753.

Smolianski, A. (2005). Finite-element/level-set/operator-splitting (felsos) approach for computing two-fluid unsteady flows with free moving interfaces, *International Journal for Numerical Methods in Fluids* **48**(3): 231–269.

Snyman, J. A. (2005). *Practical Mathematical Optimization: An Introduction to Basic Optimization Theory and Classical and New Gradient-Based Algorithms*, Springer Publishing.

Song, Y., Turton, R. and Kayihan, F. (2006). Contact detection algorithms for dem simulations of tablet-shaped particles, *Powder Technology* **161**(1): 32–40.

Sprague, M. A. and Geers, T. L. (2004). A spectral-element method for modelling cavitation in transient fluid-structure interaction, *International Journal for Numerical Methods in Engineering* **60**(15): 2467–2499.

Srinivas, H. and Nakagawa, Y. (2008). Environmental implications for disaster preparedness: Lessons learnt from the indian ocean tsunami, *Journal of Environmental Management* **89**(1): 4–13.

Stoker, J. (1957). *Water Waves: The Mathematical Theory with Applications*, Interscience Publishers, New York.

Stokes, G. (1847). On the theory of oscillatory waves, *Transactions of the Cambridge philosophical society* **8**: 441–455.

Sukumar, N., Chopp, D. L., Moës, N. and Belytschko, T. (2001). Modeling holes and inclusions by level sets in the extended finite-element method, *Computer Methods in Applied Mechanics and Engineering* **190**: 6183–6200.

Sussman, M., Smereka, P. and Osher, S. (1994). A level set approach for computing solutions to incompressible two-phase flow, *Journal of Computational Physics* **114**(1): 146–159.

Swaminathan, C. R., Voller, V. R. and Patankar, S. V. (1993). A streamline upwind control volume finite element method for modeling fluid flow and heat transfer problems, *Finite Elem. Anal. Des.* **13**(2-3): 169–184.

Syamlal, M. and O'Brien, T. (1987). A generalized drag correlation for multiparticle systems, *Doe report*, Morgantown Energy Technology Center.

Takeuchi, S., Morita, I. and Kajishima, T. (2008). Motion of particle agglomerate involving interparticle force in dilute suspension, *Powder Technology* **184**(2): 232–240.

Tanaka, M., Dold, J. W., Lewy, M. and Peregrine, D. H. (1987). Instability and breaking of a solitary wave, *Journal of Fluid Mechanics Digital Archive* **185**(-1): 235–248.

Tedesco, J. W., McDougal, W. G., Bloomquist, D. and Wise, L. A. (2003). Response of concrete armor units to wave-induced hydrodynamic loads, *Computers & Structures* **81**(8-11): 983 – 994. K.J Bathe 60th Anniversary Issue.

Tezduyar, T. E. (1998). Stabilized finite element formulations for incompressible flow computations, *Advances in Applied Mechanics* **28**: 1–44.

Thomas, P. D. and Lombard, C. K. (1979). Geometric conservation law and its application to flow computations on moving grids, *AIAA Journal* **17**: 1030.

Thompson, E. (1986). Use of pseudo-concentrations to follow creeping viscous flows during transient analysis, *International Journal for Numerical Methods in Fluids* **6**(10): 749–761.

Torrey, M. D., Cloutman, L. D., Mjolsness, R. C. and Hirt, C. W. (1985). Nasa-vof2d: a computer program for incompressible flows with free surfaces, *NASA STI/Recon Technical Report N* **86**: 30116–+.

Torrey, M. D., Mjolsness, R. C. and Stein, L. R. (1987). Nasa-vof3d: A three-dimensional computer program for incompressible flows with free surfaces, *NASA STI/Recon Technical Report N* **88**: 10288–+.

Tribbia, J. and Moser, S. C. (2008). More than information: what coastal managers need to plan for climate change, *Environmental Science & Policy* **11**(4): 315–328.

Trépanier, J. Y., Reggio, M., Zhang, H. and Camarero, R. (1991). A finite-volume method for the euler equations on arbitrary lagrangian-eulerian grids, *Computers & Fluids* **20**(4): 399–409.

Tsinker, G. (1997). *Handbook of port and harbour engineering*, Chapman & Hall, London, U.K.

Turton, R. and Clark, N. (1987). An explicit relationship to predict spherical particle terminal velocity, *Powder Technology* **53**: 127–129.

Ubbink, O. (1997). *Numerical prediction of two fluid systems with sharp interfaces*, PhD thesis, Imperial College of Science, Technology, and Medicine.

USACE (2002). *Coastal Engineering Manual*, US Army Corps of Engineers.

Van der Meer, J. W. (1988). *Rock slopes and gravel beaches under wave attack*, PhD thesis, Delft University of Technology.

Wacholder, E. and Weihs, D. (1972). Slow motion of a fluid sphere in the vicinity of another sphere or a plane boundary, *Chemical Engineering Science* **27**(10): 1817–1828.

Walkden, M. J., Wood, D. J., Bruce, T. and Peregrine, D. H. (2001). Impulsive seaward loads induced by wave overtopping on caisson breakwaters, *Coastal Engineering* **42**(3): 257 – 276.

Wallis, G. B. (1991). The averaged bernoulli equation and macroscopic equations of motion for the potential flow of a two-phase dispersion, *International Journal of Multiphase Flow* **17**(6): 683–695.

Wandinger, J. (1998). A symmetric craig-bampton method of coupled fluid-structure systems, *Engineering Computations* **15**: 450–461.

Wang, J., Liu, G. and Wu, Y. (2001). A point interpolation method for simulating dissipation process of consolidation, *Computer Methods in Applied Mechanics and Engineering* **190**: 5907–5922(16).

Weggel, J. R. and Maxwell, W. H. (1970). Numerical model for wave pressure distributions, *Journal of Waterway, Harbours and Coastal Engineering Division* **96**: 623–641.

Wells, B. V., Baines, M. J. and Glaister, P. (2005). Generation of arbitrary lagrangian-eulerian (ale) velocities, based on monitor functions, for the solution of compressible fluid equations, *International Journal for Numerical Methods in Fluids* **47**(10-11): 1375–1381.

Wen, C. and Yu, Y. (1966). Mechanics of fluidization, *Chemical Engineering Progress* **62**: 100–111.

Whitaker, N. (1990). Numerical solution of the hele-shaw equations, *Journal of Computational Physics* **90**: 176–199.

White, F. M. (1999). *Fluid Mechanics*, McGraw-Hill.

Wigley, T. and Raper, S. (1987). Thermal expansion of sea water associated with global warming, *Nature* **330**: 127–131.

Wilson, C., Collins, G., Mindel, J. and Pain, C. (2008). A compressive advection scheme for incompressible multi-material modelling on unstructured meshes, *Unpublished, in preparation*.

Young, Y. (2008). Fluid-structure interaction analysis of flexible composite marine propellers, *Journal of Fluids and Structures* **24**(6): 799–818.

Youngs, D. (1982). Time-dependent multi-material flow with large fluid distortion, *in* K. W. Morton and M. J. Baines (eds), *Numerical Methods for Fluid Dynamics*, Academic Press.

Zalesak, S. T. (1979). Fully multidimensional flux-corrected transport algorithms for fluids, *J. Comput. Phys. ()*. **31**: 335–362.

Zelt, J. A. and Raichlen, F. (1990). A lagrangian model for wave-induced harbour oscillations, *Journal of Fluid Mechanics* **213**: 203–225.

Zhang, D. and Prosperetti, A. (1994). Averaged equations for inviscid disperse two-phase flow, *Journal of Fluid Mechanics* **267**: 185–219.

Zhang, Y. and Reese, J. (2003). The drag force in two-fluid models of gas-solid flows, *Chemical Engineering Science* **58**: 1641–1644.

Zhao, Y., Galvin, K. P. and Davis, R. H. (2002a). Motion of a sphere down a rough plane in a viscous fluid, *International Journal of Multiphase Flow* **28**(11): 1787–1800.

Zhao, Y., Hui Tan, H. and Zhang, B. (2002b). A high-resolution characteristics-based implicit dual time-stepping vof method for free surface flow simulation on unstructured grids, *Journal of Computational Physics* **183**(1): 233–273.

Zienkiewicz, O. and Taylor, R. (1991). *The finite element method. Vol. I. Basic formulations and linear problems*, London: McGraw-Hill.

Zigrang, D. and Sylvester, N. (1981). An explicit equation for particle settling velocities in solid-liquid systems, *AIChE Journal* **27**: 1043–1044.

Appendix A

Numerical methods used in FSI tests

The following tables were obtained from Schäfer and Turek (1996) , which were used for comparison in the 2D and 3D validation cases of Chapter 4.

	Space discretization	Time discretization	Solver	Implementation
1	FD, blockstructured non-staggered QUICK upwinding	fully implicit 2nd ord. equidistant	artificial compressibility expl. 5-step Runge-Kutta FAS-MG (steady) line-Jacobi (unsteady)	serial Fujitsu VPP500 1600 PEAK
2	FV, blockstructured 2nd ord. upwindig	implicit Euler equidistant	ILU with algebraic MG for linear problems	serial IBM RS6000/370 37 LINP
3	FV, blockstructured non-staggered QUICK upwinding	fully implicit 2nd ord. equidistant	stream function form fixed-point iteration ILU for lin. subproblems	serial SGI-Indigo2 75 PEAK parallel Cray T3D/16 16x88 LINP
4	FE, blockstructured 4Q1-Q1 BTD stabilisation	Projection 2 (Gresho) Crank-Nicolson (diff.) explicit Euler (conv.) adaptive	pseudo time step (steady) PCG for lin. subproblems hierarch. preconditioning	parallel GC/PP32 32x13.9 LINP
5a	FV, unstructured 1st ord. upwind	STARCD software	pressure correction	serial HP 735 13 LINP
5b	FV, unstructured CDS	STARCD software	pressure correction	serial HP 735 13 LINP
6	Lattice BGK equidistant orth. grid	explicit equidistant	gaskinetic solution of BGK-Boltzmann equation evol. of distribution funct.	serial HP735 13 LINP
7a	FV, blockstructured non-staggered CDS with def. corr.	Crank-Nicolson equidistant	nonlinear MG SIMPLE smoothing ILU for lin. subproblems	serial HP735 13 LINP
7b	FV, blockstructured non-staggered CDS with def. corr.	fully implicit 2nd ord. equidistant	nonlinear MG SIMPLE smoothing ILU for lin. subproblems	parallel GC/PP128,32,8 128x13.9 LINP
8a	FE, unstructured P2-P1 (Taylor-Hood) CDS	2nd order fract. step operator splitting equidistant	nonlinear GMRES PCG for lin. subproblems	serial SGI R4000 8.3 LINP
8b	FE, unstructured P2-P1 (Taylor-Hood) CDS adaptive refinement	2nd order fract. step operator splitting equidistant	nonlinear GMRES PCG for lin. subproblems	serial SGI R4400 13.2 LINP IBM RS6000/590 58 LINP
9a	FV, blockstructured CDS	fully implicit 2nd ord. equidistant	SIMPLE ILU for lin. subproblems	serial IBM RS6000/250 34 LINP
9b	FV, unstructured CDS	fully implicit 2nd ord. equidistant	SIMPLE ILU-CGSTAB for lin. subproblems	serial IBM RS6000/590 90 LINP

Table A.1: Numerical methods used for comparison in 2D and 3D steady flow tests.

	Space discretization	Time discretization	Solver	Implementation
10	FE, blockstructured Q1(rot)-Q0 adaptive upwind	2nd order fract. step projection method adaptive	fixed-point iteration MG for lin. NS with Vanka smoother (steady) MG for scalar lin. subproblems (unsteady)	serial IBM RS6000/590 90 LINP
11	FV, structured CDS with momentum interpolation	explicit 3rd ord. Runge-Kutta equidistant	SIMPLE ILU for lin. subprobl.	serial SNI S600/20 5000 PEAK
12	FV, unstructured non-staggered adapt. 2nd ord. DISC deferred correction	–	SIMPLEC ILU-BICGSTAB for lin. subproblems	serial SUN SS10 5.5 LINP
13a	FD, structured	explicit Euler equidistant	stream function form pseudo time step SOR for lin. subprobl.	serial IBM RS6000/590 90 LINP
13b	FD, structured	explicit 4th order Runge-Kutta-Gill equidistant	stream function form SOR for lin. subprobl.	serial IBM RS6000/590 90 LINP
14a	FE, blockstructured P1-P0 (Crouzeix-Raviart) 1st order upwind	–	nonlinear MG Vanka smoother	serial HP737/125 6.6 LINP
14b	FE, blockstructured Q1(rot)-Q0 1st order upwind		fixed-point iteration MG for lin. NS with Vanka smoother	parallel GC/PP96 96x13.9 LINP
14c	FE, unstructured P1-P0 (Cr.-Rav.) Samarskij upwind adaptive refinement	BDF(2), equidistant	fixed-point iteration GMRES for pressure Schur-complement lin. MG for velocity	parallel GC/PP24 24x13.9 LINP
15a	FD, structured staggered, orthogonal CDS/UDS flux-blend.	explicit Euler adaptive	SOR for pressure	serial HP720 7.4 LINP
15b	FD, structured staggered, orthogonal CDS/UDS flux blend.	explicit Euler adaptive	SOR for pressure	parallel HP720 cluster 8x7.4 LINP
16	FV, blockstructured CDS	explicit 2nd ord. leap-frog time-lagged diff.	pressure correction Gauss-Seidel for lin. subproblems	serial SGI Indigo 9.6 LINP Convex C3820 19.2 LINP
17	FV, unstructured adaptive upwind	–	fixed-point iteration MG for lin. NS BILU$_\beta$ smoother	serial SGI R4400 8.3 LINP

Table A.2: (Continued from Table A.1) Numerical methods used for comparison in 2D and 3D steady flow tests.