

A Parallel Explicit Incompressible Smoothed Particle Hydrodynamics (ISPH) Model for Nonlinear Hydrodynamic Applications

by

Shahab Yeylaghi

B.Sc., Urmia University, 2006

M.Sc., Sharif University of Technology, 2009

A Dissertation Submitted in Partial Fulfillment of the
Requirements for the Degree of

DOCTOR OF PHILOSOPHY

in the Department of Mechanical Engineering

© Shahab Yeylaghi, 2016
University of Victoria

All rights reserved. This dissertation may not be reproduced in whole or in part, by photocopying or other means, without the permission of the author.

**A Parallel Explicit Incompressible
Smoothed Particle Hydrodynamics (ISPH) Model for Nonlinear
Hydrodynamic Applications**

by

Shahab Yeylaghi

B.Sc., Urmia University, 2006

M.Sc., Sharif University of Technology, 2009

Supervisory Committee

Dr. C. Crawford, Co-Supervisor
(Department of Mechanical Engineering)

Dr. P. Oshkai, Co-Supervisor
(Department of Mechanical Engineering)

Dr. J. Gemmrich, Outside Member
(School of Earth and Ocean Sciences)

Supervisory Committee

Dr. C. Crawford, Co-Supervisor
(Department of Mechanical Engineering)

Dr. P. Oshkai, Co-Supervisor
(Department of Mechanical Engineering)

Dr. J. Gemmrich, Outside Member
(School of Earth and Ocean Sciences)

Summary

Fluid structure interactions in the presence of a free surface includes complex phenomena, such as slamming, air entrainment, transient loads, complex free surface profiles and turbulence. Hence, an appropriate and efficient numerical method is required to deal with these type of problems (efficient both in problem setup and numerical solution). Eulerian mesh-based methods can be used to solve different types of problems, however they have difficulties in problems involving moving boundaries and discontinuities (e.g. fluid structure interactions in the presence of a free surface). Smoothed Particle Hydrodynamics (SPH) is a mesh-less Lagrangian particle method, ideal for solving problems with large deformation and fragmentation such as complex free surface flows. The SPH method was originally invented to study astrophysical applications and requires modifications in order to be applied for hydrodynamic applications. Applying solid boundary conditions for hydrodynamic applications in SPH is a key difference to the original SPH developed for astrophysics. There are several methods available in literature to apply solid boundaries in SPH. In this research, an accurate solid boundary condition is used to calculate the pressure at the boundary particles based on the surrounding fluid particles. The two main methods to calculate the pressure in the SPH method are the weakly compressible SPH (WCSPH) and

the incompressible SPH (ISPH) approaches. The WCSPH uses the equation of state while ISPH solves Poisson's equation to determine the pressure. In this dissertation, an explicit incompressible SPH (ISPH) method is used to study nonlinear free surface applications. In the explicit ISPH method, Poisson's equation is explicitly solved to calculate the pressure within a projection based algorithm. This method does not require solving a set of algebraic equations for pressure at each time step unlike the implicit method. Here, an accurate boundary condition along with an accurate source term for Poisson's equation is used within the explicit method. Also, the sub-particle turbulent calculation is applied to the explicit ISPH method (which handles large-scale turbulent structures implicitly) in order to calculate the flow field quantities and consequently forces on the device more accurately.

The SPH method is typically computationally more expensive than Eulerian-based CFD methods. Therefore, parallelization methods are required to improve the performance of the method, especially for 3D simulations. In this dissertation, two novel parallel schemes are developed based on Open Multi Processing (OpenMP) and Message Passing Interface (MPI) standards. The explicit ISPH approach is an advantage for parallel computing but our proposed method could also be applied to the WCSPH or implicit ISPH. The proposed SPH model is used to simulate and analyze several nonlinear free surface problems. First, the proposed explicit ISPH method is used to simulate a transient wave overtopping on a horizontal deck. Second, a wave impacting on a scaled oscillating wave surge converter (OWSC) is simulated and studied. Third, the performance and accuracy of the code is tested for a dam-break impacting on tall and short structures. Forth, the hydrodynamic loads from the spar of a scaled self-reacting point absorber wave energy converter (WEC) design is studied. Finally, a comprehensive set of landslide generated waves are modeled and analyzed and a new technique is proposed to calculate the motion of a slide on an inclined ramp implicitly without using a prescribed motion.

Contents

| | |
|---|-------------|
| Supervisory Committee | ii |
| Abstract | iii |
| Table of Contents | v |
| List of Tables | viii |
| List of Figures | ix |
| ACKNOWLEDGEMENTS | xiv |
| 1 Introduction | 1 |
| 1.1 Motivation | 1 |
| 1.2 Dissertation Outline | 3 |
| 1.3 Research Contributions | 5 |
| 2 A Comparison of SPH and RANS Models for Simulation of Wave Over- | 8 |
| topping | |
| 2.1 Abstract | 8 |
| 2.2 Introduction | 9 |
| 2.3 Methodology | 10 |
| 2.4 Results | 13 |
| 2.5 Conclusion | 18 |
| 2.6 Acknowledgment | 18 |
| 3 ISPH Modelling of an Oscillating Wave Surge Converter Using an | |
| OpenMP-Based Parallel Approach | 19 |
| 3.1 Abstract | 20 |

| | | |
|----------|---|-----------|
| 3.2 | Introduction | 20 |
| 3.3 | Methodology | 22 |
| 3.3.1 | Governing Equations | 22 |
| 3.3.2 | Boundary Conditions | 26 |
| 3.4 | Parallelization Scheme | 27 |
| 3.5 | Test Cases | 29 |
| 3.5.1 | Test case 1: Dam Break on a Structure | 29 |
| 3.5.2 | Test case 2: Wedge Water Entry Simulation | 33 |
| 3.5.3 | Test case 3: Wave Impacting an Oscillating Wave Surge Converter | 37 |
| 3.5.3.1 | Experiment | 37 |
| 3.5.3.2 | SPH simulations | 38 |
| 3.6 | Conclusion | 46 |
| 3.7 | Acknowledgment | 46 |
| 4 | SPH Modeling of Hydrodynamic Loads on a Point Absorber Wave Energy Converter Hull | 47 |
| 4.1 | Abstract | 47 |
| 4.2 | Introduction | 48 |
| 4.3 | Experiment | 50 |
| 4.4 | Methodology | 50 |
| 4.5 | Results | 54 |
| 4.6 | Conclusion | 62 |
| 4.7 | Acknowledgment | 62 |
| 5 | ISPH Modelling for Hydrodynamic Applications Using A New MPI-Based Parallel Approach | 63 |
| 5.1 | Abstract | 63 |
| 5.2 | Introduction | 64 |
| 5.3 | SPH Methodology | 68 |
| 5.4 | Parallelization Scheme | 71 |
| 5.4.1 | Domain Decomposition and Initial Load Balancing | 71 |
| 5.4.1.1 | Spatial Domain Decomposition | 72 |
| 5.4.1.2 | Domain Decomposition Based on Particles (Initial Load Balancing) | 73 |
| 5.4.2 | Peano-Hilbert Sorting | 74 |

| | | |
|----------|---|------------|
| 5.4.3 | Dynamic Load Balancing | 75 |
| 5.5 | Results | 77 |
| 5.5.1 | Dam Break in a Tank | 77 |
| 5.5.2 | A Tank Full of Still Water | 78 |
| 5.5.3 | Column of Water Collapsing in the Middle of a Tank | 79 |
| 5.5.4 | Dam Break on a Structure | 82 |
| 5.6 | Conclusions | 92 |
| 5.7 | Acknowledgment | 94 |
| 6 | ISPH Modelling of Landslide Generated Waves for Rigid and De- | |
| | formable Slides in Newtonian and Non-Newtonian Reservoir Fluids | 95 |
| 6.1 | Abstract | 96 |
| 6.2 | Introduction | 96 |
| 6.3 | SPH Methodology | 99 |
| 6.4 | A New Method for Sliding Solid-Solid Contact in SPH | 101 |
| 6.5 | Results | 104 |
| 6.5.1 | Rigid Slide | 104 |
| 6.5.1.1 | Submarine Landslide in 2D | 104 |
| 6.5.1.2 | 2D Subaerial Landslide-Scott Russell Wave Generator | 107 |
| 6.5.1.3 | Subaerial Landslide Generated Waves (SPHeric Test-11), 2D Channel | 110 |
| 6.5.1.4 | Subaerial Landslide Generated Waves (SPHeric Test-11), 3D Basin | 113 |
| 6.5.2 | Deformable Slide | 121 |
| 6.5.3 | Landslide in a non-Newtonian Reservoir | 122 |
| 6.6 | Conclusion | 126 |
| 6.7 | Acknowledgment | 127 |
| 7 | Conclusions and Future Work | 128 |
| 7.1 | Conclusions | 128 |
| 7.2 | Future Work | 131 |
| | Bibliography | 132 |

List of Tables

| | | |
|-----------|---|-----|
| Table 3.1 | Wall clock time for 200 time steps | 46 |
| Table 6.1 | Comparison of the parameters in Figure 6.10a at $t=0.285$ s. . . . | 110 |
| Table 6.2 | Summary of important parameters in Heller et al. [1] experiment (Figure 6.11). | 111 |

List of Figures

| | | |
|------|---|----|
| 2.1 | Sketch of the kernel support domain (S) and neighboring particles. . . | 11 |
| 2.2 | Numerical wave tank. | 13 |
| 2.4 | Time history of free surface at different locations obtained from explicit ISPH and plotted vs experimental data [2] and numerical results [3]. . | 14 |
| 2.3 | Explicit ISPH simulation of numerical wave tank with deck. | 15 |
| 2.5 | Vertical variation of the horizontal velocity u (m/s) at leading edge of the deck, $x = 8$ m. (Experimental data from [2] and VOF results from [3]) | 16 |
| 2.5 | Vertical variation of the horizontal velocity u (m/s) at leading edge of the deck, $x = 8$ m. (Experimental data from [2] and VOF results from [3]) | 17 |
| 3.1 | Two-dimensional underlying grid for the link list algorithm. Cells enclosed by red contours are handled by the same thread. | 27 |
| 3.2 | Schematic of the validation case: dam break in the vicinity of a tall structure (dimensions in cm). Experiment reported by Gomez-Gesteira et al. [4]. | 30 |
| 3.3 | Magnitude of the fluid velocity in x-direction (u) at $x = 0.754$ m, $y = 0.31$ m, $z = 0.026$ m as a function of time for dam break experiment. . | 31 |
| 3.4 | Flow due to the dam break on the tall structure at $t = 0.9$ s. | 32 |
| 3.5 | Wall clock time for 500 time steps for different thread combinations for 1, 2, 4 and 16 threads. | 33 |
| 3.6 | Wall clock time for 500 time steps for different number of particles. . . | 34 |
| 3.7 | Schematic of the wedge (dimensions in cm). | 34 |
| 3.8 | Particles are shown with their pressure at $t = 0.00435$ s. | 35 |
| 3.9 | Particles are shown with their pressure at $t = 0.0158$ s. | 35 |
| 3.10 | Pressure along the boundary of the wedge at $t = 0.00435$ s. | 36 |

| | | |
|------|---|----|
| 3.11 | Pressure along the boundary of the wedge at $t = 0.0158$ s. | 36 |
| 3.12 | Schematic of the flap (dimensions in mm). | 37 |
| 3.13 | Schematic of the wave tank (dimensions in m). | 38 |
| 3.14 | Particles in the domain are colored by their pressure at three time steps (a) shown in 3D, (b-d) shown in 2D. | 39 |
| 3.15 | Pressure field on the flap face. | 40 |
| 3.16 | Free surface elevation at ($x = 10.8$ m and $y = 3.5$ m) as a function of time. | 41 |
| 3.17 | Angular position of the flap as a function of time. | 42 |
| 3.18 | Dynamic pressure at the surface of the flap. | 45 |
| 4.1 | Pressure force evaluation on the solid boundary | 53 |
| 4.2 | Time evolution of the water-front for dam-break problem $\tau = t(g/H)^{0.5}$ | 55 |
| 4.3 | 3D domain | 56 |
| 4.4 | (a) Sketch of the WEC under water, (b) Particle representation of the WEC in SPH | 57 |
| 4.5 | Particles velocities at $t = 1.15$ s, ($1.47 \leq x \leq 1.53, 0 \leq y \leq 3, 0 \leq z \leq$ 2.2) | 58 |
| 4.6 | Particles velocities at $t = 1.75$ s, ($1.47 \leq x \leq 1.53, 0 \leq y \leq 3, 0 \leq z \leq$ 2.2) | 59 |
| 4.7 | Displacement. $A = 0.06$ m and $\omega = 4$ rad/s | 60 |
| 4.8 | Hydrodynamic force. $A = 0.06$ m and $\omega = 4$ rad/s | 61 |
| 4.9 | Total force. $A = 0.06$ m and $\omega = 4$ rad/s | 61 |
| 5.1 | Domain decomposition and initial load balancing (blue color corre- sponds to the initial particle region). (a) Spatial bounding boxes: cells (green) and processors (red), (b) <i>Departments</i> (dashed red) and <i>chairs</i> (dashed black). | 72 |
| 5.2 | Peano–Hilbert curve in 2D and 3D. (a) 2D first order, (b) 2D second order, (c) 2D fifth order, (d) 3D forth order. | 76 |

| | | |
|------|--|----|
| 5.3 | Domain decomposition and initial load balancing for a problem where the fluid particles are non-uniformly distributed. The colors in (b) and (c) from left to right corresponds to processors 1-4. (a) The non-uniformly fluid particles in the tank, (b) Domain decomposition without initial load balancing, (c) Domain decomposition with initial load balancing. | 78 |
| 5.4 | Particle numbers for 4 processors in Figure 5.3b and Figure 5.3c. | 79 |
| 5.5 | Speedup and efficiency for a tank full of still water. | 80 |
| 5.6 | Particle distribution at: (a) $t=0$ s, (b) $t=0.25$ s, (c) $t=0.35$ s, (d) $t=0.5$ s. | 81 |
| 5.7 | Particles are colored by their processor ID in the x , y and z directions (z vertical) at: (a) $t=0$ s, (b) $t=0.25$ s, (c) $t=0.35$ s, (d) $t=0.5$ s. | 81 |
| 5.8 | Particle numbers for 8 processors. | 82 |
| 5.9 | Schematic of the experiment reported by [5], dimensions in cm. | 83 |
| 5.10 | Position of pressure sensors on the box [5], dimensions in cm. | 83 |
| 5.11 | Particles are colored by their processors ID at: (a) $t=0$ s, (b) $t=0.5$ s, (c) $t=2$ s. Note that left and right ends are boundary particles (non-moving). | 85 |
| 5.12 | Snapshots of the dam-break on the short box compared with experiment reported by Kleefsman [6]. (a) Current study at $t=0.4$ s (Particles are colored by their velocity at z -direction), (b) Experiment at $t=0.4$ s, reprinted with permission from [6], (c) Current study at $t=0.56$ s (Particles are colored by their velocity at z -direction), (d) Experiment at $t=0.56$ s, reprinted with permission from [6]. The small picture on the experimental figures represent the water behind the gate ([6]). | 86 |
| 5.13 | Fluid particles are colored by their pressure at: (a) $t=0$ s, (b) $t=0.2$ s, (c) $t=0.4$ s, (d) $t=0.5$ s. | 87 |
| 5.14 | Fluid particles are colored by their velocity in x -direction (u) at: (a) $t=0$ s, (b) $t=0.2$ s, (c) $t=0.4$ s, (d) $t=0.5$ s. | 88 |
| 5.15 | Impact pressure on the box face at $t=0.5$ s. | 89 |
| 5.16 | Water height evolution in time. Experimental data and VOF results reported in [5] and previous semi-implicit ISPH results reported in [7]. | 90 |
| 5.17 | Pressure evolution in time. Experimental data and VOF results reported in [5] and previous semi-implicit ISPH results reported in [7]. | 91 |

5.18 Speedup and efficiency. 93

6.1 (a) Description of particle types, (b) Particle representation in SPH, the interface particles are enclosed by the red line. 102

6.2 Vertical displacement of the rigid body against time (Experimental data of Henrich [8], DNS simulation of Abadie et al. [9]). 103

6.3 Free surface elevation against time at (a) $t=0.5$ s (b) $t=1.0$ s by using prescribed velocity and the new proposed method (Experimental data of Henrich [8]). 103

6.4 Schematic of the experiment of Heinrich [8]. 104

6.5 Particle are colored by their velocity in the x-direction (u) at: (a) $t=0.25$ s, (b) $t=0.5$ s, (c) $t=0.75$ s, (d) $t=1.0$ s, (e) $t=1.25$ s, (f) $t=1.5$ s, (g) $t=1.75$ s, (h) $t=2.0$ s. 105

6.6 Vortex formation above the slide, particles velocity vector are colored by their velocity in the x-direction (u) at: (a) $t=0.85$ s, (b) $t=1.35$ s, (c) $t=1.75$ s, (d) $t=2.0$ s. 106

6.7 Schematic of the experiment of Monaghan and Kos [10]. 108

6.8 Particles are colored by their pressure at (a) $t=0.2$ s, (b) $t=0.3$ s, (c) $t=0.4$ s, (d) $t=0.5$ s, (e) $t=0.6$ s, (f) $t=0.75$ s. 108

6.9 Vortex formation and movement, particles are colored by their velocity in the x-direction (u) at (a) $t=0.2$ s, (b) $t=0.4$ s, (c) $t=0.6$ s, (d) $t=0.75$ s. 109

6.10 (a) Definition of parameters in the experiment of Monaghan and Kos [10], (b) Particle representation using current ISPH code at $t=0.285$ s. 109

6.11 Schematic of the experiment of Heller et al. [1]. Dimensions in m. 110

6.12 Particles are colored with their velocities at: (a) $t=0$ s, (b) $t=0.45$ s, (c) $t=1.5$ s, (d) $t=3$ s. 112

6.13 Particle velocity vectors and stream tracers are colored with their horizontal velocities at: (a and b) $t=1.5$ s, (c and d) $t=3$ s, (e and f) $t=4$ s, (g and h) $t=5$ s. 114

6.14 Free surface elevation against time at: (a) $x=1.2$ m, (b) $x=1.8$ m, (c) $x=2.4$ m, (d) $x=3.6$ m, (e) $x=5.8$ m, (f) $x=8.4$ m. 115

6.15 Particles are colored by velocity magnitude at: (a) $t=0$ s, (b) $t=0.5$ s, (c) $t=1.5$ s, (d) $t=3$ s. 116

| | | |
|------|--|-----|
| 6.16 | Free surface elevation against time at: (a) $x=0.72$ m, (b) $x=1.2$ m, (c) $x=1.8$ m, (d) $x=3.6$ m. | 117 |
| 6.17 | Top view of free surface particles colored by velocity in the y-direction at: (a) $t=0.5$ s, (b) $t=1.0$ s, (d) $t=1.5$ s, (f) $t=2$ s. | 118 |
| 6.18 | Top view of free surface particles colored by dimensionless eddy viscosity (ν/ν_0) at: (a) $t=0.5$ s, (b) $t=1.0$ s, (d) $t=1.5$ s, (f) $t=2$ s. | 119 |
| 6.19 | Top view of particles colored by dimensionless eddy viscosity and velocity at $t=1.0$ s for: (a and b) $z=0.02$ m, (c and d) $z=0.11$ m, (e and f) $z=0.22$ m. | 120 |
| 6.20 | Schematic of the experiment of Rzadkiewicz et al. [11]. Dimensions are in m. | 121 |
| 6.21 | Particles are colored with their pressures and horizontal velocities (u) at: (a and b) $t=0.4$ s, (c and d) $t=0.8$ s. | 122 |
| 6.22 | Free surface elevation against time at: (a) $t=0.4$ s, (b) $t=0.8$ s | 123 |
| 6.23 | Schematic of NHC experiment at: a) side view b) top view. Dimensions are in m. | 123 |
| 6.24 | Horizontal and vertical velocities of the slide against time in NHC experiment. | 124 |
| 6.25 | Free surface elevation for water and water–bentonite mixture against time at: (a and b) wave probe 1, (c and d) wave probe 4. | 125 |
| 6.26 | Water bentonite mixture particles are colored by their velocity in the x-direction at: (a) $t=0$ s, (b) $t=0.5$ s, (c) $t=1.0$ s, (d) $t=2.0$ s, (e) $t=3.0$ s, (f) $t=4.0$ s. | 126 |

ACKNOWLEDGEMENTS

I would like to thank my supervisors Dr. Curran Crawford and Dr. Peter Oshkai for their help, support, suggestions and insightful comments throughout my PhD studies.

I would like to thank Dr. Belaid Moa for his support and help especially for the code implementation of the MPI code.

I would like to thank Dr. Bradley Buckham for his support throughout this work.

I would like to thank my parents, my sister and my wife for their love and support throughout this work.

Also, I would like to thank the Natural Sciences and Engineering Research Council of Canada (NSERC), Natural Resources Canada (NRCan) for their financial support and Compute Canada for the computational resources throughout this research.

Chapter 1

Introduction

1.1 Motivation

There are two main approaches for studying fluid structure interaction in numerical simulations; Eulerian and Lagrangian. In the Eulerian approach, the flow quantities are determined at specific spatial locations as the fluid flows through time. In the Lagrangian approach, fluid particle motion is tracked over time as it moves in the flow, i.e. the equations are formulated in a non-inertial reference frame moving with the flow. Eulerian based methods have been thoroughly studied over the last few decades and many finite difference/volume and finite element commercial CFD codes (e.g. Fluent, CFX, etc.) have been developed based on this approach. Eulerian mesh-based methods can be used to solve different types of problems, however they have limitations in problems involving moving boundaries and flow discontinuities, such as fluid structure interactions in the presence of free surface [12] (e.g. applications in ocean and coastal engineering such as wave interacting a wave energy converter, floating breakwaters or any floating object). Numerical treatments, such as moving grids [13] and dynamic over-set grids [14] are proposed to overcome these limitations, but they are computationally expensive and not straightforward to apply [15]. Also, an additional flow quantity tracking technique, such as volume of fluid (VOF) [16] or level set methods [17] is required to accurately determine the free surface position in these methods.

Mesh-less particle methods were developed to solve problems involving free surface breaking and fragmentation for which conventional CFD methods could not easily be applied [12]. Mesh-less particle methods refer to mesh-less methods in which a set

of discrete particles are scattered over the domain and its boundaries [18]. In these methods, there is no connectivity required between particles to solve the PDEs governing the problem, which is an advantage when solving problems involving breaking and fragmentation of the interface [18, 19]. The oldest mesh-less particle method is Smoothed Particle Hydrodynamics (SPH), which was invented in the 1970s to study the astrophysical problems [20, 21]. The unique features of SPH (i.e. Lagrangian and mesh-less) have drawn attention in different engineering fields, especially hydrodynamics. A comprehensive review of the method and its applications is recently reported by Shadloo et al. [22]. SPH was originally developed to study compressible flows with no solid boundaries. Therefore, the method requires modifications for bounded incompressible fluid simulations. In SPH, there are two main methods to determine pressure: weakly compressible SPH (WCSPH) by Monaghan [23] and incompressible SPH (ISPH) by Cummins et al. [24]. WCSPH calculates the pressure by using a stiff equation of state, whereas ISPH solves Poisson’s equation to determine the pressure. The WCSPH approach has been found to exhibit large fluctuations in pressure due to the way that formulation calculates the pressure [25–29]. In order to reduce these fluctuations, several methods have been proposed such as using ISPH method. A common way to solve Poisson’s equation in ISPH method is to use implicit solvers. In this dissertation, the Poisson’s equation is solved explicitly using the method first proposed by Hosseini et al. [30]. Although using an implicit ISPH method would allow a larger time step, but it requires building and solving a sparse matrix at every time step (since position of particles is not fixed). The available explicit ISPH methods use the density invariance for the source term of Poisson’s equation [30, 31]. In this dissertation, the more accurate source term proposed by Khayyer et al. [32] is used for the explicit ISPH method. Also, an eddy viscosity model is added to the method. Adami et al. [33] proposed an accurate boundary condition within the WCSPH approach. This boundary condition is used with the explicit ISPH method in order to avoid boundary conditions requiring many tuning parameters.

The SPH method is typically computationally more expensive than Eulerian-based CFD methods. Therefore, parallelization methods are required to improve the performance of the method, especially for 3D simulations. CPU-based and GPU-based parallelizations are the two main techniques that can be employed for SPH parallelization to improve the performance of the method in simulating computationally

expensive problems. CPU-based parallelization can be applied in SPH by using Open Multi Processing (OpenMP) or Message Passing Interface (MPI) standards. In this dissertation, two novel and easy to implement parallel schemes are developed based on OpenMP and MPI standards. An easy way of OpenMP parallelization is developed for less expensive problems. Also, an efficient and easy MPI-based parallel design is developed for expensive 3D applications. GPU-based parallelization offers optimum hardware acceleration, but it has the issue of fitting all the data in the GPU memory [34]. Therefore, to overcome the memory limitation for computationally expensive problems, multi-GPU systems, which employ the MPI standard for communication between devices, are required. Hence, developing the MPI-based program is a necessary precursor to massively parallel GPU implementation. The developed codes in this dissertation showed a good agreement with the available experimental and previous numerical simulations for several 2D and 3D applications, such as the effect of a transient waves on a deck, wave impacting an Oscillating Wave Surge Converter (OWSC), hydrodynamic loads from the spar of a self-reacting point absorber Wave Energy Converter (WEC) design, dam breaking on short and tall structures, and landslide generated waves. The available code can be easily converted to WCSPH or implicit ISPH. Also, the code is an appropriate base for the future GPU parallelization.

1.2 Dissertation Outline

This dissertation includes five papers which are presented separately in chapters 2–6. These papers have been accepted/submitted in academic Journals or presented in international conferences. Each paper includes its own abstract, introduction, methodology, test cases and conclusions. The chapters 2-6 are outlined as follows:

In chapter 2, a serial 2D code is developed based on an explicit ISPH method with an accurate boundary condition and source term for Poisson’s equation. The purpose of this chapter is to compare the performance of the proposed ISPH method with the available experimental and previous numerical results for simulation of a wave overtopping on a horizontal deck. The time history of the free surface at 5 different locations is compared against experimental and previous numerical results. In addition, the vertical variation of the horizontal velocity at the leading edge of the deck in the presence and absence of the deck, at 4 different time instants is compared

with the experimental data and previous numerical results. The current explicit ISPH study can be compared with analytical solutions in terms of velocity under the wave in a future work.

In Chapter 3, a new Open Multi-Processing (OpenMP)-based parallel SPH code is developed and tested for a wave impacting an Oscillating Wave Surge Converter (OWSC). The performance and accuracy of the new OpenMP parallel SPH code is first reported for the water surge from a dam-break impacting a tall structure, and a wedge water entry problem, prior to the simulation of wave-OWSC interactions. The detailed discussion of the results and performance of the proposed OpenMP code are provided in this chapter.

Chapter 4 presents another application for the OpenMP code developed in chapter 3. In this chapter, an in-house 3D OpenMP explicit ISPH code is used to calculate the hydrodynamic loads from the spar of a 1:25 scale self-reacting point absorber Wave Energy Converter (WEC) design. The simulation results are compared with the available experimental data and detailed discussion of the results are provided in this chapter.

In Chapter 5, a novel parallel design based on a Message Passing Interface (MPI) standard for distributed memory programming is presented for the SPH method. A novel and easy domain decomposition is developed to reach an efficient parallelization. The Peano-Hilbert ordering of the underlying cells is employed to take the advantage of memory locality. Also, a novel dynamic load balancing method is developed in 3D for the parallelization of the SPH method. In addition, an eddy viscosity turbulence model is added to the available explicit ISPH method. The performance of the proposed parallel code was tested for several test cases including a tank full of still water, a column of water collapsing in the middle of a tank, and a dam-break impacting on a short structure.

In Chapter 6, a comprehensive modeling of landslide generated waves is presented by using an in-house parallel explicit ISPH code. A new method is proposed to calculate the motion of a rigid slide on an inclined ramp implicitly, without resorting to a prescribed motion. In this chapter, both subaerial and submarine landslides in 2D and in more realistic 3D applications are simulated by using both rigid and deformable slides. A general Cross rheological model is used to simulate the deformable slide, assuming the slide is non-Newtonian fluid. A landslide case is simulated where a slide is falling into a non-Newtonian reservoir fluid (water-bentonite mixture).

Chapter 7 summarizes the key developments and results from the work, and suggests avenues for continued development of the SPH method and implementation.

1.3 Research Contributions

The contributions of the current dissertation are summarized in the following:

1. **Unique assembly of SPH sub-models**

After a critical appraisal of the various SPH models for incompressibility (pressure calculation), diffusion calculation and boundary conditions in literature, the accurate and efficient numerical ones are brought together under one model. An accurate source term in Poisson's equation is used along with accurate boundary conditions within the explicit ISPH method. The proposed method is used to simulate the effect of a waves overtopping a deck in 2D in chapter 2, the wave-OWSC interactions in chapter 3, and the hydrodynamic loads on a point absorber hull in chapter 4. A turbulence eddy viscosity model is added to the explicit ISPH method. The method is then applied to study the effect of a dam break on a short structure in chapter 5 and a comprehensive set of landslide generated waves modeling in chapter 6.

2. **A new OpenMP-based parallelization scheme for ISPH**

A pure OpenMP-based ISPH parallelization scheme is developed for hydrodynamic applications and its efficiency is determined for two test cases of uniform (e.g. the wave-OWSC interactions) and non-uniform (e.g. dam-break) initial particle distributions. The OpenMP is a standard for implementing the shared memory parallelization by adding directives to parallelize an existing serial code. The proposed method is used to simulate the wave-OWSC interactions, a dam-break on a tall structure and the hydrodynamic loads on a point absorber hull. The OpenMP-based parallelization scheme was published at the Journal of Ocean Engineering and Marine Energy [35] and is presented in chapter 2.

3. **A new MPI-based parallelization scheme for SPH**

A novel MPI standard for distributed memory programming was developed to parallelize the SPH method to achieve the efficiency of several times higher than OpenMP scheme and as a necessary precursor for future multi-GPU implementation. The new scheme has a novel domain decomposition, load balancing and

applies the Peano-Hilbert ordering schemes to order the background cells in SPH. The proposed method is accepted for publication at the Journal of Ocean Engineering and Marine Energy [36] and is presented in chapter 5.

In previous parallel ISPH codes, Poisson's equation is solved implicitly using specific solvers and algorithms where almost half of the total computation time is spent on solving Poisson's equation [37]. In the proposed MPI-based method an explicit solver is used as an alternative to previous implicit solution to Poisson's equation. In SPH, the domain decomposition is performed either by using particle decomposition [37, 38] or spatial decomposition [39]. In the proposed method, a domain decomposition is presented by using both spatial and particle decompositions. The Peano-Hilbert curve is employed in SPH to perform the whole domain decomposition [37, 38]. Our scheme orders only the cells instead of performing the whole domain decomposition and, as such, it is efficient and is performed only when necessary (applied with dynamic load balancing). In the proposed method the dynamic load balancer works as a feedback system that recognizes particle imbalances and applies the load balancing accordingly rather than applying the dynamic load balancing after every fixed n time steps, as done in SPHysics code [40] or by Marrone et al. [41].

4. **A new technique to calculate the motion of a rigid slide on an inclined ramp implicitly**

The previous SPH methods were restoring to a prescribed motion to impose the motion of a rigid slide on an inclined ramp. In this work, a new technique is developed to calculate the motion of a slide on an inclined ramp implicitly without using the prescribed motion in SPH. By using the new technique, the motion of the slide is calculated based on the surrounding fluid. The proposed method is presented in chapter 6.

5. **A comprehensive sub-particle turbulent ISPH analysis of landslide generated waves**

The previous SPH simulations of landslide generated waves were performed by either WCSPH or implicit ISPH method. While the majority of ISPH simulations of landslide generated waves are limited to 2D, here in addition to 2D simulations, a 3D study of landslide generated waves is presented and analyzed by using an explicit ISPH method. The previous 3D simulation of landslide gen-

erated waves, presented in this work, used a WCSPH with artificial viscosity [1], here an ISPH method is applied with sub-particle turbulent model for diffusion. Also, a study of vortex generation for landslide generated waves in SPH are presented and compared with available DNS simulation. A comprehensive set of landslide generated waves simulations is submitted for publication to the Journal of Advances in Water Resources and is presented in chapter 6.

Chapter 2

A Comparison of SPH and RANS Models for Simulation of Wave Overtopping

This paper is presented at the 5th International Conference on Ocean Energy in Halifax, Canada.

Yeylaghi, Shahab, Curran Crawford, Peter Oshkai, and Bradley Buckham. "A comparison of SPH and RANS models for simulation of wave overtopping.", International Conference on Ocean Energy (ICOE) 2014, Halifax, Canada.

This chapter is the first step in validating our in-house explicit ISPH code. The code is used to simulate the effect of a transient wave overtopping a horizontal deck in 2D. In this chapter, The current study of explicit ISPH simulations are compared with the experimental data of Cox and Ortega [2] and numerical results (VOF) of Lu et al. [3].

2.1 Abstract

Fluid-structure interactions associated with wave overtopping are of significance to many engineering applications. Unsteady wave overtopping loads can lead to significant damage or even failure of the structure. Wave overtopping is a challenging

problem to study, since it includes complex phenomenon, such as large deformation of the free surface, air entrainment and turbulence.

Recently, Lagrangian meshless particle methods such as Smoothed Particle Hydrodynamics (SPH) have become an alternative to conventional Eulerian mesh-based methods for studying complex free surface flows. The purpose of the current study is to compare the performance of the explicit incompressible Smoothed Particle Hydrodynamics (SPH) method with experimental data and previous available numerical results for simulation of wave overtopping on a horizontal deck. The free surface profile at different locations and horizontal velocities at the leading edge of the deck, calculated by explicit ISPH method, show good agreement with the experiment and previous numerical study.

2.2 Introduction

Mesh-less methods were developed to solve problems such as free surface flows with breaking and fragmentation for which conventional CFD methods could not easily be applied [12]. In these methods, a set of discrete particles are scattered over the domain and its boundaries [18]. There is no connectivity (grid) required between particles to solve the PDEs governing the problem [18]. The oldest mesh-less particle method is Smoothed Particle Hydrodynamics (SPH), which was invented in 1970s to study the astrophysical problems including compressible, inviscid flows [20, 21]. Using the unique features of SPH (Lagrangian and mesh-less) make it attractive in different engineering fields, especially hydrodynamics. The main advantage of the relatively computationally expensive SPH over RANS solvers is its ability to capture complex free surface flows [7]. SPH has proved to be a successful numerical method in modeling violent free surface flows [42].

Incompressible fluids in SPH can be modeled either by relating the fluid pressure to particle density by using a stiff equation of state, or by solving Poisson's equation to determine the pressure; the two methods are known as weakly compressible SPH (WCSPH) [23] and incompressible SPH (ISPH) [24], respectively. In the ISPH method, a common approach to solve the Poisson's equation is to solve a set of algebraic equations implicitly, but Hosseini et al. [30] proposed an explicit method to solve the Poisson's equation that doesn't require solving a set of algebraic equations at each time step. Eulerian based CFD methods on other hand do not directly link

density variation to pressure. Rather the mass conservation and momentum equations are used to calculate the pressure. WCSPH and ISPH methods are both adept at implicitly including complex free-surfaces, without the need for volume-of-fluid (VOF) or level-set (LS) approaches in RANS. On the other hand, viscosity terms and solid boundary conditions can be more challenging to implement in the SPH methods compared to Eulerian based CFD methods.

The wave overtopping on a horizontal deck is of interest to many engineering applications, such as ship decks, FPSO units [43] and coastal structures [44]. The unsteady loads from the wave overtopping can significantly damage an offshore device [45] and better understanding of wave overtopping loads is needed. More information on wave overtopping can be found in [2, 3].

In this paper, the numerical results of explicit ISPH modeling of wave overtopping on a horizontal deck are compared with the experimental data presented earlier by Cox et al. [2]. The results are also compared to volume-of-fluid (VOF) numerical method available in [3].

2.3 Methodology

In the SPH method, the Lagrangian description of the Navier-Stokes equations are solved for each single fluid particle which represents a finite volume of fluid about a point. The governing equations include the conservation of mass and momentum equations:

$$\frac{1}{\rho} \frac{D\rho}{Dt} + \nabla \cdot \mathbf{u} = 0 \quad (2.3.1)$$

$$\frac{D\mathbf{u}}{Dt} = -\frac{1}{\rho} \nabla p + \mathbf{g} + \nu \nabla^2 \mathbf{u} \quad (2.3.2)$$

where ρ is the fluid particle density, \mathbf{u} is the particle velocity vector, t is time, p is the particle pressure, \mathbf{g} is the gravitational acceleration vector and ν is the kinematic viscosity. In this method, the domain and its boundaries are represented by discrete volumes of fluid (particles), in which each particle carries material properties, such as velocity, density and viscosity. The flow quantities are interpolated over the predefined neighboring particles using a smoothing function (kernel). The kernel determines the

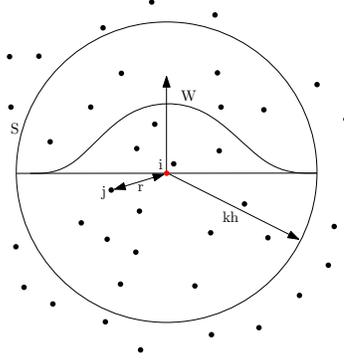


Figure 2.1: Sketch of the kernel support domain (S) and neighboring particles.

contribution of neighboring particles on the particle of interest's property. The kernel approximation of $A(\mathbf{r})$ in SPH is written as:

$$A(\mathbf{r}) \approx \int_{\Omega} A(\mathbf{x})W(\mathbf{r} - \mathbf{x}, h)d\mathbf{x} \quad (2.3.3)$$

where \mathbf{r} is the location vector, $A(\mathbf{r})$ is the function of interest at location \mathbf{r} , W is smoothing function or kernel defined over domain of interest Ω , and h is the smoothing length which defines the kernel radius (Figure 2.1). The particle approximation of the above integral is obtained by replacing the integral with summation (finite number of particles) as:

$$A_i = \sum_{j=1}^N A_j W(\mathbf{r}_i - \mathbf{r}_j, h) \frac{m_j}{\rho_j} \quad (2.3.4)$$

where m_j is the mass of particle j in the neighborhood of particle i and ρ_j is the density of particle j (Figure 2.1). In this study, the fifth order Wendland kernel [46] is used and smoothing length is set to $h = 1.2dr$, where dr is the interparticle distance (Figure 2.1). Also, the following formulations are adopted for the first order and second order spatial derivatives.

The first order derivative is defined as: [47]:

$$(\nabla A)_i = \rho_i \sum_j m_j \left(\frac{A_j}{\rho_j^2} + \frac{A_i}{\rho_i^2} \right) \nabla_i W_{ij} \quad (2.3.5)$$

The second order derivative adopted here is [24, 30]:

$$\nabla \cdot \left(\frac{1}{\rho} \nabla A \right)_i = \sum_{j=1}^N \left(\frac{8m_j}{(\rho_i + \rho_j)^2} \frac{A_{ij} \mathbf{r}_{ij} \cdot \nabla_i W_{ij}}{r_{ij}^2 + \eta^2} \right) \quad (2.3.6)$$

where $\mathbf{r}_{ij} = \mathbf{r}_i - \mathbf{r}_j$, $A_{ij} = A_i - A_j$ and $\nabla_i W_{ij}$ is the gradient of the kernel function at particle i .

The explicit ISPH method [30] is used for this study. This approach is based on the two step projection method that is widely used in Eulerian based methods [24]. In the prediction step, the intermediate velocity is calculated using the viscous and body forces, without pressure force:

$$\mathbf{u}^* = \mathbf{u}(t) + \Delta t(\mathbf{g} + \nu \nabla^2 \mathbf{u}) \quad (2.3.7)$$

$$\mathbf{r}^* = \mathbf{r}(t) + \Delta t \mathbf{u}^* \quad (2.3.8)$$

where \mathbf{u}^* is the intermediate velocity and \mathbf{r}^* is the intermediate position. In the correction step, the incompressibility condition is achieved by correcting the velocity using the pressure force as:

$$\mathbf{u}(t + \Delta t) = \mathbf{u}^* + \Delta t \left(-\frac{1}{\rho} \nabla p \right) \quad (2.3.9)$$

By taking the divergence of equation 2.3.9 and forcing $\nabla \cdot \mathbf{u}(t + \Delta t) = 0$, one can obtain the pressure Poisson's equation which should be solved for pressure at each time step.

$$\nabla \cdot \left(\frac{\nabla p}{\rho} \right) = \left(\frac{\nabla \cdot \mathbf{u}^*}{\Delta t} \right) \quad (2.3.10)$$

By using equation 2.3.6, the Poisson's equation 2.3.10 in particle form can be written as:

$$\sum_{j=1}^N 8m_j \left(\frac{1}{\rho_i + \rho_j} \right)^2 \frac{(p_{ij} \mathbf{r}_{ij}) \cdot \nabla_i W_{ij}}{r_{ij}^2 + \eta^2} = \frac{-1}{\Delta t} \sum_{j=1}^N V_j \mathbf{u}_{ij}^* \cdot \nabla_i W_{ij} \quad (2.3.11)$$

where $p_{ij} = p_i - p_j$, $\mathbf{r}_{ij} = \mathbf{r}_i - \mathbf{r}_j$ and η is a small number. The standard approach to solve the Poisson's equation pressure is an implicit approach. In the implicit approach, at each time step, a set of algebraic equations is solved to find the pressure: i.e. equation 2.3.11 will be a system of equations such as, $Ap = b$, where $p_{ij} = p_i^{n+1} - p_j^{n+1}$

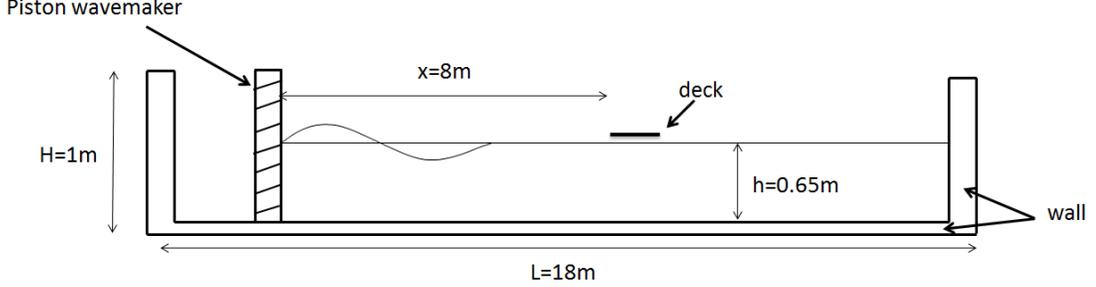


Figure 2.2: Numerical wave tank.

(n is the current time step), b is the right-hand side vector and A is the coefficient matrix for each particle.

In this paper, the explicit approach [30] is adapted to solve the Poisson's equation. In the explicit approach, at time $n + 1$ the pressure for particle i ; p_i^{n+1} , is calculated based on the pressure at neighboring particles at time n ; p_j^n (i.e. in equation 2.3.11, $p_{ij} = p_i^{n+1} - p_j^n$ and the only unknown will be p_j^{n+1}). By using the explicit approach, solving a set of algebraic equations is not required in each time step; however, a smaller time step must be adopted for accuracy.

2.4 Results

The important parameters of Cox's experiment [2] are summarized herein-more details can be found in [2]. The experiment includes a piston wavemaker and the still water depth is 0.65 m. The horizontal deck is 0.61 m long and 0.0115 m thick. The leading edge of the deck is located at $x = 8$ m from the wavemaker and at $y = 0.0525$ m above still water. The wavemaker signals consists two waves of period $T = 1.0$ s followed by two and a half waves of period $T = 1.5$ s [2]. In order to save computational costs, dimensions of the numerical wave tank are set to $L = 18$ m and $H = 1$ m, Figure 2.2. The fluid particle dimensions are set to $dx = 0.0125$ m and $dy = 0.0125$ m which leads to 79932 total particles. Fixed dummy particles are used for the boundary particles as in [30]. A damping zone used in [48] was adopted in this work at the end of the wave tank in order to absorb the wave reflection. The time step is set to $\Delta t = 0.0005$ s for numerical simulations.

The explicit ISPH simulation of the numerical wave tank for the case with deck at $t = 5.7$ s and $t = 10.7$ s are shown in Figure 2.3. The particles are colored for each

case by their pressures and velocities, respectively.

The numerical free surface positions at five locations for the case without the deck are calculated based on the explicit ISPH method and shown in Figure 2.4. It is shown that the time histories of the free surface at these locations are in good agreement with experimental measurements [2] and available numerical data [3].

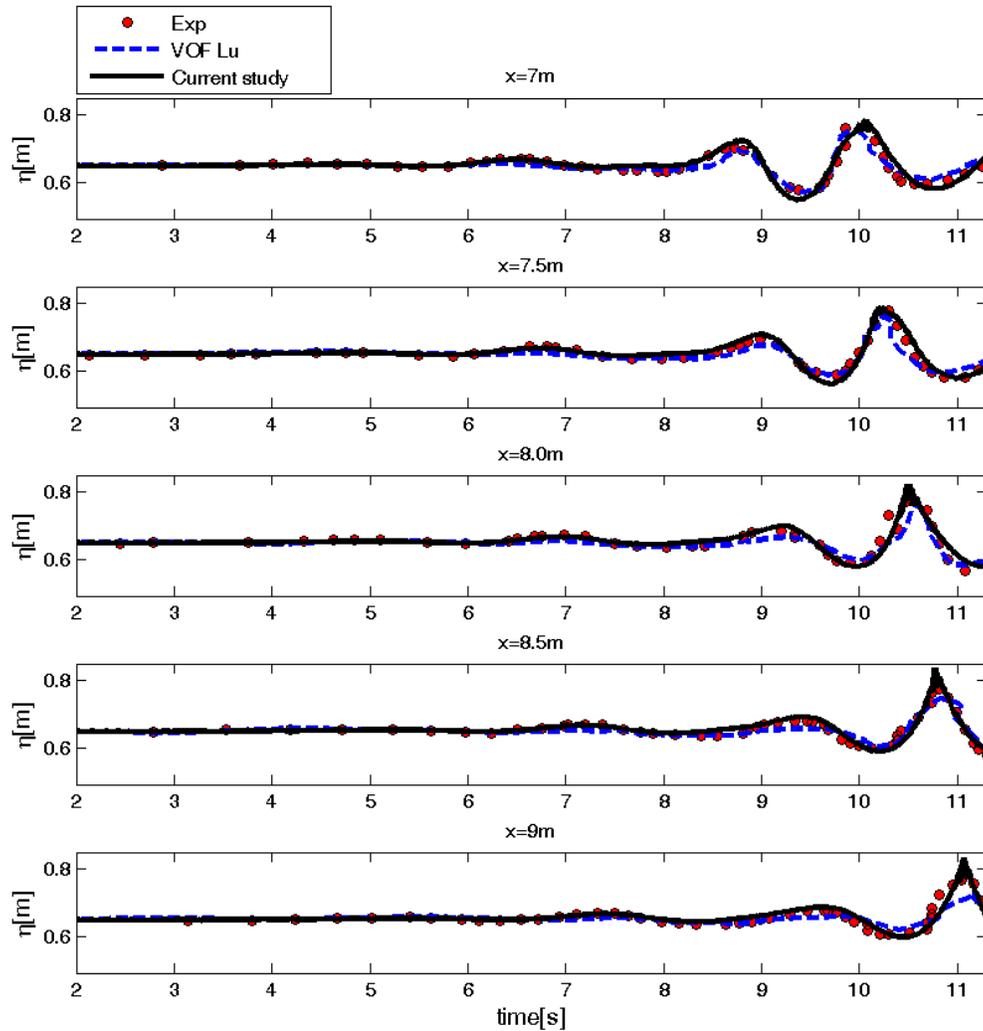


Figure 2.4: Time history of free surface at different locations obtained from explicit ISPH and plotted vs experimental data [2] and numerical results [3].

Figure 2.5a shows the vertical variation of the horizontal velocity in the absence of the deck at four time steps compared with the experimental measurements [2] and previous numerical [3] data. In order to extract the horizontal velocity, the

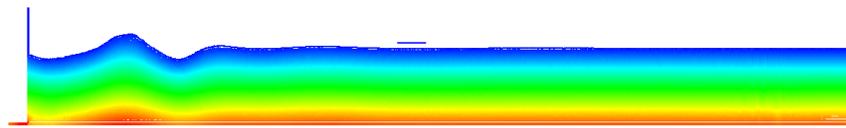
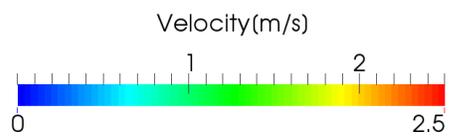
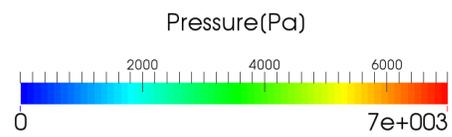
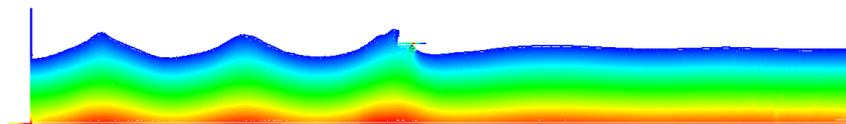
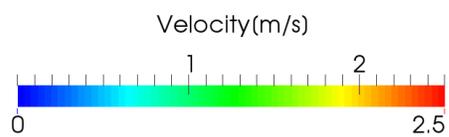
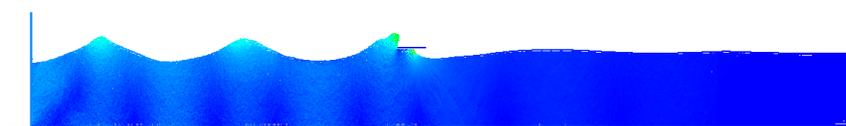
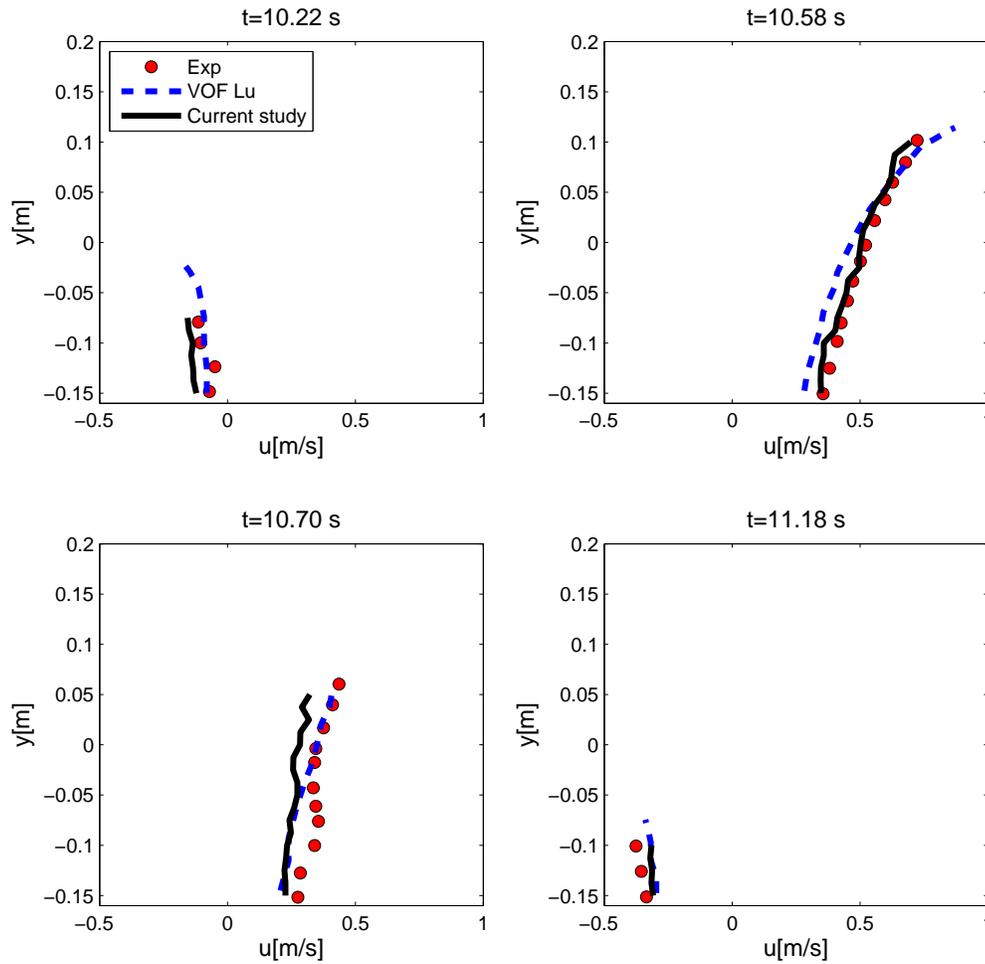
(a) pressure at $t = 5.7$ s(b) velocity at $t = 5.7$ s(c) pressure at $t = 10.7$ s(d) velocity at $t = 10.7$ s

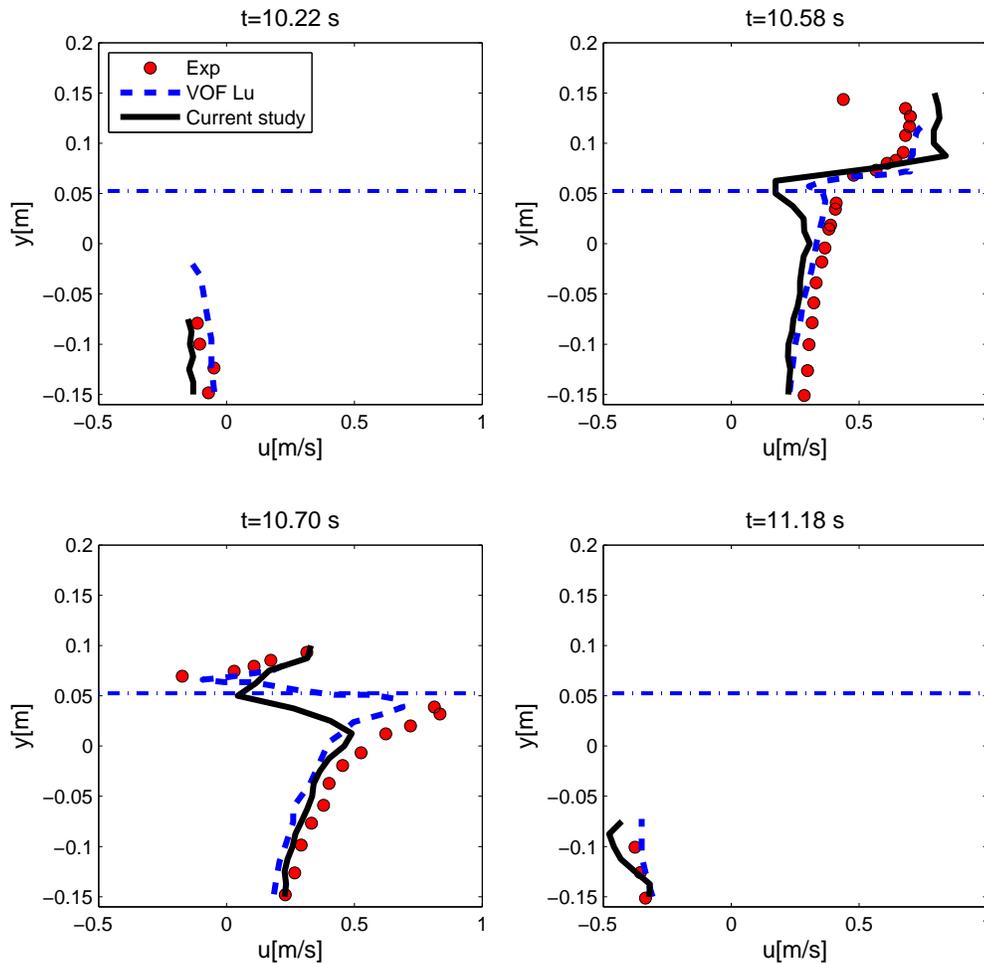
Figure 2.3: Explicit ISPH simulation of numerical wave tank with deck.

interpolation points were added at the leading edge of the deck, $x = 8 \text{ m}$, and the Wendland kernel [46] was applied in order to obtain the velocities at those points.



(a) without deck

Figure 2.5: Vertical variation of the horizontal velocity u (m/s) at leading edge of the deck, $x = 8 \text{ m}$. (Experimental data from [2] and VOF results from [3])



(b) with deck

Figure 2.5: Vertical variation of the horizontal velocity u (m/s) at leading edge of the deck, $x = 8$ m. (Experimental data from [2] and VOF results from [3])

Figure 2.5b shows the vertical variation of the horizontal velocity in the presence of the deck at four time steps compared with previous numerical data in [3] and the experimental measurements in [2]. At $t = 10.22$ s the wave hasn't reached the deck and the horizontal velocity is close to the case without the deck. At $t = 10.58$ s and $t = 10.7$ s the wave has already reached to the deck and the separation in the velocity is clear. At $t = 10.58$ s the maximum velocity occurs above the deck while at $t = 10.7$ s maximum occurs below the deck. The horizontal velocity below the

deck at $t = 10.58$ s is smaller than the case without deck while this is vice versa at $t = 10.7$ s. At $t = 11.18$ s wave has passed the deck and the velocities are close to the values without the deck.

2.5 Conclusion

An explicit Incompressible Smoothed Particle Hydrodynamics (ISPH) method was applied to model the interaction of two waves with different amplitudes and frequencies and a fixed deck. This method takes advantage of both WCSPH and ISPH methods. It was shown that numerical simulations are in good agreement with experimental data and previous numerical results. This study is part of the future work in simulation of the wave energy converter under wave spectrum.

2.6 Acknowledgment

The authors thank the Natural Sciences and Engineering Research Council, Natural Resources Canada, the Pacific Institute for Climate Solutions and the University of Victoria for their financial support.

Chapter 3

ISPH Modelling of an Oscillating Wave Surge Converter Using an OpenMP-Based Parallel Approach

This paper was published at the Journal of Ocean Engineering and Marine Energy.

Yeylaghi, Shahab, Belaid Moa, Peter Oshkai, Bradley Buckham, and Curran Crawford. "ISPH modelling of an oscillating wave surge converter using an OpenMP-based parallel approach." *Journal of Ocean Engineering and Marine Energy* (2016): 1-12. [<http://dx.doi.org/10.1007/s40722-016-0053-7>].

This chapter presents an OpenMP-based parallel approach for the SPH method. The parallel code is used to simulate the impact of a wave on an OWSC. The experimental data for the OWSC experiment is provided by Resolute Marine Energy. In order to test the accuracy and performance of the current parallel explicit ISPH simulations, two test cases including a dam breaking on a tall structure and a symmetric wedge impacting still water are compared with the available experimental data and previously available numerical simulations.

3.1 Abstract

Fluid-structure interactions occurring between a wave train and an Oscillating Wave Surge Converter (OWSC) are studied in this paper using Smoothed Particle Hydrodynamics (SPH). SPH is an alternative numerical method to conventional Computational Fluid Dynamics (CFD) for studying complex free surface flows. A new Open Multi-Processing (OpenMP) based parallel SPH code is developed and tested on a wave impacting an OWSC. An incompressible SPH (ISPH) method is implemented here to avoid spurious pressure oscillations, and an OpenMP approach is employed due to its relative ease of coding. The simulation results show good agreement with the experimental data. The performance of the new parallel SPH code is also reported for the water surge from a canonical dam break impinging on a tall square structure.

3.2 Introduction

Meshless methods date back almost four decades. The oldest meshless particle method is Smoothed Particle Hydrodynamics (SPH), which was developed in the 1970s to study compressible inviscid problems in astrophysics by Gingold and Monaghan [21] and Lucy [20]. SPH is of interest to hydrodynamic problems due to its Lagrangian and meshless characteristics. The interactions between waves and Wave Energy Converters (WECs), commonly lead to a large motion of the WEC. In order to model these large motions, conventional CFD methods require expensive and complicated grid moving algorithms. Particle based SPH is favored for simulating flows with large motion since a grid is not required when solving the Lagrangian formulation of the Navier-Stokes equations. The free surface is captured implicitly in the SPH method without the need for solving an additional equation, such as volume-of-fluid (VOF) or level-set (LS) approaches in conventional CFD methods. On the other hand, the relatively expensive SPH method was developed to study compressible inviscid flows (astrophysics), therefore, modifications are necessary for hydrodynamic applications. SPH has been applied to a diverse range of free surface flows from wave propagation in [23], wave breaking in [49] to dam break problems in [50] and fluid-structure interactions in [51] and [52]. See [42] for a review of SPH applications in free surface flows.

The pitching flap Wave Energy Converter (WEC) hinged close to the bottom of the ocean is known as an Oscillating Wave Surge Converter (OWSC) [53]. It is mentioned by Babarit et al. [53] and Folley et al. [54] that OWSCs are designed for shallow waters in order to attain higher horizontal velocities and pitching motions. Several experimental, analytical and numerical studies of OWSCs have been reported. The experimental studies of OWSCs are mostly performed for scaled model devices in the wave tank. Henry et al. [51] reported the experimental study of a 1/25th-scale OWSC model along with numerical simulations, using both OpenFOAM (conventional CFD) and a SPH method. The time histories of the OWSC rotation angle between the three methods showed good agreement. The time evolution of pressure on two sensors from the SPH simulation was compared with the experimental data and good agreement was achieved. Henry et al. [55] performed two dimensional experiments on an 1/40th-scale OWSC model and compared the results with numerical simulations. It was concluded that the slamming of the model is related to the classic wedge water entry problem [56, 57]. Clabby and Tease [58] performed a series of experiments to explore the extreme events related to a 1/20th-scale OWSC model. It was reported that the extreme pressure occurs during the breaking waves or re-entry slamming of the flap.

It is worth pointing out that the experimental studies are extremely important to study OWSC devices due to complex phenomena involved and to validate numerical simulations. However, it is costly to perform parametric studies, such as changing flap size, wave conditions and tank dimensions in the experimental campaigns. Therefore, numerical simulations are also extremely important to efficiently design the experiments. Potential flow methods have been extensively applied to ocean engineering problems. Although these methods are restricted to solving linear inviscid equations, they provide valuable insight for the problem in a reasonable time. Studies based on potential methods applied to OWSCs can be found in [59] and [60, 61]. OWSCs are designed for shallow waters, hence they may experience extreme wave loads as mentioned in [51, 62]. Also, the interactions between wave and OWSC may include complex phenomena such as slamming, wave over-topping, air entrainment and turbulence [62]. Therefore, potential flow methods have limitations in capturing the details, especially the nonlinearities, involved in the interactions of OWSCs and shallow waters.

Computational Fluid Dynamics (CFD) simulation using the full Navier-Stokes equations is an alternative approach to potential methods. It provides more accurate

description of the whole flow including the interactions between waves and OWSC. However, Navier-Stokes solvers are computationally more expensive than potential flow methods. Wei et al. [62] performed numerical simulations using ANSYS FLUENT to investigate the viscous and scaling effects on OWSCs. The time histories of different pressure sensors, rotation angle and vortex shedding from the wave impact for two wave conditions were compared with the experimental data. It was concluded that the viscous effects are negligible for wide flaps. Schmitt and Elsaesser [63] proposed a new moving mesh approach for OWSCs using OpenFOAM and compared simulations results with experiment for regular and irregular waves.

Rafiee and Dias [64] performed 2D and 3D simulations of wave interactions with OWSC using SPH. The $k - \epsilon$ turbulence model was used along with the SPH method in order to study the effects of wave loads on the OWSCs. It was concluded that 3D simulations provide more accurate estimation of the pressure peaks and angle of rotations compared to the 2D simulations.

In this paper, we report our work on wave interaction with an OWSC device. A custom SPH implemented using parallel computing and an incompressible formulation of the governing equations. The methodology we followed and the parallel scheme used to implement a new OpenMP SPH are discussed in Sections 3.3.1 and 3.4 respectively. A classical wedge water entry problem is presented in Section 3.5.2 as a slamming benchmark test case. The experimental setup simulated is described in Section 3.5.3.1. The numerical results of the simulations are compared with the available experimental data. The performance of the new parallel SPH code is also reported for a dam break on a tall square structure (Section 3.5.1).

3.3 Methodology

3.3.1 Governing Equations

In the SPH method, the Navier-Stokes equations are solved in a Lagrangian formulation, thus they read:

$$\frac{D\mathbf{u}}{Dt} = -\frac{1}{\rho}\nabla p + \mathbf{g} + \nu\nabla^2\mathbf{u}, \quad (3.3.1)$$

where ρ is the fluid particle density, \mathbf{u} is the particle velocity vector, t is time, p is the particle pressure, \mathbf{g} is the gravitational acceleration vector and ν is the kinematic viscosity (it should be mentioned that the bold parameters represent vector forms in

this paper). In SPH, a general function is approximated as [47]

$$A(\mathbf{r}) \approx \int A(\mathbf{x})W(\mathbf{r} - \mathbf{x}, h)dx, \quad (3.3.2)$$

where $A(\mathbf{r})$ is the function of interest at position \mathbf{r} , $W(\mathbf{r} - \mathbf{x}, h)$ is the smoothing or kernel function, \mathbf{r} is the position vector, h is the smoothing length by which neighboring particles are defined. The particle approximation of the above integral is written by replacing the integral with a summation as

$$A_i \approx \sum_{j=1}^N A_j W(\mathbf{r} - \mathbf{r}_j, h) V_j, \quad (3.3.3)$$

where N is the number of neighboring particles and V_j is the volume of particle j (particle j is a neighbor to i). In this paper, the fifth order Wendland kernel proposed by Wendland [46] is used:

$$W(q) = W_c \begin{cases} (1 + 2q)(1 - \frac{q}{2})^4 & 0 \leq q \leq 2 \\ 0 & q \geq 2, \end{cases} \quad (3.3.4)$$

where $W_c = 21/16\pi h^3$ in 3D and $q = |\mathbf{r}_i - \mathbf{r}_j|/h$. The smoothing length is set to $h = 1.5\Delta r$ for 3D cases, where Δr is the initial particle spacing. The gradient operator is used from [47] as

$$\left(\frac{1}{\rho}\nabla A\right)_i = \sum_j m_j \left(\frac{A_i}{\rho_i^2} + \frac{A_j}{\rho_j^2}\right) \nabla_i W_{ij}. \quad (3.3.5)$$

In this paper, the Laplacian operator is adopted from [65] as

$$\nabla \cdot \left(\frac{\nabla A}{\rho}\right)_i = \sum_j \frac{8m_j}{(\rho_i + \rho_j)^2} \frac{A_{ij} \mathbf{r}_{ij} \cdot \nabla_i W_{ij}}{r_{ij}^2 + \eta^2}, \quad (3.3.6)$$

where $A_{ij} = A_i - A_j$, $\eta = 0.1h$ and $\mathbf{r}_{ij} = \mathbf{r}_i - \mathbf{r}_j$. In SPH, there are two main approaches to calculate pressure: weakly compressible SPH (WCSPH) and incompressible SPH (ISPH). WCSPH is the original SPH method applied to simulate free surface flows. In this approach, the fluid is considered to be weakly compressible and the equation of state is used to calculate pressure. A promising alternative to

WCSPH is incompressible SPH (ISPH) proposed by [24]. This approach is based on a two step projection method widely used in Eulerian based CFD methods [66]. In this approach, in the prediction step the intermediate velocity (\mathbf{u}^*) is calculated using the viscous and body forces, without pressure forces as

$$\mathbf{u}_i^* = \mathbf{u}_i(t) + \Delta t(\mathbf{g} + \nu \nabla^2 \mathbf{u}_i). \quad (3.3.7)$$

Subsequently, the intermediate position is calculated by

$$\mathbf{r}_i^* = \mathbf{r}_i(t) + \Delta t \mathbf{u}_i^*. \quad (3.3.8)$$

In the ISPH method, Poisson's equation is solved for pressure at each time step

$$\nabla \cdot \left(\frac{\nabla p}{\rho} \right)_i = \frac{\nabla \cdot \mathbf{u}_i^*}{\Delta t}. \quad (3.3.9)$$

The left hand side of Poisson's equation is discretized using equation 3.3.6 as

$$\nabla \cdot \left(\frac{\nabla p}{\rho} \right)_i = \sum_j \frac{8m_j}{(\rho_i + \rho_j)^2} \frac{p_{ij} \mathbf{r}_{ij} \cdot \nabla_i W_{ij}}{r_{ij}^2 + \eta^2}, \quad (3.3.10)$$

and for the right hand side, the term proposed by Khayyer et al. [67] is implemented as

$$\nabla \cdot \mathbf{u}_i^* = \sum_j V_j (\mathbf{u}_j^* - \mathbf{u}_i^*) \nabla_i W_{ij}. \quad (3.3.11)$$

Hence, Poisson's equation is written as

$$\sum_j \frac{8m_j}{(\rho_i + \rho_j)^2} \frac{p_{ij} \mathbf{r}_{ij} \cdot \nabla_i W_{ij}}{r_{ij}^2 + \eta^2} = \frac{-1}{\Delta t} \sum_j V_j (\mathbf{u}_i^* - \mathbf{u}_j^*) \nabla_i W_{ij}. \quad (3.3.12)$$

Poisson's equation is then solved explicitly using the same procedure proposed by Hosseini et al. [30]. Hence, the pressure for a fluid particle, p_i , is obtained by

$$p_i = \frac{\sum_j A_{ij} p_j + RHS_i}{\sum_j A_{ij}}, \quad (3.3.13)$$

where,

$$RHS_i = \frac{-1}{\Delta t} \sum_j \frac{m_j}{\rho_j} (\mathbf{u}_i^* - \mathbf{u}_j^*) \nabla_i W_{ij}, \quad (3.3.14)$$

$$A_{ij} = \sum_j \frac{8m_j}{(\rho_i + \rho_j)^2} \frac{\mathbf{r}_{ij} \cdot \nabla_i W_{ij}}{r_{ij}^2 + \eta^2}. \quad (3.3.15)$$

In the correction step, the incompressibility is achieved by correcting the velocity using the pressure force as

$$\mathbf{u}_i(t + \Delta t) = \mathbf{u}_i^* + \Delta t \left(-\frac{1}{\rho} \nabla p_i \right). \quad (3.3.16)$$

The positions are updated at the end of each time step as

$$\mathbf{r}_i(t + \Delta t) = \mathbf{r}_i(t) + \Delta t \left(\frac{\mathbf{u}_i(t + \Delta t) + \mathbf{u}_i(t)}{2} \right). \quad (3.3.17)$$

SPH is a general 3D method, but the scale OWSC model is constrained only to pitch motion (one degree of freedom (DOF)). To calculate the motions and forces in the fluid-structure interaction, the rigid body equations are solved for one DOF (pitch motion) along with the Navier-Stokes equations in particle form as

$$\frac{d\mathbf{U}}{dt} = \frac{\sum_i \mathbf{f}_i}{M_{body}} + \mathbf{g}, \quad (3.3.18)$$

$$\frac{d\omega}{dt} = \frac{\sum_i \tau_i}{I_{body}}, \quad (3.3.19)$$

$$\tau_i = (\mathbf{r}_i - \mathbf{R}) \times \mathbf{f}_i, \quad (3.3.20)$$

where M_{body} is the mass of the rigid-body, I_{body} is the moment of inertia about the fixed rotation axis, \mathbf{R} is the position of center of mass for the rigid-body, \mathbf{U} is the transitional velocity, ω is the rotational velocity, $\sum_i \tau_i$ is the sum of the moments on the rigid-body particles and $\sum_i \mathbf{f}_i$ is the sum of the forces on the rigid-body particles which is calculated based on the surrounding fluid particles. Subsequently, the velocity of the rigid-body particles are obtained as

$$\mathbf{u}_i = \mathbf{U} + \omega \times (\mathbf{r}_i - \mathbf{R}). \quad (3.3.21)$$

The positions of the rigid-body particles are updated by using equation 3.3.17.

3.3.2 Boundary Conditions

Applying solid boundary conditions is the most challenging task in the SPH method. The three main methods reported to simulate solid boundaries in SPH are: repulsive boundary particles [23, 68], dummy boundary particles [69, 70] and ghost boundary particles [26]. In the repulsive boundary particles approach, a single line of boundary particles are placed on the edge of the solid boundary, exerting a repulsive force on the fluid particles approaching them. In the dummy boundary particles approach, several layers of particles are placed on the edge and inside the solid boundary. In the ghost boundary particles approach, the position of ghost particles is determined by reflection of the fluid particles position through the solid boundary. The pressure of the ghost particles are the same as their corresponding fluid particles (in the presence of the gravity, there will be an additional hydrostatic pressure). Each of these methods has advantages and drawbacks in terms of accuracy and computational complexity.

In this paper, fixed dummy particles are used both for solid boundary particles on tank walls and on the OWSC flap. The dummy particles have the advantage of being easy to implement, especially in a parallel SPH. In the current work, we used the method described by Adami et al. [33] to calculate the pressure for boundary particles from the surrounding fluid particles as

$$p_s = \frac{\sum_i p_i W_{si} + (\mathbf{g} - \mathbf{a}_s) \sum_i \rho_i \mathbf{r}_{si} W_{si}}{\sum_i W_{si}}, \quad (3.3.22)$$

where p_s is the pressure for a solid boundary particle, i is the index for the surrounding fluid particle and a_s is acceleration of the wall.

The free surface particles are identified to apply the Dirichlet boundary condition ($p = 0$) in Poisson's equation. In the ISPH method the density for each particle is constant. However, in order to determine free surface particles a fake density (ρ_f) is calculate for each fluid particle as

$$\rho_f(i) = \sum_j m_j W_{ij}. \quad (3.3.23)$$

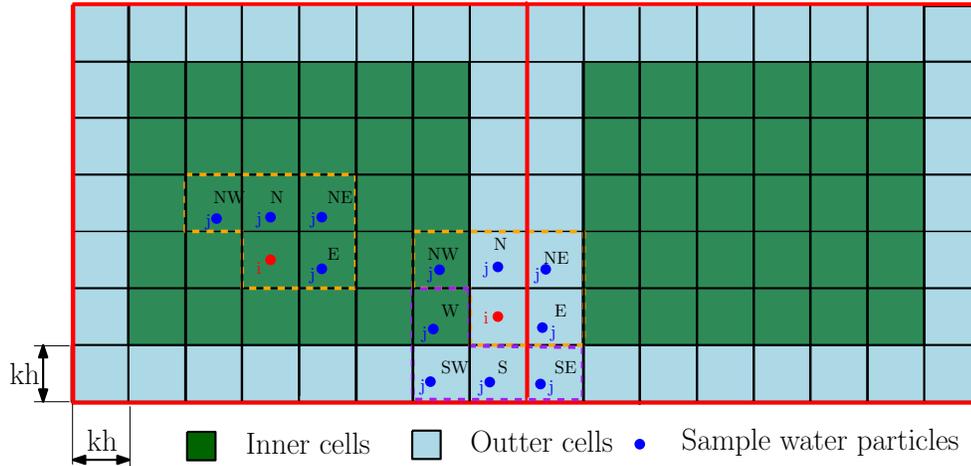


Figure 3.1: Two-dimensional underlying grid for the link list algorithm. Cells enclosed by red contours are handled by the same thread.

ρ_f is not involved in solving the governing equations and is only used to determine the free surface particles. The criteria used for defining a free surface particle is

$$\rho_f \leq 0.9\rho_0, \quad (3.3.24)$$

where $\rho_0 = 1000 \text{ kg/m}^3$. The particles that meet the above criteria are considered to be free surface particles.

3.4 Parallelization Scheme

The SPH method is typically computationally more expensive than Eulerian-based CFD methods. Therefore, parallelization methods are required to improve the performance of the method, especially for 3D simulations. CPU-based and GPU-based parallelizations are the two main techniques that can be employed for SPH parallelization [71]. The CPU-based parallelization is divided into shared-memory and distributed memory parallelizations. The shared-memory approach assumes that the processing units share a common memory (as is the case for multi-core processors) that the parallel tasks can use to communicate and share variables with each other. The thread model is usually used when implementing a shared memory parallelization. More specifically, OpenMP, a standard for implementing the thread model by

adding directives to the code, is a relatively easy way to parallelize an existing serial code. The distributed memory method uses the common memory assumption and requires the parallel tasks to communicate by exchanging messages. MPI (Message Passing Interface) is a standard for distributed memory parallelization. GPU-based parallelization relies on GPUs to schedule and execute the parallel tasks. CUDA, openCL and openACC are the common programming standards for GPU-based implementations. Several approaches have been applied to parallelize the SPH method using these standards. Ferrari et al. [72] proposed a parallelization schemes using the MPI standard to study free surface flows. Marrone et al. [41] studied ship wave breaking patterns using a 3D hybrid MPI and OpenMP standards. A review of CPU-based parallelization implementations for the SPH method in free surface flows is available by Gomez-Gesteira et al. [73]. GPUs have been applied to SPH methods recently. A review of GPU-based parallelization implementations for the SPH method is available by Crespo et al. [74].

The SPH method is both Lagrangian and meshless. Although these two features are attractive in modelling complex free surface flows, they cause difficulties in parallelization scheme [41]. Unlike the fixed grids in CFD mesh-based methods, particles move due to the Lagrangian nature of the method and the neighboring particles do not remain the same throughout the simulation. Hence, as mentioned by Marrone et al. [41], the parallel scheme applied to the SPH method must take into account this specific characteristic.

To save computational costs in SPH, only the contribution of neighboring particles ($r_{ij} \leq kh$) are calculated in the simulation. The link list searching algorithm reported in [73] is adopted here to search for the neighboring particles. In this algorithm, the computational domain is divided into square cells of side kh (kernel radius). The particle in each cell only interacts with particles in neighboring cells; 8 cells in 2D are shown in Figure 3.1. The sweep of the link list search starts from the lower left end and in each sweep, only the E, N, NW, NE cells are involved in order to prevent repeating particle interactions (4 cells out of 8 neighboring cells [73]). The same procedure will be applied in 3D; interactions of 13 cells out of 26 neighboring cells will be calculated. In the current work, we take the advantage of this approach in order to parallelize the code using an OpenMP standard.

Due to the Lagrangian nature of the method special treatments are required at the particles along the processor domain boundaries. These particles may require

information from the neighboring particles located on an other processor domain. This is handled by introducing ghost cells by Gomez-Gesteira et al. [73] or buffer particles by Marrone et al. [41]. In this paper, the domain decomposition is performed spatially in 3D as shown in Figure 3.1 for a 2D case (for simplicity) but the same applies for the 3D case. Here, we divide the cells in each thread to be the inner cells and the outer cells. The last cell in each thread is assigned to be the outer cell. The outer cells are available for both threads. The domain decomposition is performed in this way in order to avoid two or more parallel threads have access to the same data. Since each thread will update its particles, we need to make sure that the other thread has access to the old values instead of the new ones.

For the inner cells, the same procedure that was performed for a single core is performed. The sweep starts from the lower left end and in each sweep, only the E, N, NW, NE cells are involved as shown in Figure 3.1.

For the outer cells, the interaction between particles is considered in all 8 neighboring cells in 2D and 26 cells in 3D. The first sweep is performed on E, N, NW, NE cells (yellow sweep in Figure 3.1). If the cells are within the same thread then both particles i and j will be updated. If one of the cells is from the adjacent thread only particle i will be updated. The second sweep is performed on W, S, SW, SE cells (pink sweep in Figure 3.1). If cells belong to the same thread, none of the particles i or j are updated, but if they belong to the adjacent thread only particle i will be updated. By using this approach, the same procedure of finding neighboring particles is applied for parallelization without introducing any ghost cells or buffer particles.

3.5 Test Cases

3.5.1 Test case 1: Dam Break on a Structure

Dam-break problems are typically used as benchmark test cases for SPH codes. In this paper, the dam-break on a tall structure is first simulated to test the performance of the new OpenMP SPH code. The dam-break benchmark studies are important in order to investigate the influence of severe flooding events such as tsunamis on the shoreline structures. The experimental set up of Yeh and Petroff reported by Gesteira and Dalrymple [4] is used to validate the parallel OpenMP SPH code. Dimensions of the experiment and the tall square structure are shown in Figure 3.2. A layer

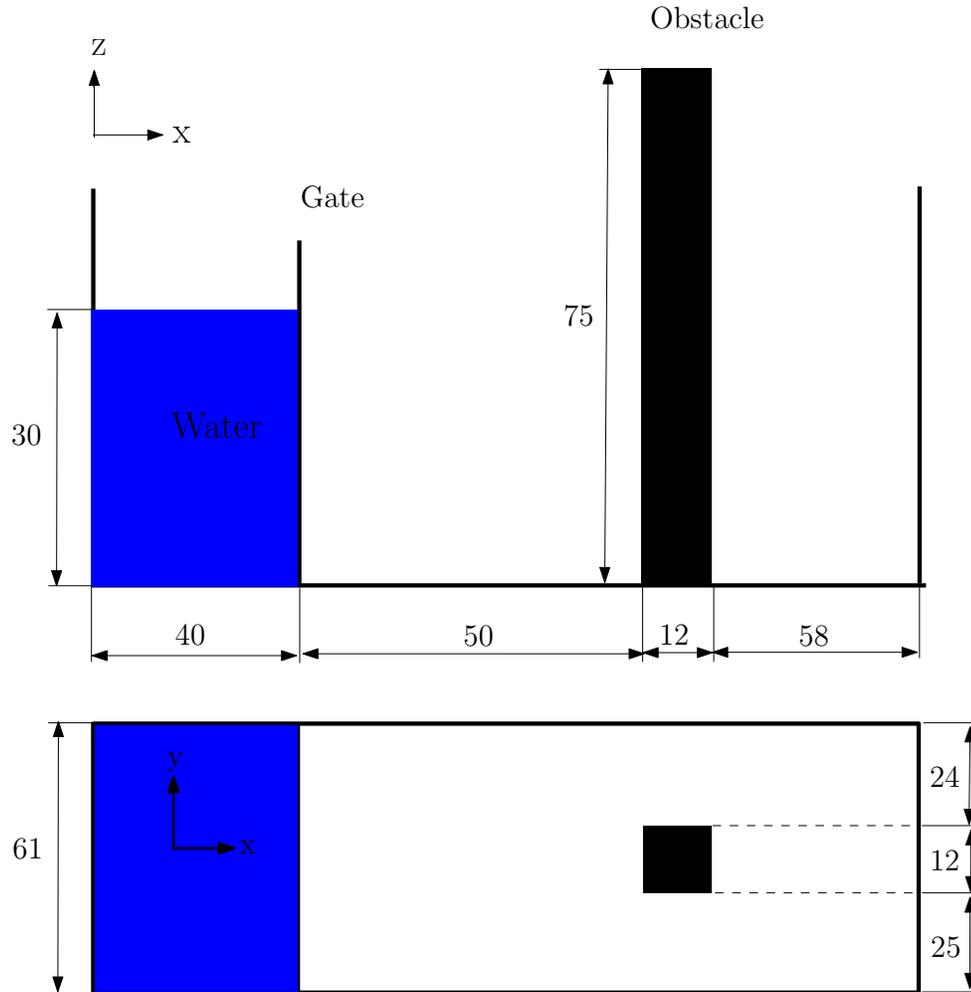


Figure 3.2: Schematic of the validation case: dam break in the vicinity of a tall structure (dimensions in cm). Experiment reported by Gomez-Gesteira et al. [4].

of approximately 1 cm of water existed on the bottom of the tank, before the dam breaks, at $t = 0$ s. In the experiment, as mentioned in [4], the velocity in the x-direction was measured at 2.6 cm from the bottom of the tank and 14.6 cm upstream of structure center.

For this simulation, the particle size was set to $\Delta x = 0.01$ m, $\Delta y = 0.01$ m, $\Delta z = 0.01$ m which resulted in 298463 total particles. The simulations were performed on a local machine using 16 processors (model Intel(R) Xeon(R) CPU E5-2630 0 @ 2.30GHz).

The time history of water velocity in the x-direction at the measured point is compared with experimental data in Figure 3.3. In order to calculate the velocity,

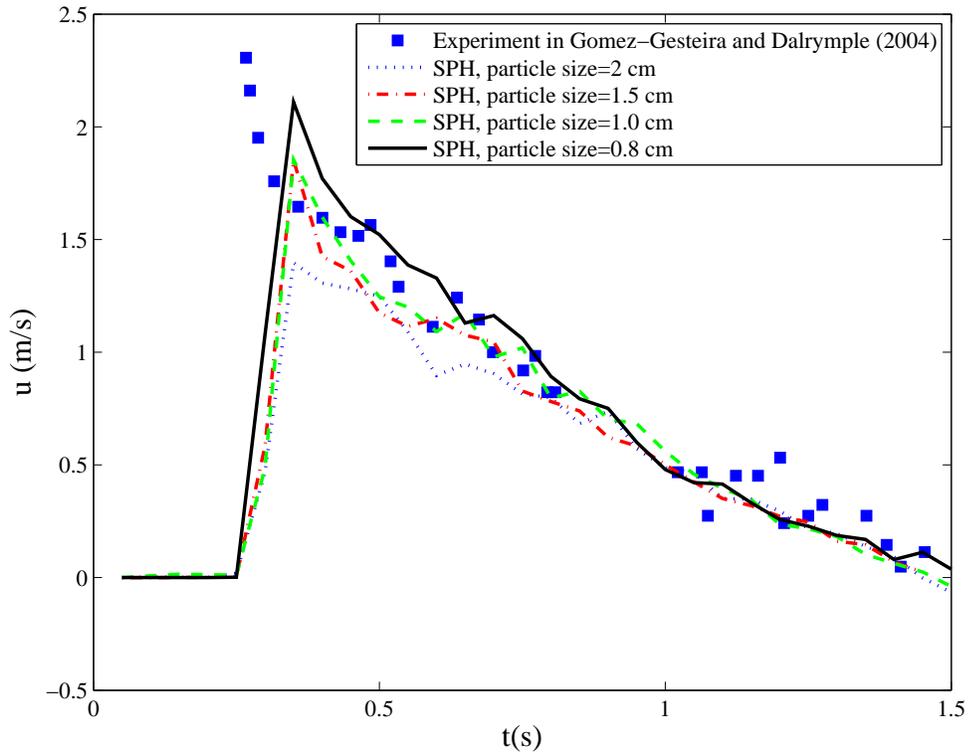


Figure 3.3: Magnitude of the fluid velocity in x-direction (u) at $x = 0.754$ m, $y = 0.31$ m, $z = 0.026$ m as a function of time for dam break experiment.

the same kernel function for simulation is used to interpolate the neighboring particle's velocity at the specified point. It is shown in Figure 3.3 that the SPH results are in good agreement with the experimental data. The convergence study is performed for four particle sizes and the results are presented in Figure 3.3. It is shown that by increasing the particle resolution the maximum velocity value is captured more accurately.

The particle representation of the domain (water particles and the structure) at $t = 0.9$ s is shown in Figure 3.4. Particles are colored with their velocity magnitude. At $t = 0.9$ s, water has impacted the structure and getting closer to the end of the wave tank. The non-uniformly distribution of the particles near the front of the dam can be due to the kernel truncation in the ISPH method which was mentioned by Lind et al. [48] and Xu et al. [75].

The wall clock time (execution time) for 500 time steps of the simulation with 298463 particles for 1, 2, 4 and 16 threads is shown in Figure 3.5. The wall clock time is compared for different combinations of threads in this figure (the x and y-directions

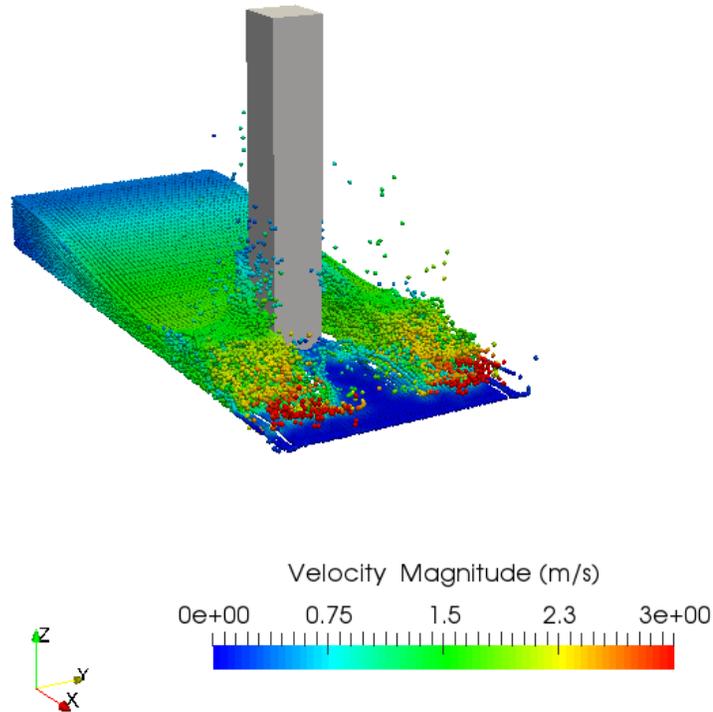


Figure 3.4: Flow due to the dam break on the tall structure at $t = 0.9$ s.

are chosen since the initial water depth is 0.3 m in z-direction). It is shown that by increasing the number of threads from 1 to 16 the execution time decreases. The speedup for a parallel code is defined by Quinn [76] as $S = T_{serial}/T_{parallel}$, where the T_{serial} is the serial execution time and $T_{parallel}$ is the parallel execution time. As expected, the speedup does not scale particularly well for the OpenMP approach. Possible explanations for this include: the time spent on the serial part of the code, joining and forking and the time wasted by the threads waiting for other threads to perform their jobs due to load imbalance. As shown in Figure 3.6, by increasing the number of particles, decomposing threads in the y-direction decreases the execution time more than decomposing them in the x-direction. The reason is at 500 time steps ($t = 0.25$ s), water particles accelerate toward the tall structure but they have not reached the structure yet. Therefore, there are more particles in the y-direction and decomposing more threads in that direction reduces the wall clock time (Figure 3.6).

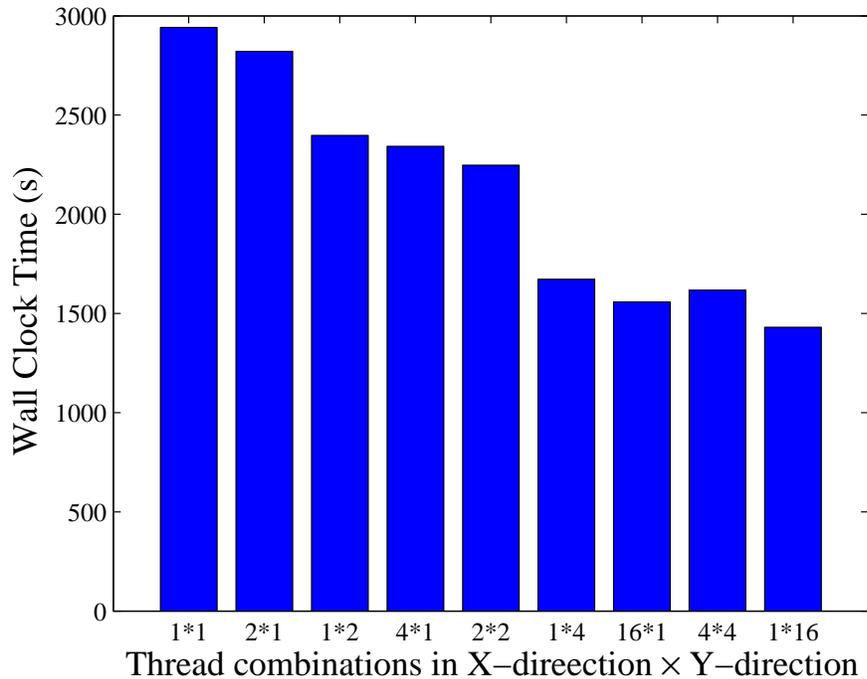


Figure 3.5: Wall clock time for 500 time steps for different thread combinations for 1, 2, 4 and 16 threads.

3.5.2 Test case 2: Wedge Water Entry Simulation

The 2D wedge water entry problem is chosen for simulation in order to test the accuracy of the ISPH method before modeling the wave impacting an OSWC. The experiment of Zhao et al. [57] is simulated in this paper where a symmetric wedge is dropped on a free surface of a water tank at rest. The schematic of the wedge and locations of pressure sensors are shown in Figure 3.7. At $t = 0$ s, the wedge with total mass of 241 kg is dropped with the initial vertical velocity of -6.15 m/s. The total number of particles used for ISPH simulation is 85983 in 2D. In Figures 3.8 and 3.9 fluid particles are shown with by their pressure at $t = 0.00435$ s and $t = 0.0158$ s, respectively. The formation of two jets along the boundaries of the wedge and the free surface deformation are clear in both figures.

In Figures 3.10 and 3.11 the pressure along the boundary of the wedge are shown at $t = 0.00435$ s and $t = 0.0158$ s, respectively. In these figures, as mentioned by Liu et

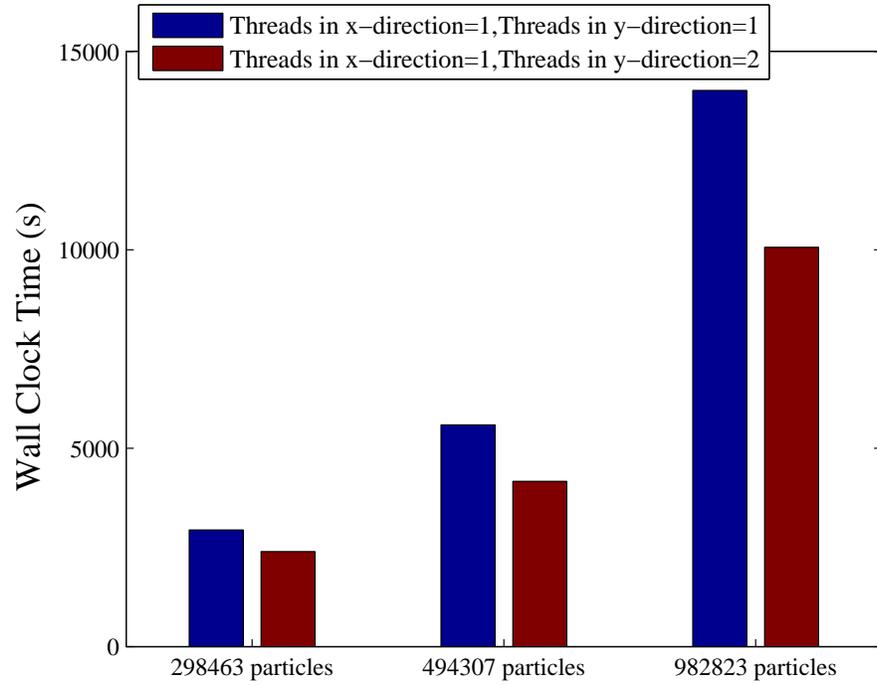


Figure 3.6: Wall clock time for 500 time steps for different number of particles.

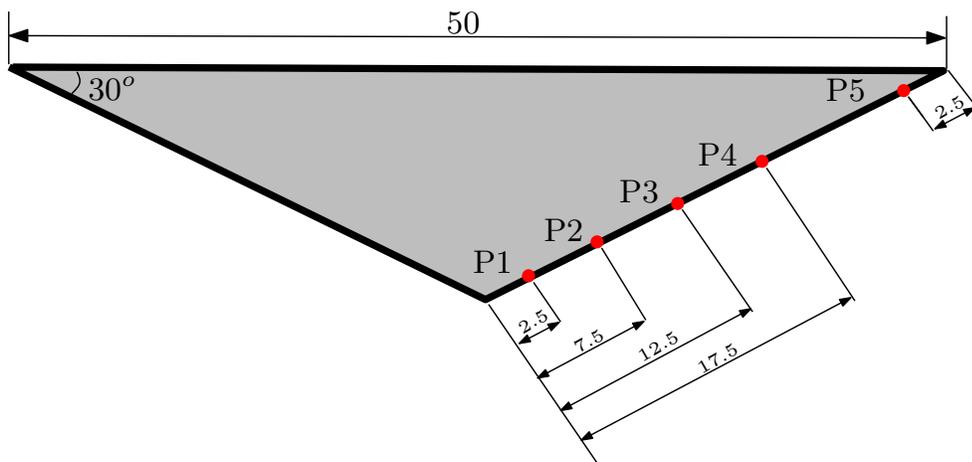


Figure 3.7: Schematic of the wedge (dimensions in cm).

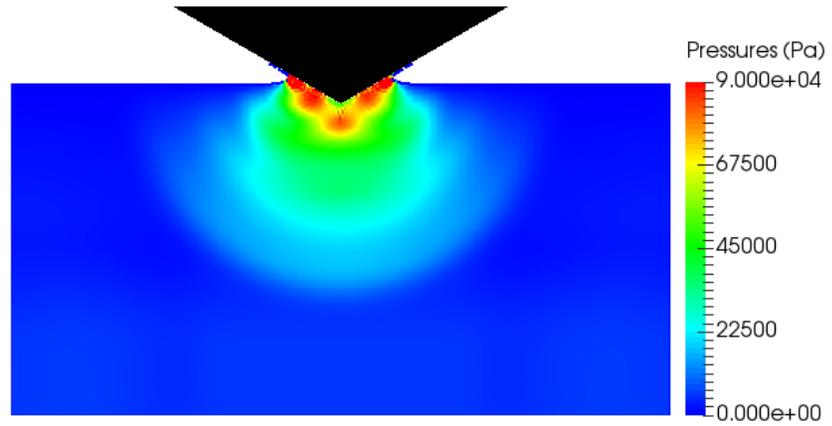


Figure 3.8: Particles are shown with their pressure at $t = 0.00435$ s.

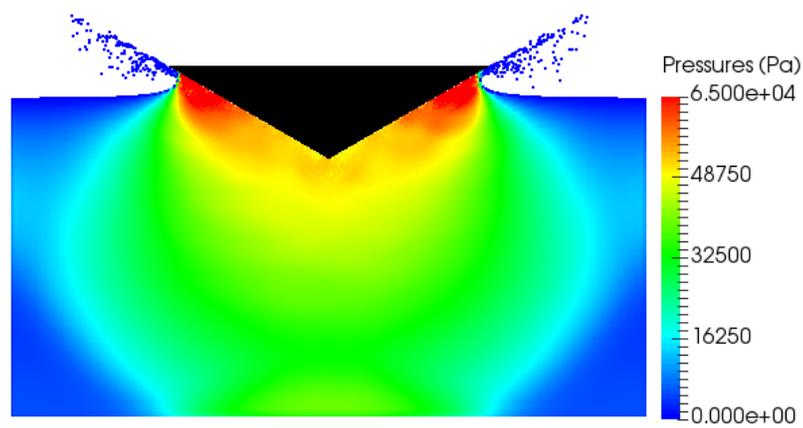


Figure 3.9: Particles are shown with their pressure at $t = 0.0158$ s.

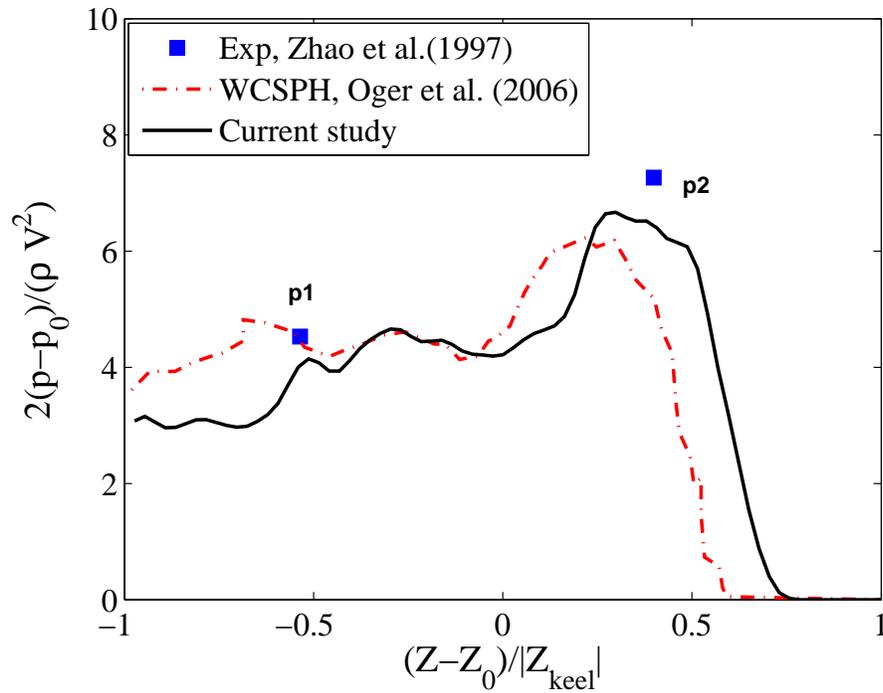


Figure 3.10: Pressure along the boundary of the wedge at $t = 0.00435$ s.

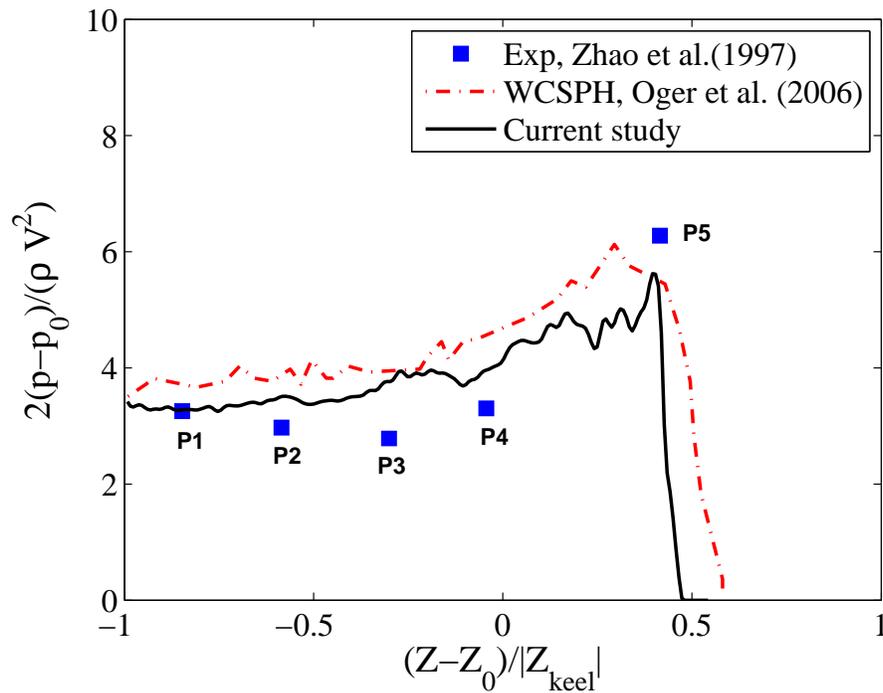


Figure 3.11: Pressure along the boundary of the wedge at $t = 0.0158$ s.

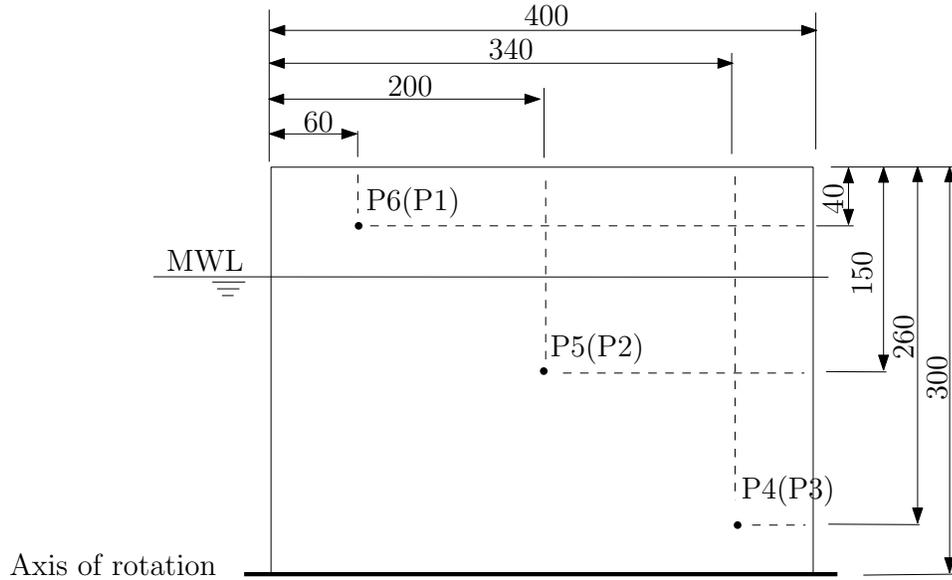


Figure 3.12: Schematic of the flap (dimensions in mm).

al. [77], the $p_0 = 0$, V is the vertical velocity of the wedge, Z is the vertical coordinate of the wedge boundary particles, Z_0 is the initial vertical coordinate of the keel and $|Z_{keel}|$ is the total displacement of the keel. In Figures 3.10 and 3.11, results of the current simulation are compared with the experimental data of Zhao et al. [57] and WCSPH study of Oger et al. [56]. At $t = 0.00435$ s both SPH studies underestimates the pressure. The reason was mentioned by Oger et al. [56] to be compressible effect caused by abrupt change of flow at the beginning of the simulation. At $t = 0.0158$ s both methods slightly overestimates the experimental data up to sensor P5. This is due to the 3D effects mentioned by Zhao et al. [57] and Oger et al. [56].

3.5.3 Test case 3: Wave Impacting an Oscillating Wave Surge Converter

3.5.3.1 Experiment

The 1:20 scale model of Resolute Marine Energys SurgeWEC is chosen for simulation in this paper. As described in [58], this device is smaller than the Aquamarines Oyster, therefore the device response to the wave impact behavior might be different than the already studied Oyster.

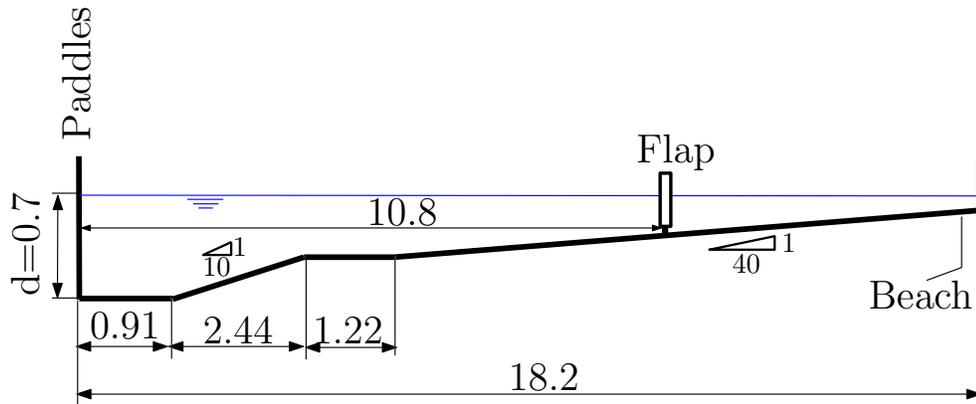


Figure 3.13: Schematic of the wave tank (dimensions in m).

The experiments were conducted at Orion Energy Centre’s wave tank. The wave tank is 18.2 m long and 6 m wide (Figure 3.13). The waves are generated using eight wavemakers across the width on the left end of the tank (Figure 3.13). The OWSC model is located at 10.8 m from the wavemakers. The waves are absorbed at the end of the tank by a sloped beach (Figure 3.13).

The 1:20 scale model is a flap of 0.4 m width, 0.3 m height, and 0.06 m thickness and it was hinged 0.05 m above the tank bottom (Figure 3.13). The model is made of 6.3 mm thick steel plate with aluminum skeleton and foam body (for more details about the model see [58]). Six Keller series 10 pressure sensors were placed on the front and rear faces of the flap. Sensors 4-6 were placed on the front face of the flap and sensors 1-3 placed on the rear face of the flap. The position of the pressure sensors are shown in Figure 3.12.

A series of experiments were performed to study the extreme loads on the SurgeWEC 1:20 scale model by Clabby and Tease [58]. Three water depths were tested under four extreme sea conditions, ten monochromatic wave and three irregular waves. In this paper, experimental results for a monochromatic wave was chosen to validate the SPH results.

3.5.3.2 SPH simulations

The 1:20 scale model of the SurgeWEC that was described in Section 3.5.3.1 is simulated herein using the ISPH approach. The numerical flap has the same dimensions of the experimental flap: 0.4 m width, 0.3 m height, and 0.06 m thickness. The SPH simulation is performed in a 18.2 m (length) \times 4 m (width) \times 1.2 m (height)

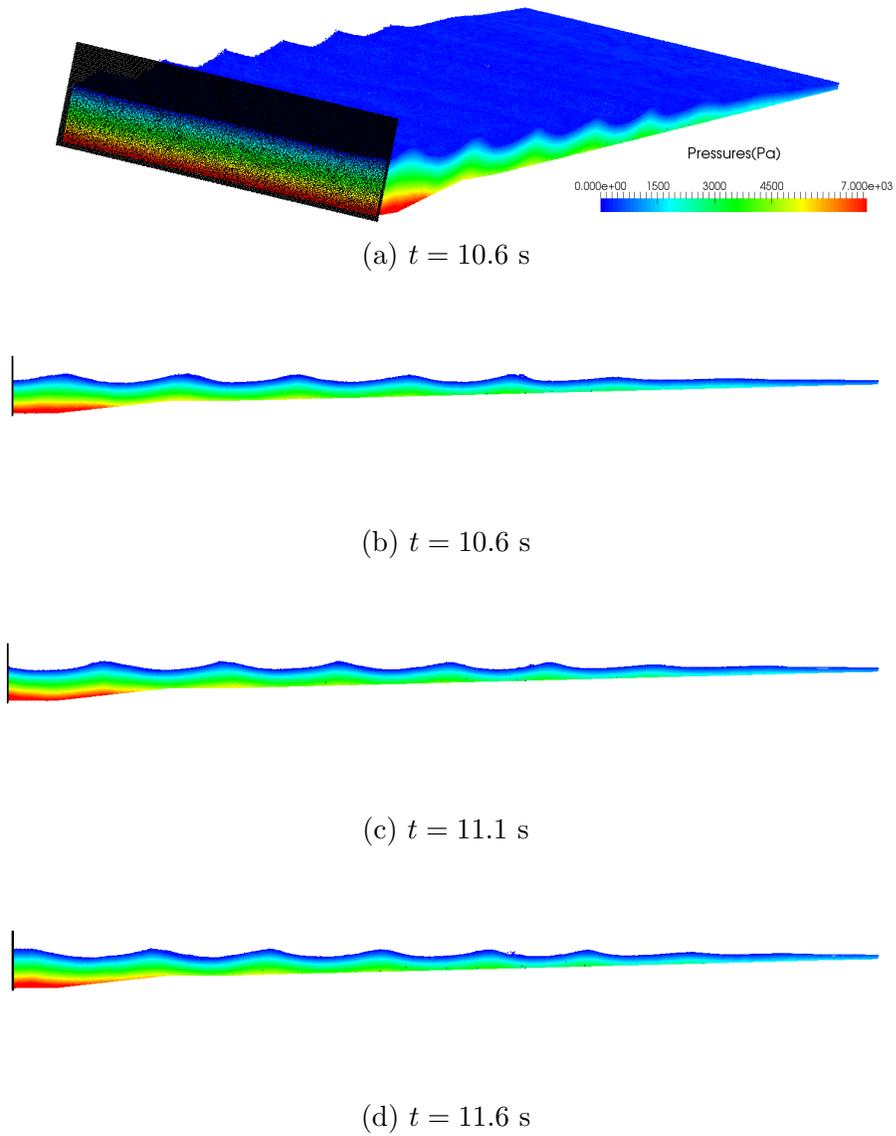
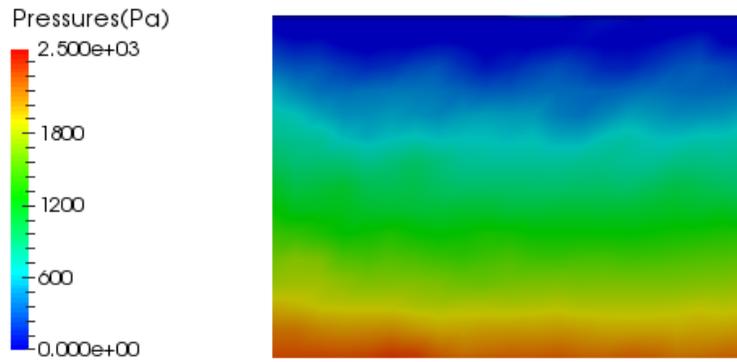
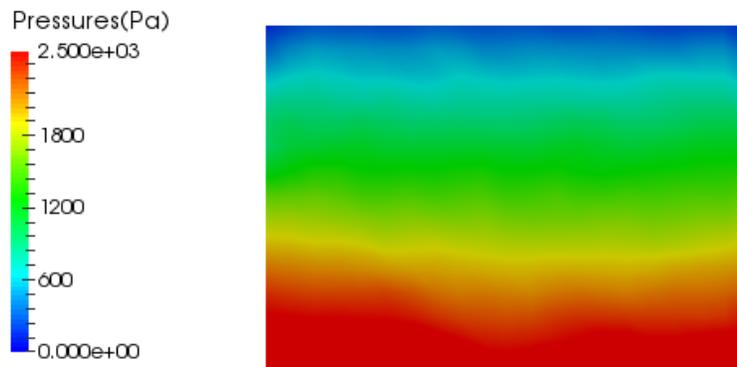


Figure 3.14: Particles in the domain are colored by their pressure at three time steps (a) shown in 3D, (b-d) shown in 2D.

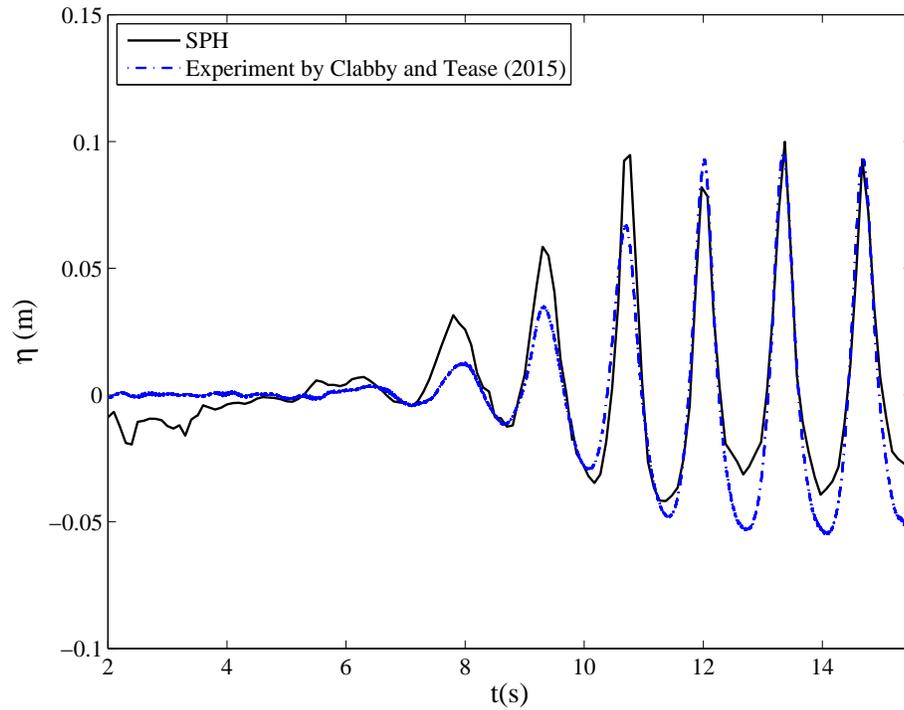


(a) $t = 11.6$ s where $\theta \approx -36^\circ$.

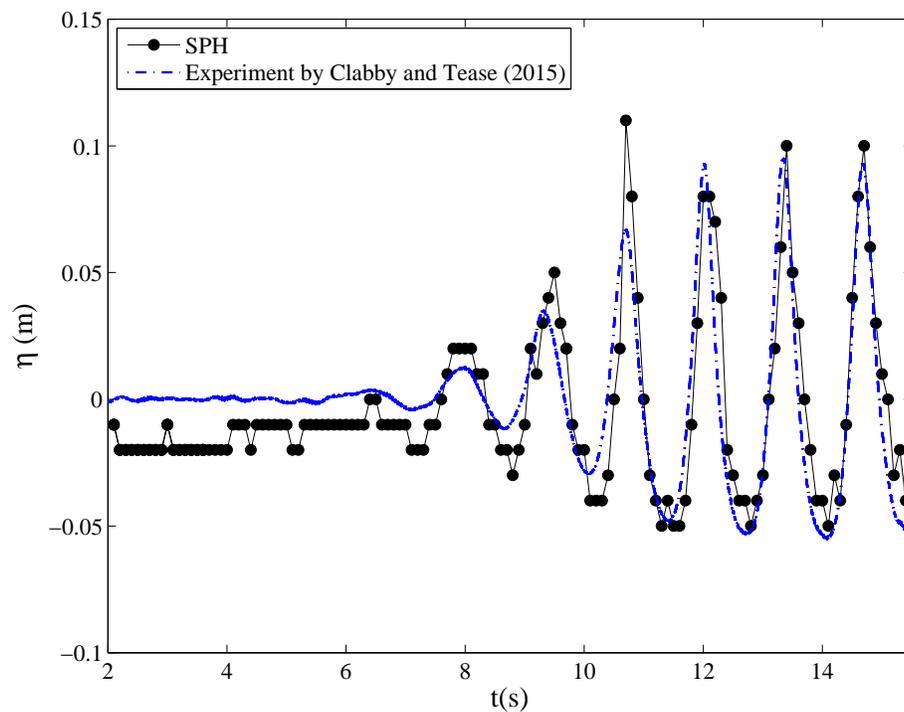


(b) $t = 12.1$ s where $\theta \approx 30^\circ$.

Figure 3.15: Pressure field on the flap face.



(a) Free surface location determined by averaging probe's neighboring particles z positions.



(b) Free surface location determined by using fixed interpolation points at the probe location.

Figure 3.16: Free surface elevation at $(x = 10.8 \text{ m}$ and $y = 3.5 \text{ m})$ as a function of time.

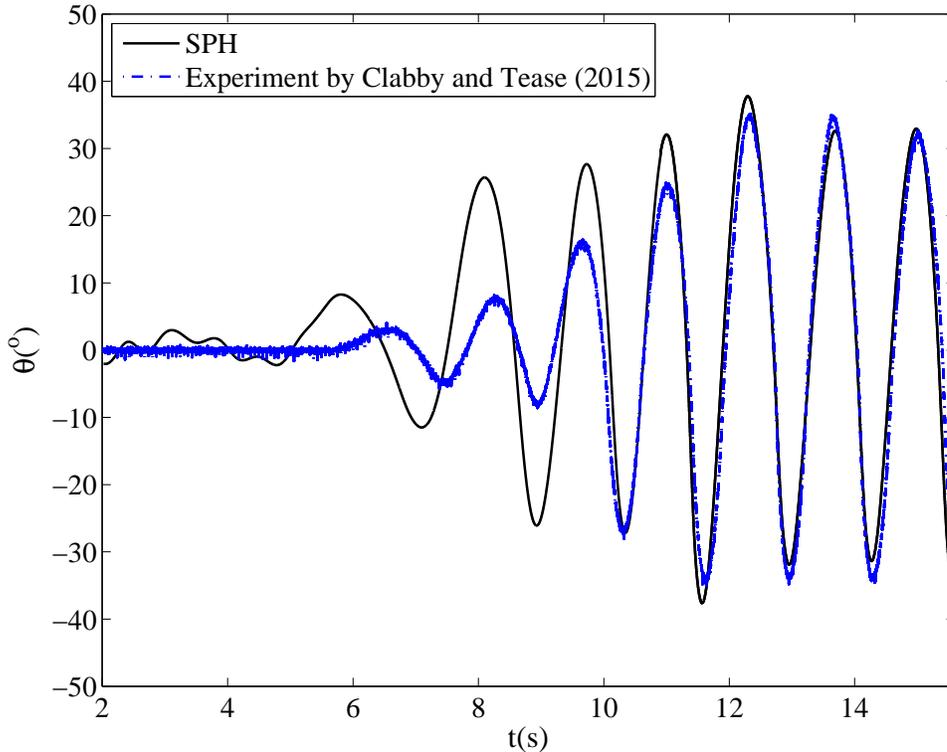


Figure 3.17: Angular position of the flap as a function of time.

numerical tank where the flap is positioned at 10.8 m from the wavemaker. To save computational cost, we choose the numerical wave tank's width to be 2.0 m smaller than the experimental width. The numerical wavemaker is forced to have sinusoidal motion based on the theory given by Dalrymple and Dean [78]:

$$x = \frac{S}{2} \sin\left(\frac{2\pi}{T}t\right), \quad (3.5.1)$$

where x is the motion of the wavemaker in the x-direction, T is the wave period and S is the stroke of the wavemaker. The wave profile used for the simulation has a height of $H_w = 0.15$ m and a period of $T = 1.34$ s.

The still water depth is 0.7 m and the numerical wavemaker was calibrated to produce the wave of height 0.15 m. The particle dimensions are set to $\Delta x = \Delta y = \Delta z = 0.02$ m, which leads to 4222746 total particles. To avoid wave reflection, the damping region similar to the relaxation zone proposed by Lind et al. [48] is applied to the zone close to the far downstream wall.

The code was parallelized using OpenMP as described in Section 3.4 and the runs

were performed on the Hermes WestGrid cluster. The 16 processors were used for the simulation and it took approximately 12 days for 15 seconds of simulation time. Figure 3.14 shows the particles in the domain colored by their pressure at several snapshots. Figures 3.15a and 3.15b show pressure field on the face of the flap at $t = 11.6$ s and $t = 12.1$ s, respectively. At $t = 11.6$ s the flap is at $\theta \approx -36^\circ$ while at $t = 12.1$ s the flap is at $\theta \approx 30^\circ$. It is shown that when the flap is at the trough ($t = 11.6$ s), the pressure has smaller values than when the flap is close to the wave crest ($t = 12.1$ s).

The free surface elevation (η) of the SPH simulations is compared with the experimental data at the wave probe location ($x = 10.8$ m and $y = 3.5$ m) for the wave profile described ($H_w = 0.15$ m, $T = 1.34$ s). The data for eight wave probes is available from the experiment in the range of $x = 10.1$ m to $x = 11.1$ m. The wave probe at the flap position ($x = 10.8$ m) is chosen to validate the SPH results. In order to determine the free surface location, two different methods are used in this paper. In the first method, the particles that fall within a specified range (particle size) around the probe location are chosen and the average of their z positions is used as the free surface position (Figure 3.16a). In the second method, a line of fixed interpolation points is added to the probe location and the density is calculated on each point by using the same kernel used for SPH simulations (Figure 3.16b). The interpolation points are placed in a 1 cm distance from each other and the free surface location is determined by applying the criteria mentioned in Section 3.3.2. The interpolation method is computationally more expensive than the former averaging method. The comparison between simulation and the experiment shows good agreement after the startup time (after $t = 10$ s) as shown in Figures 3.16a and 3.16b. The discrepancy between the SPH simulations and the experiment during startup is due to the differences in the real and simulated wavemakers' motion. The SPH results deviation from the experimental data between 2-6 s, is due to the fact that particles are settling at the probe location in the wave tank during this time. The deviation of SPH results from the experimental data is approximately within a particle size ($\Delta x = 0.02$ m). This indicates increasing resolution will improve the accuracy of the simulation. It is shown that part of the shallower wave troughs in Figure 3.16a is removed by using the interpolation points scheme in Figure 3.16b. Another reason of shallower troughs in the simulation could be the wave reflection from the side wall. The velocity in the trough is smaller than the crest hence it can be more affected. Applying the damping

zone to the side walls will improve results in the trough.

The pitch angle (θ) evolution is compared with experimental data in Figure 3.17. The positive pitch angle corresponds to the angle between the initial position of the OWSC and landward (clockwise), and the negative angle represents the angle between the initial angle of the OWSC and seaward (counter clockwise). The pitch angle amplitude from the experimental data is approximately 30° . The pitch angle in the SPH simulations follows the free surface elevations in Figure 3.16 and is in an excellent agreement with the experimental data after startup time.

In Figure 3.18a the dynamic pressure at sensor 5 (positioned at the center of the flap face) is compared with the experimental data. In order to calculate the dynamic pressure, the hydrostatic pressure (mean pressure) is subtracted from the total pressure. The hydrostatic pressure is the dominant pressure for buoyant OWSCs, but the dynamic pressure is a better metric to evaluate the performance of the method. The deviation in pressure between the experimental data and SPH simulations is expected during startup as the wave height and flap rotations are different. Henry et al. [51] reported that the highest pressure occurs in the center of the flap face. It was mentioned by Henry et al. [51] and Wei et al. [62] that the wave impact has a stochastic nature accompanied with spikes in the pressure. It is clear that the current code is capable of capturing the slamming phenomena. In Figure 3.18a, the dynamic pressure is in a good agreement with the experimental data after startup time.

In Figure 3.18b the dynamic pressure at sensor 6 is compared with the experimental data. Sensor 6 is located above the free surface and it reads zero pressure where it is exposed to the air throughout the simulation. The dynamic pressure at sensor 6, is in a fair agreement with the experiment. The performance of the SPH simulations in reproducing the dynamic pressure for both pressure probes is excellent where the free surface displacements are in good agreement with the experimental data (Figure 3.16).

Table 3.1 shows the wall clock time for 200 time steps of the OWSC simulation. The total number of particles are 4222746 and the Hermes WestGrid Cluster is used for the simulation. It is shown that by increasing the number of threads from 1 to 16 the wall clock time decreases from 21189 s to 5341 s. Although the OpenMP approach is relatively easier approach to implement in comparison with the MPI approach, MPI parallelization will be essential for simulations with larger particle numbers.

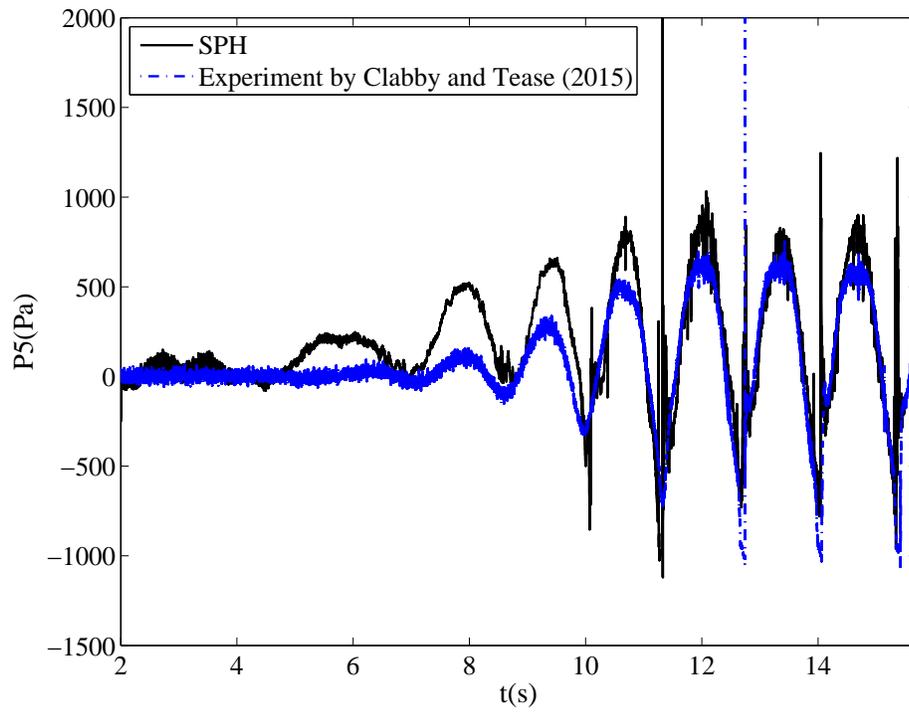
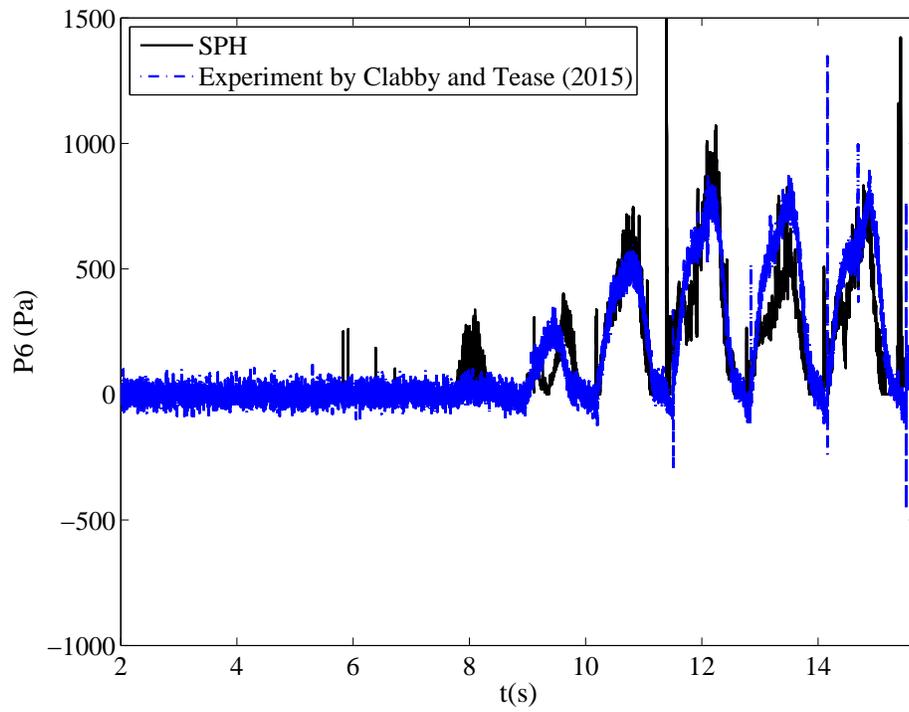
(a) P_5 as a function of time.(b) P_6 as a function of time.

Figure 3.18: Dynamic pressure at the surface of the flap.

Table 3.1: Wall clock time for 200 time steps

| Threads in x, y and z direction | $1 \times 1 \times 1$ | $3 \times 2 \times 1$ | $4 \times 4 \times 1$ |
|---------------------------------|-----------------------|-----------------------|-----------------------|
| Wall clock time | 21426 s | 8825 s | 5341 s |

3.6 Conclusion

A new OpenMP parallelization scheme for the SPH method is proposed. This method has the ability to use variable thread combinations in different directions. The performance of the OpenMP SPH code is reported for the impact of a dam break on a tall square structure. The code was also tested to simulate the impact of a monochromatic wave on the Resolute Marine Energys SurgeWEC. Time history of free surface, pitch angle and two pressure sensors are compared with experimental data and reasonable agreement is achieved after startup. However, modifications are required to improve the numerical simulations. The current SPH code is capable of simulating a single fluid. The air entrainment effect is an important phenomena in the interaction between wave and the structures which is not accounted for in the current work. Finer particle resolution close to the OWSC is required, to provide more details about the interaction. It is shown that using a higher number of threads will reduce the wall clock time, but the future work must implement the MPI standard to perform simulations with larger number of particles more efficiently.

3.7 Acknowledgment

The authors thank the Natural Sciences and Engineering Research Council, Natural Resources Canada, the Pacific Institute for Climate Solutions and the University of Victoria for their financial support. The authors thank the Compute Canada and WestGrid for their computational resources. The authors also thank the Resolute Marine Energy for providing the experimental data.

Chapter 4

SPH Modeling of Hydrodynamic Loads on a Point Absorber Wave Energy Converter Hull

This paper is presented at the 11th European Wave and Tidal Energy Conference (EWTEC) 2015 in Nantes, France.

Yeylaghi, Shahab, Bradley Buckham, Belaid Moa, Peter Oshkai, Scott Beatty, and Curran Crawford. "SPH modeling of hydrodynamic loads on a point absorber wave energy converter hull." 11th European Wave and Tidal Energy Conference (EWTEC) 2015, Nantes, France.

The scope of this chapter is to use the OpenMP code to simulate the hydrodynamic load on a point absorber hull. The hydrodynamic and total forces are compared with the previously available experimental data. The experiments for the point absorber wave energy converter were previously performed by Scott Beatty and used in this paper for comparison with the ISPH simulations.

4.1 Abstract

For over forty years Smoothed Particle Hydrodynamics (SPH) has been used to simulate a wide variety of problems ranging from astrophysical (compressible, inviscid) to

hydrodynamic (incompressible, viscous) flows. The Lagrangian and meshless features of SPH have made it attractive in different engineering fields, especially hydrodynamics. In the context of hydrodynamic applications, the advantage of SPH over conventional Eulerian mesh-based CFD methods is its ability to capture complex free surface flows involving air, water and possibly a surface piercing rigid body. Unfortunately, the performance in such three-phase problems comes with increased computational expense. In this paper, an in-house 3D parallel explicit Incompressible-SPH (ISPH) code is used to simulate the hydrodynamic loads from the spar of a self-reacting point absorber Wave Energy Converter (WEC) design. The WEC considered in the experimental and numerical study is a 1:25 physical scale model. In the experiment, radiation tests in heave were considered for the WEC design in which the displacement of the WEC was controlled. Sinusoidal trajectories with different amplitudes and frequencies were imposed in otherwise still water. In the current numerical work, the hydrodynamic loads from an explicit ISPH simulation for one of the extreme cases, in terms of frequency $\omega=4$ rad/s and amplitude ($A=6.1$ cm) , is compared with the available experimental data for the WEC design.

4.2 Introduction

Smoothed Particle Hydrodynamics (SPH) is a Lagrangian meshless particle method, which was invented in 1970s to study astrophysical problems including compressible, inviscid flows [20, 21]. The Lagrangian and meshless features of SPH make it attractive in different engineering fields, especially hydrodynamics. The SPH method is adept at problems including free surface flows with fragmentation and breaking, without the need for Volume of Fluid (VoF) or Level-Set (LS), Eulerian based CFD methods. On the other hand, viscosity terms and solid boundary conditions can be more challenging to implement in the SPH method compared to Eulerian based CFD methods.

Incompressible fluids in SPH can be modelled either by relating the fluid pressure to particle density by using a stiff equation of state, or by solving Poisson's equation to determine the pressure. The two methods are known as weakly compressible SPH (WCSPH) [23] and incompressible SPH (ISPH) [24], respectively. In the ISPH method, a common approach to solve Poisson's equation is to solve a set of algebraic equations implicitly, but Hosseini et al. [30] proposed an explicit method to solve the

Poisson's equation that doesn't require solving a set of algebraic equations at each time step.

In order to simulate viscous flows, the original SPH formulation requires viscosity treatment. Several methods of modeling viscosity are reported in literature and they can be divided into three main categories. Introducing artificial viscosity term was the first attempt to model viscosity in SPH [47]. Several artificial viscosity forms were proposed to be used in shock wave simulations [47]. The most common model was applied to free surface flows in [23]. The second method was applied in [79] to solve the Navier-Stokes equations. The proposed method used second derivatives of velocity with a constant viscosity to model viscous diffusion in the Navier-Stokes equations. The third method is the viscosity term introduced by Morris [80] to model diffusion in the Navier-Stokes equations. This model combines the first derivative of standard SPH with a finite difference approximation of the first derivative [80]. This method is reported to be more accurate, and efficient in terms of computational cost among the two aforementioned methods in [81].

SPH was initially invented to simulate astrophysical problems where no solid boundary condition is required. The first attempt to model the solid boundary condition was reported by Monaghan [23]. In this method, a single line of stationary boundary particles are placed on the edge of the solid boundary, exerting a repulsive force to the fluid particles approaching them. Improved versions of this method were proposed in [68, 82]. Due to adjustments of many parameters involved in this method, these types of boundary conditions are not suitable for accurate calculation of the hydrodynamic forces on a structure [83]. Dummy boundary particles were introduced to SPH method by Shao [65]. In this method, several layers of particles are placed on the edge and inside of the solid boundary [69]. The momentum equation is solved for all particles (solid particles held fixed) which is suitable for computer programming and code parallelization [30]. Another method of simulating solid boundaries are ghost boundary particles. These particles are placed outside of the domain along the solid boundary (position determined by reflection through the solid boundary). The pressure and density of the ghost particles are the same as their corresponding fluid particles [82]. Their velocity has the opposite sign of the correspondent fluid particle's velocity (no-slip boundary condition) [82]. Applying this method is challenging for complex geometries (sharp boundaries) and in 3D [82].

In this paper, an in-house parallel 3D explicit ISPH code is used to simulate the

hydrodynamic loads from the spar of a self-reacting point absorber Wave Energy Converter (WEC) design. The hydrodynamic loads from an explicit ISPH simulation for one of the extreme cases, in terms of frequency ($\omega = 4$ rad/s) and amplitude ($A = 6.0$ cm), is compared with the available experimental data for the WEC design.

4.3 Experiment

The tank facility (4.5 m wide, 58 m long, 2 m water depth) used to undertake the experiment was Memorial University's Ocean Engineering Research Center located in St. John's Newfoundland, Canada.

A position feedback mode enables the linear actuator to carry out preprogrammed motions in still water for hydrodynamic coefficient extraction. The control system was developed in NI Labview and implemented in real time on an NI PXI chassis running at 50 Hz for sampling and control. The linear motor (LinMot PS01-37x120, Elkhorn, Wisconsin, USA), exerts 250 N maximum force, has a maximum travel of 0.28 m, and weighs under 3 kg. Relative displacements are measured with a non-contacting laser displacement sensor (Micro-Epsilon optoNCDT-1402-600 with a range of 600 mm and resolution of 80 μm). The force was measured by a 500 N capacity, S-type, tension/compression load cell. Acceleration was measured via accelerometer (Analog Devices ADXL203) with ± 1.7 g range.

The tests comprised pre-programmed heave motions of 2-8 cm in height and 120 seconds in duration using the linear motor as a position-controlled actuator in a quiescent tank. Four quantities were logged: displacement $x(t)$, velocity $\dot{x}(t)$, acceleration $\ddot{x}(t)$, and actuator force $F(t)$. In this particular work, experimental results from the case $\omega = 4$ rad/s and motion amplitude 0.06 m were chosen for comparison to the SPH results.

4.4 Methodology

The Lagrangian description of the Navier-Stokes equations are solved for each single fluid particle in the SPH method. The governing equations include the conservation of mass and momentum equations:

$$\frac{1}{\rho} \frac{D\rho}{Dt} + \nabla \cdot \mathbf{u} = 0 \quad (4.4.1)$$

$$\frac{D\mathbf{u}}{Dt} = -\frac{1}{\rho} \nabla p + \mathbf{g} + \nu \nabla^2 \mathbf{u} \quad (4.4.2)$$

where ρ is the fluid particle density, \mathbf{u} is the particle velocity vector, t is time, p is the particle pressure, \mathbf{g} is the gravitational acceleration vector and ν is the kinematic viscosity. In this method, the domain and its boundaries are represented by discrete volumes of fluid (particles), in which each particle carries material properties, such as velocity, density, viscosity etc. The flow quantities are interpolated over the predefined neighboring particles using a smoothing function (kernel). The kernel determines the contribution of neighboring particles on the particle of interest's properties. The kernel approximation of $A(\mathbf{r})$ in SPH is written as:

$$A(\mathbf{r}) \approx \int_{\Omega} A(\mathbf{x}) W(\mathbf{r} - \mathbf{x}, h) d\mathbf{x} \quad (4.4.3)$$

where \mathbf{r} is the location vector, $A(\mathbf{r})$ is the function of interest at location \mathbf{r} , W is smoothing function (kernel) defined over domain of interest Ω , and h is the smoothing length which defines the kernel radius. The particle approximation of the above integral is obtained by replacing the integral with summation (finite number of particles) as:

$$A_i = \sum_{j=1}^N A_j W(\mathbf{r}_i - \mathbf{r}_j, h) \frac{m_j}{\rho_j} \quad (4.4.4)$$

where m_j is the mass of particle j in the neighborhood of particle i and ρ_j is the density of particle j . In this study, the fifth order Wendland kernel [46] is used,

$$W(q) = k \begin{cases} (1 + 2q)(1 - \frac{q}{2})^4 & 0 \leq q \leq 2 \\ 0 & q \geq 2, \end{cases} \quad (4.4.5)$$

where $k = 7/4\pi h^2$ in 2D and $k = 21/16\pi h^3$ in 3D and $q = |d_r|/h$ and d_r is the inter-particle distance. Here, the smoothing length is set to $h = 1.5d_r$ for 3D simulations. The first order spatial derivative is defined as [47]:

$$(\nabla A)_i = \rho_i \sum_j m_j \left(\frac{A_j}{\rho_j^2} + \frac{A_i}{\rho_i^2} \right) \nabla_i W_{ij} \quad (4.4.6)$$

Also, the second order derivative adopted here is [24]:

$$\nabla \cdot \left(\frac{1}{\rho} \nabla A \right)_i = \sum_{j=1}^N \left(\frac{8m_j}{(\rho_i + \rho_j)^2} \frac{A_{ij} \mathbf{r}_{ij} \cdot \nabla_i W_{ij}}{r_{ij}^2 + \eta^2} \right) \quad (4.4.7)$$

where $\mathbf{r}_{ij} = \mathbf{r}_i - \mathbf{r}_j$, $A_{ij} = A_i - A_j$ and $\nabla_i W_{ij}$ is the gradient of the kernel function at particle i . The explicit ISPH method [30] is used for this study. This approach is based on the two step projection method widely used in Eulerian based CFD methods [24]. In the prediction step, the intermediate velocity is calculated using the viscous and body forces, without pressure force as:

$$\mathbf{u}^* = \mathbf{u}(t) + \Delta t(\mathbf{g} + \nu \nabla^2 \mathbf{u}) \quad (4.4.8)$$

$$\mathbf{r}^* = \mathbf{r}(t) + \Delta t \mathbf{u}^* \quad (4.4.9)$$

where \mathbf{u}^* is the intermediate velocity and \mathbf{r}^* is the intermediate position. In the correction step, the incompressibility condition is achieved by correcting the velocity using the pressure force as:

$$\mathbf{u}(t + \Delta t) = \mathbf{u}^* + \Delta t \left(-\frac{1}{\rho} \nabla p \right) \quad (4.4.10)$$

By taking the divergence of Eq. 4.4.10 and forcing $\nabla \cdot \mathbf{u}(t + \Delta t) = 0$, one obtains the pressure Poisson's equation which should be solved for pressure at each time step:

$$\nabla \cdot \left(\frac{\nabla p}{\rho} \right) = \left(\frac{\nabla \cdot \mathbf{u}^*}{\Delta t} \right) \quad (4.4.11)$$

By using Eq. 4.4.7, Eq. 4.4.11 in particle form can be written as:

$$\sum_{j=1}^N \frac{8m_j}{(\rho_i + \rho_j)^2} \frac{(p_{ij} \mathbf{r}_{ij}) \cdot \nabla_i W_{ij}}{r_{ij}^2 + \eta^2} = \frac{-1}{\Delta t} \sum_{j=1}^N V_j \mathbf{u}_{ij}^* \cdot \nabla_i W_{ij} \quad (4.4.12)$$

where $p_{ij} = p_i - p_j$, $\mathbf{r}_{ij} = \mathbf{r}_i - \mathbf{r}_j$, $\mathbf{u}_{ij}^* = \mathbf{u}_i^* - \mathbf{u}_j^*$ and η is a small number ($\eta = 0.1h$). In the explicit approach, at time $n + 1$ the pressure for particle i ; p_i^{n+1} , is calculated based on the pressure at neighboring particles at time n ; p_j^n (i.e. in Eq. 4.4.12,

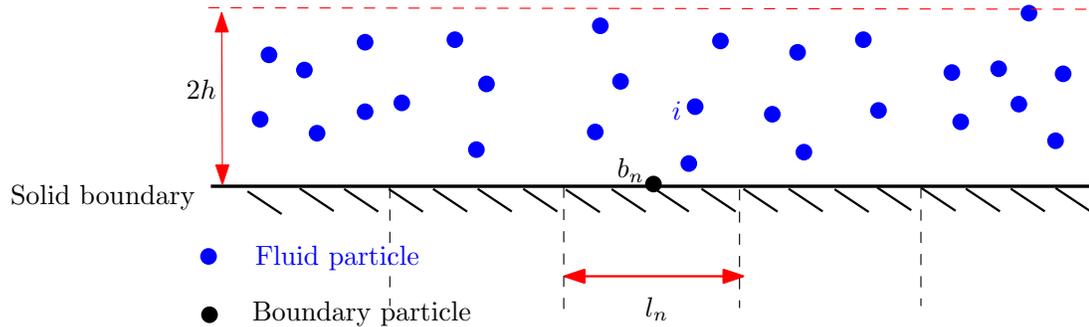


Figure 4.1: Pressure force evaluation on the solid boundary

$p_{ij} = p_i^{n+1} - p_j^n$ and the only unknown will be p_i^{n+1}). By using the explicit approach, solving a set of algebraic equations is not required in each time step; however, a smaller time step must be adopted for accuracy.

Fixed dummy particles are used for the boundary particles as in [30]. The pressure on the boundary particles was calculated from their neighbouring fluid particles in addition to the $\rho g \Delta z$ term. Δz is the vertical distance between the fluid particle and the boundary particle [33]. A similar damping zone to the one used in [84] was adopted in this work close to the tank wall in order to absorb the wave reflection.

The free surface particles are identified in order to apply the Dirichlet boundary condition ($p = 0$). The criteria used for defining a free surface particle is:

$$\rho = k\rho_0 \quad (4.4.13)$$

where k is set to 0.8 for the simulations. The particles that have the densities equal or smaller than the above criteria are set to be a free surface particle.

A sketch describing the pressure force evaluation on the solid boundary in 2D is shown in Figure 4.1 and has been extended to 3D. In order to calculate the forces on solid boundaries, the contribution of the fluid particles (i) within a distance of $2h$ from the solid boundary are considered. The solid boundary has been divided into segments l_n as represented in Figure 4.1. The pressure p_{bn} is calculated on the boundary particle b_n , which is located at the centre of each segment. This pressure is calculated based on the methodology described in [33].

4.5 Results

The dam-break problem is typically used as benchmark test cases for SPH studies. In the current work also, the validation of accuracy of ISPH has been done by comparing the results of dam break problem with the experimental results of [85]. Figure 4.2 shows the comparison of normalized water front as obtained from current analysis to previously published experimental and numerical studies. For this simulation, the water column has the dimension of $L = H = 1$ m, the particle size is set to $\Delta x = 0.02$ m, $\Delta y = 0.02$ m, smoothing length $h = 1.2d_r$, and time step $\Delta t = 0.0002$ s. The time evolution of the water front is presented in Figure 4.2 and compared with the experimental data [85], the Level set method (LS) [26] and a previous WCSPH method [26]. The calculated results are in a reasonable agreement with the experiment and mentioned numerical methods. The discrepancy between all numerical simulations and the experimental data is reported to be due to the surface roughness which makes the water front move slower in the experiment than the mentioned numerical methods [26].

The SPH simulation of the scale model WEC was performed in a $3\text{ m} \times 3\text{ m} \times 2.2\text{ m}$ numerical tank as shown in Figure 4.3. The still water depth is 2 m and the draft of the WEC is 1.4 m . The sketch of the WEC model and the particle representation of it in the SPH method are shown in Figures 4.4a and 4.4b, respectively. The WEC is forced to have sinusoidal motion of:

$$z = A \sin(\omega t) \quad (4.5.1)$$

where A is the amplitude of the oscillation ($A = 0.06\text{ m}$) and ω is the angular frequency ($\omega = 4\text{ rad/s}$). The fluid particle dimensions for this simulation are set to $dx = 0.02\text{ m}$, $dy = 0.02\text{ m}$ and $dz = 0.02\text{ m}$ which leads to 2707963 total particles. The smoothing length (h) concept in the SPH method is similar to grid resolution in Eulerian based CFD methods. The $h = 1.5d_r$ is chosen for this simulation based on the preliminarily runs and similar to the smoothing length used for the Wendland kernel in [51]. The time step is set to $\Delta t = 0.0005\text{ s}$ for the numerical simulations. In order to avoid wave reflection from the walls, the absorption (damping) region is applied to the zone close to the tank walls (0.5 m) similar to the relaxation zone proposed in [84].

The simulations were performed on the Hermes WestGrid cluster. The code was

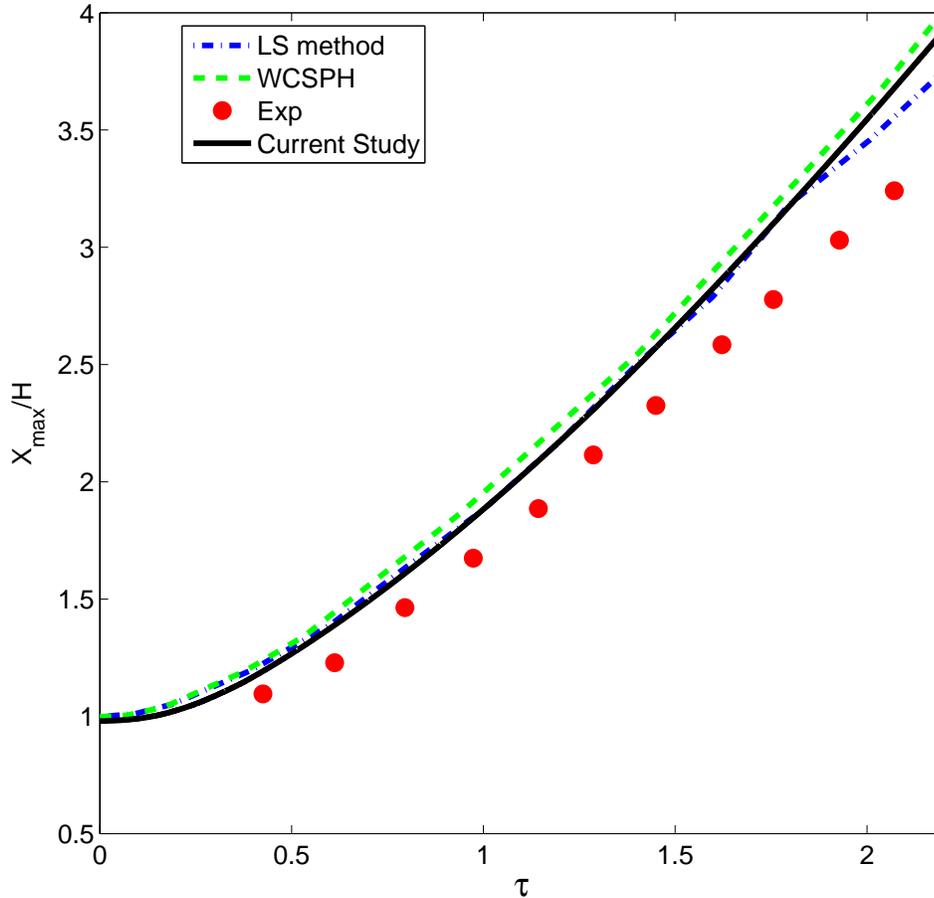


Figure 4.2: Time evolution of the water-front for dam-break problem $\tau = t(g/H)^{0.5}$

parallelized in 3D using OpenMP. The SPH simulations were carried out using 16 processors (model Intel(R) Xeon(R) CPU E5-2660 0 @ 2.20GHz). The memory required was 4Gb and it took approximately 72 hours for 2 periods of physical simulation time.

The vertical velocity component (V_z) and the velocity magnitude for a vertical cut plane passing through the centre of numerical tank at $t = 1.15$ s and $t = 1.75$ s are shown in Figure 4.5a and Figure 4.5a respectively. The dimensions of the cut plane are $0 \leq y \leq 3, 0 \leq z \leq 2.2$. The WEC moves downward with the negative velocity in the z-direction in Figure 4.5a, and it moves towards the free surface in Figure 4.6a. The velocity on the angled surfaces and sharp edges is higher which will lead to higher dynamic pressures in these regions. It is shown in Figures 4.5a and 4.5b that there is

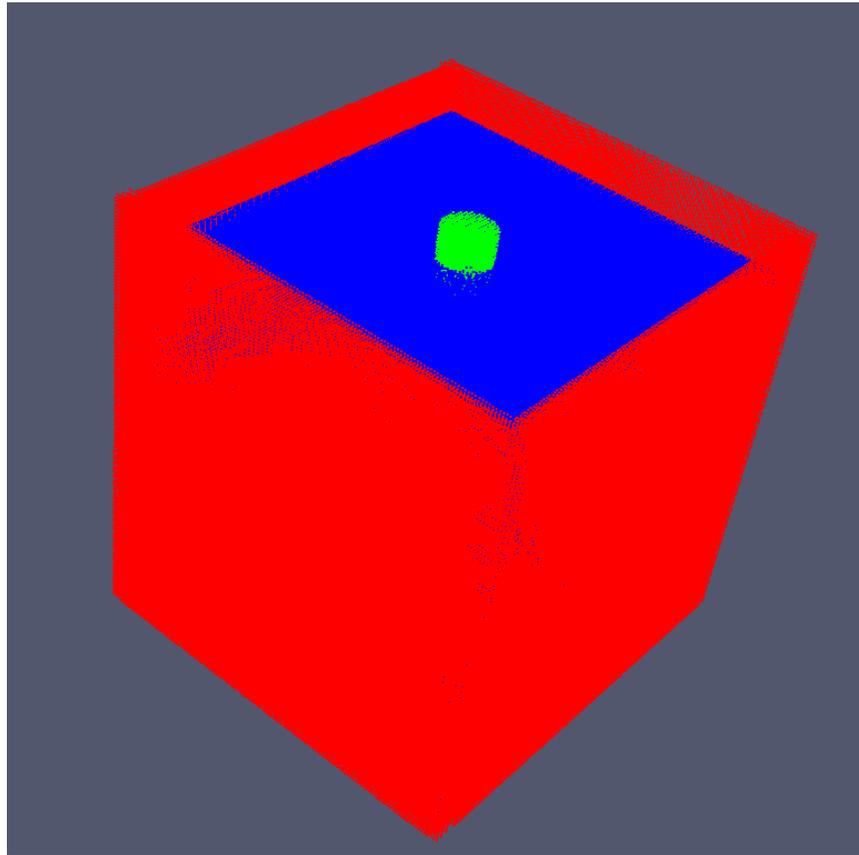


Figure 4.3: 3D domain

a stagnation region on the upper part of the WEC where the particles between angled surfaces meet each other. As shown in Figure 4.6a, the stagnation region occurs in the lower part in comparison with Figure 4.5a at ($t = 1.15$ s). The small distortion of free surface close to the solid walls is due to the fact that the criteria defined to define the free surface particles is not satisfied for the free surface particles close to the walls. The kernel support domain is complete for these particles because of the dummy boundary particles. The damping zone is applied to 0.5 m from the tank wall. It can be seen from the velocity magnitudes at 1.15 s, 1.75 s, in Figures 4.5b and 4.6b, that no reflection from the walls are found. The velocity of free surface particles were also checked at 4 s and it was found that the velocities close to the tank walls indicated no reflection.

The experiment involved gradual ramping of spar from a stationary to a steady periodic displacement. The experiment was performed for 120 seconds, which included ramping and periodic displacement of the spar. The periodic profile in the

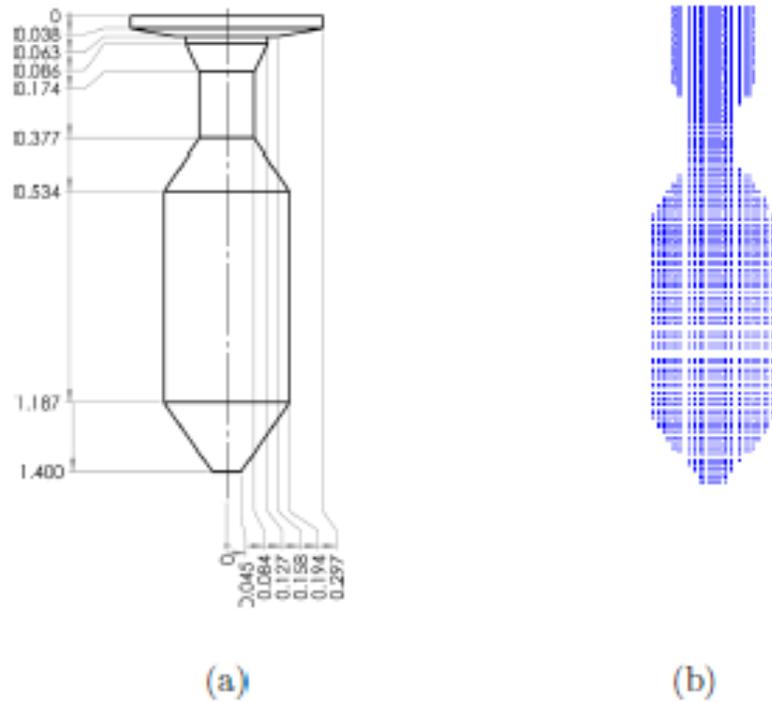


Figure 4.4: (a) Sketch of the WEC under water, (b) Particle representation of the WEC in SPH

experiment immediately after the completion of ramping is chosen for comparison to avoid possible wave reflection from the tank walls. The hydrodynamic force from the experiment is obtained by subtracting the inertia force from the measured force:

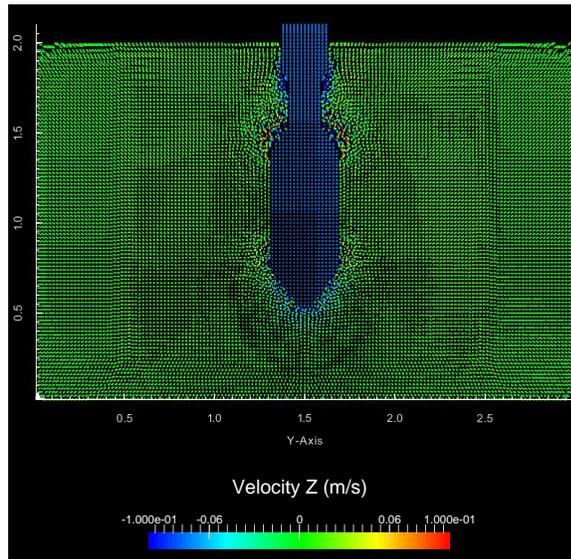
$$F_{exp} = F_{Hydrodynamic} + F_{Inertia} \quad (4.5.2)$$

$$F_{Inertia} = -z\omega^2 M \quad (4.5.3)$$

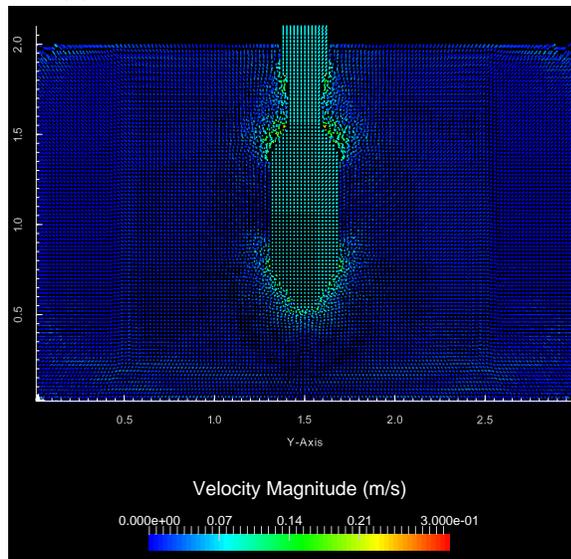
where $M = 115 \text{ kg}$ is the mass of the spar. The hydrodynamic force in the SPH simulations includes the hydrostatic force and should be subtracted as well.

In Figure 4.7, it is shown that the WEC motion in the simulation follows the same motion as in the experiment.

In Figure 4.8, the time history of the hydrodynamic force and the motion of the WEC from the SPH simulations are compared with 4.0 s of the periodic part in the experiment. In order to let the particles settle down in the tank, the simulation was

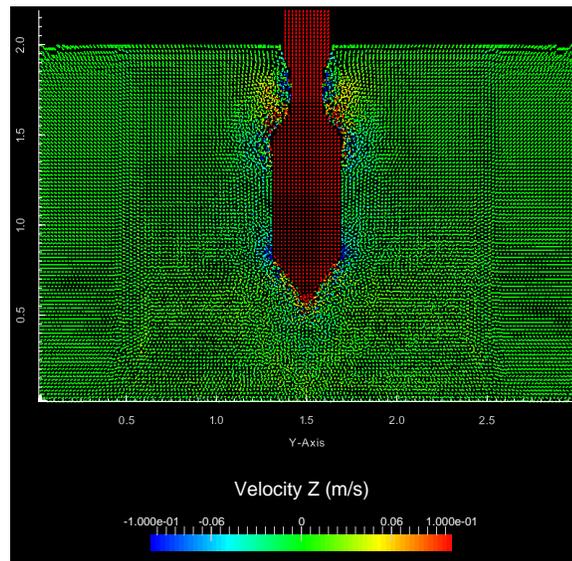


(a) Velocity of particles in z-direction

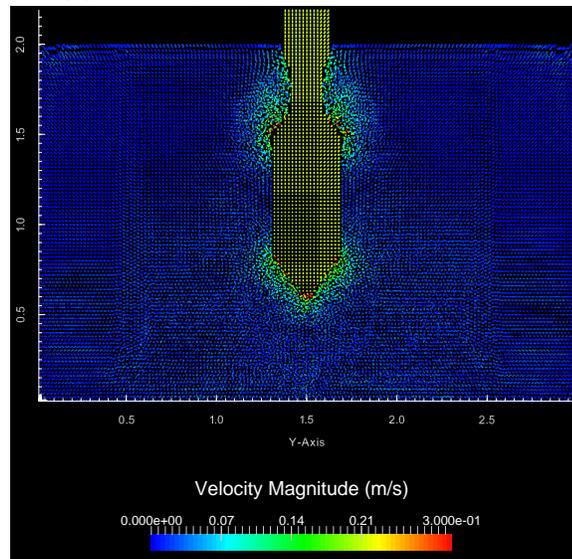


(b) Velocity magnitude of particles

Figure 4.5: Particles velocities at $t = 1.15$ s, ($1.47 \leq x \leq 1.53, 0 \leq y \leq 3, 0 \leq z \leq 2.2$)



(a) Velocity of particles in z-direction



(b) Velocity magnitude of particles

Figure 4.6: Particles velocities at $t = 1.75$ s, ($1.47 \leq x \leq 1.53, 0 \leq y \leq 3, 0 \leq z \leq 2.2$)

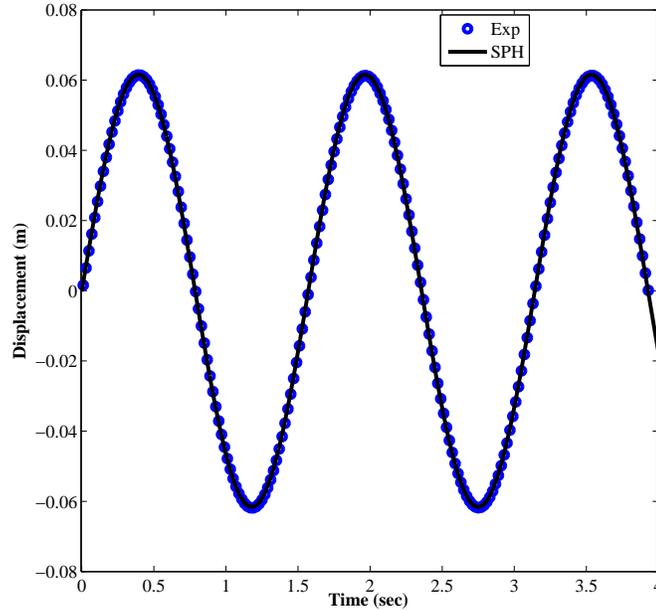


Figure 4.7: Displacement. $A = 0.06 \text{ m}$ and $\omega = 4 \text{ rad/s}$

run for the first 2000 time steps without moving the WEC. The hydrodynamic force obtained from the SPH simulation is in reasonable agreement with the experimental data. The high amplitude spikes in the force at the beginning of the simulation is due to transient effects. The WEC particles move based on the Eq. 4.5.1 and consequently their velocities take $V_z = A\omega\cos(\omega t)$. At the beginning of the simulation, t is small and the WEC particles velocities read $V_z \approx A\omega$ while the fluid particles are in rest. This leads to the high amplitude spikes in the pressure.

The total force from SPH simulation is compared with the experimental data and is shown in Figure 4.9. In order to obtain the total force, Eq. 4.5.2, the inertia force is calculated by Eq. 4.5.3 and also adopted from experiment. The total force when the inertia force is calculated by Eq. 4.5.3 is shown in solid black line in Figure 4.9 and when the inertia force is adopted from experiment ($F_{inertia} = Ma_{exp}$) it is shown in solid green line. The discrepancy between two methods is due to the experimental and theoretical acceleration. The hydrodynamic force has less effect when the larger inertia force is included. It is shown that the simulation result for total force is in good agreement with experimental data when the inertia force is adopted from experiment.

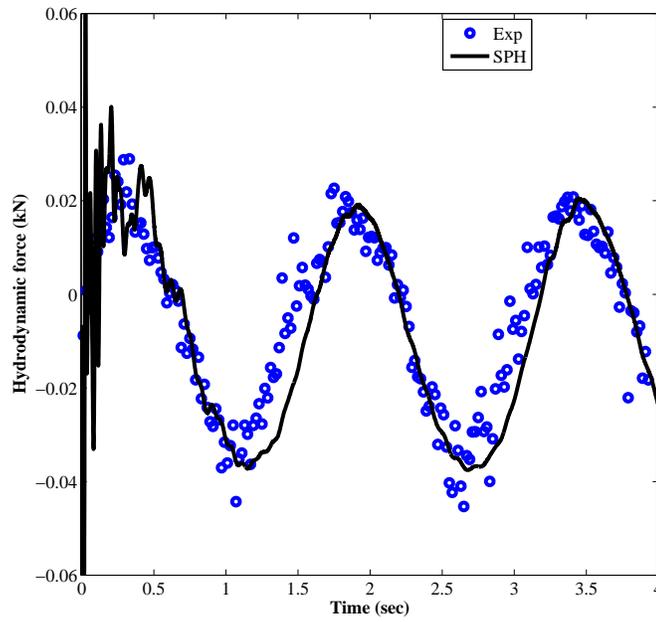


Figure 4.8: Hydrodynamic force. $A = 0.06\text{ m}$ and $\omega = 4\text{ rad/s}$

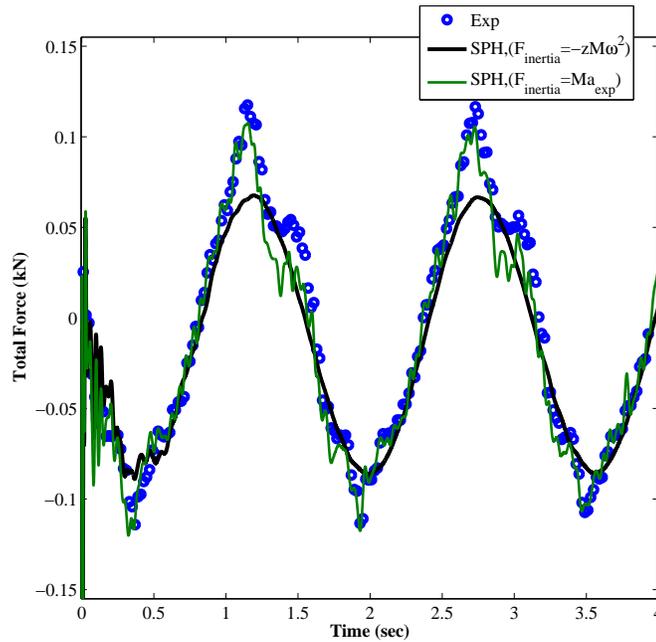


Figure 4.9: Total force. $A = 0.06\text{ m}$ and $\omega = 4\text{ rad/s}$

4.6 Conclusion

In this paper, an explicit incompressible Smoothed Particle Hydrodynamics (ISPH) method was applied to calculate the hydrodynamic force from the spar of 1:25 scale model WEC. It is shown that the hydrodynamic force calculated from the simulation is small compared to the inertia force. The simulation results are in a reasonable agreement with the experimental data. The discrepancy between the simulation and experimental data can be reduced by using a larger numerical tank and increasing number of particles both of which will increase the computational costs. This study is the initial part of the future work in simulation of this WEC in 6 degrees of freedom motion.

4.7 Acknowledgment

The authors thank the Natural Sciences and Engineering Research Council, Natural Resources Canada, the Pacific Institute for Climate Solutions and the University of Victoria for their financial support. The authors thank the Compute Canada and WestGrid for their computational resources.

Chapter 5

ISPH Modelling for Hydrodynamic Applications Using A New MPI-Based Parallel Approach

This paper is accepted for publication at the Journal of Ocean Engineering and Marine Energy.

Yeylaghi, Shahab, Belaid Moa, Peter Oshkai, Bradley Buckham, and Curran Crawford. "ISPH modelling for hydrodynamic applications using a new MPI-Based parallel approach." *Journal of Ocean Engineering and Marine Energy*.
[http://dx.doi.org/\[10.1007/s40722-016-0070-6\]](http://dx.doi.org/[10.1007/s40722-016-0070-6]).

This chapter presents a new MPI-based parallel approach for SPH. The focus in this chapter is on different parts of the parallelization scheme. Several test cases are presented in this chapter to test different aspects of the proposed scheme.

5.1 Abstract

This paper presents a novel parallel design for Smoothed Particle Hydrodynamics (SPH). The Message Passing Interface (MPI), standard for distributed memory programming, is used to parallelize the code as a necessary precursor to future multi-GPU implementation. In the proposed parallelization scheme, the domain decomposition

is performed based on both spatial and particle decompositions in order to reach efficient and well-balanced parallelization. To take the advantage of memory locality, the Peano-Hilbert ordering of the underlying cells, which allows particles that are spatially close to also be close in memory, is adopted. In our scheme, the dynamic load balancing is performed in three Cartesian dimensions as a feedback system that recognizes the particle imbalance and applies the load balancing accordingly. The incompressible SPH (ISPH) method along with an eddy viscosity turbulence model is solved explicitly in the proposed parallel scheme, rather than the typical weakly compressible or implicit incompressible Poisson formulations. The performance of the code is tested for several test cases including a dam-break problem impacting on a short box. The simulation results for water depth at two locations in the tank and two pressure sensors on the box are compared with experimental data and reasonable agreement is achieved.

5.2 Introduction

Smoothed Particle Hydrodynamics (SPH) is well-suited for solving problems with large deformations such as complex free surface flows due to its Lagrangian and grid-less characteristics. SPH has been applied to a wide range of engineering problems such as solid mechanics [86, 87], internal viscous flows [79, 80, 88], heat transfer [89, 90], wave propagation [23, 48, 68], wave breaking [49, 83], and fluid-structure interactions [51, 56, 64, 83, 91–93]. A recent review of the method for free surface flows is reported by Violeau and Rogers [94]. The main advantage of relatively computationally expensive SPH over Eulerian grid-based CFD methods is the ability of the former to capture the interface in free surface flows without the need to solve an additional equation such as volume-of-fluid method. On the other hand, solid boundary conditions can be more challenging to implement in SPH compared to Eulerian CFD methods.

SPH was originally developed to study compressible flows with no solid boundaries. Therefore, the method requires modifications for fluid simulations. Incompressible fluids in SPH can be modeled either by relating the pressure to the density of a particle by using a stiff equation of state, or by solving Poisson’s equation to determine the pressure. The two methods are referred to as weakly compressible SPH (WCSPH) by Monaghan [23] and incompressible SPH (ISPH) by Cummins and Rud-

man [24], respectively. WCSPH approach has proved to exhibit large fluctuations in pressure due to the way that formulation calculates the pressure. In order to reduce these fluctuations, several methods have been proposed such as using ISPH method. Cummins and Rudman [24] was the first to use a projection based method in SPH. Shao and Lo [65] proposed an ISPH method with the density variation term for the source term in Poisson’s equation to simulate Newtonian and non-Newtonian free surface flows. Gotoh et al. [95] presented an ISPH-LES modeling approach for simulating wave-breakwater interactions. Lee et al. [50] proposed a new way of discretization for Poisson’s equation, and standard boundary conditions for ISPH method. Khayyer et al. [32] presented a corrected ISPH (CISPH) method based on a variational approach to preserve the angular momentum. The CISPH is used to simulate breaking waves accurately. Gotoh et al. [96] proposed an enhanced ISPH method with a higher order Laplacian and an error compensating source of Poisson’s equation to simulate violent sloshing flows. Ren et al. [97] proposed a coupled SPH method for flows with low Reynolds number in which the improved ISPH method is used in the interior regions and the WCSPH is used near the free surface. Violeau and Leroy [98] investigated the critical time step for numerical stability in the ISPH method algorithm and proposed a maximum CFL number as a function of viscosity, velocity and spatial discretization. Gotoh and Khayyer [99] presented a review on projection based particle methods for ocean engineering applications in terms of stability, accuracy enhancement, and boundary condition improvements. Lind et al. [100] presented an approach to determine 3D loads on floating bodies in both breaking and non-breaking waves by using ISPH method and Froude–Krylov forcing. In all the above mentioned papers, ISPH is proven to be an appropriate method for problems involving force calculations and fluid-structure interactions.

SPH is computationally more expensive than the conventional Navier–Stokes solvers, especially in 3D [7]. Additionally, the ISPH method has more complexity compared to the WCSPH method due to the solving of Poisson’s equation [75], however as mentioned above, it is proven to have a smoother pressure field. In order to apply ISPH for realistic engineering problems, a large number of particles, often in the order of 10^6 – 10^9 particles are required to perform an accurate simulation (e.g. see [101], [102], [103]). Without a parallelization method, it would not be feasible to perform such simulations. In a parallel simulation, we divide large-scale problem into smaller problems and solve them simultaneously, therefore enormous computational

time can be saved. For instance, Dominguez et al. [104] reported the simulation time for a dam-break impacting on a tall structure by using approximately 1 million particles for 26,493 time steps. The simulation times reported for a single-core CPU, CPU with 8 threads, GPU with 240 cores and GPU with 480 cores are 40.7 hours, 9.1 hours, 1.5 hours and 0.7 hours, respectively [104]. Using the parallelized SPH method allowed simulations of real-life 3D problems, such as ship wave breaking [41], large wave interacting with an oil rig [101], landslide generated waves [102] and realistic wave-coastal structures interaction [103]. Such studies show that parallelization is essential for large-scale 3D SPH simulations and thus for practical ocean engineering applications using ISPH.

Recently, Central Processing Unit (CPU) and Graphics Processing Unit (GPU) based parallelizations have been employed in SPH to improve the performance of the method in simulating computationally expensive problems. SPHysics (<http://www.sphysics.org>) is an open-source program available for 2D and 3D applications in both serial and parallel versions ([40, 73, 74]). Ferrari et al. [72] used an MPI-based SPH code to study free surface flows such as 3D dam-break on structures. Marrone et al. [41] used a 3D hybrid MPI and OpenMP-based standards to study ship wave breaking. Cherfils et al. [105] proposed a parallel SPH program to solve unsteady free surface flows such as a stretching liquid drop and a dam-break problem. Several important features of massively parallelized particle methods (e.g. SPH) such as domain decomposition, particle interactions and load balancing on distributed memory systems are discussed by Oger et al. [39].

CPU-based parallelization can be applied in SPH by using Open Multi Processing (OpenMP) or MPI standards. While the OpenMP standard offers an easier programming environment, improvements in computational speedup achieved to date using this framework have been limited [35]. Consequently, CPU-based parallelization is mostly employed in SPH by using the distributed memory MPI standard. Strong computational power and the lower cost of GPUs in comparison to CPUs make them a suitable parallel design for SPH [34]. A few GPU-based parallelization studies have been reported for SPH [34, 106, 107]. DualSPHysics is an open-source GPU-based WCSPH program developed for free surface flow problems [74]. Although GPU-based parallelization offers optimum hardware acceleration, it could also bring some challenges [106]. One of the main issues in using GPU-based codes is that all the data should fit in the GPU memory [34]. Therefore, to overcome the memory limitation for

computationally expensive problems, (e.g. more than 30 million particles, as reported by Valdez-Balderas et al. [108]), multi-GPU systems are required. The multi-GPUs systems employ the MPI standard for communication between devices [108]. Hence, developing the MPI program base in this paper is a necessary precursor to massively parallel GPU implementation.

Although ISPH allows larger time steps, it is computationally more expensive than WCSPH due to the solving of Poisson’s equation. In previous parallel ISPH codes, Poisson’s equation is solved implicitly using specific solvers and algorithms [37]. It was mentioned by Guo et al. [37] that almost 47% of the total computation time is spent on solving Poisson’s equation. In SPH, the domain decomposition is performed either by using particle decomposition [37, 38] or spatial decomposition [39]. The Peano-Hilbert curve is employed in SPH by Springel [38] in Gadget2 to order all the particles for astrophysical problems. Also, Guo et al. [37] performed the whole domain decomposition by using the Peano-Hilbert curve in their ISPH code. In SPH, particles change positions due to the Lagrangian nature of the method. This requires a special parallelization scheme with good load balancing and communications between processors [37, 72]. Several load balancing methods are applied in SPH, such as the Orthogonal Recursive Bisection (ORB) method [39], block partitioning [40, 73, 74], and the Peano-Hilbert space filling curve [37, 38]. These methods attempt to load balance on each processor but require sophisticated algorithms or generate complicated sub-domain shapes.

This paper introduces a novel 3D in-house MPI-based parallel ISPH code for free surface flows. For computationally expensive problems, having an MPI program is a necessity regardless of the choice of processing hardware (CPU or GPU). The goals of this new parallel ISPH code development are as follows: (i) An explicit alternative is used to Guo et al. [37]’s implicit solution to Poisson’s equation. As a trade off, the time steps in the explicit ISPH should be smaller than in the implicit ISPH. However, by solving Poisson’s equation explicitly the code remains totally explicit for further extensions to GPU parallelization. (ii) A simple domain decomposition is presented by using both spatial and particle decompositions. The spatial domain decomposition is performed based on assigning the cells to processors in a balanced manner and the particle domain decomposition is carried out by a proposed departmental algorithm. (iii) The Peano-Hilbert ordering of the underlying cells is used to take advantage of memory locality. This allows particles that are spatially close to also be close in

memory. Our scheme orders only the cells instead of performing the whole domain decomposition [37, 38] and, as such, it is efficient and is performed only when necessary (applied with dynamic load balancing). (iv) The dynamic load balancing is performed as a feedback system that senses the particle imbalance and applies the load balancing accordingly. The above mentioned developments are organized in the current paper as follows: The governing equations for incompressible SPH are presented in Sect. 5.3. The domain decomposition and initial load balancing are discussed in Sect. 5.4.1. The Peano-Hilbert sorting and the dynamic load balancing are presented in Sects. 5.4.2 and 5.4.3, respectively. Finally, in Sect. 5.5, the code is validated for a few test cases such as the impact problem of a dam that breaks on a short structure.

5.3 SPH Methodology

In SPH, particles represent macroscopic volume of fluids. Each particle has mass, density, velocity and other material properties. A function of interest, e.g. f , in SPH is approximated as

$$f(\mathbf{r}) \approx \int_{\Omega} f(\mathbf{x})W(\mathbf{r} - \mathbf{x}, h)dx, \quad (5.3.1)$$

where $W(\mathbf{r} - \mathbf{x}, h)$ is a predefined smoothing function (kernel), Ω is the domain of support for the kernel, \mathbf{r} is the position vector and h is the smoothing length. To determine a property at a certain particle, interpolation between the particle's property and that of its neighboring particles has to be performed. The interpolation is conducted by using the kernel ($W(\mathbf{r} - \mathbf{x}, h)$). The particle approximation of Eq. 5.3.1 in SPH is defined as

$$f_i \approx \sum_{j=1}^N f_j W(\mathbf{r}_i - \mathbf{r}_j, h) V_j, \quad (5.3.2)$$

where N is the number of neighboring particles and V_j is the particle volume. In this study, we use the fifth order Wendland kernel [46] as

$$W(q) = W_{\alpha} \begin{cases} (1 + 2q)(1 - \frac{1}{2}q)^4 & 0 \leq q \leq 2 \\ 0 & q \geq 2, \end{cases} \quad (5.3.3)$$

where $W_\alpha = 21/16\pi h^3$ in 3D, $q = |\Delta l|/h$ and Δl is the interparticle distance. In SPH, the Lagrangian description of the Navier-Stokes equations are solved for each particle. The Lagrangian description of the governing equations are formulated as

$$\nabla \cdot \mathbf{v} = 0, \quad (5.3.4)$$

$$\frac{D\mathbf{v}}{Dt} = -\frac{1}{\rho}\nabla p + \mathbf{g} + \nu_0\nabla^2\mathbf{v} + \frac{1}{\rho}\nabla \cdot \vec{\tau}, \quad (5.3.5)$$

where ρ is the fluid particle density, \mathbf{v} is the particle velocity vector, t is time, p is the particle pressure, \mathbf{g} is the gravitational acceleration vector and ν_0 is the laminar kinematic viscosity. $\vec{\tau}$ is the sub-particle scale (SPS) turbulence stress tensor in particle form, which is analogous to the sub-grid scale (SGS) in grid-based CFD methods [109]. In the SGS model, the large-scale turbulence is resolved while the small-scale is modeled. The same technique is applied in SPH, except in SPH, we deal with particle size instead of grid cell size. The eddy viscosity assumption is often used to model the SPS turbulence stress, $\vec{\tau}_{ij}$, as

$$\vec{\tau}_{ij} = 2\rho(\nu_t S_{ij} - \frac{1}{3}\delta_{ij}k), \quad (5.3.6)$$

where S_{ij} is SPS strain tensor, k is the turbulence kinetic energy, δ_{ij} is the Kronecker delta function and ν_t is the eddy viscosity. The eddy viscosity based on Smogorinsky sub-grid scale model [110] was applied to represent the effect of small scale turbulence in SPH [109]. Therefore, ν_t is calculated as

$$\nu_t = (C_s\Delta)^2|S|, \quad (5.3.7)$$

where

$$|S| = (2S_{ij}S_{ij})^{1/2}, \quad (5.3.8)$$

and C_s is the Smagorinsky constant, which is taken to be 0.12 in this paper. The momentum equation in SPH particle form is written as (see [65])

$$\begin{aligned}
\frac{D\mathbf{v}_i}{Dt} = & \mathbf{g} - \sum_j m_j \left(\frac{p_i}{\rho_i^2} + \frac{p_j}{\rho_j^2} \right) \cdot \nabla_i W_{ij} \\
& + \sum_j \frac{4m_j(\mu_i + \mu_j)}{(\rho_i + \rho_j)^2} \frac{\mathbf{r}_{ij} \cdot \nabla_i W_{ij}}{r_{ij}^2 + \eta^2} \mathbf{v}_{ij} \\
& + \sum_j m_j \left(\frac{\vec{\tau}_i}{\rho_i^2} + \frac{\vec{\tau}_j}{\rho_j^2} \right) \cdot \nabla_i W_{ij},
\end{aligned} \tag{5.3.9}$$

where $\mathbf{r}_{ij} = \mathbf{r}_i - \mathbf{r}_j$, $\mathbf{v}_{ij} = \mathbf{v}_i - \mathbf{v}_j$, ρ_i is the density of particle i with velocity \mathbf{v}_i , m_j is the mass of particle j , \mathbf{v}_j is the velocity of particle j , μ is the laminar dynamic viscosity, $\eta = 0.1h$ and W_{ij} is the kernel function based on the $\mathbf{r}_i - \mathbf{r}_j$.

The explicit ISPH method by Hosseini et al. [30] adopted in this paper is based on the two-step projection method widely used in Eulerian-based CFD methods [24]. In this paper, the divergence free velocity is used for the source term in Poisson's equation in the explicit ISPH, based on the correction proposed by khayyer et al. [32], while the previous explicit ISPH methods, such as [30, 31] used the density invariance method. Also, the LES turbulence model is added in the current explicit ISPH method. In the prediction step, the intermediate velocity is calculated using the viscous and body forces as

$$\mathbf{v}^* = \mathbf{v}(t) + \Delta t(\mathbf{g} + \nu_0 \nabla^2 \mathbf{v} + \frac{1}{\rho} \nabla \cdot \vec{\tau}), \tag{5.3.10}$$

$$\mathbf{r}^* = \mathbf{r}(t) + \Delta t \mathbf{v}^*, \tag{5.3.11}$$

where \mathbf{v}^* is the intermediate velocity and \mathbf{r}^* is the intermediate position. Poisson's equation needs to be solved for pressure at each time step is written as

$$\nabla \cdot \left(\frac{\nabla p}{\rho} \right) = \frac{\nabla \cdot \mathbf{v}^*}{\Delta t}, \tag{5.3.12}$$

Hence, Poisson's equation in SPH particle form can be formulated as

$$\sum_j \frac{8m_j}{(\rho_i + \rho_j)^2} \frac{p_{ij} \mathbf{r}_{ij} \cdot \nabla_i W_{ij}}{r_{ij}^2 + \eta^2} = \frac{-1}{\Delta t} \sum_j V_j (\mathbf{v}_i^* - \mathbf{v}_j^*) \nabla_i W_{ij}, \tag{5.3.13}$$

and is solved explicitly by using the method described by Yeylaghi et al. [35]. In the correction step, the velocity is corrected as

$$\mathbf{v}^{t+1} = \mathbf{v}^* + \Delta t \left(-\frac{1}{\rho} \nabla p \right). \quad (5.3.14)$$

The criteria used for defining a free surface particle involves calculation of an artificial density (ρ_f) for each fluid particle. If $\rho_f < 0.95\rho_0$, the particle is marked as a free surface particle. In this paper, fixed dummy particles are used to model the solid wall. The method described by Adami et al. [33] is used to calculate the pressure for boundary particles from the surrounding fluid particles as

$$p_w = \frac{\sum_i p_i W_{wi} + (\mathbf{g} - \mathbf{a}_w) \sum_i \rho_i \mathbf{r}_{wi} W_{wi}}{\sum_i W_{wi}}, \quad (5.3.15)$$

where p_w is the boundary particle's pressure and \mathbf{a}_w is acceleration of the wall ($\mathbf{a}_w = 0$ for stationary walls).

5.4 Parallelization Scheme

The Message Passing Interface (MPI) standard for distributed memory programming is used to parallelize our existing OpenMP-based SPH code [35]. It was mentioned by Yeylaghi et al. [35] that the OpenMP approach is easier to implement, but the efficiency of the code is limited for the ocean engineering problems. Hence, an MPI parallelization is required for the simulation of realistic problems. The majority of Eulerian-based CFD methods computations are performed on a fixed mesh, whereas, SPH computes the properties of the fluid based on a set of moving particles (since it is a Lagrangian-based method). Such moving particles are required to be tracked and assigned to different processors. Unfortunately, even if the initial particle distribution is uniform across multiple processors, the number of particles assigned to a processor can vary substantially and, therefore, cause a large load imbalance compared to the mesh-based methods. Hence, any parallel code in SPH should take care of load imbalance and try to minimizing the communications [37]. In our code, ghost cells are used for overlapping between MPI communications same as [40, 74]. The domain decomposition and load balancing are discussed in the following sections.

5.4.1 Domain Decomposition and Initial Load Balancing

In SPH, the domain decomposition is performed either by using a particle (data) decomposition [37, 38] or spatial decomposition [39]. Data decomposition is mostly used by employing the Peano-Hilbert curve in SPH [37, 38]. The N-body SPH codes [38] use (gravitational) trees and Peano-Hilbert curve-fitting for performing domain decomposition and distributing the loads between processors. In this paper, the domain decomposition is performed based on both spatial and particle decompositions. The Peano-Hilbert curve is used only to order the cells for each processor as opposed to the previous studies in [37, 38] where the whole domain is decomposed using the space filling curve.

5.4.1.1 Spatial Domain Decomposition

The domain decomposition algorithm has two passes. In the first pass of the algorithm, the domain of simulation is decomposed into cells based on its three geometrical dimensions (green lines in Figure 5.1a). The cells are chosen to be of size $2h$, where h is the particle smoothing length. We then assign the cells to the MPI processor in a balanced manner (red lines in Figure 5.1a). Then, the temporary creation and counting of particles are carried out for each processor (sub-domain) and in each cell. If the number of cells in direction i is denoted by nc_i and the number of processors by $proc_i$, then the number of cells assigned to each processor is $\approx nc_i/proc_i$. As such, our first initial decomposition is purely geometric and is based on the load balancing of the cells. The blue area in Figure 5.1 represents initial particle positions, so that clearly further processor balancing should be done based on actual particle positions.

5.4.1.2 Domain Decomposition Based on Particles (Initial Load Balancing)

Several domain decomposition (load balancing) methods are applied in SPH, such as the Orthogonal Recursive Bisection (ORB) method [39], block partitioning [40, 73, 74], and the Peano-Hilbert space filling curve [37, 38]. As opposed to these available load balancing approaches that attempt to load balance on each processor but require sophisticated algorithms and generate complicated sub-domain shapes, in our proposed methodology the load is simply balanced on the *departments* instead

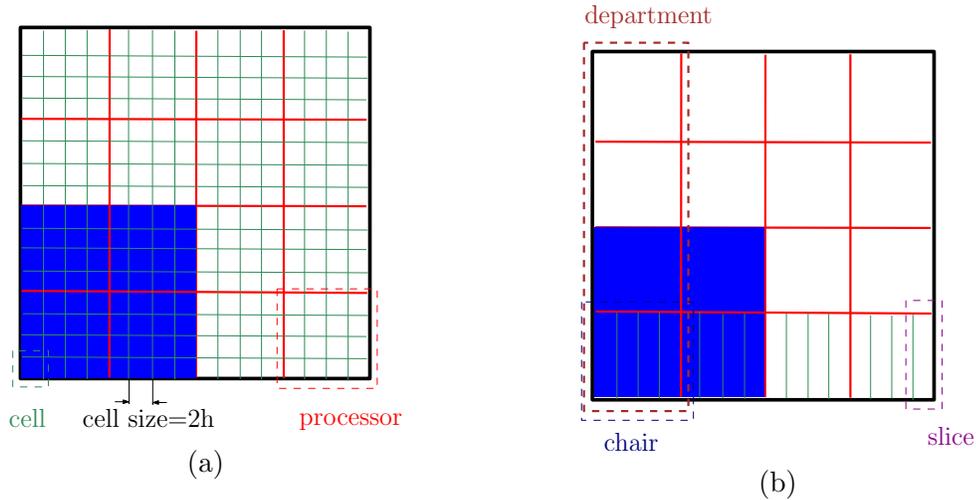


Figure 5.1: Domain decomposition and initial load balancing (blue color corresponds to the initial particle region). (a) Spatial bounding boxes: cells (green) and processors (red), (b) *Departments* (dashed red) and *chairs* (dashed black).

and is therefore much more efficient. Moreover, it respects the 3D Cartesian nature of the geometry.

In the current code, the initial load balancing is based on particles, and relies on the MPI's communicator partitioning capability. An MPI 3D Cartesian topology is used to associate 3D coordinates to processors, enabling us to easily identify the 26 neighbors for each processor. The processors are grouped into a 2D Cartesian topology, which we call *departments* (Figure 5.1b). For each processor, the number of particles in each slice and in each direction (see Figure 5.1b) is reduced to the root processor of the 2D Cartesian topology, called the *chair* (Figure 5.1b). Each chair then computes the number of cells it should hold so that the number of particles in each department is comparable to the rest of departments in a given direction. Using that information, the initial load balancing of particles in each direction, x , y and z is performed. The boundaries of the sub-domains are then modified accordingly and the required memory is allocated. The initial load balancing is summarized in Algorithm 1.

Algorithm 1 Load balancing (initial)

- 1: Group the processors into *departments* in each direction (Figure 5.1b)
 - 2: For each processor, reduce the number of particles in each slice and in each direction to the *chairs* (see Figure 5.1b)
 - 3: Each chair then computes the number of cells it should hold to balance loads between *departments*
 - 4: Update the sub-domain boundaries for each processor
-

In summary, the first pass of the domain decomposition algorithm is used to gather the information to allocate memory and perform the initial load balancing. In the second pass, the cells are ordered within each processor by using a Peano-Hilbert sorting, the first particle ID in each cell is determined, particles properties are set, and cells are classified into ghost, inner and outer cells. This approach allows us to combine both spatial and particle decomposition to obtain efficient and balanced parallelization. The domain decomposition is summarized in Algorithm 2.

Algorithm 2 Domain decomposition

```

for pass= 1,2
  if pass==1
    1: Create the MPI 3D Cartesian topology
    2: Partition the domain into cells of width  $2h$ 
    3: Perform the spatial domain decomposition based on the cells
    4: Count the particles in each cell on each processor
    5: Carry out the initial load balancing
    6: Update the sub-domain boundaries for each processor
    7: Allocate the memory required
  else
    8: Use the Peano-Hilbert to order the cells
    9: Determine the first particle ID in each cell
    10: Set the particles properties
    11: Classify the cells to: ghost, inner and outer
    12: Determine the particle global and local ID
    13: Go to the main loop
  end if
end for

```

5.4.2 Peano-Hilbert Sorting

In our previous OpenMP-based SPH code [35], a multi-dimensional array was used to associate each particle’s global ID to the cell (c) in which it is located in, and to its local index in the cell. To take advantage of memory locality efficiency, an ordering scheme of the cells is chosen that allows particles that are close together in space to also be close in memory. Figure 5.2 depicts Peano-Hilbert curves in 2D and 3D. The Peano-Hilbert space filling curve maps the 3D space to 1D. Extensive details about the Peano-Hilbert curve and its employment to SPH are reported by Springel [38]. Springel [38] and Guo et al. [37] used the Peano-Hilbert space filling curve for domain decomposition in SPH.

In the current code, each processor sorts and indexes its cells, including the ghost ones, according to the Peano-Hilbert ordering. The particle index on each processor is then computed based on the Peano-Hilbert index of the cell to which the particle belongs, and an incremental local count is used to track the particles inside each cell. In fact in our case, there is no need to redo the Peano-Hilbert sorting every time a particle moves (the Peano-Hilbert curve is not used for domain decomposition). The sorting is redone only when the dynamic load balancer is called. The effect of using the Peano-Hilbert sorting is reported by Springel [38] and Guo et al. [37].

5.4.3 Dynamic Load Balancing

To deal with the Lagrangian nature of fluid particles in SPH, dynamic load balancing is required to achieve an acceptable parallel efficiency. In fact, although the initial load balancing may produce an equal particle partitioning among the processors at the start of the simulation, as the particles move between the processors, the partitions will become imbalanced and make the initial load balancing inefficient. The same argument holds when applying the load balancing after a pre-fixed number of time steps. Therefore, in the proposed method the dynamic load balancer works as a feedback system that recognizes particle imbalances and applies the load balancing accordingly rather than applying the dynamic load balancing after every fixed n time steps, as done in [40, 41].

Our dynamic load balancer senses the imbalance when the particle drift (`maxPartCntDrift`), on a processor reaches a certain threshold. The threshold is defined as a multiplier (`LBPartDriftFract`) of the number of particles at each processor

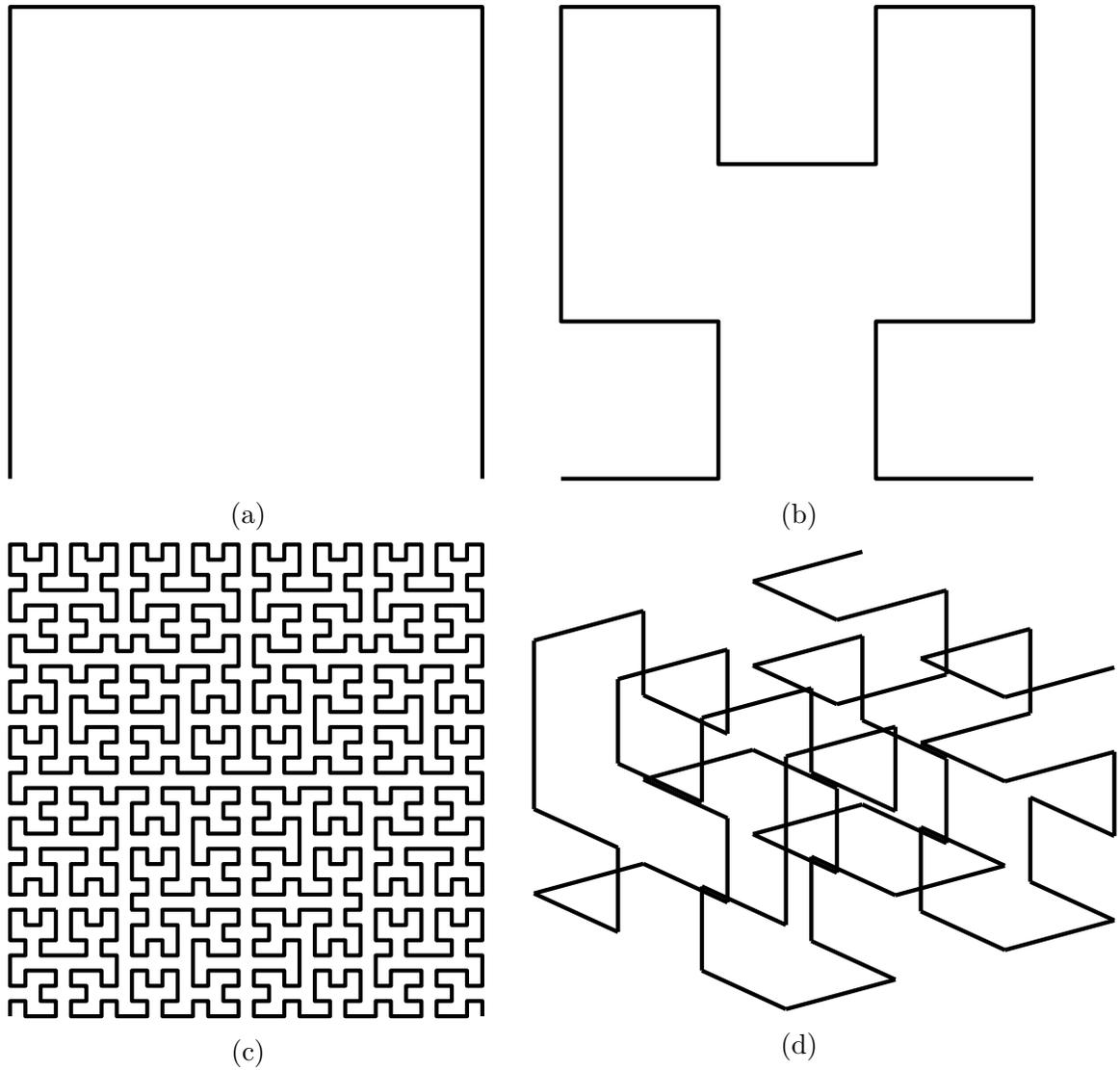


Figure 5.2: Peano–Hilbert curve in 2D and 3D. (a) 2D first order, (b) 2D second order, (c) 2D fifth order, (d) 3D fourth order.

($np/numprocs$). Although the threshold has a fixed value currently, it can be easily made dynamic by introducing the effectiveness of the load balancing into the feedback system. For instance, we could make `LPartDriftFract` dynamic by checking whether the previous load balancing did a major exchange of cells. If yes, decrease `LPartDriftFract`, else increase it. Also, rather than moving one slice (i.e. $2h$) at a time between processors when an imbalance is detected (e.g. [40]), we move as many slices as required.

The dynamic load balancing is summarized in Algorithm 3. In this algorithm, `maxPartCntDrift` is the maximum particle drift among all the processors, `LPartDriftFract` is the coefficient for the load imbalance, and `timestepforLB` is the n time step at which the dynamic load balancing is forced to happen.

Algorithm 3 Dynamic load balancing

- 1: The history of particle numbers at each processor is saved
 - 2: Calculate the particle drift at each processor
 - if** ($(\text{maxPartCntDrift} > \text{LPartDriftFract} \times np/numprocs)$ or remainder of (time step/`timestepforLB`)=0)
 - 3: Call load balancing
 - end if**
-

5.5 Results

In this paper, water properties are set to $\rho=1000 \text{ kg/m}^3$, $\mu=0.001 \text{ kg/m.s}$ and Guillimin¹, Compute Canada’s supercomputer, is used to perform the following ISPH simulations.

5.5.1 Dam Break in a Tank

The proposed domain decomposition and initial load balancing are first tested for a problem of non-uniformly distributed fluid particles. The tank dimensions are $3.22 \text{ m} \times 0.6 \text{ m} \times 1 \text{ m}$ in the x , y and z directions (z vertical) and water column dimensions are $0.6 \text{ m} \times 0.6 \text{ m} \times 0.6 \text{ m}$. In this case, the particle size is set to $\Delta l = 0.005 \text{ m}$ resulting in 3558212 total particles. It is shown in Figure 5.3a that the fluid particles

¹<http://www.hpc.mcgill.ca/index.php/starthere/81-doc-pages/215-guillimin-hardware>

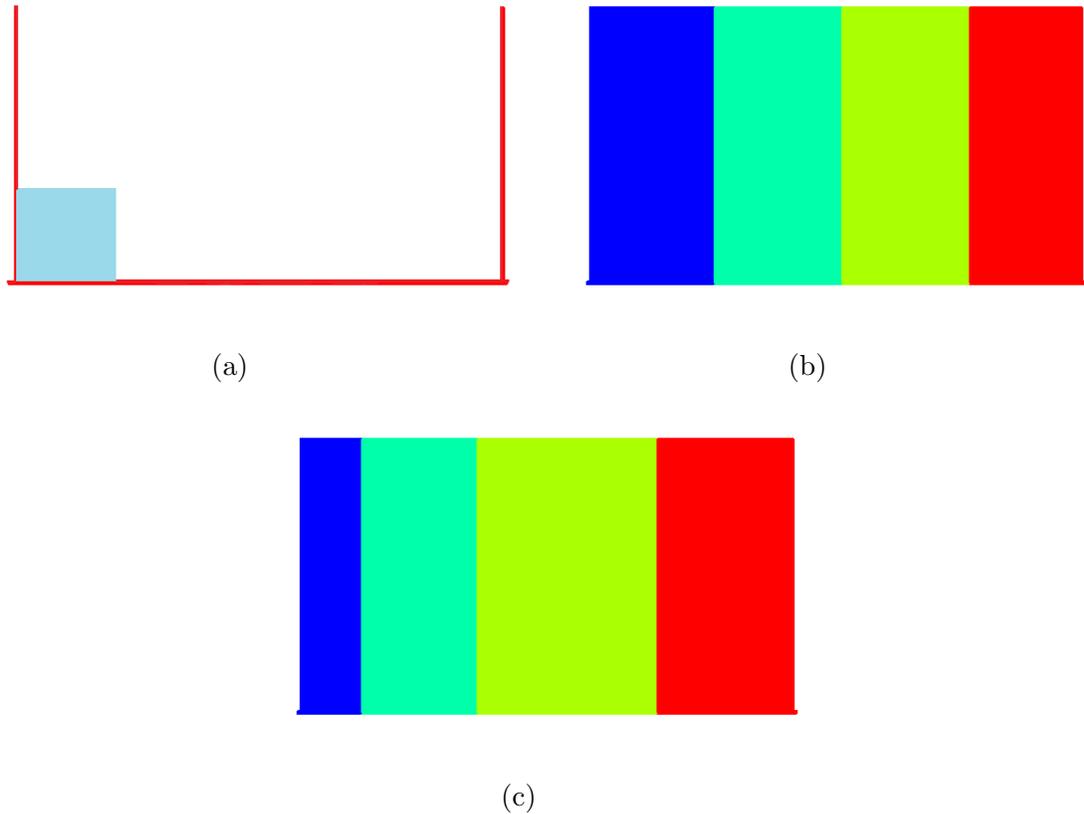


Figure 5.3: Domain decomposition and initial load balancing for a problem where the fluid particles are non-uniformly distributed. The colors in (b) and (c) from left to right corresponds to processors 1-4. (a) The non-uniformly fluid particles in the tank, (b) Domain decomposition without initial load balancing, (c) Domain decomposition with initial load balancing.

are on the left side of the tank at the initial time. The domain decomposition without the initial load balancing is shown in Figure 5.3b. As expected, by applying the initial load balancing the boundaries of processors are moved to the left end of the tank in Figure 5.3c.

In Figure 5.4, the number of particles are compared for four processors at the two steps (b) and (c) described in Figures 5.3b and 5.3c. If we define the load imbalance to be the maximum load divided by the average load, same as [37], the load imbalance is decreased from 2.5 at step (b) to 1.01 at step (c) showing the effectiveness of the proposed domain decomposition and initial load balancing.

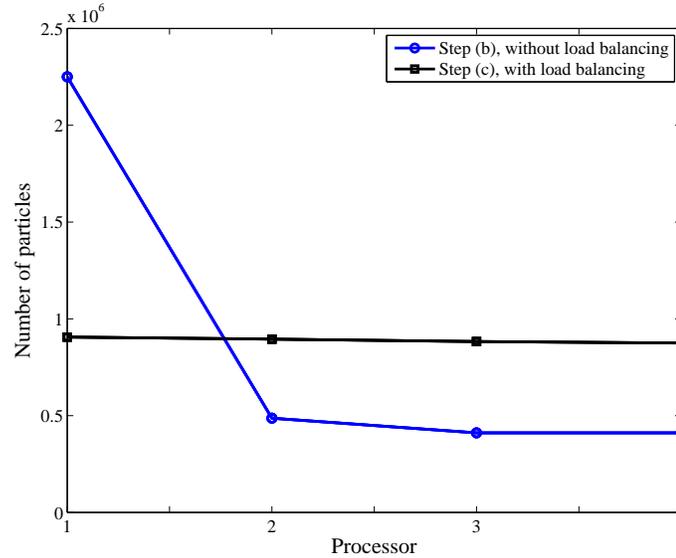


Figure 5.4: Particle numbers for 4 processors in Figure 5.3b and Figure 5.3c.

5.5.2 A Tank Full of Still Water

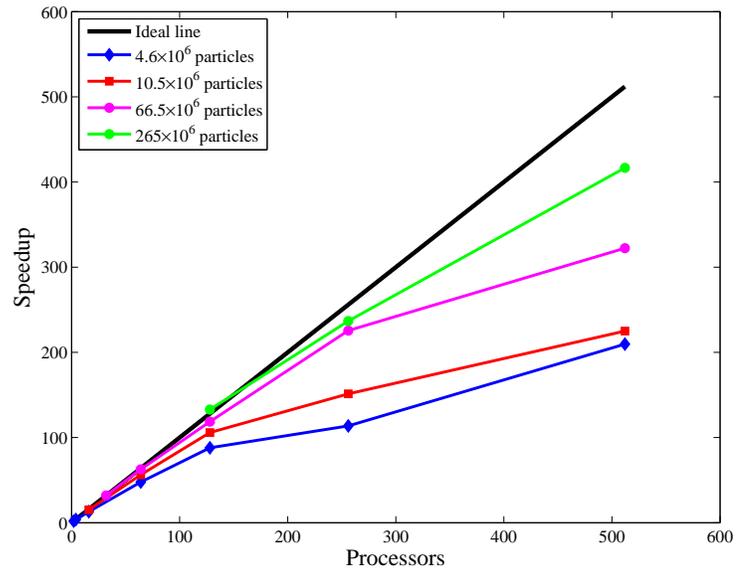
The performance of the code is tested without involving the dynamic load balancing by simulating a tank full of still water. The tank dimensions are 2 m \times 2 m \times 1 m in the x , y and z directions (z vertical) and is filled with water. In order to report the performance of the code, the speedup is measured either by (strong scaling)

$$S = \frac{t_{p=1}}{t_p}, \quad (5.5.1)$$

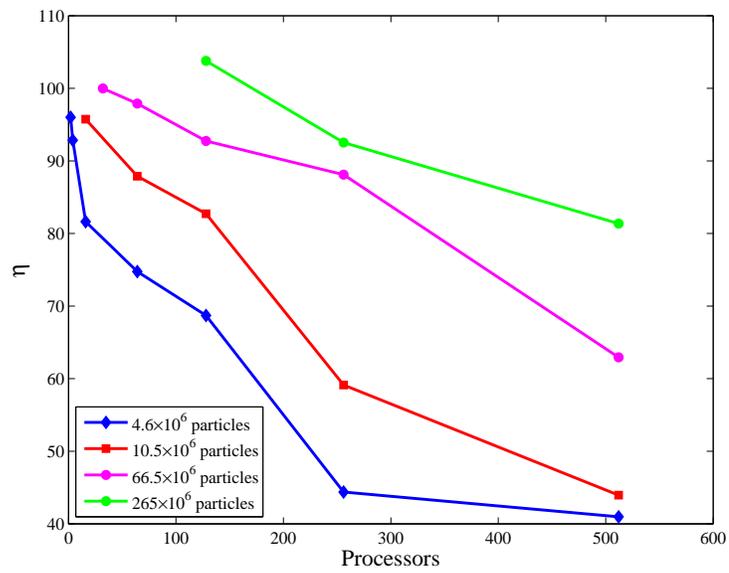
or by (weak scaling)

$$S = \frac{t_{N_{minimum}} N_{minimum}}{t_p}, \quad (5.5.2)$$

where $t_{p=1}$ is the CPU wall time for the serial run, $N_{minimum}$ is the minimum number of processors used, and t_p is the CPU wall time by employing p processors. The speedup and efficiency are shown for 4.6, 10.5, 66.5 and 265 million particles using up to 512 processors in Figures 5.5a and 5.5b, respectively. The strong scaling speedup is reported for the case of 4.6 million particles while weak scaling is reported for 10.5, 66.5 and 265 million particles. It was not possible to run the code in serial mode for the cases with large number of particles. It is shown that by increasing the number of particles the speedup increases. However, the efficiency is lower for the larger number of processors due to increased communications between processors.



(a) Speedup



(b) Efficiency

Figure 5.5: Speedup and efficiency for a tank full of still water.

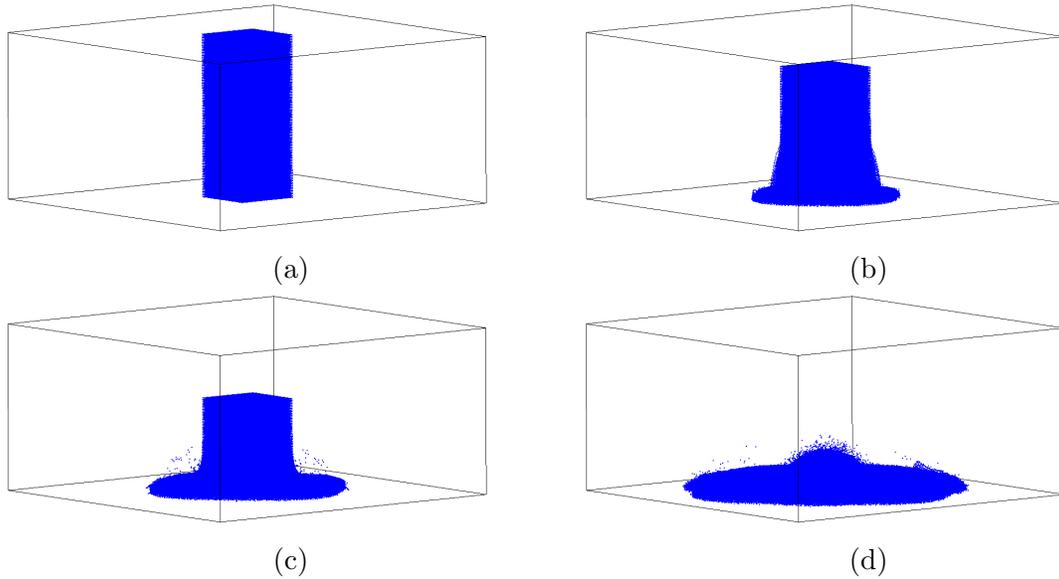


Figure 5.6: Particle distribution at: (a) $t=0$ s, (b) $t=0.25$ s, (c) $t=0.35$ s, (d) $t=0.5$ s.

5.5.3 Column of Water Collapsing in the Middle of a Tank

In order to test the effectiveness of the proposed dynamic load balancing, we considered a column of water collapsing under gravity force in a rectangular tank. The tank dimensions are $2\text{ m} \times 2\text{ m} \times 1\text{ m}$ in the x , y and z directions (z vertical). The water column dimensions are $0.4\text{ m} \times 0.4\text{ m} \times 1\text{ m}$ in the x , y and z directions and located in the middle of the tank. In this case, the particle size is set to $\Delta l = 0.01\text{ m}$ resulting in 731860 particles. The evolution of the water particles are shown in Figure 5.6 at $t=0$ s, $t=0.25$ s, $t=0.35$ s and $t=0.5$ s. Accordingly, the evolution of processors boundaries due to the water particle's motions are shown in Figure 5.7. In Figure 5.7a, the initial domain decomposition and load balancing is performed by employing 8 processors ($2 \times 2 \times 2$). It is shown in Figures 5.7a–5.7d that the processors boundaries are contracting in z direction according to the water particle's motions. The particle numbers are shown for all 8 processors at $t=0$ s, $t=0.25$ s, $t=0.35$ s and $t=0.5$ s in Figure 5.8. It is shown that the load imbalance (the maximum load divided by the average load) is changed from 1.07 at $t=0$ s, where the water column is vertical, to 1.08 at $t=0.5$ s, where the water column is spread on the tank bottom, showing the effectiveness of the proposed dynamic load balancing.

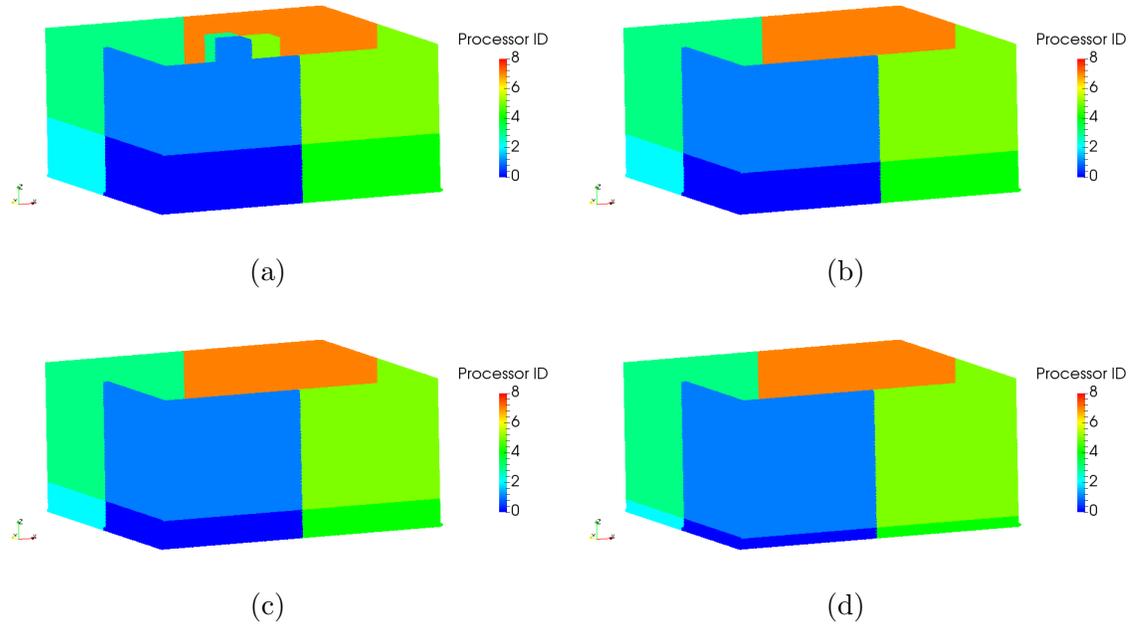


Figure 5.7: Particles are colored by their processor ID in the x , y and z directions (z vertical) at: (a) $t=0$ s, (b) $t=0.25$ s, (c) $t=0.35$ s, (d) $t=0.5$ s.

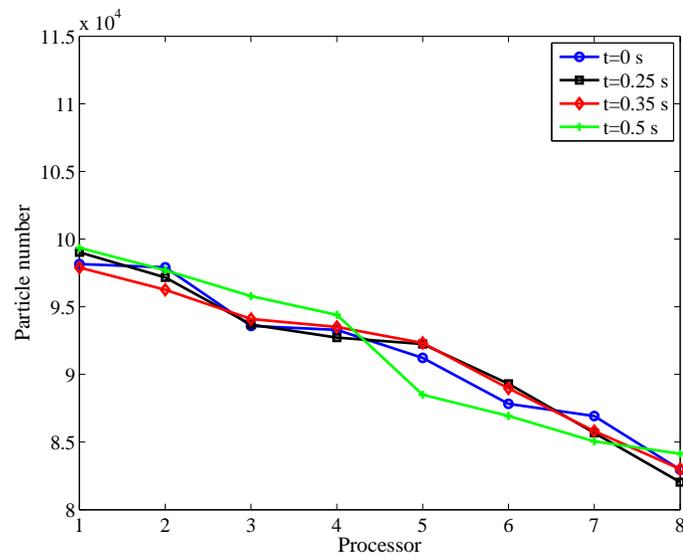


Figure 5.8: Particle numbers for 8 processors.

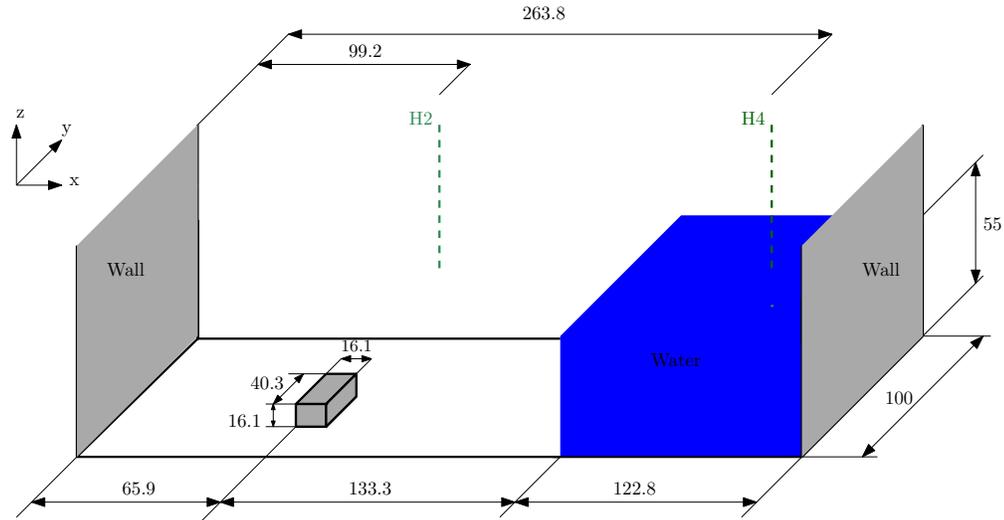


Figure 5.9: Schematic of the experiment reported by [5], dimensions in cm.

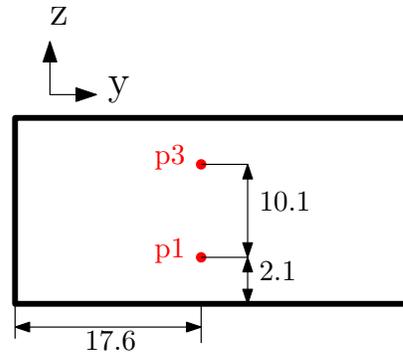


Figure 5.10: Position of pressure sensors on the box [5], dimensions in cm.

5.5.4 Dam Break on a Structure

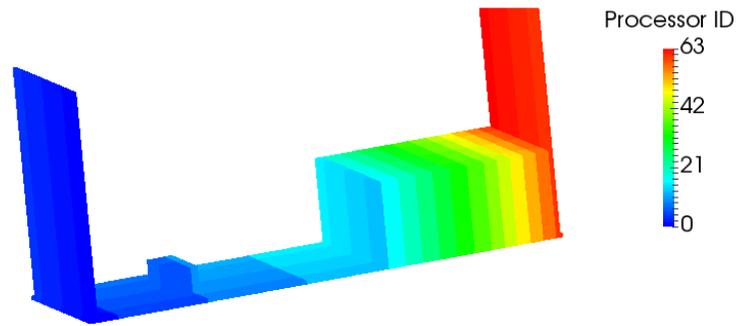
The dam break on a short box is considered here to test the performance of the code. The experiment reported by [5] is chosen to validate the current ISPH results. A schematic of the experiment is shown in Figure 5.9. The current ISPH results are compared for water depth at two probe locations (H4 and H2) and at two pressure sensors on the box (p1 and p3). The location of the pressure sensors (p1 and p3) on the box are shown in Figure 5.10. The pressure sensor p1 is closer to the tank bottom while p3 is closer to the top of the box (Figure 5.10). More details about the experiment is reported by [5].

In Figure 5.11, particles are colored by their assigned processor IDs in x and y directions. The evolution of 64 processors (16 in x -direction and 4 in y -direction) are

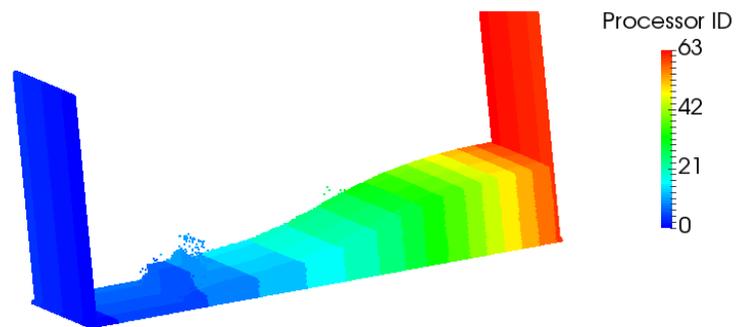
shown at $t=0$ s, $t=0.5$ s and $t=2$ s. At $t=0$ s, more processors are assigned for the location of fluid particles as mentioned in Sect. 5.4.1. It is shown in Figure 5.11b that by applying dynamic load balancing the boundaries of the processors have accordingly moved with fluid particles. At $t=2$ s, the wave impacted on the tank's left end, reflected and impinged the box. Consequently, there are more water particles in the left end of the tank in contrast to $t=0$ s where all the water particles were in the right end. In Fig 5.12, two snapshots of the dam-break on the short structure from current study is compared at $t=0.4$ s and $t=0.56$ s with the corresponding experimental snapshots reported by Kleefsman [6]. The slightly slower motion at $t=0.4$ s in the ISPH results compared to the experimental data ([6]) could be due to the boundary conditions used in the simulations to model the tank floor, similar slower motions are reported by Lee et al. [7]. At $t=0.56$ s, comparing the particle representation with the experimental results reported by Kleefsman [6], shows the jet height is slightly shorter, given that in the simulation water impacts the box slightly later than the experiment as mentioned above. The particles are colored by their velocity at the z -direction in this figure. Comparing this figure with same snapshot ($t=0.56$ s) presented by Khayyer et al. [111], shows a good agreement for the velocity magnitude at the z -direction.

Particles are colored by their pressure and velocity in x -direction (u) at $t=0$ s, $t=0.2$ s, $t=0.4$ s and $t=0.5$ s in Figures 5.13 and 5.14, respectively. The pressure of particles is set to hydrostatic pressure at the initial time ($t=0$ s). At $t=0.5$ s, the water wave moved due to gravity and impinged the box. The impact pressure on the box face is shown in Figure 5.15 at the impact time of $t=0.5$ s. The non-uniform particle distribution and the noise in pressure is due to purely Lagrangian approximations by imperfectly consistent differential operator models as well as the kernel truncation near the free surface as highlighted in [75], [48], [111] and [96]. A particle shifting approach is proposed by Xu et al. [75], and Lind et al. [48] to obtain more uniform particle distribution. An optimized particle shifting to improve uniformity of particle distribution near the free surface is proposed by Khayyer et al. [112] that also leads to smoother pressure distribution.

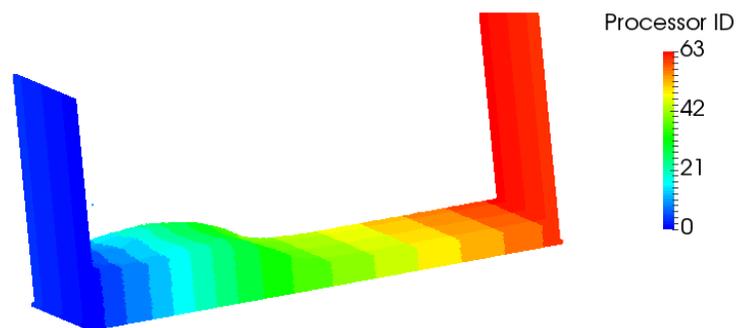
In Figures 5.16a and 5.16b, the time history of water height at probe locations H4 and H2 are compared with experimental data and VOF results reported by Kleefsman et al. [5] and previous semi-implicit ISPH results reported by Lee et al. [7]. The same methodology used by [35] is applied here to locate the free surface positions. A set of



(a)



(b)



(c)

Figure 5.11: Particles are colored by their processors ID at: (a) $t=0$ s, (b) $t=0.5$ s, (c) $t=2$ s. Note that left and right ends are boundary particles (non-moving).

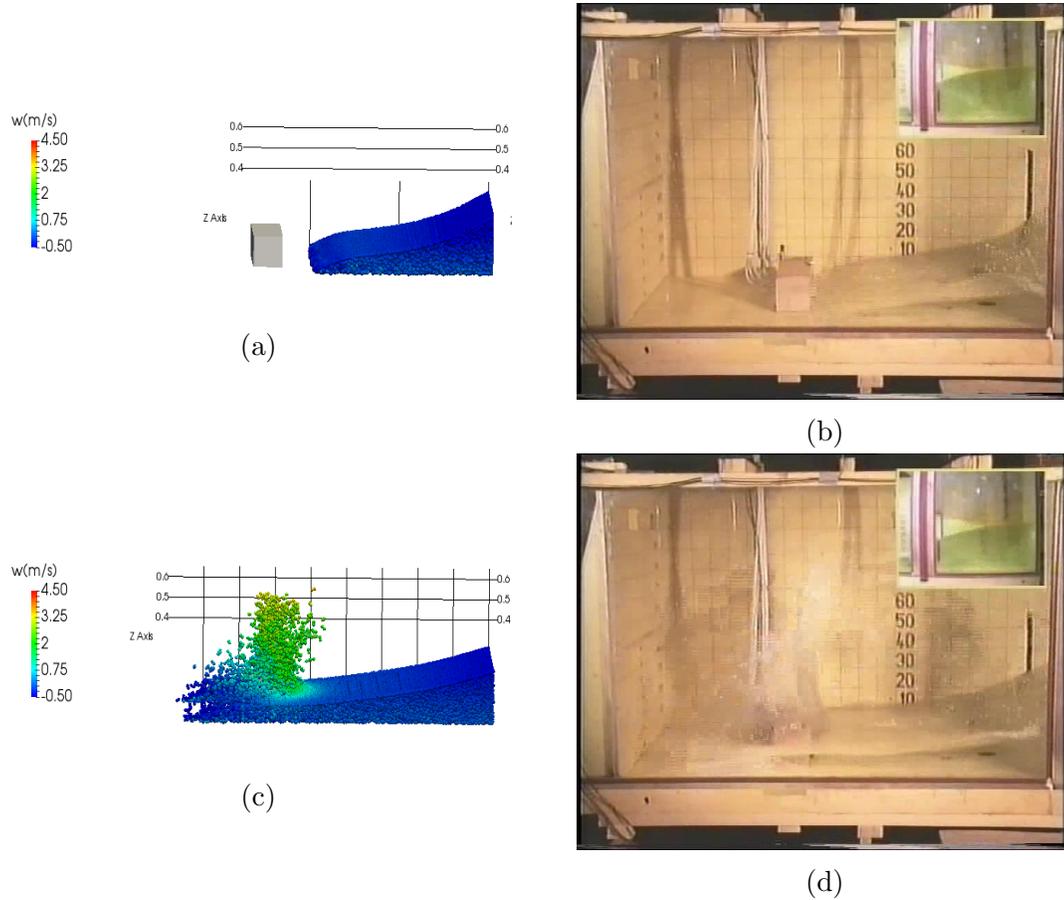
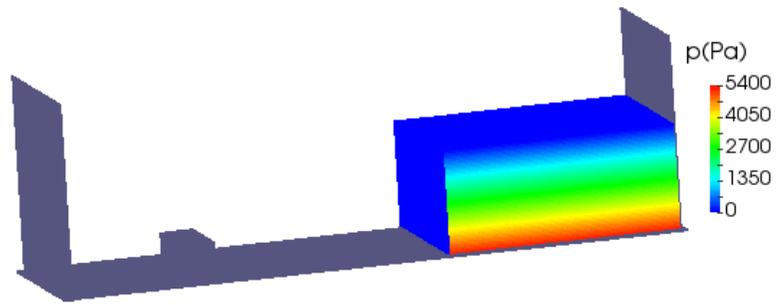
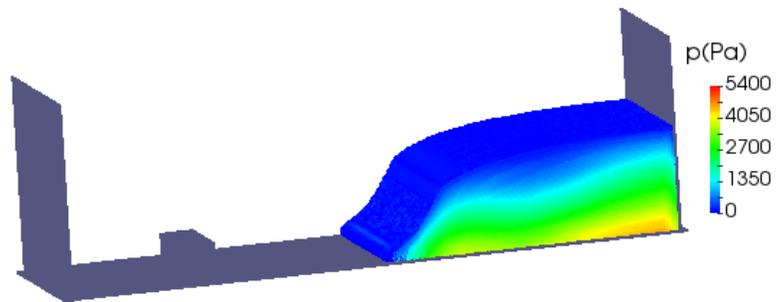


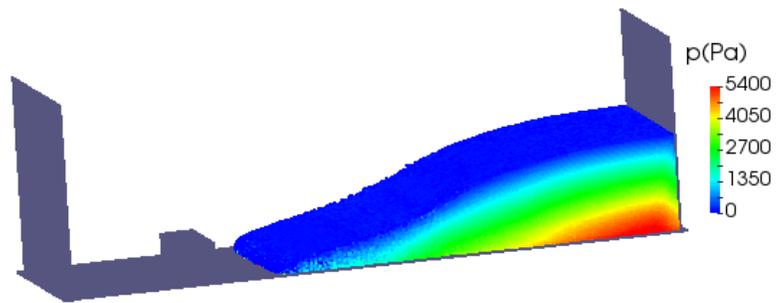
Figure 5.12: Snapshots of the dam-break on the short box compared with experiment reported by Kleefsman [6]. (a) Current study at $t=0.4$ s (Particles are colored by their velocity at z -direction), (b) Experiment at $t=0.4$ s, reprinted with permission from [6], (c) Current study at $t=0.56$ s (Particles are colored by their velocity at z -direction), (d) Experiment at $t=0.56$ s, reprinted with permission from [6]. The small picture on the experimental figures represent the water behind the gate ([6]).



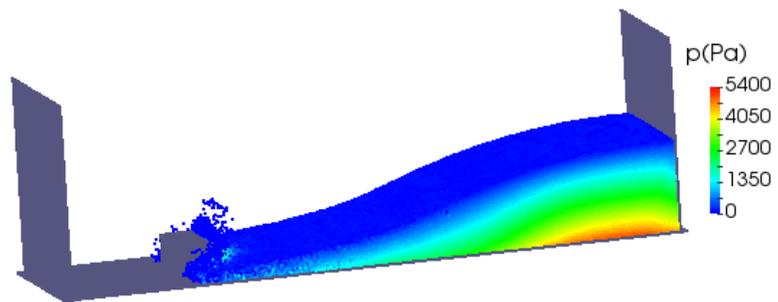
(a)



(b)

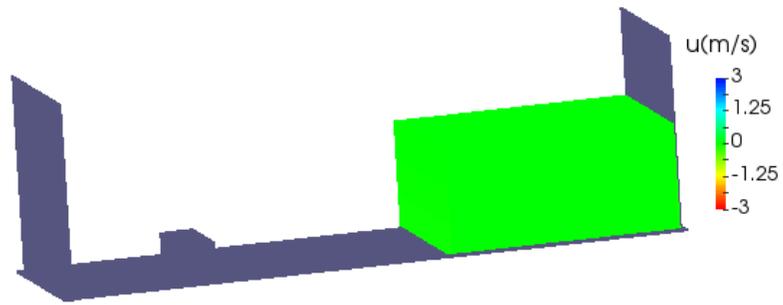


(c)

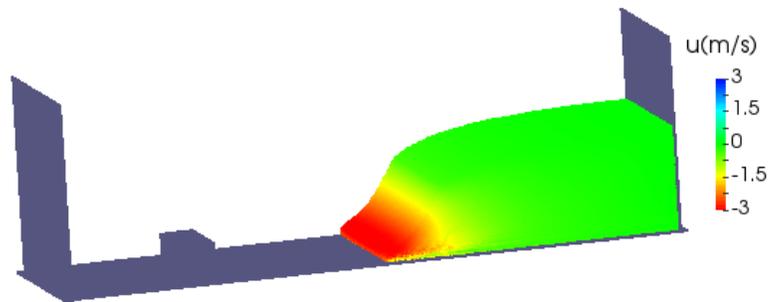


(d)

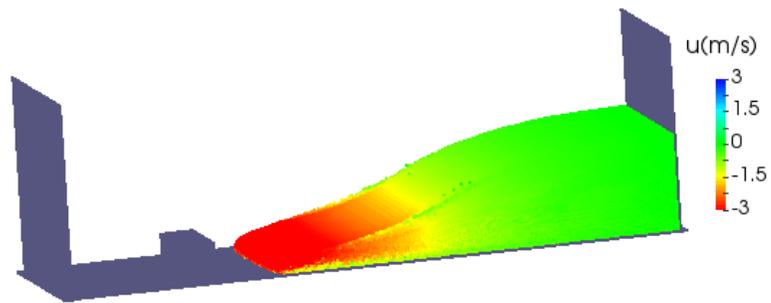
Figure 5.13: Fluid particles are colored by their pressure at: (a) $t=0$ s, (b) $t=0.2$ s, (c) $t=0.4$ s, (d) $t=0.5$ s.



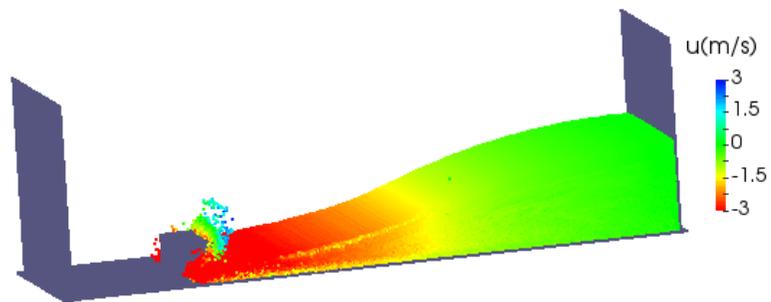
(a)



(b)



(c)



(d)

Figure 5.14: Fluid particles are colored by their velocity in x -direction (u) at: (a) $t=0$ s, (b) $t=0.2$ s, (c) $t=0.4$ s, (d) $t=0.5$ s.

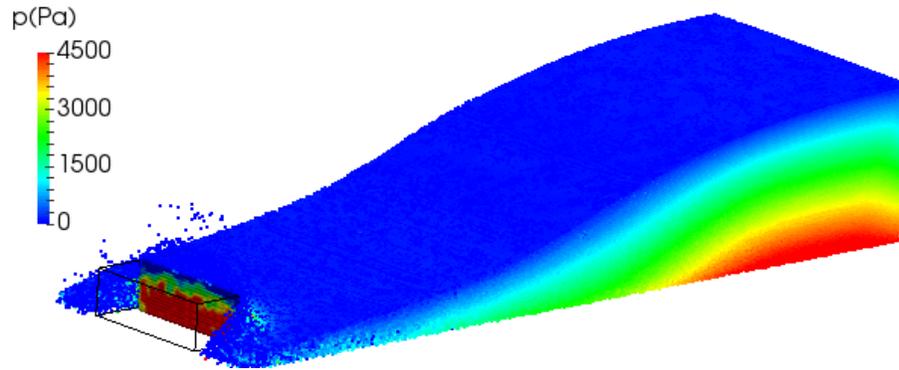


Figure 5.15: Impact pressure on the box face at $t=0.5$ s.

fixed particles is introduced at the probe locations and the same kernel function used for the simulation (Eq. 5.3.3) is applied to determine the free surface locations. It is shown in Figure 5.16a that the current ISPH results are in a good agreement with the experimental data and previous numerical simulations for water height at H4. In Figure 5.16b, current ISPH results are in a good agreement with the experimental data at H2 location, up to approximately $t=1.5$ s. The delay of 0.2 s after $t=1.5$ s of physical time is clear which could be due to the unphysical behavior of boundary particles that results in a slower motion of fluid particles in the current simulations. Similar delay behavior was shown in semi-implicit ISPH results of Lee et al. [7].

The time history of the pressure at p1 and p3 is compared with experimental data of kleefsman et al. [5] and semi-implicit ISPH results of Lee et al. [7] in Figures 5.17a and 5.17b, respectively. At p1, the current ISPH results are in a good agreement with the experimental data after the spike in the pressure. The spike in pressure is underestimated by both ISPH methods, however our explicit ISPH method captures pressure spike more accurately than the semi-implicit ISPH method. It is shown that by increasing the resolution from $\Delta l = 0.02$ m to $\Delta l = 0.01$ m the pressure spike is captured more accurately. At p3, all the simulation results are in a good agreement with the experimental data after $t=1.2$ s, but the spike in pressure is underestimated by all the numerical methods. The proposed explicit ISPH method captures the pressure spike more accurately than the two aforementioned numerical methods.

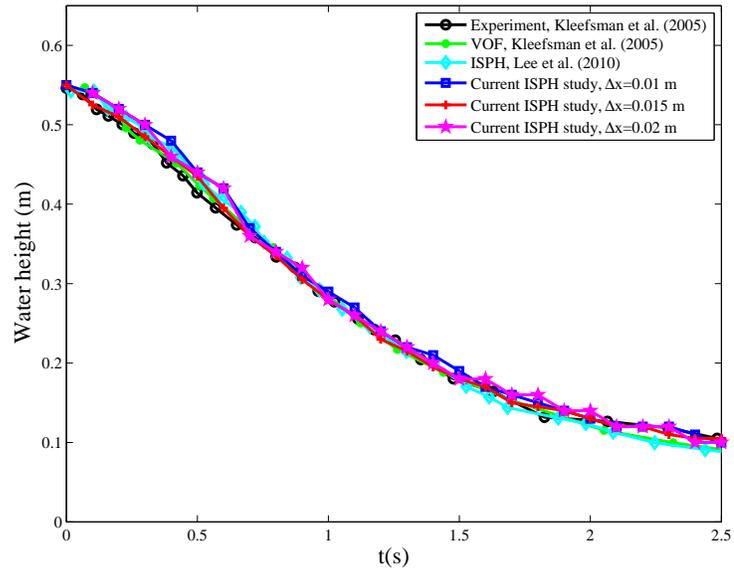
The convergence study is performed for three particle sizes of $\Delta l = 0.01$ m, $\Delta l = 0.015$ m and $\Delta l = 0.02$ m at both pressure and water height studies reported in Figures 5.16a–5.17b. It is shown that by increasing the resolution, the pressure spike due to slamming of the water wave, is captured more accurately by the current ISPH

simulations at both pressure sensors (p1 and p3).

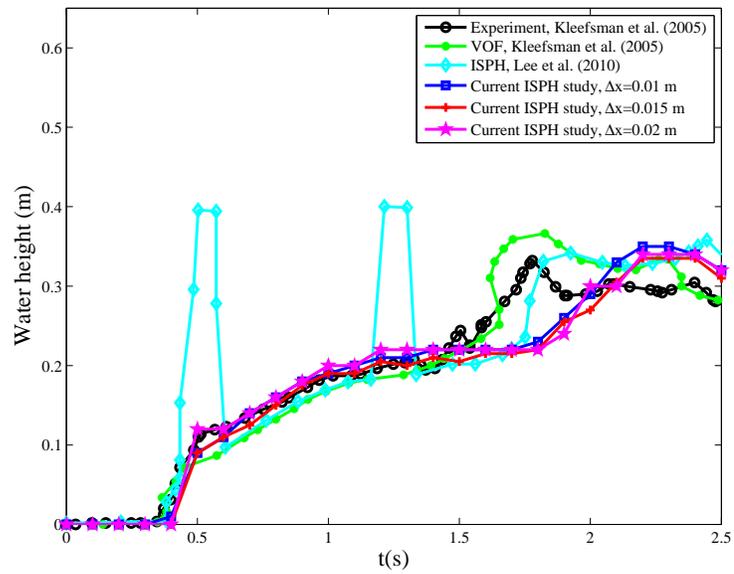
In Figures 5.18a and 5.18b, the speedup and efficiency (η) is reported for four particle resolutions of 0.75, 1.9, 10.5 and 40 million particles by using up to 256 processors. The full incompressible turbulent Navier-Stokes equations are used to measure the speedup. It is shown that the speedup increases by using a higher number of processors for all four resolutions. However, for 0.75 and 1.9 million particles, the strong scaling speedup is measured which shows saturation at 64 processors with an efficiency of approximately 35 %. The weak scaling speedup is measured for 10.5 and 40 million particles, showing that the speedup is saturated at 256 processors with efficiencies of approximately 60 % and 90%, respectively. As expected, by increasing the number of particles, the speedup and consequently efficiency of the proposed code will increase. One should keep in mind that in Figures 5.18a and 5.18b the measurement is performed for the explicit incompressible formulation of SPH. In the explicit ISPH method, Poisson's equation is solved explicitly which involves more computations than the WCSPH where the state equation is solved easily by using the already calculated density. In this paper, the SPS turbulence was used along with the Navier-Stokes equations which is computationally more expensive than the popular laminar or artificial viscosity schemes extensively used in SPH. Also, the geometry used for the simulation is such that some processors are assigned to the boundary particles with less interactions than the fluid particles [39].

Guo et al. [37] presented their result for the static water test case using a weak scaling with $N_{minimum} = 32$ for 2 million particles, while in Fig.18, we present our result based on a strong scaling for 1.9 million particles. Due to the equation used, a weak scaling usually would lead on a higher efficiency. However, due to the differences in the machine characteristics, test cases used, and the efficiency calculation schemes, the comparison between these two codes is not in fact possible directly.

It was mentioned by Yeylaghi et al. [35] that the OpenMP approach is easier to implement, but the efficiency of the OpenMP code is limited for the ocean engineering problems. In this paper, an easy MPI-based parallelization approach is proposed for the simulation of realistic 3D problems with the efficiency of several times higher than the OpenMP approach. The domain decomposition is performed in an efficient and easy way which is an asset for dynamic load balancing in simulating violent engineering problems as shown in this section.

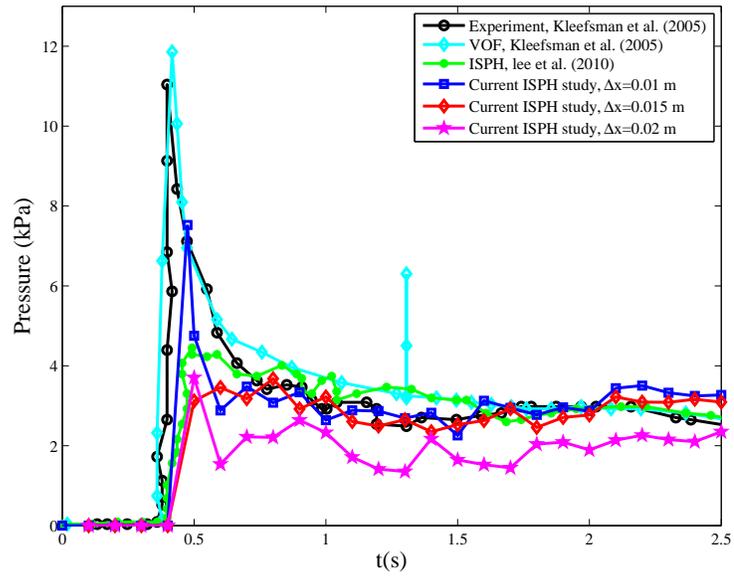


(a) H4

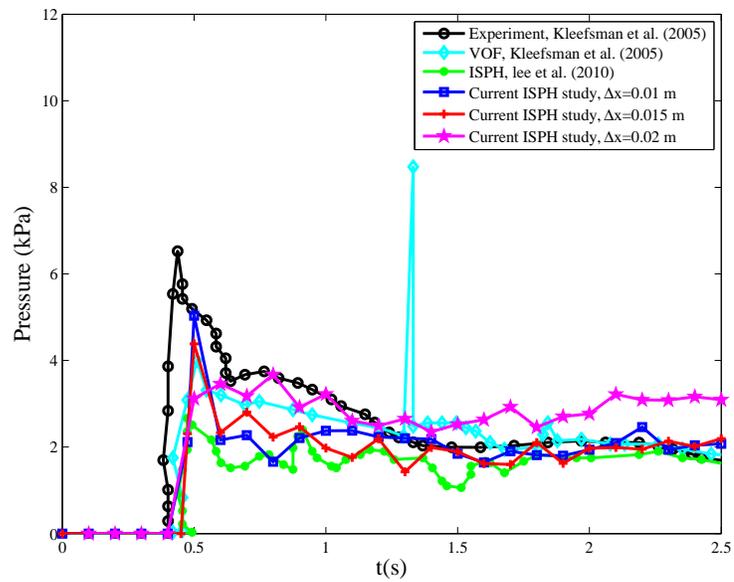


(b) H2

Figure 5.16: Water height evolution in time. Experimental data and VOF results reported in [5] and previous semi-implicit ISPH results reported in [7].

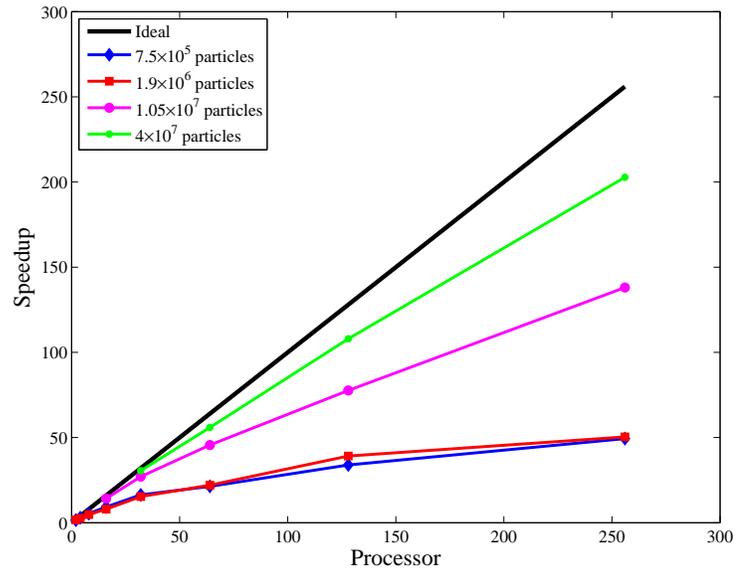


(a) p1

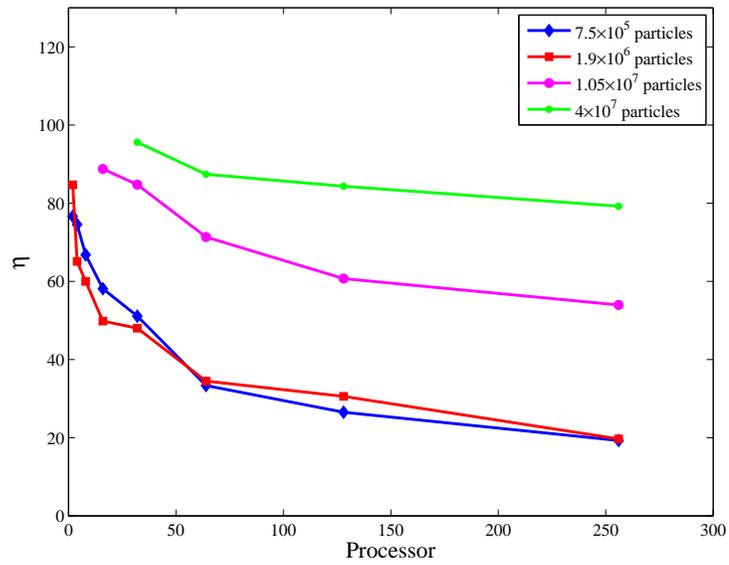


(b) p3

Figure 5.17: Pressure evolution in time. Experimental data and VOF results reported in [5] and previous semi-implicit ISPH results reported in [7].



(a) Speedup



(b) Efficiency

Figure 5.18: Speedup and efficiency.

5.6 Conclusions

A new MPI-based parallel ISPH code is presented in this paper. An explicit solver along with the eddy viscosity turbulence model is used to solve the incompressible Navier-Stokes equations in SPH particle form. By solving the explicit ISPH equations, the code remains totally explicit for further GPU implementation. The proposed code has a simple new domain decomposition and load balancing based on both spatial and particle decompositions. In this study, the Peano-Hilbert sorting algorithm is adopted to order the cells within a processor. By using the Peano-Hilbert sorting for cells instead of particles, it is possible for particles which are close spatially, also be close in the memory. The new dynamic load balancing is proposed with a feedback system that checks the particle imbalances between processors and applies the load balancing accordingly. The effectiveness of the proposed parallelization scheme was reported for several cases. The speedup and efficiency of the new code were also reported for two test cases including static water in a full tank and a dam break on a short structure. The comparison between simulation results and experimental data for the later case showed reasonable agreement for water depth at two wave gauges in the wave tank and pressure probes on the box. The new code will be applied for problems involving open boundary conditions in the near future.

5.7 Acknowledgment

The authors would like to gratefully thank Professor Arthur Veldman of the University of Groningen, and Dr. Theresa Kleefsman for kindly allowing to reuse snapshots from their experiment.

The authors would also like to thank the Natural Sciences and Engineering Research Council, Natural Resources Canada and the University of Victoria for their financial support. The authors also gratefully thank the Compute Canada's WestGrid and Calcul Qubec for their computational resources.

Chapter 6

ISPH Modelling of Landslide Generated Waves for Rigid and Deformable Slides in Newtonian and Non-Newtonian Reservoir Fluids

This paper is submitted for publication to the Journal of Advances in Water Resources.

The scope of this chapter is to study different scenarios of landslide generated waves. Several test cases in 2D and 3D are simulated by the current ISPH method and compared with previously available experimental and numerical results. Also, a new technique is presented to calculate the motion of a rigid slide on an inclined ramp implicitly without using a prescribed velocity. In this work, the experimental data for the case of non-Newtonian reservoir fluid was provided by Northwest Hydraulic Consultant (NHC).

6.1 Abstract

A comprehensive modeling of landslide generated waves using an in-house parallel incompressible Smoothed Particle Hydrodynamics (ISPH) code is presented in this paper. The generated waves due to the landslides can have different scales and sometimes lead to tsunamis that put human lives at risk and have devastating consequences on the environment. The study of landslide generated waves is challenging due to the involvement of several complex physical phenomena, such as slide-water interaction, turbulence and complex free surface profiles. Numerical simulations play an important role in better understanding of this phenomenon. Mesh-less numerical methods, such as Smoothed Particle Hydrodynamics (SPH), could handle the slide motion and the complex free surface profile with ease, thanks to its Lagrangian and mesh-less features. In this paper, an in-house parallel explicit ISPH code is used to simulate both subaerial and submarine landslides in 2D and in more realistic 3D applications of the method. Both rigid and deformable slides are used to generate the impulsive waves. A landslide case is simulated where a slide falls into a non-Newtonian reservoir fluid (water-bentonite mixture). Also, a new technique is proposed to calculate the motion of a rigid slide on an inclined ramp implicitly, without using the prescribed motion in SPH. In all the presented test cases, results from our proposed ISPH method are in good agreement with the available experimental data.

6.2 Introduction

The waves generated due to landslides in restricted waters bodies, such as lakes, bays, fjords and reservoirs, can cause environmental disaster and put human lives at risk [1, 9]. The generated impulsive waves could have different scales sometimes referred as tsunamis [1, 9]. The initial position of the slide with respect to the water determines the type of the landslide: subaerial, submerged or submarine landslide. By increasing the submergence of the slide, the generated impulsive wave height decreases [1]. The most extreme subaerial landslide generated wave reported, was the megatsunami in 1958 at Lituya Bay, Alaska [113]. This landslide resulted in a wave runup of greater than 500 m due to the slide of 30 million m³ of rocks into the Lituya Bay [113]. The 1963 landslide in the Vajont dam in Italy caused the 240 m high waves to overtop the dam and took the lives of 2000 people [114]. The Papua New

Guinea submarine landslide in 1998 resulted in a wave height of 15 m and took the lives of 2200 people [115]. The number of dams, rivers, lakes and reservoirs existing globally, and the potential danger that landslide generated waves pose to human life, environment and coastal infrastructure have led extensive study of this phenomenon.

The study of landslide generated waves is challenging due to the involvement of several complex physical phenomena, such as slide-water interaction, multiphase flow, turbulence and complex free surface profiles [1]. There are many experimental and numerical studies reported on investigating landslide generated waves using both rigid and deformable slides. Experimental studies are performed in both 2D and 3D, often accompanied by empirical equations for wave height, wave period and propagation velocity, etc. [1, 116, 116–120]. The physical experiments are important for both understanding the complexity of the physics involved in landslide generated waves, and validating of numerical simulations. A recent review of landslide generated waves studies through numerical modeling is presented by Yavari et al. [121].

Traditional mesh-based methods such as finite volume and finite difference schemes have been frequently used to simulate landslides. These methods have to employ complicated grid moving and interface tracking algorithms. Heinrich [8] simulated 2D submarine and subaerial landslides solving the incompressible Navier-Stokes equations by using the finite difference method. Liu et al. [122] employed 3D Large Eddy Simulations (LES) along with a volume-of-fluid (VOF) method to study waves generated by a sliding mass. It was concluded that the maximum runup is a function of initial position and the specific weight of the slide. Abadie et al. [9] used a Direct Numerical Simulation (DNS) of the Navier-Stokes equations to simulate 2D and 3D landslide generated tsunamis. In their proposed method, water, air and slide are simulated as fluids and the interface between phases is captured using the VOF method [9]. Serrano-Pacheco et al. [123] simulated the landslide generated waves using the finite volume method for the solution of the shallow water equations. In the mesh-based finite volume or finite difference methods mentioned above, the interface is captured by using an additional equation.

SPH is a Lagrangian meshfree method developed in 1977 by Lucy [20] and Gingold and Monaghan [21] to study astrophysical applications. In SPH, the free surface is determined implicitly. The method has been applied to various engineering fields due to its Lagrangian and meshless characteristics. In SPH, there are two general methods to calculate the pressure: weakly compressible SPH (WCSPH) and incompressible

SPH (ISPH). The pressure is calculated by using a state equation in WCSPH, while in ISPH, Poisson's equation is used to calculate pressure. A recent comprehensive review of the SPH method is reported by Shadloo et al. [22]. Monaghan and Kos [10] were first to use weakly compressible SPH to simulate subaerial landslide generated waves in 2D. Ataie-Ashtiani [124] used an implicit incompressible SPH (ISPH) method to study 2D landslide generated waves for both rigid and deformable slides. The Bingham Cross rheological model was used to simulate the deformable non-Newtonian material. Capone et al. [125] used WCSPH to simulate submarine landslides where the Bingham model was employed to simulate the deformable slide. Viroulet et al. [126] used WCSPH SPHysics to simulate a 2D subaerial landslide. Heller et al. [1] used the WCSPH DualSPHysics code [74] to reproduce their subaerial landslide experiment in both 2D and 3D. Shi et al. [127] used WCSPH along with a elasto-plastic constitutive model to simulate the 2D landslides for the soil slides. The fast interaction between slide and water, and the complex free surface involved in landslide generated wave problems, make SPH an appropriate method to study these problems thanks to its Lagrangian and mesh-less features.

In this paper, our in-house parallel incompressible code [36] is used to simulate landslide generated waves and compared to existing experimental and computational test cases. In our code, Poisson's equation is solved explicitly along with boundary conditions proposed by Adami et al. [33]. The proposed code is used to simulate the landslide generated waves in 2D and 3D, for both rigid and deformable slides. The deformable slide is assumed to be a Bingham fluid and the general Cross model is used to simulate the non-Newtonian fluid. Also, a new method is proposed to calculate the slide motion on the inclined ramp implicitly for more realistic applications. This paper is organized as follows: the governing equations for incompressible turbulent SPH are summarized in (Sect. 6.3). The experimental test cases simulated in this paper include: submarine landslide in 2D (Sect. 6.5.1.1), subaerial Scott Russell wave generator (Sect. 6.5.1.2), the SPHERic test case-11 in 2D (Sect. 6.5.1.3) and in 3D (Sect. 6.5.1.4), deformable landslide (Sect. 6.5.2), and the landslide in a non-Newtonian reservoir fluid (Sect. 6.5.3). Also, a new method to calculate a rigid slide's motion on an inclined ramp implicitly, without prescribing its velocity, is presented in Sect. 6.4.

6.3 SPH Methodology

In SPH, the Lagrangian formulation of the governing equations are solved for each particle as:

$$\nabla \cdot \mathbf{u} = 0, \quad (6.3.1)$$

$$\rho \frac{D\mathbf{u}}{Dt} = -\nabla p + \rho \mathbf{g} + \mu \nabla^2 \mathbf{u} + \nabla \cdot \vec{\tau}, \quad (6.3.2)$$

where ρ is the particle density, t is time, p is the particle pressure, \mathbf{g} is the particle gravitational acceleration vector, \mathbf{u} is the particle velocity vector, μ is the laminar dynamic viscosity, and $\vec{\tau}$ is the sub-particle scale (SPS) turbulence stress tensor in particle form ([95]). The eddy viscosity assumption is often used to model the SPS turbulence stress ($\vec{\tau}_{ij}$) as:

$$\vec{\tau}_{ij} = 2\rho(\nu_t S_{ij} - \frac{1}{3}\delta_{ij}k), \quad (6.3.3)$$

where S_{ij} is SPS strain tensor, k is the turbulence kinetic energy, δ_{ij} is the Kronecker delta function and ν_t is the turbulence eddy viscosity. The eddy viscosity model of Smagorinsky (smagorinsky1963general) is applied to represent the effect of small scale turbulence in SPH ([95]) as:

$$\nu_t = (C_s \Delta)^2 (2S_{ij}S_{ij})^{1/2}, \quad (6.3.4)$$

where Δ is the filter width and C_s is the Smagorinsky constant, which is taken to be 0.15 in this paper. The pressure gradient term in SPH particle form is discretized as (see [65]):

$$\left(-\frac{1}{\rho}\nabla p\right)_i = -\sum_j m_j \left(\frac{p_i}{\rho_i^2} + \frac{p_j}{\rho_j^2}\right) \cdot \nabla_i W_{ij}, \quad (6.3.5)$$

where W_{ij} is the kernel or smoothing function. In SPH, the interpolation between particles is performed based on a kernel function that depends on the distances between particles ($\mathbf{r}_i - \mathbf{r}_j$). The Wendland [46] kernel is used for all simulations presented in this paper. The diffusion term is written as (see [65]):

$$(\mu \nabla^2 \mathbf{u})_i = \sum_j \frac{4m_j(\mu_i + \mu_j)}{(\rho_i + \rho_j)^2} \frac{\mathbf{r}_{ij} \cdot \nabla_i W_{ij}}{r_{ij}^2 + \eta^2} \mathbf{u}_{ij}, \quad (6.3.6)$$

where $\mathbf{r}_{ij} = \mathbf{r}_i - \mathbf{r}_j$, $\mathbf{u}_{ij} = \mathbf{u}_i - \mathbf{u}_j$, ρ_i is the density of particle i with velocity \mathbf{u}_i , m_j is the mass of particle j , \mathbf{u}_j is the velocity of particle j , μ is the laminar dynamic

viscosity and η is a small number. The sub-particle scale (SPS) turbulence stress is written as:

$$\left(\frac{1}{\rho}\nabla\cdot\vec{\tau}\right)_i = \sum_j m_j \left(\frac{\vec{\tau}_i}{\rho_i^2} + \frac{\vec{\tau}_j}{\rho_j^2}\right) \cdot \nabla_i W_{ij}, \quad (6.3.7)$$

The explicit ISPH method by Hosseini et al. [30], used here, is based on the two-step projection method widely used in Eulerian based CFD methods ([24]). In the prediction step, the intermediate velocity is calculated using the viscous and body forces as:

$$\mathbf{u}^* = \mathbf{u}(t) + \Delta t(\mathbf{g} + \nu\nabla^2\mathbf{u} + \frac{1}{\rho}\nabla\cdot\vec{\tau}), \quad (6.3.8)$$

$$\mathbf{r}^* = \mathbf{r}(t) + \Delta t\mathbf{u}^*, \quad (6.3.9)$$

where \mathbf{u}^* is the intermediate velocity and \mathbf{r}^* is the intermediate position. Poisson's equation needs to be solved for pressure at each time step is written as:

$$\nabla\cdot\left(\frac{\nabla p}{\rho}\right) = \frac{\nabla\cdot\mathbf{u}^*}{\Delta t}, \quad (6.3.10)$$

Hence, Poisson's equation in SPH particle form can be formulated as:

$$\sum_j \frac{8m_j}{(\rho_i + \rho_j)^2} \frac{p_{ij}\mathbf{r}_{ij}\cdot\nabla_i W_{ij}}{r_{ij}^2 + \eta^2} = \frac{-1}{\Delta t} \sum_j V_j(\mathbf{u}_i^* - \mathbf{u}_j^*)\nabla_i W_{ij}, \quad (6.3.11)$$

and is solved explicitly by using the method described by [30, 35]. Finally, the velocity is corrected as:

$$\mathbf{u}^{t+1} = \mathbf{u}^* + \Delta t\left(-\frac{1}{\rho}\nabla p\right). \quad (6.3.12)$$

In order to mark the free surface particles, an artificial density is calculated ($\rho_f = \sum_j m_j \rho_0$), and if $\rho_f < 0.95\rho_0$ the particle is marked as a free surface particle. In this paper, fixed particles are used to model the solid wall. The method described by Adami et al. [33] is used to calculate the pressure for boundary particles from the surrounding fluid particles as:

$$p_w = \frac{\sum_i p_i W_{wi} + (\mathbf{g} - \mathbf{a}_w) \sum_i \rho_i \mathbf{r}_{wi} W_{wi}}{\sum_i W_{wi}}, \quad (6.3.13)$$

where p_w is the boundary particle's pressure and \mathbf{a}_w is the acceleration of the wall.

A new parallel SPH code is used to simulate the 3D test cases. The detailed description of the code and parallelization scheme is available in [36]. Compute Canada’s supercomputer, Guillimin¹, was used to perform the simulations.

6.4 A New Method for Sliding Solid-Solid Contact in SPH

In this section, a new method for the motion of a slide on an inclined ramp in SPH is proposed without using a prescribed velocity. In realistic landslide problems, the velocity of the slide is neither known nor prescribed. A common way to deal with this in numerical simulation models is to prescribe the velocity, which is available from the experiment, as an input variable. Examples of such studies based on the mesh-based methods are presented by Heinrich [8], Liu et al. [122] and Yul et al. [128]. Abadie et al. [9] presented a method in which all three phases are modeled as fluids in their DNS simulation. The slide is assumed to be a non-Newtonian fluid with a high viscosity and its motion is calculated implicitly. In previous SPH studies, velocity is prescribed based on the available experimental data [10, 124, 125]. Hence, having a method that calculates the motion of the slide implicitly is of interest for practical applications.

In the proposed methodology, there are four types of particles: the fluid particles, the fixed boundary particles, the slide particles and the interface particles between slide and fixed particles. These four types of particles are shown in Figure 6.1a and the particle representation for the Heinrich [8] test case described in Sect. 6.5.1.1 is shown in Figure 6.1b. In the proposed method, a layer of fluid particles (interface particles in Figure 6.1a) are glued on the slide particles in between slide and fixed particles. The interface particles (green particles in Figure 6.1a) have the same properties as the water particles and the same equations for fluid particle are solved for these particles. However, they move with the slide particles. The slide motion is calculated based on the forces and moments from surrounding fluid particles (the same methodology described by Yeylaghi et al. [35] is used). The explicit incompressible equation is used to calculate the pressure (Eq. 6.3.11). Therefore, the pressure at each time step is calculated based on the previous time step’s pressure. The criteria to impose the

¹<http://www.hpc.mcgill.ca/index.php/starthere/81-doc-pages/215-guillimin-hardware>

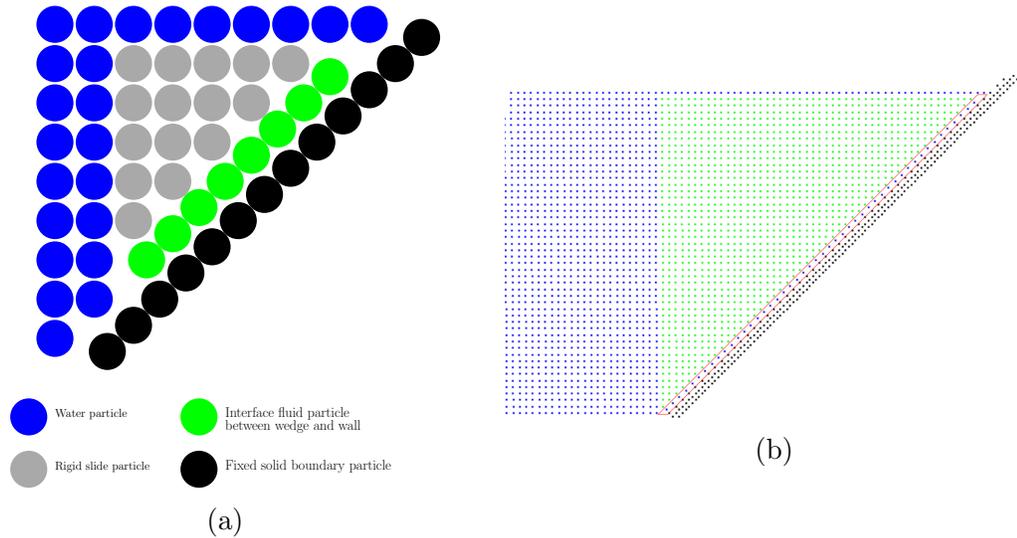


Figure 6.1: (a) Description of particle types, (b) Particle representation in SPH, the interface particles are enclosed by the red line.

free surface is not applied to the interface particles in order to avoid imposing zero pressure on these particles. The pressure on the fixed particles (black particles in Figure 6.1a) are calculated based on Eq. 6.3.13 from the surrounding fluid particles. The second term in the numerator ($\mathbf{g} \sum_i \rho_i \mathbf{r}_{wi} W_{wi}$) would result in higher pressure in the fixed boundary particles (black particles in Figure 6.1a) since there is a layer of fluid particle on the slide face (interface particles). By using this method, there is no need to prescribed the slide velocity and the motion of the slide is calculated implicitly.

The experimental data of Heinrich [8] is used to test the accuracy of the proposed method. The vertical slide displacement from the proposed method is compared with the experimental data of Heinrich [8] and the previous DNS numerical simulation of Abadie et al. [9] in Figure 6.2. It is shown in Figure 6.2 that both simulation methods are in a good agreement with the experimental data, however they show an overestimation after $t=0.8$ s. The reason could be the compressibility effects. In the SPH simulation, we just solve for water particles, but in the DNS simulations both air and water phases are considered. This could explain the larger deviation in the SPH simulation after $t=0.8$ s.

The free surface at $t=0.5$ s and $t=1.0$ s are compared with the experimental data of Heinrich [8] and shown in Figure 6.3. In these figures, the SPH results are

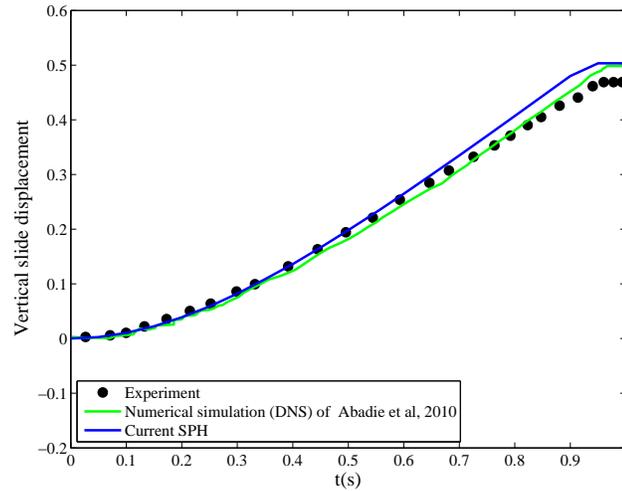


Figure 6.2: Vertical displacement of the rigid body against time (Experimental data of Henrich [8], DNS simulation of Abadie et al. [9]).

presented by using both the prescribed motion, and the methodology prescribed in this section. It is shown that both SPH simulations are in good agreement with the experimental data. The practical advantage of the freely falling motion is that it is useful for predictive analysis applications, while remaining as equally valid as the priori prescribed motion method.

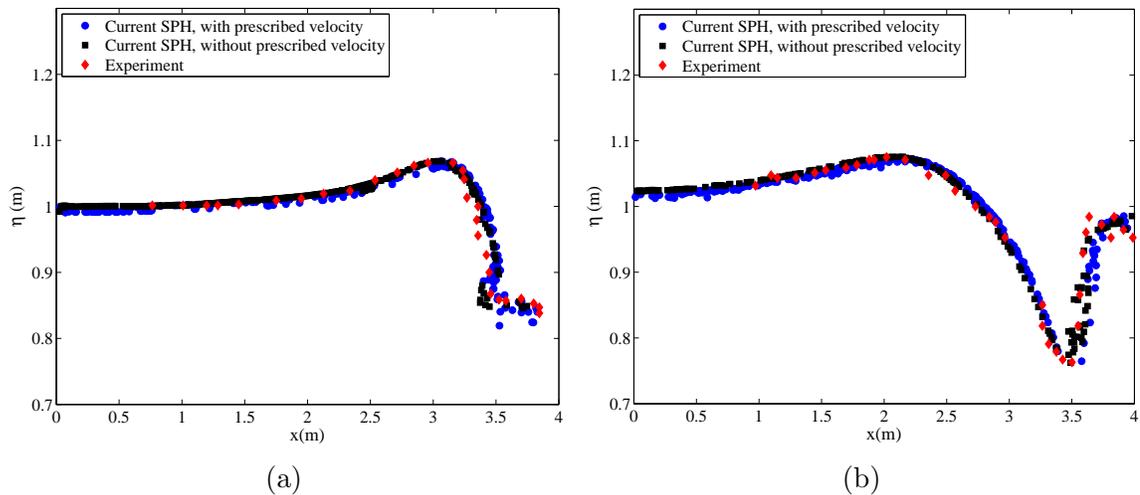


Figure 6.3: Free surface elevation against time at (a) $t=0.5$ s (b) $t=1.0$ s by using prescribed velocity and the new proposed method (Experimental data of Henrich [8]).

6.5 Results

6.5.1 Rigid Slide

Landslide generated waves are simulated in this paper in both 2D and 3D for rigid and deformable slides and in Newtonian and non-Newtonian reservoir fluids.

6.5.1.1 Submarine Landslide in 2D

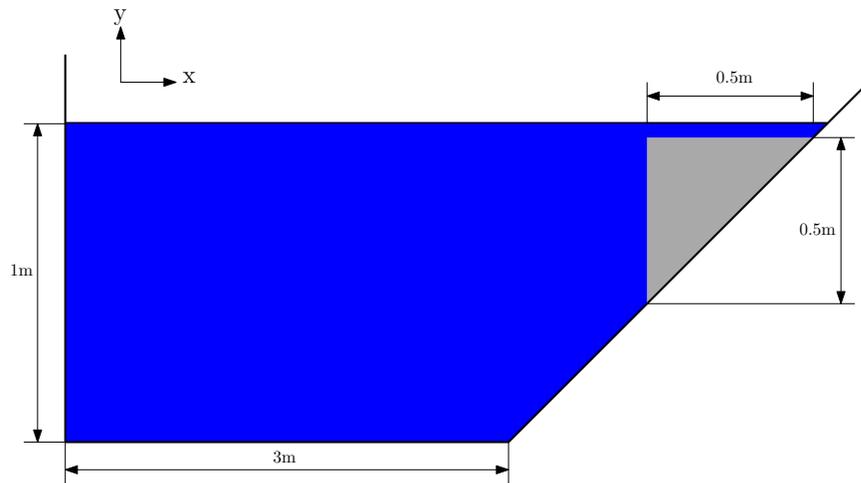


Figure 6.4: Schematic of the experiment of Heinrich [8].

In this section, the experiment of Heinrich [8] for a submarine landslide is simulated to validate our SPH results. In this experiment, the rigid wedge with a density of 2000 kg/m^3 moved down a ramp with a slope of 45° . The schematic of the experiment including the dimensions of the slide is shown in Figure 6.4. The still water depth is 1 m and the particle size is set to 0.01 m for this simulation, resulting in 38680 total particles. The velocity of the slide is specified in this simulation using the function prescribed by Ataie-Ashtiani et al. [124].

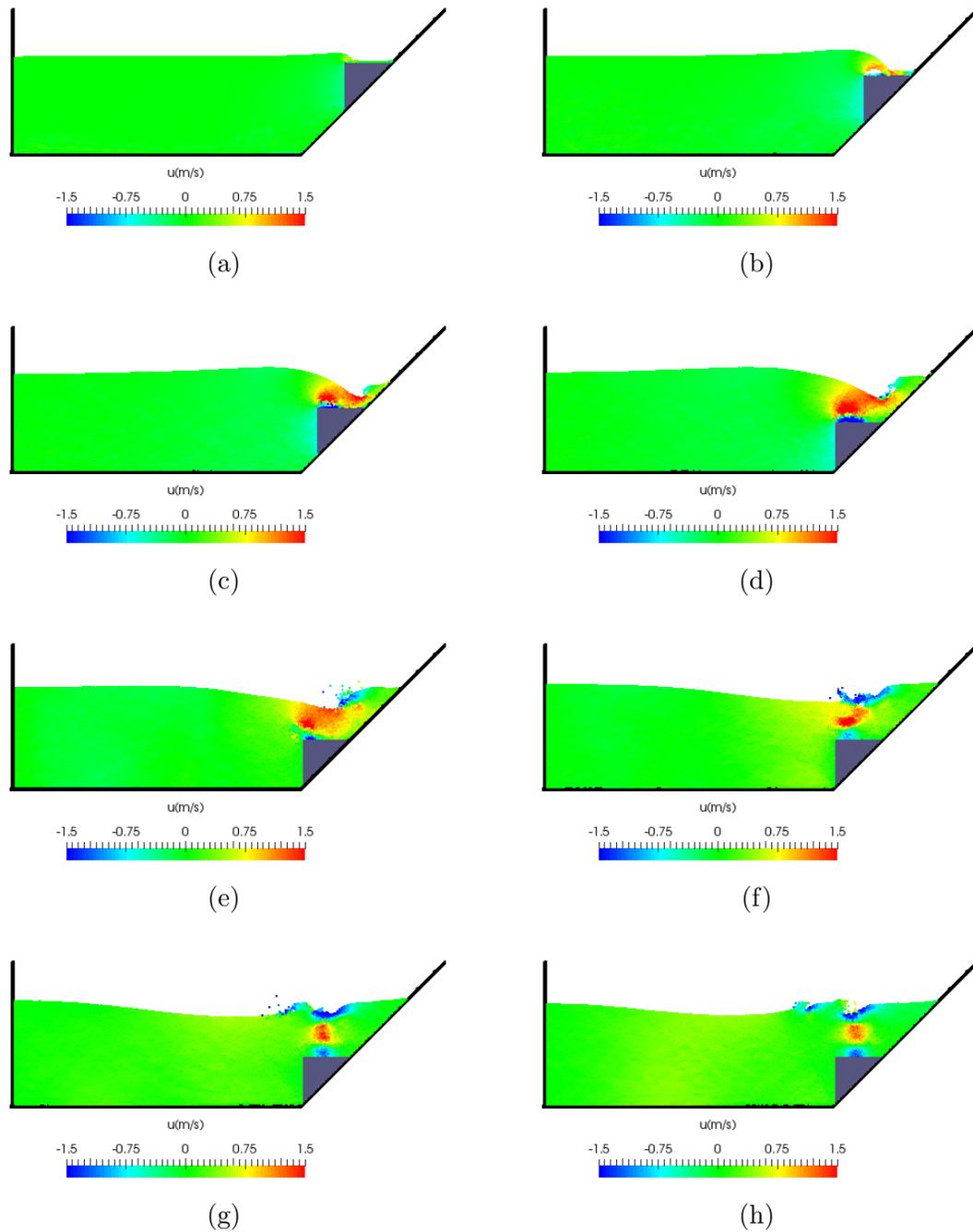


Figure 6.5: Particle are colored by their velocity in the x-direction (u) at: (a) $t=0.25$ s, (b) $t=0.5$ s, (c) $t=0.75$ s, (d) $t=1.0$ s, (e) $t=1.25$ s, (f) $t=1.5$ s, (g) $t=1.75$ s, (h) $t=2.0$ s.

In Figure 6.5, particles are colored by their velocities in the x-direction (u) at $t=0.25$ s, $t=0.5$ s, $t=0.75$ s, $t=1.0$ s, $t=1.25$ s, $t=1.5$ s, $t=1.75$ s and $t=2$ s. The rigid

body with a triangular shape slides down the ramp for approximately 1 s and then stops at the break, down the ramp (Figure 6.5d). It was mentioned by Abadie et al. [9] that two waves are generated due to the motion of the slide in water. These two waves are shown in our SPH simulation in Figure 6.5. The first wave is generated when the slide moves downward (Figures 6.5a- 6.5c), and the second wave is generated as a plunging wave on the ramp (Figures 6.5d- 6.5e). After the wave breaks, the free surface shape becomes more complex as shown in Figures 6.5f-6.5h.

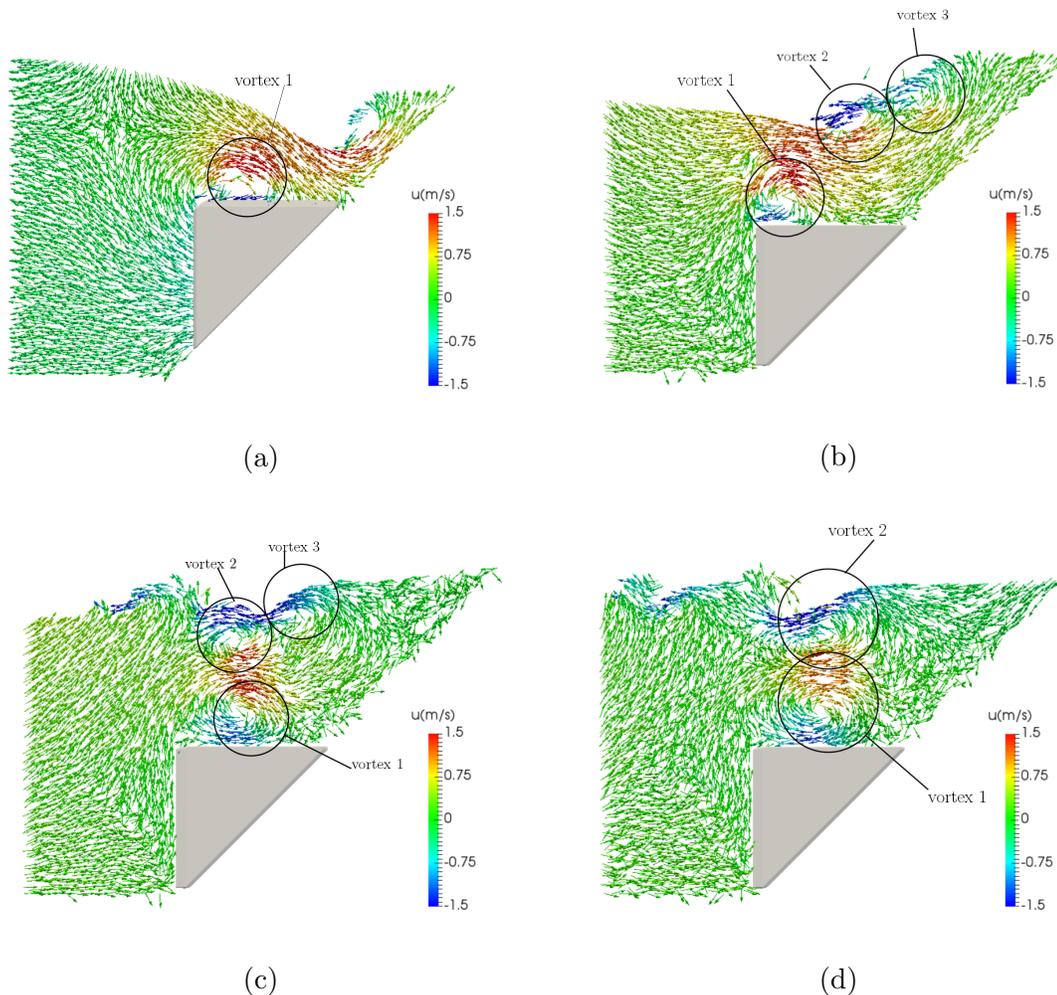


Figure 6.6: Vortex formation above the slide, particles velocity vector are colored by their velocity in the x-direction (u) at: (a) $t=0.85$ s, (b) $t=1.35$ s, (c) $t=1.75$ s, (d) $t=2.0$ s.

The velocity vector for particles are colored by their velocity in the x-direction and

are shown in Figure 6.6 at $t=0.85$ s, $t=1.35$ s, $t=1.75$ s and $t=2$ s. It was mentioned by Abadie et al. [9] in their DNS model that three vortices are generated as the slide moves down the ramp under the water. The same vortices are shown in our SPH simulations in Figure 6.6. A large vortex is generated above the left side of the slide as it starts to move down the ramp. This vortex is shown by the negative and positive x-direction velocities in Figures 6.5e and 6.5f above the left side of the slide. (the large vortex is marked vortex 1 in Figure 6.6b). Two other vortices are generated after the plunging wave breaks in between $t=1$ s and $t=1.5$ s; these vortices are reported in [9]. These two vortices are marked vortex 2 and 3 in Figures 6.6a and 6.6c. The three vortices are interacting, which results in two vortices at $t=2$ s shown in Figure 6.6d.

6.5.1.2 2D Subaerial Landslide-Scott Russell Wave Generator

In this section, the subaerial landslide is simulated using an experiment of Monaghan and Kos [10] which is similar to the wave generator described by Russel [129]. The schematic of the experiment is shown in Figure 6.7. In this experiment, a heavy box is dropped vertically into still water of 0.21 m depth. The dimensions of the box are given in Figure 6.7. The tank length in the experiment is 9 m but in our simulation we use a tank with a length of 4 m to save computational time. The details about the experiment are available in [10]. Monaghan and Kos [10] performed the experiment along with WCSPH simulation. Here, the explicit incompressible formulation of SPH is used with a particle size of 0.008 m resulting in 17000 total particles.

In Figure 6.8, water particles are colored by their pressure at $t=0.2$ s, $t=0.3$ s, $t=0.4$ s, $t=0.5$ s, $t=0.6$ s and $t=0.75$ s. The pressure under the heavy box increases initially, which moves water particles rightward and also through the small space between the heavy box and the left wall. In addition, the jet formation in between the left wall and the heavy box forms as is clearly shown in Figure 6.8.

It was mentioned in [9, 10] that a vortex will be developed at the right corner of the block as soon as the box enters the still water and will be accompanied by a plunging wave. The vortex formation and movement in our SPH simulation is shown in Figure 6.9. In this figure the particles, velocity vector, are colored by their velocity in the x-direction (u). The formation of the vortex at the right corner of the block is clear in Figures 6.9a-6.9b. After the vortex is generated, it moves to the right end of the tank (Figures 6.9c-6.9d).

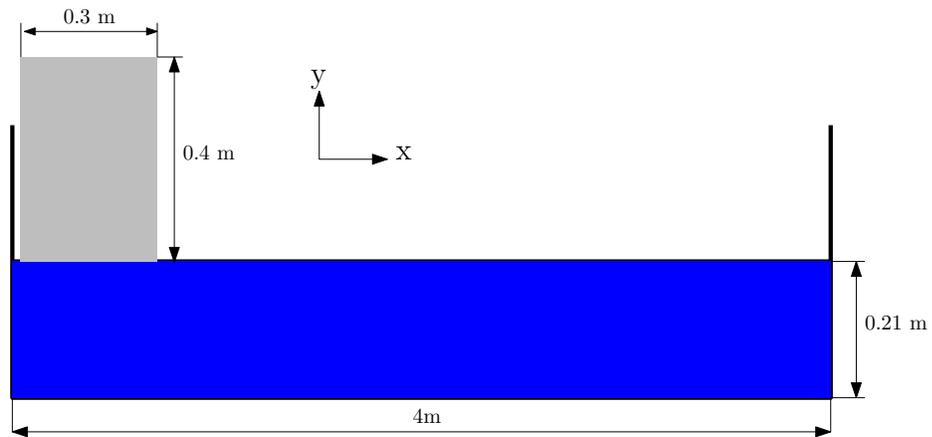


Figure 6.7: Schematic of the experiment of Monaghan and Kos [10].

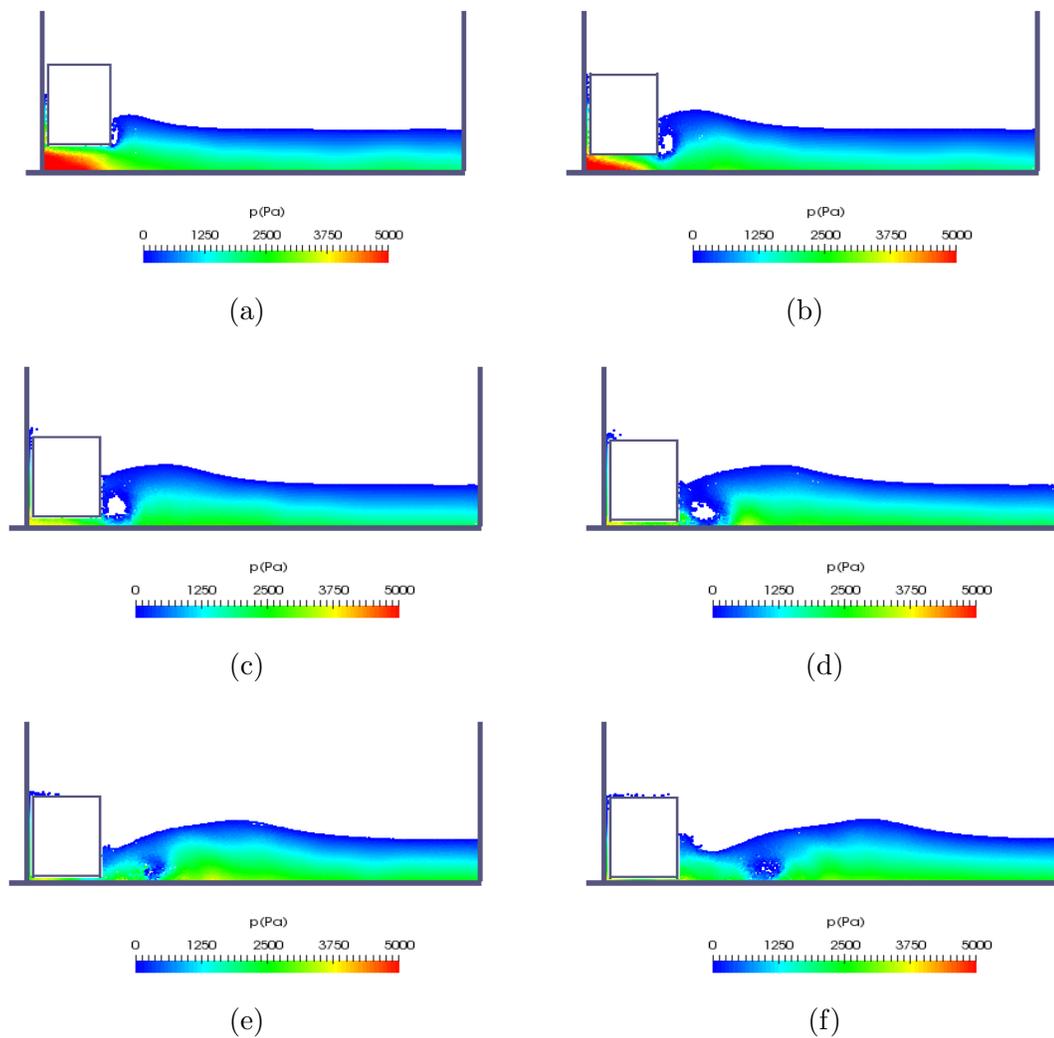


Figure 6.8: Particles are colored by their pressure at (a) $t=0.2$ s, (b) $t=0.3$ s, (c) $t=0.4$ s, (d) $t=0.5$ s, (e) $t=0.6$ s, (f) $t=0.75$ s.

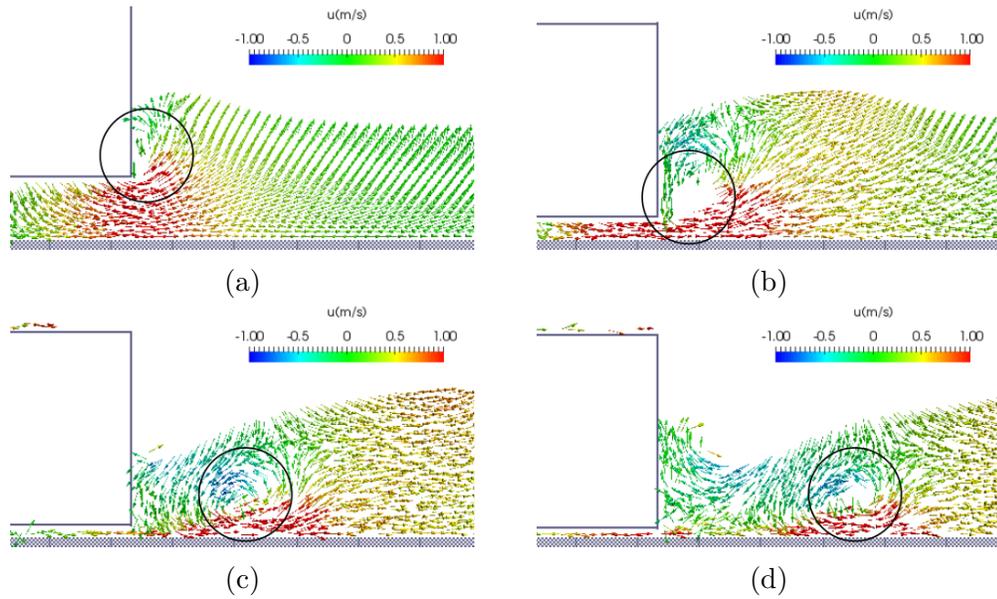


Figure 6.9: Vortex formation and movement, particles are colored by their velocity in the x-direction (u) at (a) $t=0.2$ s, (b) $t=0.4$ s, (c) $t=0.6$ s, (d) $t=0.75$ s.

A few parameters measured in the experiment of Monaghan and Kos [10] at $t=0.285$ s and are shown in Figure 6.10a. Particle representation from the current ISPH simulation at the same time ($t=0.285$ s) are shown in Figure 6.10b. A comparison of these parameters from the current ISPH results with the experimental data of Monaghan and Kos [10] and the previous SPH study of Ataie-Ashtiani et al. [124] is given in Table 6.1. Both SPH results reproduce the experimental data with a reasonable accuracy.

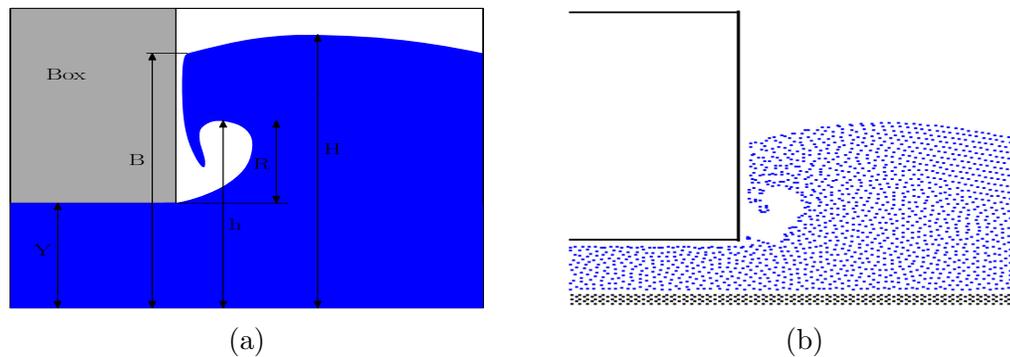


Figure 6.10: (a) Definition of parameters in the experiment of Monaghan and Kos [10], (b) Particle representation using current ISPH code at $t=0.285$ s.

Table 6.1: Comparison of the parameters in Figure 6.10a at $t=0.285$ s.

| Method | H(m) | R (m) | h (m) | B (m) |
|---------------------|---------------|---------------|---------------|----------------|
| Experiment [10] | $0.333\pm.01$ | $0.133\pm.02$ | $0.227\pm.02$ | 0.303 ± 0.02 |
| Previous ISPH [124] | 0.329 | 0.13 | 0.216 | 0.255 |
| Current ISPH | 0.3006 | 0.1159 | 0.214 | 0.2707 |

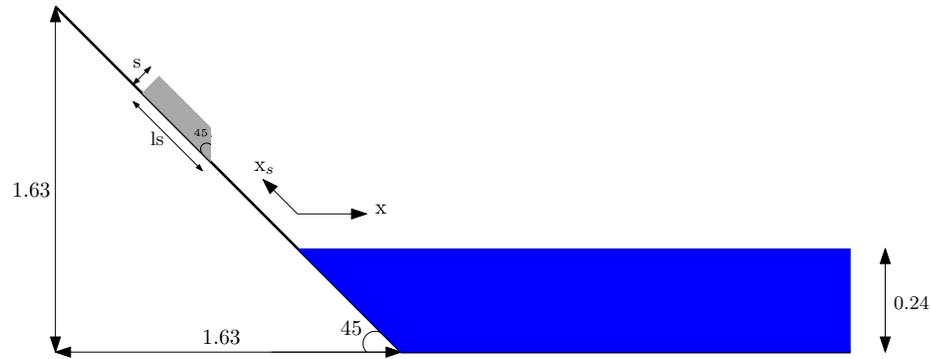


Figure 6.11: Schematic of the experiment of Heller et al. [1]. Dimensions in m.

6.5.1.3 Subaerial Landslide Generated Waves (SPHeric Test-11), 2D Channel

The SPHeric² test case 11 is simulated here for the 2D channel. The experiment reported by Heller et al. [1] is simulated in this section. In this experiment, a rigid slide impacts still water in a tank. The experiment is performed in a 2D channel and a 3D basin for two different scenarios reported in [1]. In this paper, we perform the simulation in 2D and 3D for scenario 1 reported in [1], where the still water depth is 0.24 m and the impact results in an impulsive solitary wave. The important parameters of the experiment are summarized in Table 6.2. The detailed description of the experiment can be found in [1].

²<http://spheric-sph.org/tests/test-11>

Table 6.2: Summary of important parameters in Heller et al. [1] experiment (Figure 6.11).

| Parameter | Value |
|---|---------------------------------------|
| slide length (l_s) | 0.599 m |
| slide width (w) | 0.577 m |
| slide thickness (s) | 0.12 m |
| slide density (ρ_s) | 1540 kg/m ³ |
| slide mass (m_s) | 60.14 kg |
| still water depth (h_{sw}) | 0.24 m |
| slide front initial position (x_s) | -0.55 m (2D), -0.47 m (3D) |
| channel width | 0.6 m |
| channel length | 24 m |
| basin width | 20 m |
| basin length | 12 m |
| relative wave probe distances (x/h_{swl}) in 2D and (r/h_{swl}) in 3D | 3.0, 5.0, 7.5, 10.0, 15.0, 22.5, 35.0 |

The velocity and position of the slide in the 2D and 3D experiments are reported in [1]. The same velocity is used in our ISPH simulation. The ramp is curved close to the tank bottom for both the 2D channel and 3D basin, however the sharp ramp is used in this paper since the velocity is prescribed. Heller et al. [1] used DualSPHysics [74] open source code to validate their experiment. DualSPHysics is WCSPH code that uses the artificial viscosity for the diffusion term and dynamic boundary particles to represent the solid boundary. Here, we use our in-house parallel explicit ISPH code that implements the SPH-LES formulation for the diffusion term, and the Adami et al. [33] boundary condition (Eq. 6.3.13) to calculate the pressure on the boundary particles.

In Figure 6.12, the particles are colored with their velocities at $t=0$ s, $t=0.45$ s, $t=1.5$ s and $t=3$ s. At $t=0$ s, the slide is in the initial position. At $t=0.45$ s, the slide has just impacted the still water and after $t=1.5$ s the solitary wave is propagating rightward in the tank.

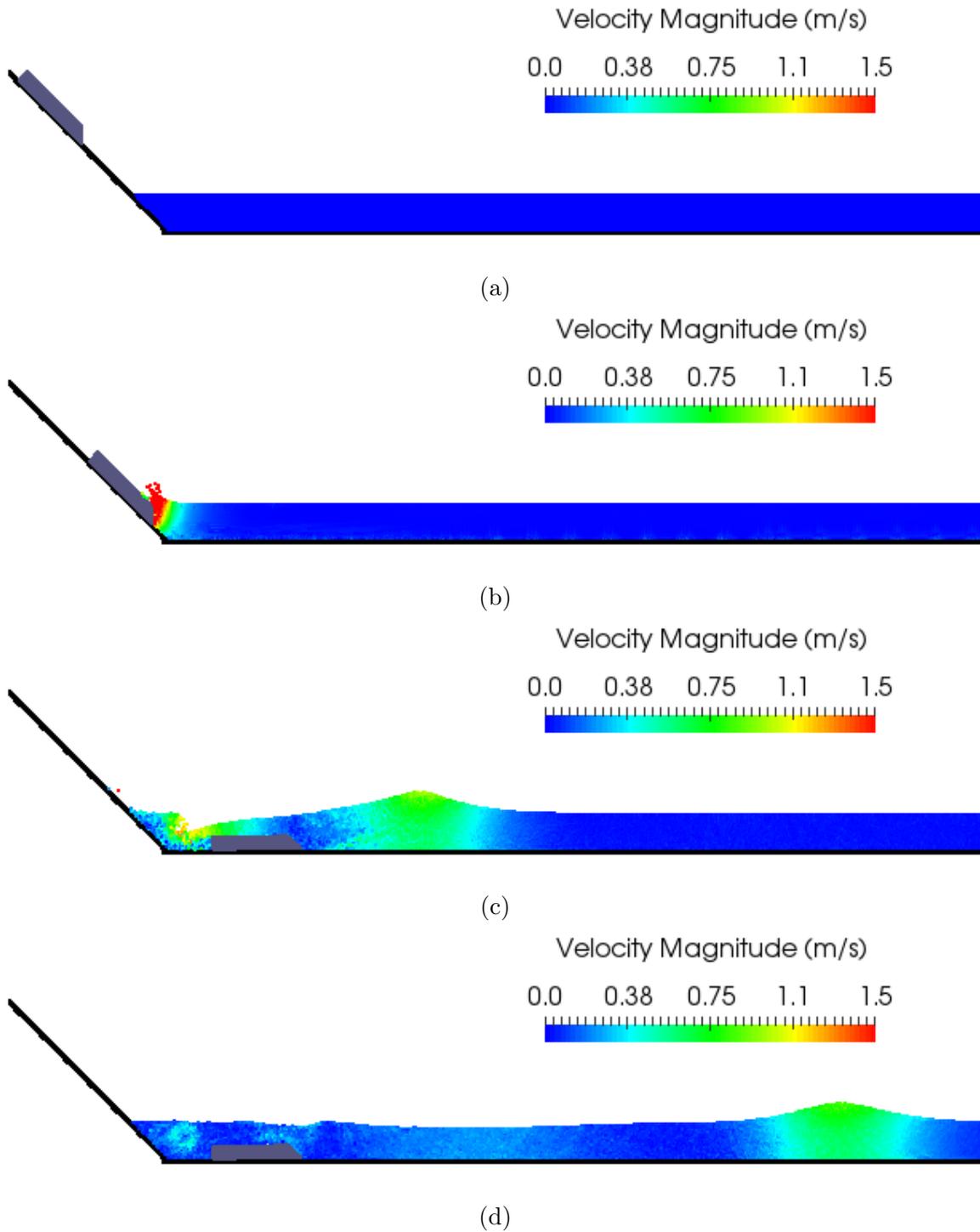


Figure 6.12: Particles are colored with their velocities at: (a) $t=0$ s, (b) $t=0.45$ s, (c) $t=1.5$ s, (d) $t=3$ s.

In Figure 6.13, the velocity vectors of the particles are colored by their velocity in the x-direction (u) at $t=1.5$ s, $t=3$ s, $t=4$ s and $t=5$ s for the region between the stationary slide and the inclined ramp. The corresponding stream tracer for each time instant is shown alongside each velocity vector and colored by the velocity in the x-direction in Figure 6.13. In Figures 6.13a and 6.13b, the plunging wave on the ramp is breaking, leading into a large vortex formation. A smaller vortex is clear in the wake of the slide in these two figures. At $t=3$ s (Figures 6.13b and 6.13b), there are three vortices in between the slide and the inclined ramp. At $t=4$ s (Figures 6.13c and 6.13c), a large vortex is clear in the middle of the region along with smaller one on the left. The left vortex in Figures 6.13c and 6.13c is divided into two smaller vortices shown in Figures 6.13d and 6.13d.

In Figure 6.14, the free surface at six wave probes (positions are given at Table 6.2) are compared with the experimental data of Heller et al. [1]. The same method used in [35] is adopted here to determine the free surface. It is shown that the solitary wave's height and shape agrees well with the experimental data for all six wave probes. The particle resolution study is performed for two particle sizes of $\Delta l = 0.01$ m and $\Delta l = 0.005$ m at two wave probes locations ($x=1.2$ m and $x=1.8$ m) and shown in Figure 6.14a and 6.14b, respectively. It is shown that the results have converged for particle resolutions of $\Delta l = 0.01$ m.

6.5.1.4 Subaerial Landslide Generated Waves (SPHeric Test-11), 3D Basin

The ISPH code is used to simulate the SPHeric test case 11 in a 3D basin. The tank length and width in 3D simulations are set to 10 m and 7.4 m, respectively. The particle length is set to 0.02 m resulting in 3163005 particles. Particles are colored with their velocities at $t=0$ s, 0.5 s, 1.5 s and 3 s in Figure 6.15. At $t=0$ s, the slide is in the initial position and at $t=0.5$ s has impacted the still water. In 3D, the wave is moving in both x and y directions as shown in Figure 6.15c-6.15d.

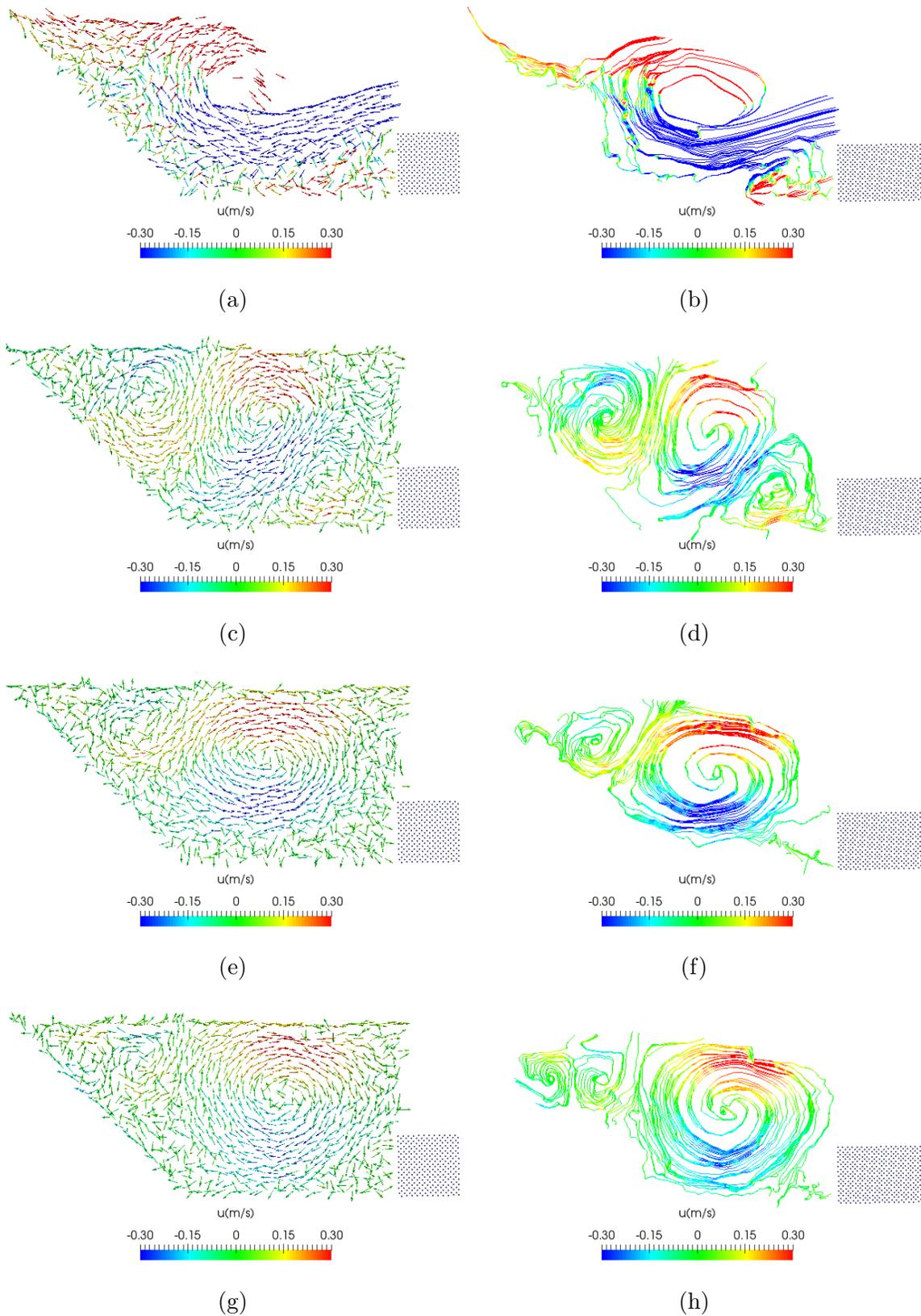


Figure 6.13: Particle velocity vectors and stream tracers are colored with their horizontal velocities at: (a and b) $t=1.5$ s, (c and d) $t=3$ s, (e and f) $t=4$ s, (g and h) $t=5$ s.

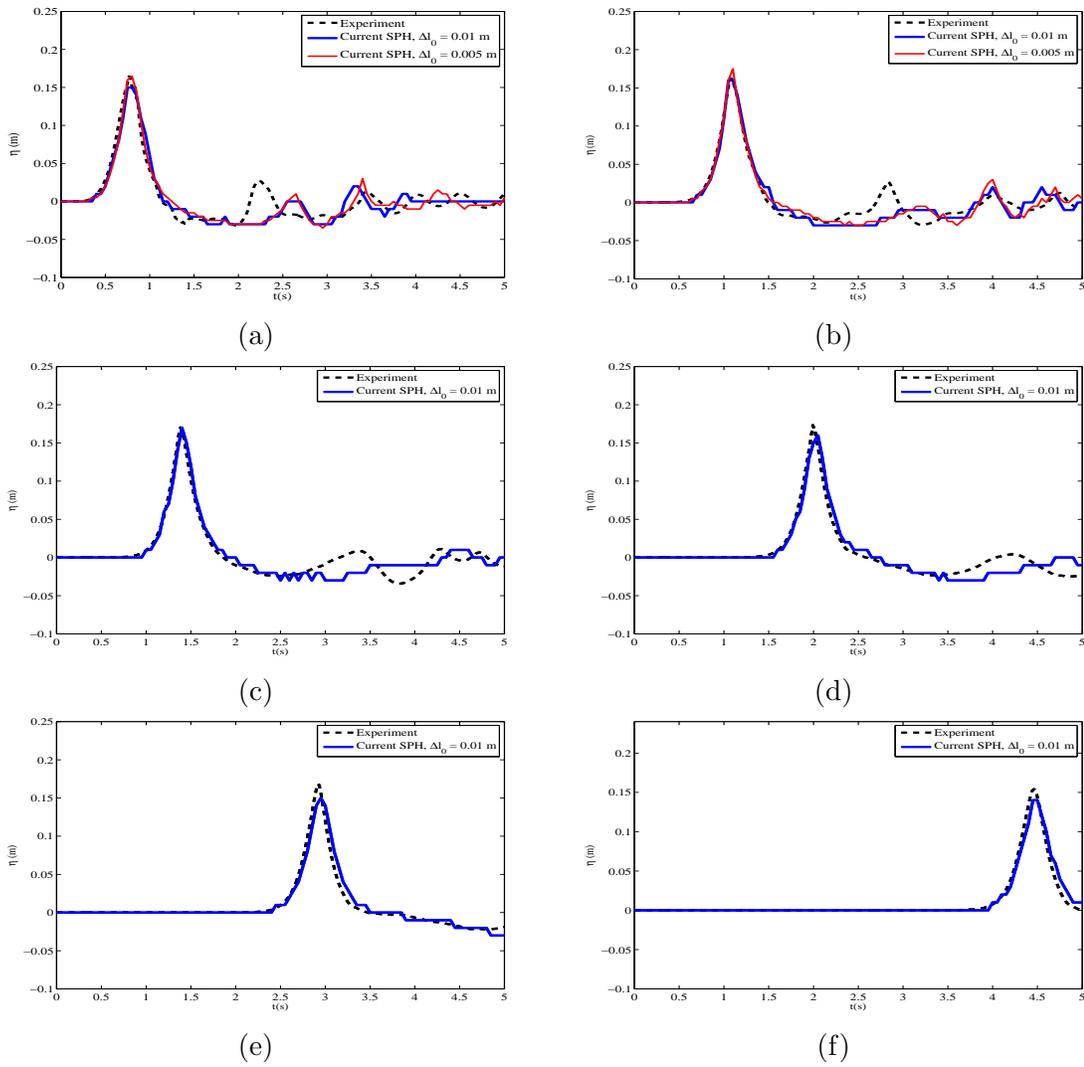
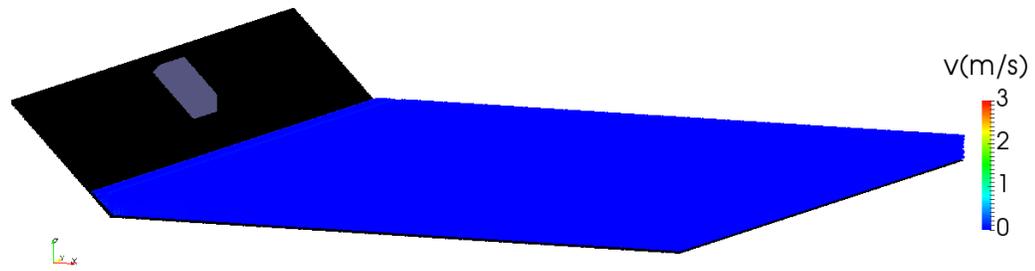
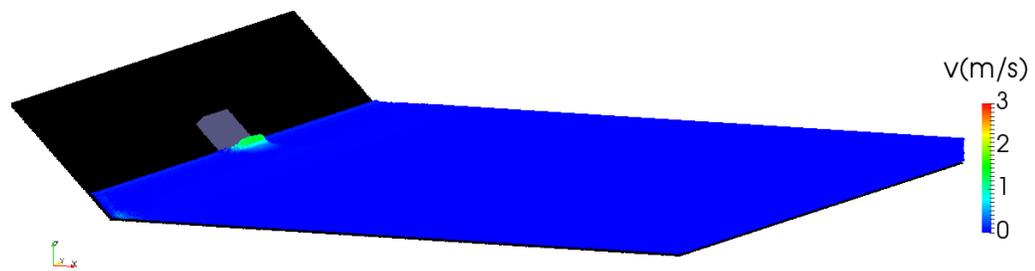


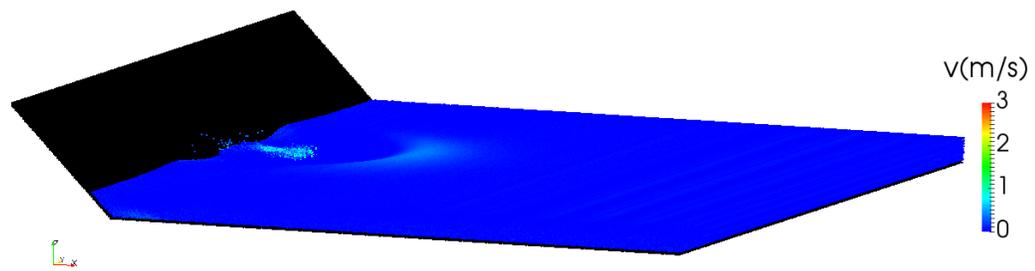
Figure 6.14: Free surface elevation against time at: (a) $x=1.2$ m, (b) $x=1.8$ m, (c) $x=2.4$ m, (d) $x=3.6$ m, (e) $x=5.8$ m, (f) $x=8.4$ m.



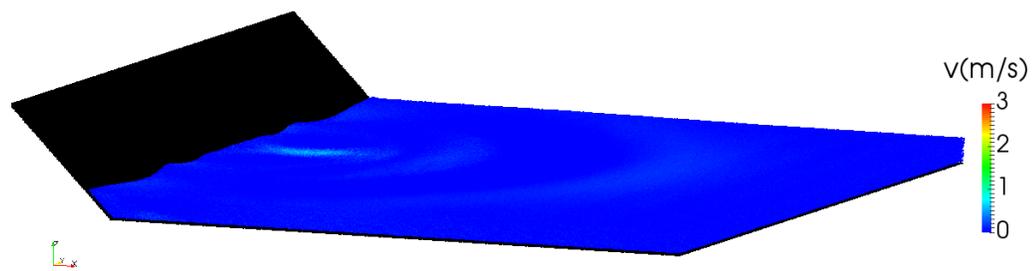
(a)



(b)



(c)



(d)

Figure 6.15: Particles are colored by velocity magnitude at: (a) $t=0$ s, (b) $t=0.5$ s, (c) $t=1.5$ s, (d) $t=3$ s.

The free surface position in 3D at four wave probes (positions are given at Table. 6.2) are compared with the experimental data provided by Heller et al. [1] in Figure 6.16. It is shown that there is a good agreement between our SPH simulations and the experimental data for all four wave probes. By comparing the wave height at the probes, location $x=1.2$ m, $x=1.8$ m and $x=3.6$ m between 2D and 3D simulations, it is clear that the wave in 3D decays faster than 2D. It is shown that at $x=3.6$ m the wave height is a reduced more than six times for the 3D basin simulation, but it is the same in 2D channel.

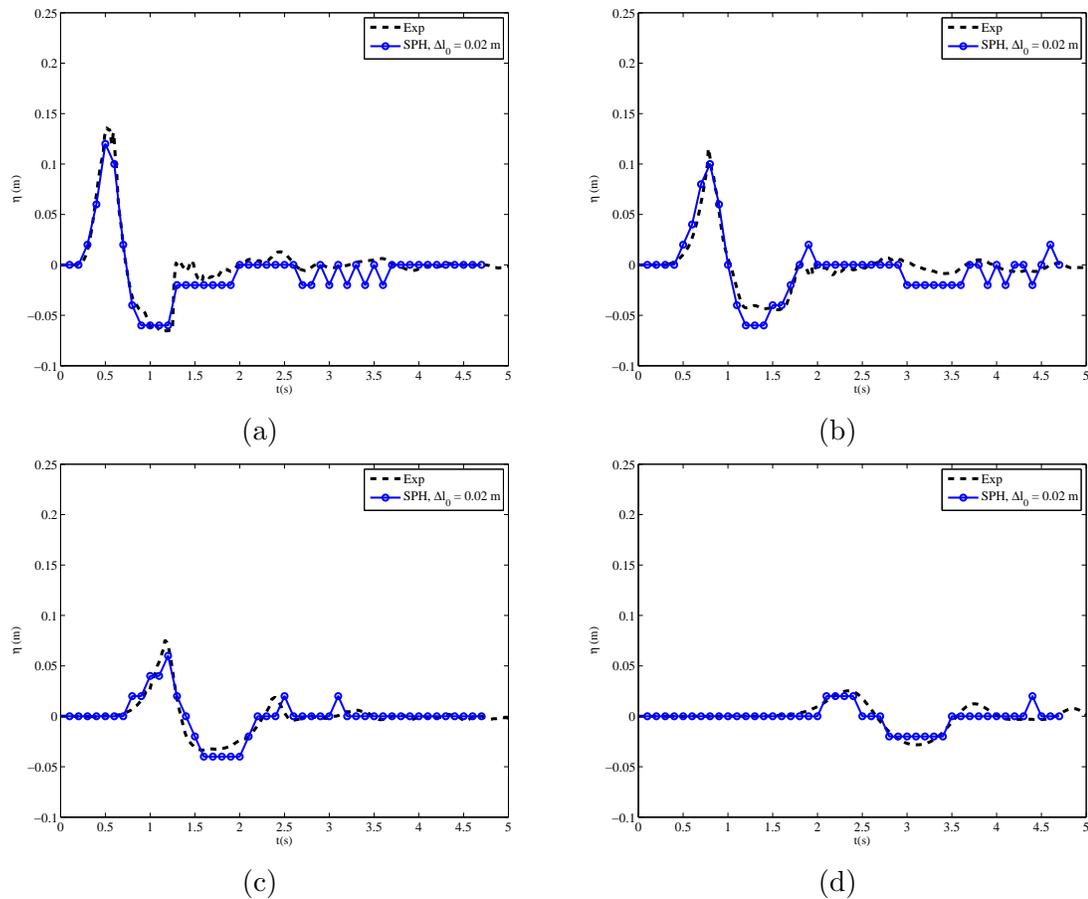


Figure 6.16: Free surface elevation against time at: (a) $x=0.72$ m, (b) $x=1.2$ m, (c) $x=1.8$ m, (d) $x=3.6$ m.

The velocity in the y -direction (shoreline velocity) on the free surface at four time instances of $t=0.5$ s, $t=1.0$ s, $t=1.5$ s and $t=2$ s are shown in Figure 6.17. This clarifies the decay of the wave height in 3D compared to 2D simulations. At $t=0.5$ s

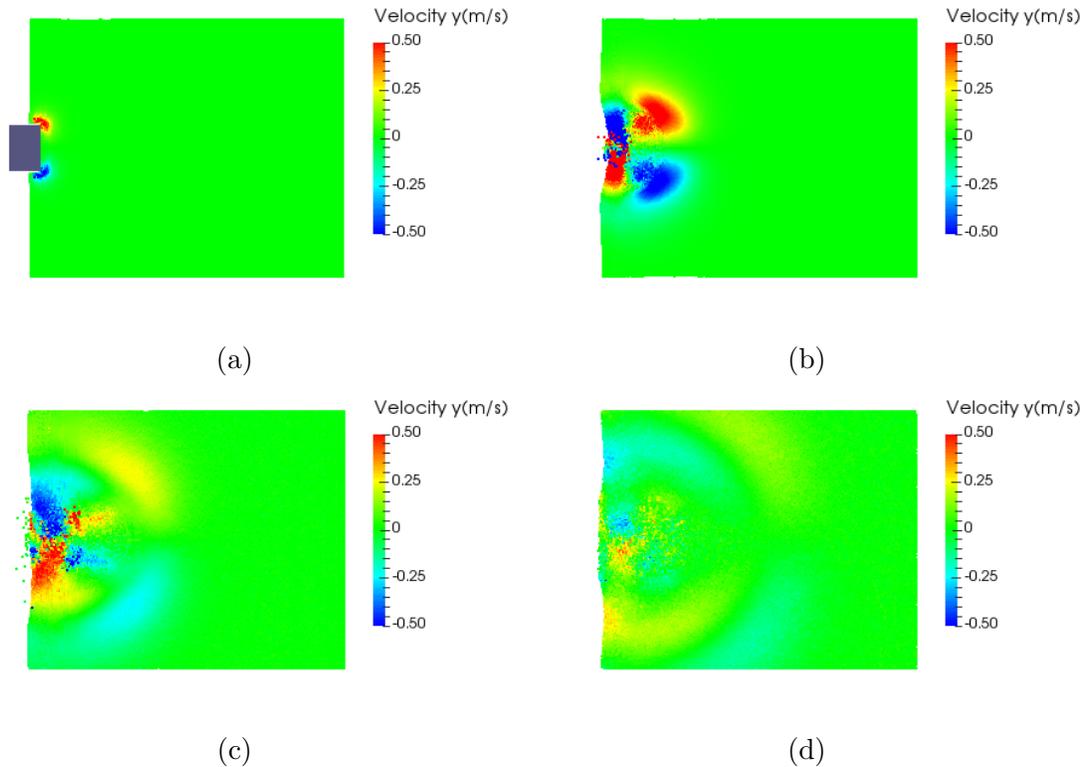


Figure 6.17: Top view of free surface particles colored by velocity in the y -direction at: (a) $t=0.5$ s, (b) $t=1.0$ s, (c) $t=1.5$ s, (d) $t=2$ s.

and $t=1.0$ s, the velocity in the y -direction is relatively higher. However, as time increases the wave decays, as such, the shoreline velocity decreases.

In Figure 6.18, particles are colored by their turbulence eddy viscosity (ν/ν_0) normalized by kinematic viscosity ($\rho=1000$ kg/m^3 , $\mu=0.001$ $kg/m.s$, $\nu_0 = \mu/\rho$). It's shown that the eddy viscosity is three order of magnitude higher than laminar kinematic viscosity between $t=0.5-2$ s. The maximum turbulence eddy viscosity occurs at $t=0.5$ s, when the slide impacts the still water (Figure 6.18a). The turbulence effects are noticeably higher close to the impact origin. These effects propagate in circles and decrease with time.

In Figure 6.19, the plan view of the normalized turbulence eddy viscosity and velocity magnitude are shown for different planes at $t=1$ s, where the slide has just stopped on the bottom of the tank. On the $z=0.02$ m plane, the particles are close to the tank bottom and the turbulent eddy viscosity and corresponding velocity magnitude are higher close to the slide and in particular behind the slide in the wake (Fig-

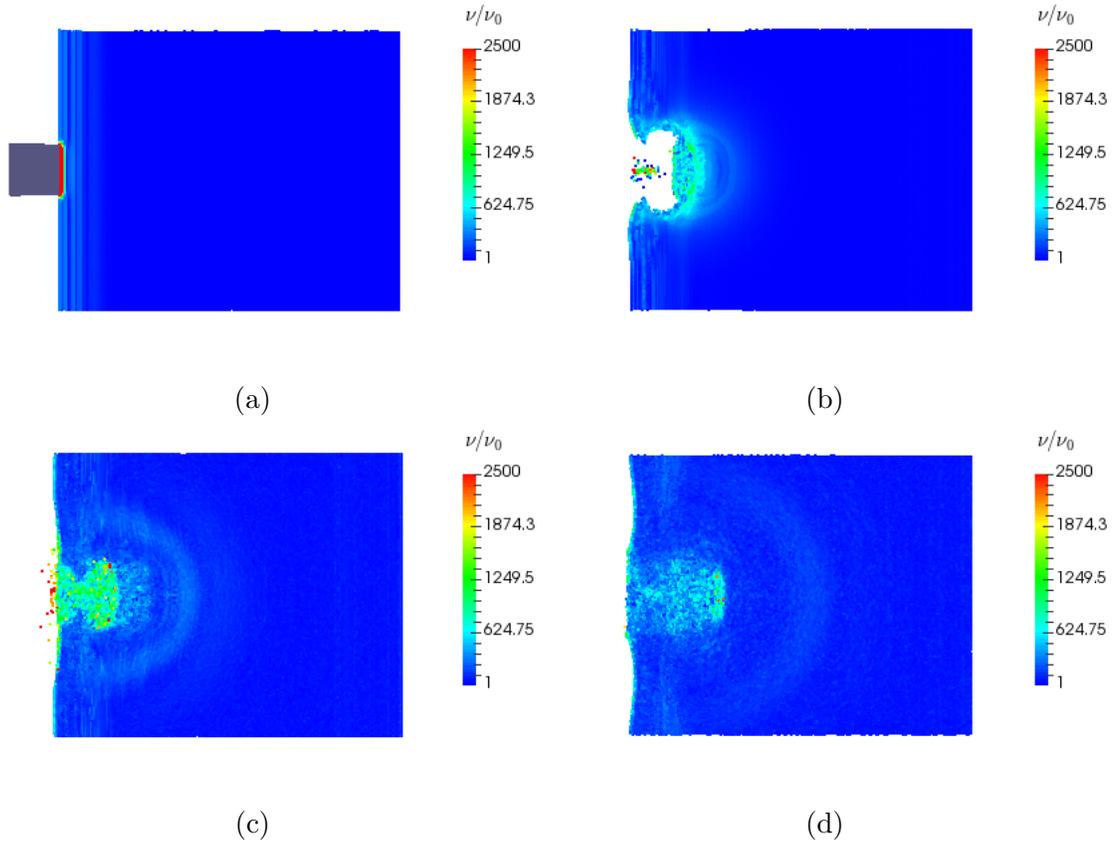


Figure 6.18: Top view of free surface particles colored by dimensionless eddy viscosity (ν/ν_0) at: (a) $t=0.5$ s, (b) $t=1.0$ s, (c) $t=1.5$ s, (d) $t=2$ s.

ures 6.19a and 6.19b). The $z=0.11$ m plane is close to the middle of the still water depth and the turbulent eddy viscosity decreases compared to $z=0.02$ m (Figures 6.19c and 6.19d). The $z=0.22$ m plane is close to the free surface and has smaller eddy viscosity and velocity, compared to the previous mentioned planes (Figures 6.19e and 6.19f). It is shown in Figure 6.19 that at $t=1$ s, the velocity and turbulence effects are decreasing as we get closer to the free surface.

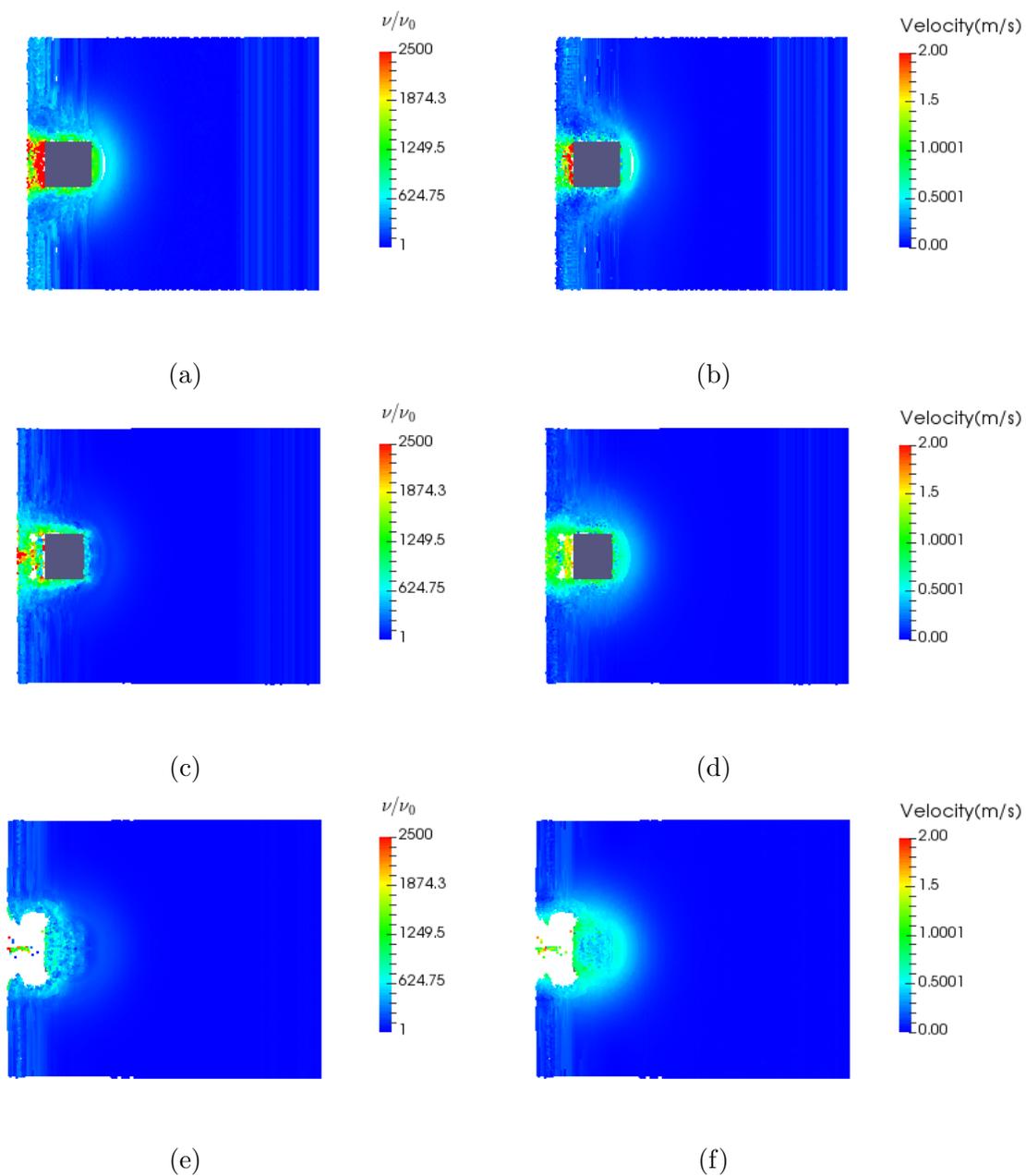


Figure 6.19: Top view of particles colored by dimensionless eddy viscosity and velocity at $t=1.0$ s for: (a and b) $z=0.02$ m, (c and d) $z=0.11$ m, (e and f) $z=0.22$ m.

6.5.2 Deformable Slide

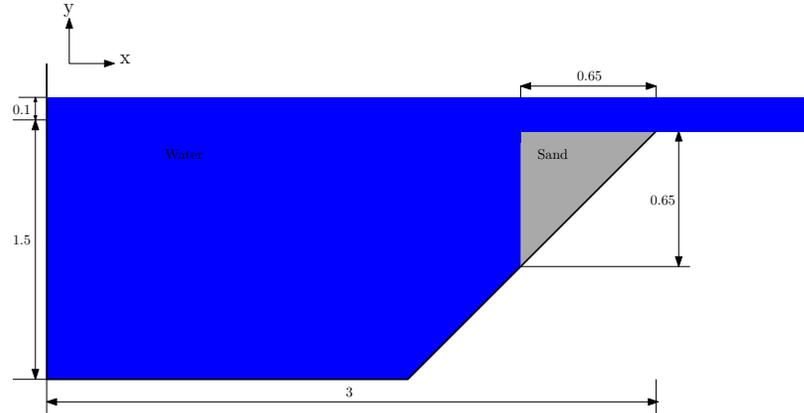


Figure 6.20: Schematic of the experiment of Rządkiwicz et al. [11]. Dimensions are in m.

In this section, the experiment of Rządkiwicz et al. [11], where the slide itself is deformable, is simulated. The deformable slide is simulated by assuming that the slide is a non-Newtonian fluid. The schematic of the experiment is shown in Figure 6.20 where the slide is sand with density of 1950 kg/m^3 . The general Cross model is used as a rheological model to represent the deformable slide [130]. In this model, the effective viscosity is calculated as:

$$\mu_{ef} = \frac{\mu_0 + \frac{\mu_0 \mu_\infty}{\tau_B} \dot{\gamma}}{1 + \frac{\mu_0}{\tau_B} \dot{\gamma}}, \quad (6.5.1)$$

where μ_0 is the viscosity at the very low shear rate, μ_∞ is the viscosity at the very high shear rate, τ_B is the Bingham yield stress and $\dot{\gamma}$ is the shear rate. These parameters are not measured in this experiment and are determined by trial and error [11, 124, 125]. Hence, previous SPH studies [11, 124, 125] used different parameters for μ_∞ and τ_B . In this paper, we set $\mu_0 = 1000\mu_\infty$, $\mu_\infty = 1 \text{ Pa}\cdot\text{s}$ and $\tau_B = 1000 \text{ Pa}$, the same as used by Capone et al. [125].

The particle size for this experiment is set to 0.02 m. The particles are colored by their pressures at $t=0.4 \text{ s}$ (Figure 6.21b) and $t=0.8 \text{ s}$ (Figure 6.21d) and their horizontal velocities (u) at $t=0.4 \text{ s}$ (Figure 6.21a) and $t=0.8 \text{ s}$ (Figure 6.21c).

The generated wave profile at $t=0.4 \text{ s}$ and $t=0.8 \text{ s}$ is compared with the experimental data in Figure 6.22. It is shown that the wave profile is in reasonable agreement

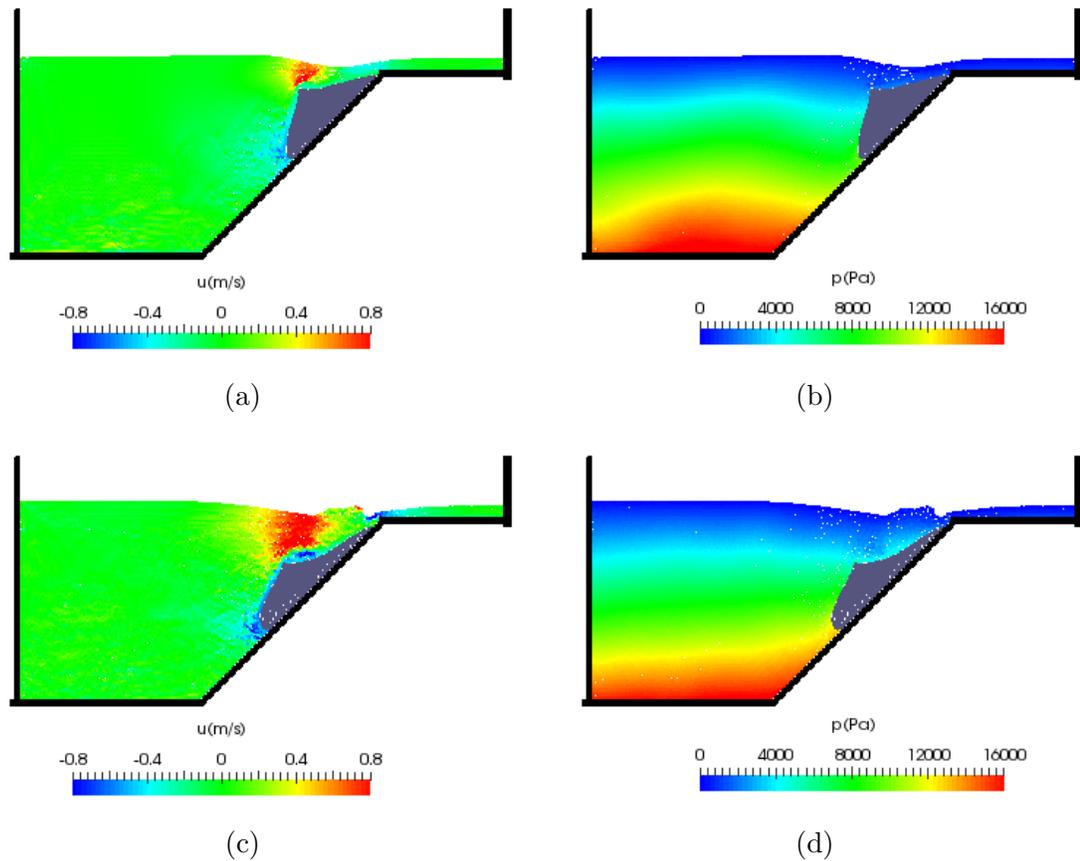


Figure 6.21: Particles are colored with their pressures and horizontal velocities (u) at: (a and b) $t=0.4$ s, (c and d) $t=0.8$ s.

with experimental data at $t=0.4$ s but has some discrepancies at $t=0.8$ s. As was mentioned by Ataie-Ashtiani et al. [124], by increasing τ_B and μ_∞ smaller waves will be generated and hence the experimental data will be reproduced more accurately.

6.5.3 Landslide in a non-Newtonian Reservoir

In this section, a physical experiment, where a large slide falls into a flume of a non-Newtonian fluid, is modeled. The experiment was performed in Northwest Hydraulic Consultant's (NHC) physical lab for two small and large slides of length 0.31 m and 1.2 m, respectively. Here, the large slide is modeled falling into the mixture of water and 6% bentonite. These experiments were carried out to study waves generated by landslides in tailings dams, where tailings is a mining waste product that behaves as

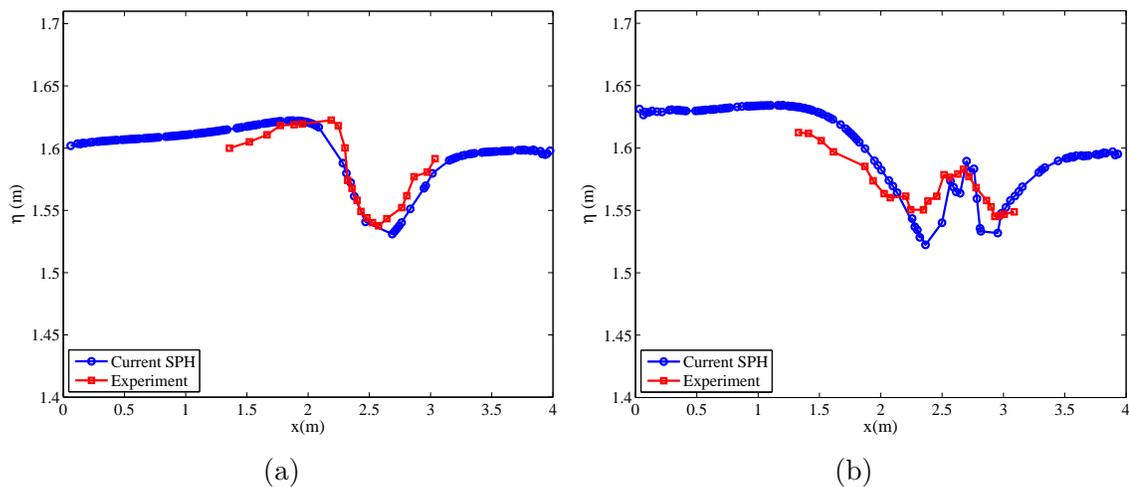


Figure 6.22: Free surface elevation against time at: (a) $t=0.4$ s, (b) $t=0.8$ s

non-Newtonian fluid. The schematic of the experiment with the slide slope of 30° and the dam slope of 45° is shown in Figure 6.23. In this experiment, the particle size is set to 0.01 m leading in 1716656 total particles.

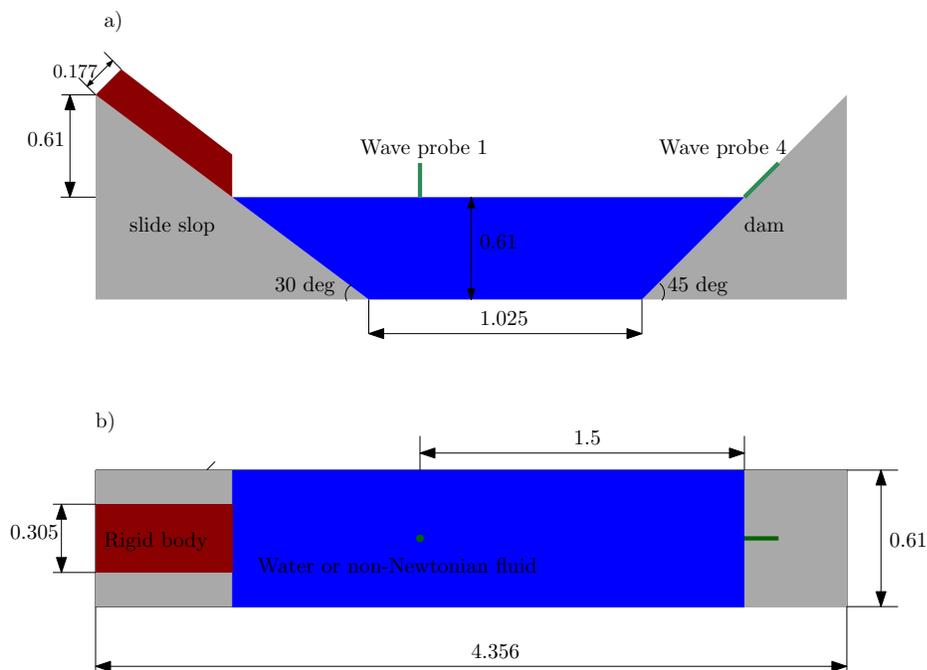


Figure 6.23: Schematic of NHC experiment at: a) side view b) top view. Dimensions are in m.

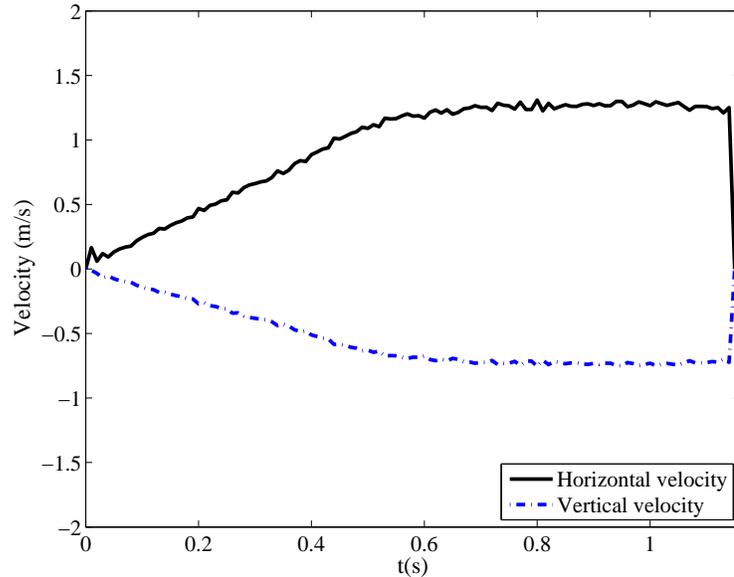


Figure 6.24: Horizontal and vertical velocities of the slide against time in NHC experiment.

The velocity of the slide is measured in the experiment and plotted in Figure 6.24. The yield stress and the viscosity of the water-bentonite mixture is used based on the data measured at a specialized lab. The yield stress is set to $\tau_B=3$ Pa and the viscosity is set $\mu_\infty=0.006$ Pa.s for the SPH simulation.

The experimental data at two wave probes (1 and 4) are used to validate our SPH simulations (Figure 6.23). The simulations are performed for both water, and a water-bentonite mixture. The general Cross model (Eq. 6.5.1) is used to calculate the viscosity of the mixture. The comparison between SPH simulation and experimental data at wave probe 1 is shown for water and water-bentonite mixture in Figures 6.25a and 6.25b, respectively. In both figures, the SPH results are in good agreement up to $t=1.5$ s. After that time, the effect from the side walls lead to deviation of SPH from the experimental results. The wave dynamics are quite complex in this experiment and waves do not seem to follow a regular pattern due to the extreme wave reflection from the walls. However, the overall agreement are fair for both the water and water-bentonite mixture experiments. The wave runup on the dam is measured at wave probe 4 (Figure 6.23). The SPH simulation is plotted against the experimental data for both water and water-bentonite mixture at wave probe 4 in Figures 6.25c and 6.25d, respectively. The maximum runup is captured well by SPH in both experiments, however deviations are clear for the smaller peaks in both

figures due to wall effects. It should be mentioned that the bentonite content in this experiment was not enough to make a noticeable difference between water and water-bentonite mixture. However, this is a start and more experiments can be done using non-Newtonian fluids with higher viscosity and yield stress.

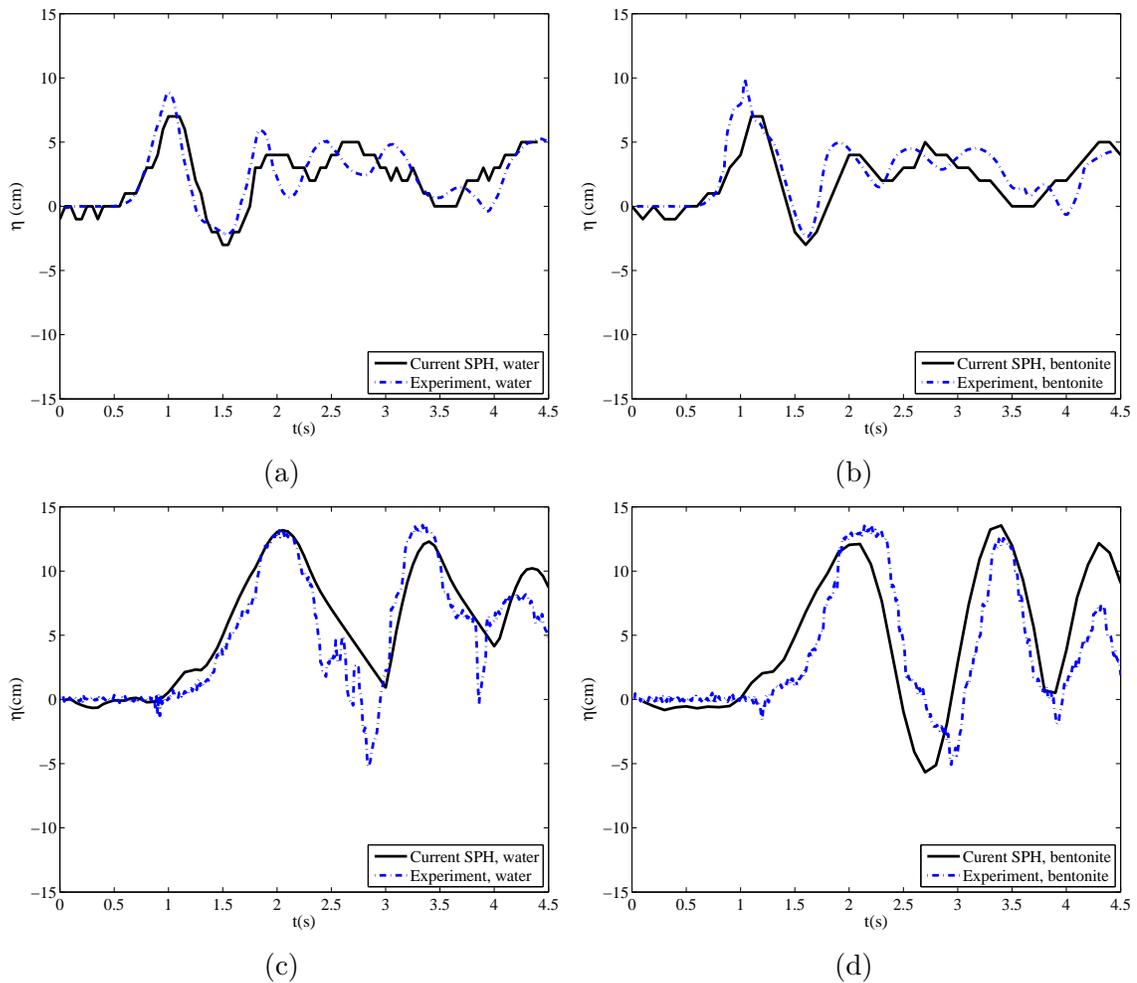


Figure 6.25: Free surface elevation for water and water–bentonite mixture against time at: (a and b) wave probe 1, (c and d) wave probe 4.

The water-bentonite mixture particles are colored by their velocity in the x -direction (u) at $t=1.0$ s, $t=2.0$ s, $t=3.0$ s, $t=4.0$ s. The rigid body slides down the ramp for 1.14 s and stops abruptly as shown in Figure 6.24. Beside the side wall, it is clear from this figure that the wave is reflecting from the dam and leads into the complex wave profiles for both water and water-bentonite mixture as presented in Figure 6.25.

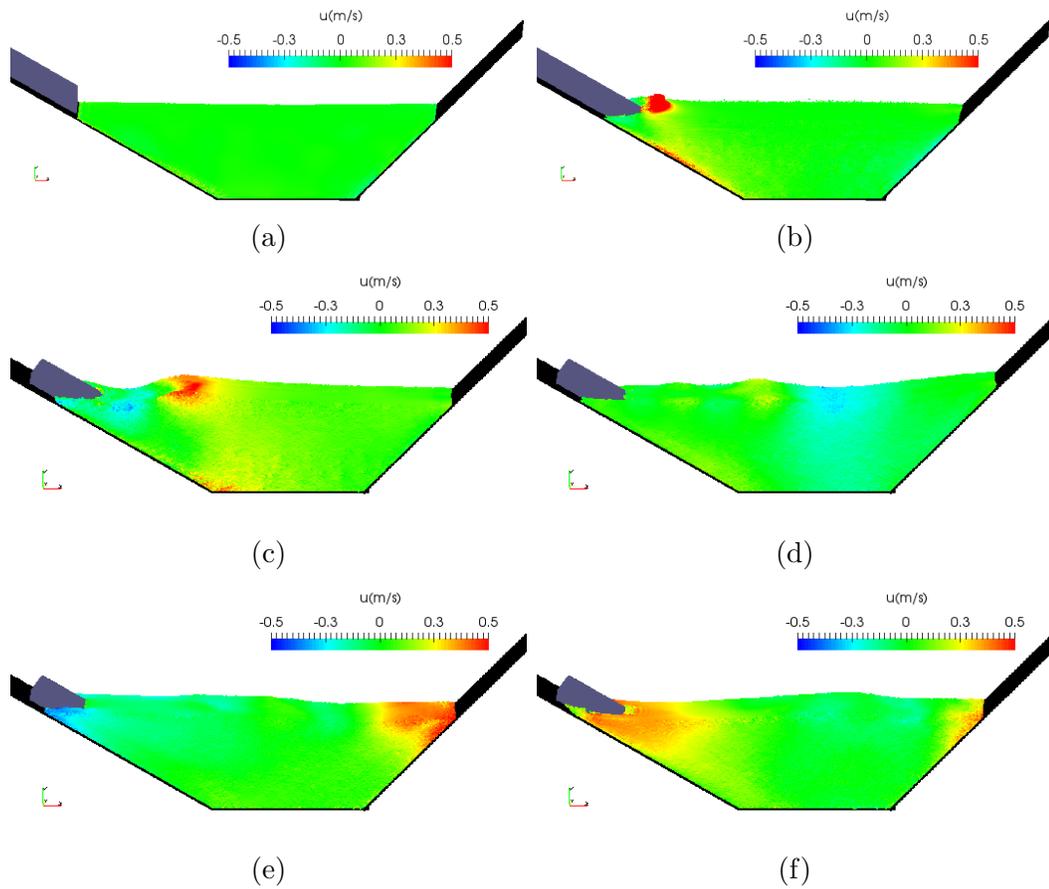


Figure 6.26: Water bentonite mixture particles are colored by their velocity in the x-direction at: (a) $t=0$ s, (b) $t=0.5$ s, (c) $t=1.0$ s, (d) $t=2.0$ s, (e) $t=3.0$ s, (f) $t=4.0$ s.

6.6 Conclusion

Several landslide test cases are simulated in this paper using an in-house parallel incompressible SPH code. The code uses the explicit incompressible turbulent SPH formulation while adopting an accurate boundary condition. The vortex study is presented for subaerial and submarine landslides in 2D. It is shown that the simulation results from the current ISPH code are in a good agreement with experimental data. A comparison between a landslide case in 2D and 3D is performed and compared with the experimental data. It is reported that the wave generated due to the landslide in 3D are several times more dissipative than the 2D case. The normalized turbulence eddy viscosity is reported for the 3D landslide. It is shown that the turbulence effect

are noticeably higher (3 order of magnitude) through the slide-water impact period and in the region close to the impact area.

A new technique is proposed to simulate the dynamics of a slide on an inclined ramp without using the prescribed motion in SPH. By using the new technique, the motion of the slide is calculated based on the surrounding fluid. The comparison between the proposed new technique and the experimental data shows a good agreement.

A deformable slide is simulated by assuming the slide is a Bingham fluid and applying the general Cross model. As expected, the deformable landslide generates waves with smaller amplitude comparing to the rigid slide. Also, by applying the general Cross model, a landslide test case where a rigid slide is falling into the non-Newtonian water-bentonite mixture, is modeled. The results of our proposed ISPH method seems promising for different landslide scenarios.

6.7 Acknowledgment

The financial support for this project is provided by Natural Sciences and Engineering Research Council of Canada (NSERC) under NSERC Engage grant no 495072-16. The authors would also like to thank Compute Canada's WestGrid and Calcul Qubec for their computational resources.

Chapter 7

Conclusions and Future Work

7.1 Conclusions

The key conclusions of this dissertation are as follows:

1. **The unique assembly of SPH sub-models yields a robust method capable of tackling a wide range of applications**

An explicit ISPH method along with accurate boundary conditions is proposed to be used for free surface applications. The ISPH method is used to reduce the WCSPH pressure fluctuation. The fully explicit method is used to respect the explicit nature of the SPH method, for parallelization of the code, and future multi-GPU implementation. The method is verified on chapter 2 by simulating the effect of a transient wave on a deck. The free surface profile and the vertical variation of the horizontal velocity showed a good agreement in comparison with available experimental and previous numerical simulations. The proposed method was tested for a wave-OWSC interaction in chapter 3. The turbulence eddy viscosity model is added to the explicit ISPH method and the proposed model is tested for a comprehensive set of landslide generated waves in chapter 6. In all the cases presented in this dissertation using the proposed explicit ISPH method good agreement has achieved. The explicit ISPH code can be easily converted to the WCSPH or the implicit ISPH method.

2. **The unique OpenMP scheme leads to a maximum speedup of four times**

A new OpenMP parallelization is proposed for the SPH method in chapter 3.

The performance of the code is tested for a dam-break impinging on a tall structure where the initial particle distribution is non-uniform. The horizontal velocity in front of the structure is compared with the available experimental data and good agreement is achieved. The speedup of two times is reported for this case by using the OpenMP scheme. The proposed OpenMP method is validated for the effect of a wave on an OWSC. The accuracy of the method is tested against the experimental data for the angular rotation of the flap, the pressure at two pressure sensors on the flap, and the free surface profile at the flap location and good agreement has achieved. It is concluded that a more efficient parallel design is required for such 3D realistic applications. The OpenMP code is also used to calculate the hydrodynamic loads on the spar of a point absorber WEC in chapter 4. It is shown that the hydrodynamic force is very smaller compared to the inertia force. The OpenMP scheme showed the maximum speedup of almost four times for test cases where the initial particle distribution is uniform. It is concluded that the proposed OpenMP scheme is a relatively easy design to implement and it could be applied for applications with less computational demand.

3. The efficiency of several times higher than the OpenMP scheme is achieved by using a unique MPI-ISPH scheme which is a precursor for future multi-GPU parallelization of ISPH

In chapter 5, a new MPI parallelization method is designed for the SPH method with novel domain decomposition and load balancing schemes. It is shown that by employing the proposed domain decomposition, the initial load imbalance (the maximum load divided by the average load) reduced from 2.5 to 1.01 for a case with non-uniform particle distribution (i.e. dam-break). The effectiveness of the proposed dynamic load balancing is tested for a column of water collapsing in the middle of a tank. It is shown that the load imbalance is only changed from 1.07 where the water column is vertical to 1.08 where the water column is spread on the tank bottom, showing the effectiveness of the proposed method. The speedup and efficiency of the proposed MPI scheme are reported for a tank full of still water showing the efficiency of approximately 83% by using up to 265 million particles and employing 512 processors. The accuracy of the proposed parallel code is tested for a dam break impinging on a short structure. The time history of water height at two wave probes and the pressure

at two pressure sensors on the box are compared with experimental data, VOF and previous implicit ISPH numerical results and showed a good agreement with the experimental data, and an improvement compared to previous implicit ISPH method. For this case, the strong scaling speedup for 1.9 million particles showed a saturation at 64 processors with an efficiency of approximately 35 %. Additionally, the weak scaling speedup for 40 million particles showed the speedup saturation at 256 processors with efficiency of approximately 90%. In summary, the proposed scheme is tested for several test cases and showed a promising accuracy, efficiency and speedup.

4. Slide motion on an inclined ramp is calculated implicitly without prescribing the motion

A novel technique is developed to calculate the motion of a slide on an incline ramp implicitly in chapter 6. The free surface elevation for a rigid slide on an inclined ramp is compared against the experimental data and the case with prescribed motion at two time instants and an excellent agreement is achieved. Also, the vertical slide displacement from the proposed method is compared with the experimental data and previous DNS numerical results. It is shown that both simulation methods are in a good agreement with the experimental data, however an overestimation is observed as time passes. It is concluded that the overestimation could be due to the compressibility effects since in the SPH simulation only one phase is solved but in DNS simulations both air and water phases are considered.

5. Turbulence effects could be three orders of magnitude higher around the impact area for landslide generated waves and the dissipation of the generated waves in 3D is higher compared to 2D simulation

A comprehensive set of simulations for landslide generated waves are performed in both 2D and 3D for rigid and deformable slides by using the unique parallel ISPH method. It is concluded that the wave dissipation is very higher (more than 6 times) in 3D compared to a 2D simulation. Also, it was shown that the turbulence effect is noticeably higher within the impact period, however it is limited around the impact area. For the first time, a landslide case is modeled where a rigid slide falls into a non-Newtonian water-bentonite mixture. This experiments showed a promising application of the SPH method for simulating

tailings dams, where tailings is a mining waste product which behaves as a non-Newtonian fluid.

7.2 Future Work

Although the current parallel explicit method looks promising, the followings studies and modifications should be done as an extension of the current work to make a deployable tool.

- A particle shifting algorithm can be implemented to reduce the non-uniform particle distribution and obtain smoother pressure distribution especially near the free surface with the cost of adding more diffusion to the scheme.
- A dynamic model can be used to calculate the coefficient in the Smagorinsky eddy viscosity turbulence model during the simulation to avoid prescribing or tuning of the coefficient, with the cost of more computational time and complexity.
- More accurate rheological models can be applied to capture the dynamics of the deformable slide instead of the general Cross model used for a deformable slide in this dissertation.
- The efficiency of developed code in this dissertation is verified for several test cases and showed that it can be an appropriate precursor for a future parallel multi-GPU ISPH code.

Bibliography

- [1] V. Heller, M. Bruggemann, J. Spinneken, and B. D. Rogers, “Composite modelling of subaerial landslide–tsunamis in different water body geometries and novel insight into slide and wave kinematics,” *Coastal Engineering*, vol. 109, pp. 20–41, 2016.
- [2] D. T. Cox and J. A. Ortega, “Laboratory observations of green water overtopping a fixed deck,” *Ocean Engineering*, vol. 29, no. 14, pp. 1827 – 1840, 2002.
- [3] H. Lu, C. Yang, R. Löhner, *et al.*, “Numerical studies of green water impact on fixed and moving bodies,” *International Journal of Offshore and Polar Engineering*, vol. 22, no. 2, pp. 123–132, 2012.
- [4] M. Gómez-Gesteira and R. A. Dalrymple, “Using a three-dimensional Smoothed Particle Hydrodynamics method for wave impact on a tall structure,” *Journal of waterway, port, coastal, and ocean engineering*, vol. 130, no. 2, pp. 63–69, 2004.
- [5] K. Kleefsman, G. Fekken, A. Veldman, B. Iwanowski, and B. Buchner, “A volume-of-fluid based simulation method for wave impact problems,” *Journal of Computational Physics*, vol. 206, no. 1, pp. 363–393, 2005.
- [6] K. M. T. Kleefsman, *Water impact loading on offshore structures, a numerical study*. PhD thesis, University of Groningen, 2005.
- [7] E.-S. Lee, D. Violeau, R. Issa, and S. Ploix, “Application of weakly compressible and truly incompressible SPH to 3-D water collapse in waterworks,” *Journal of Hydraulic Research*, vol. 48, no. S1, pp. 50–60, 2010.

- [8] P. Heinrich, “Nonlinear water waves generated by submarine and aerial landslides,” *Journal of Waterway, Port, Coastal, and Ocean Engineering*, vol. 118, no. 3, pp. 249–266, 1992.
- [9] S. Abadie, D. Morichon, S. Grilli, and S. Glockner, “Numerical simulation of waves generated by landslides using a multiple-fluid Navier–Stokes model,” *Coastal engineering*, vol. 57, no. 9, pp. 779–794, 2010.
- [10] J. Monaghan and A. Kos, “Scott russells wave generator,” *Physics of Fluids (1994-present)*, vol. 12, no. 3, pp. 622–630, 2000.
- [11] S. A. Rzadkiewicz, C. Mariotti, and P. Heinrich, “Numerical simulation of submarine landslides and their hydraulic effects,” *Journal of Waterway, Port, Coastal, and Ocean Engineering*, vol. 123, no. 4, pp. 149–157, 1997.
- [12] D. Li, *Encyclopedia of microfluidics and nanofluidics*, vol. 1. Springer, 2008.
- [13] I. Hadi, J. Hennig, M. Peri, and Y. Xing-Kaeding, “Computation of flow-induced motion of floating bodies,” *Applied Mathematical Modelling*, vol. 29, no. 12, pp. 1196 – 1210, 2005.
- [14] P. M. Carrica, R. V. Wilson, R. W. Noack, and F. Stern, “Ship motions using single-phase level set with dynamic overset grids,” *Computers & Fluids*, vol. 36, no. 9, pp. 1415–1433, 2007.
- [15] B. Bouscasse, A. Colagrossi, S. Marrone, and M. Antuono, “Nonlinear water wave interaction with floating bodies in SPH,” *Journal of Fluids and Structures*, vol. 42, pp. 112–129, 2013.
- [16] C. W. Hirt and B. D. Nichols, “Volume of fluid (VOF) method for the dynamics of free boundaries,” *Journal of computational physics*, vol. 39, no. 1, pp. 201–225, 1981.
- [17] M. Sussman, P. Smereka, and S. Osher, “A level set approach for computing solutions to incompressible two-phase flow,” *Journal of Computational physics*, vol. 114, no. 1, pp. 146–159, 1994.
- [18] G. R. Liu and M. Liu, *Smoothed Particle Hydrodynamics: a meshfree particle method*. World Scientific, 2003.

- [19] M. Hashemi, R. Fatehi, and M. Manzari, “A modified SPH method for simulating motion of rigid bodies in Newtonian fluid flows,” *International Journal of Non-Linear Mechanics*, vol. 47, no. 6, pp. 626–638, 2012.
- [20] L. B. Lucy, “A numerical approach to the testing of the fission hypothesis,” *The astronomical journal*, vol. 82, pp. 1013–1024, 1977.
- [21] R. A. Gingold and J. J. Monaghan, “Smoothed Particle Hydrodynamics-theory and application to non-spherical stars,” *Monthly notices of the royal astronomical society*, vol. 181, pp. 375–389, 1977.
- [22] M. Shadloo, G. Oger, and D. Le Touzé, “Smoothed Particle Hydrodynamics method for fluid flows, towards industrial applications: Motivations, current state, and challenges,” *Computers & Fluids*, vol. 136, pp. 11–34, 2016.
- [23] J. J. Monaghan, “Simulating free surface flows with SPH,” *Journal of computational physics*, vol. 110, no. 2, pp. 399–406, 1994.
- [24] S. J. Cummins and M. Rudman, “An SPH projection method,” *Journal of computational physics*, vol. 152, no. 2, pp. 584–607, 1999.
- [25] A. Rafiee, S. Cummins, M. Rudman, and K. Thiagarajan, “Comparative study on the accuracy and stability of SPH schemes in simulating energetic free-surface flows,” *European Journal of Mechanics-B/Fluids*, vol. 36, pp. 1–16, 2012.
- [26] A. Colagrossi and M. Landrini, “Numerical simulation of interfacial flows by Smoothed Particle Hydrodynamics,” *Journal of Computational Physics*, vol. 191, no. 2, pp. 448–475, 2003.
- [27] M. Antuono, A. Colagrossi, S. Marrone, and D. Molteni, “Free-surface flows solved by means of SPH schemes with numerical diffusive terms,” *Computer Physics Communications*, vol. 181, no. 3, pp. 532–549, 2010.
- [28] D. Molteni and A. Colagrossi, “A simple procedure to improve the pressure evaluation in hydrodynamic context using the SPH,” *Computer Physics Communications*, vol. 180, no. 6, pp. 861–872, 2009.
- [29] S. Marrone, M. Antuono, A. Colagrossi, G. Colicchio, D. Le Touzé, and G. Graziani, “ δ -SPH model for simulating violent impact flows,” *Computer*

- Methods in Applied Mechanics and Engineering*, vol. 200, no. 13, pp. 1526–1542, 2011.
- [30] S. Hosseini, M. Manzari, and S. Hannani, “A fully explicit three-step SPH algorithm for simulation of non-Newtonian fluid flow,” *International Journal of Numerical Methods for Heat & Fluid Flow*, vol. 17, no. 7, pp. 715–735, 2007.
- [31] A. Rafiee and K. P. Thiagarajan, “An SPH projection method for simulating fluid-hypoeelastic structure interaction,” *Computer Methods in Applied Mechanics and Engineering*, vol. 198, no. 33, pp. 2785–2795, 2009.
- [32] A. Khayyer, H. Gotoh, and S. Shao, “Corrected incompressible SPH method for accurate water-surface tracking in breaking waves,” *Coastal Engineering*, vol. 55, no. 3, pp. 236–250, 2008.
- [33] S. Adami, X. Hu, and N. Adams, “A generalized wall boundary condition for Smoothed Particle Hydrodynamics,” *Journal of Computational Physics*, vol. 231, no. 21, pp. 7057–7075, 2012.
- [34] A. H erault, G. Bilotta, and R. A. Dalrymple, “SPH on GPU with CUDA,” *Journal of Hydraulic Research*, vol. 48, no. S1, pp. 74–79, 2010.
- [35] S. Yeylaghi, B. Moa, P. Oshkai, B. Buckham, and C. Crawford, “ISPH modelling of an oscillating wave surge converter using an OpenMP-based parallel approach,” *Journal of Ocean Engineering and Marine Energy*, pp. 1–12, 2016.
- [36] S. Yeylaghi, B. Moa, P. Oshkai, B. Buckham, and C. Crawford, “ISPH modelling for hydrodynamic applications using a new MPI-based parallel approach,” *Journal of Ocean Engineering and Marine Energy*, pp. 1–17.
- [37] X. Guo, S. Lind, B. Rogers, P. Stansby, and M. Ashworth, “Efficient massive parallelisation for incompressible Smoothed Particle Hydrodynamics with 10^8 particles,” in *Proceedings of the 8th International SPHERIC Workshop, Trondheim, Norway*, pp. 397–402, 2013.
- [38] V. Springel, “The cosmological simulation code GADGET-2,” *Monthly Notices of the Royal Astronomical Society*, vol. 364, no. 4, pp. 1105–1134, 2005.

- [39] G. Oger, D. Le Touzé, D. Guibert, M. de Leffe, J. Biddiscombe, J. Soumagne, and J.-G. Piccinali, “On distributed memory MPI-based parallelization of SPH codes in massive HPC context,” *Computer Physics Communications*, vol. 200, pp. 1–14, 2016.
- [40] B. Rogers, R. Dalrymple, M. Gómez-Gesteira, and A. Crespo, “User guide for the parallelphysics code using mpi v2. 0,” 2011.
- [41] S. Marrone, B. Bouscasse, A. Colagrossi, and M. Antuono, “Study of ship wave breaking patterns using 3D parallel SPH simulations,” *Computers & Fluids*, vol. 69, pp. 54–66, 2012.
- [42] J. Monaghan, “Smoothed Particle Hydrodynamics and its diverse applications,” *Annual Review of Fluid Mechanics*, vol. 44, pp. 323–346, 2012.
- [43] B. Buchner *et al.*, “The impact of green water on FPSO design,” in *Offshore technology conference*, Offshore Technology Conference, 1995.
- [44] C. Franco and L. Franco, “Overtopping formulas for caisson breakwaters with nonbreaking 3D waves,” *Journal of Waterway, Port, Coastal, and Ocean Engineering*, vol. 125, no. 2, pp. 98–108, 1999.
- [45] M. Gómez-Gesteira, D. Cerqueiro, C. Crespo, and R. Dalrymple, “Green water overtopping analyzed with a SPH model,” *Ocean Engineering*, vol. 32, no. 2, pp. 223–238, 2005.
- [46] H. Wendland, “Piecewise polynomial, positive definite and compactly supported radial functions of minimal degree,” *Advances in computational Mathematics*, vol. 4, no. 1, pp. 389–396, 1995.
- [47] J. J. Monaghan, “Smoothed Particle Hydrodynamics,” *Annual review of astronomy and astrophysics*, vol. 30, pp. 543–574, 1992.
- [48] S. Lind, R. Xu, P. Stansby, and B. Rogers, “Incompressible Smoothed Particle Hydrodynamics for free-surface flows: A generalised diffusion-based algorithm for stability and validations for impulsive flows and propagating waves,” *Journal of Computational Physics*, vol. 231, no. 4, pp. 1499–1523, 2012.

- [49] S. Shao and C. Ji, “SPH computation of plunging waves using a 2D sub-particle scale (SPS) turbulence model,” *International Journal for numerical methods in fluids*, vol. 51, no. 8, pp. 913–936, 2006.
- [50] E.-S. Lee, C. Moulinec, R. Xu, D. Violeau, D. Laurence, and P. Stansby, “Comparisons of weakly compressible and truly incompressible algorithms for the SPH mesh free particle method,” *Journal of Computational Physics*, vol. 227, no. 18, pp. 8417–8436, 2008.
- [51] A. Henry, A. Rafiee, P. Schmitt, F. Dias, and T. Whittaker, “The characteristics of wave impacts on an oscillating wave surge converter,” *Journal of Ocean and Wind Energy*, vol. 1, no. 2, pp. 101–110, 2014.
- [52] B. D. Rogers, R. A. Dalrymple, and P. K. Stansby, “Simulation of caisson breakwater movement using 2D SPH,” *Journal of Hydraulic Research*, vol. 48, no. S1, pp. 135–141, 2010.
- [53] A. Babarit, J. Hals, M. Muliawan, A. Kurniawan, T. Moan, and J. Krokstad, “Numerical benchmarking study of a selection of wave energy converters,” *Renewable Energy*, vol. 41, pp. 44–63, 2012.
- [54] M. Folley, T. Whittaker, and J. Vant Hoff, “The design of small seabed-mounted bottom-hinged wave energy converters,” in *Proceedings of the 7th European Wave and Tidal Energy Conference, Porto, Portugal*, vol. 455, 2007.
- [55] A. Henry, O. Kimmoun, J. Nicholson, G. Dupont, Y. Wei, F. Dias, *et al.*, “A two dimensional experimental investigation of slamming of an oscillating wave surge converter,” in *The Twenty-fourth International Ocean and Polar Engineering Conference*, International Society of Offshore and Polar Engineers, 2014.
- [56] G. Oger, M. Doring, B. Alessandrini, and P. Ferrant, “Two-dimensional SPH simulations of wedge water entries,” *Journal of Computational Physics*, vol. 213, no. 2, pp. 803–822, 2006.
- [57] R. Zhao, O. Faltinsen, and J. Aarsnes, “Water entry of arbitrary two-dimensional sections with and without flow separation,” in *Proceedings of the 21st symposium on naval hydrodynamics*, pp. 408–423, Trondheim, Norway, National Academy Press, Washington, DC, USA, 1996.

- [58] D. Clabby and K. Tease, “Extreme loads and pressures applied to SurgeWEC: a small oscillating wave surge converter,” in *Proceedings of the 11th European Wave and Tidal Energy Conference, Nantes, France, 2015*.
- [59] M. Folley, T. Whittaker, and A. Henry, “The effect of water depth on the performance of a small surging wave energy converter,” *Ocean Engineering*, vol. 34, no. 8, pp. 1265–1274, 2007.
- [60] E. Renzi and F. Dias, “Hydrodynamics of the oscillating wave surge converter in the open ocean,” *European Journal of Mechanics-B/Fluids*, vol. 41, pp. 1–10, 2013.
- [61] E. Renzi and F. Dias, “Resonant behaviour of an oscillating wave energy converter in a channel,” *Journal of Fluid Mechanics*, vol. 701, pp. 482–510, 2012.
- [62] Y. Wei, A. Rafiee, A. Henry, and F. Dias, “Wave interaction with an oscillating wave surge converter, part I: Viscous effects,” *Ocean Engineering*, vol. 104, pp. 185–203, 2015.
- [63] P. Schmitt and B. Elsaesser, “On the use of OpenFOAM to model oscillating wave surge converters,” *Ocean Engineering*, vol. 108, pp. 98 – 104, 2015.
- [64] A. Rafiee and F. Dias, “Two-dimensional and three-dimensional simulation of wave interaction with an oscillating wave surge converter,” in *International workshop on water waves and floating bodies (IWWWF) 2013, Marseille, France, 2013*.
- [65] S. Shao and E. Y. Lo, “Incompressible SPH method for simulating Newtonian and non-Newtonian flows with a free surface,” *Advances in Water Resources*, vol. 26, no. 7, pp. 787–800, 2003.
- [66] A. J. Chorin, “Numerical solution of the Navier-Stokes equations,” *Mathematics of computation*, vol. 22, no. 104, pp. 745–762, 1968.
- [67] A. Khayyer, H. Gotoh, and S. Shao, “Enhanced predictions of wave impact pressure by improved incompressible SPH methods,” *Applied Ocean Research*, vol. 31, no. 2, pp. 111–131, 2009.

- [68] J. Monaghan and A. Kos, “Solitary waves on a Cretan beach,” *Journal of Waterway, Port, Coastal, and Ocean Engineering*, vol. 125, no. 3, pp. 145–155, 1999.
- [69] S. Koshizuka, A. Nobe, and Y. Oka, “Numerical analysis of breaking waves using the moving particle semi-implicit method,” *International Journal for Numerical Methods in Fluids*, vol. 26, no. 7, pp. 751–769, 1998.
- [70] E. Y. Lo and S. Shao, “Simulation of near-shore solitary wave mechanics by an incompressible SPH method,” *Applied Ocean Research*, vol. 24, no. 5, pp. 275–286, 2002.
- [71] M. Hermanns, “Parallel programming in Fortran 95 using openmp,” *Universidad Politecnica de Madrid, Spain*, 2002.
- [72] A. Ferrari, M. Dumbser, E. F. Toro, and A. Armanini, “A new 3D parallel SPH scheme for free surface flows,” *Computers & Fluids*, vol. 38, no. 6, pp. 1203–1217, 2009.
- [73] M. Gomez-Gesteira, B. D. Rogers, A. J. Crespo, R. A. Dalrymple, M. Narayanaswamy, and J. M. Dominguez, “SPHysics—development of a free-surface fluid solver—part 1: Theory and formulations,” *Computers & Geosciences*, vol. 48, pp. 289–299, 2012.
- [74] A. Crespo, J. Domínguez, B. Rogers, M. Gómez-Gesteira, S. Longshaw, R. Canelas, R. Vacondio, A. Barreiro, and O. García-Feal, “DualSPHysics: Open-source parallel CFD solver based on Smoothed Particle Hydrodynamics (SPH),” *Computer Physics Communications*, vol. 187, pp. 204–216, 2015.
- [75] R. Xu, P. Stansby, and D. Laurence, “Accuracy and stability in incompressible SPH (ISPH) based on the projection method and a new approach,” *Journal of Computational Physics*, vol. 228, no. 18, pp. 6703–6725, 2009.
- [76] M. J. Quinn, *Parallel Programming*, vol. 526. TMH CSE, 2003.
- [77] X. Liu, P. Lin, and S. Shao, “An ISPH simulation of coupled structure interaction with free surface flows,” *Journal of Fluids and Structures*, vol. 48, pp. 46–61, 2014.

- [78] R. A. Dalrymple and R. G. Dean, *Water wave mechanics for engineers and scientists*. Prentice-Hall, 1991.
- [79] H. Takeda, S. M. Miyama, and M. Sekiya, “Numerical simulation of viscous flow by Smoothed Particle Hydrodynamics,” *Progress of Theoretical Physics*, vol. 92, no. 5, pp. 939–960, 1994.
- [80] J. P. Morris, P. J. Fox, and Y. Zhu, “Modeling low Reynolds number incompressible flows using SPH,” *Journal of computational physics*, vol. 136, no. 1, pp. 214–226, 1997.
- [81] R. Fatehi and M. Manzari, “Error estimation in Smoothed Particle Hydrodynamics and a new scheme for second derivatives,” *Computers & Mathematics with Applications*, vol. 61, no. 2, pp. 482–498, 2011.
- [82] J. Monaghan and J. Kajtar, “SPH particle boundary forces for arbitrary boundaries,” *Computer Physics Communications*, vol. 180, no. 10, pp. 1811–1820, 2009.
- [83] A. Colagrossi, “A meshless Lagrangian method for free-surface and interface flows with fragmentation,” 2005.
- [84] N. G. Jacobsen, D. R. Fuhrman, and J. Fredsøe, “A wave generation toolbox for the open-source CFD library: OpenFoam®,” *International Journal for Numerical Methods in Fluids*, vol. 70, no. 9, pp. 1073–1088, 2012.
- [85] J. Martin and W. Moyce, “Part iv. an experimental study of the collapse of liquid columns on a rigid horizontal plane,” *Philosophical Transactions of the Royal Society of London. Series A, Mathematical and Physical Sciences*, vol. 244, no. 882, pp. 312–324, 1952.
- [86] L. D. Libersky, A. G. Petschek, T. C. Carney, J. R. Hipp, and F. A. Allahdadi, “High strain Lagrangian hydrodynamics: A three-dimensional SPH code for dynamic material response,” *Journal of Computational Physics*, vol. 109, no. 1, pp. 67 – 75, 1993.
- [87] C. Zhou, G. Liu, and K. Lou, “Three-dimensional penetration simulation using Smoothed Particle Hydrodynamics,” *International Journal of Computational Methods*, vol. 4, no. 04, pp. 671–691, 2007.

- [88] R. Issa, E. S. Lee, D. Violeau, and D. R. Laurence, “Incompressible separated flows simulations with the Smoothed Particle Hydrodynamics gridless method,” *International journal for numerical methods in fluids*, vol. 47, no. 10–11, pp. 1101–1106, 2005.
- [89] J. Jeong, M. Jhon, J. Halow, and J. Van Osdol, “Smoothed Particle Hydrodynamics: Applications to heat conduction,” *Computer Physics Communications*, vol. 153, no. 1, pp. 71–84, 2003.
- [90] K. Szewc, J. Pozorski, and A. Taniere, “Modeling of natural convection with Smoothed Particle Hydrodynamics: non-Boussinesq formulation,” *International Journal of Heat and Mass Transfer*, vol. 54, no. 23, pp. 4807–4816, 2011.
- [91] J. Vandamme, Q. Zou, and D. E. Reeve, “Modeling floating object entry and exit using Smoothed Particle Hydrodynamics,” *Journal of Waterway, Port, Coastal, and Ocean Engineering*, vol. 137, no. 5, pp. 213–224, 2011.
- [92] S. Marrone, A. Colagrossi, M. Antuono, C. Lugni, and M. Tulin, “A 2D+t SPH model to study the breaking wave pattern generated by fast ships,” *Journal of Fluids and Structures*, vol. 27, no. 8, pp. 1199–1215, 2011.
- [93] R. B. Canelas, A. J. Crespo, J. M. Domínguez, R. M. Ferreira, and M. Gómez-Gesteira, “SPH–DCDEM model for arbitrary geometries in free surface solid–fluid flows,” *Computer Physics Communications*, vol. 202, pp. 131–140, 2016.
- [94] D. Violeau and B. D. Rogers, “Smoothed Particle Hydrodynamics (SPH) for free-surface flows: past, present and future,” *Journal of Hydraulic Research*, pp. 1–26, 2016.
- [95] H. Gotoh, S. Shao, and T. Memita, “SPH-LES model for numerical investigation of wave interaction with partially immersed breakwater,” *Coastal Engineering Journal*, vol. 46, no. 01, pp. 39–63, 2004.
- [96] H. Gotoh, A. Khayyer, H. Ikari, T. Arikawa, and K. Shimosako, “On enhancement of incompressible SPH method for simulation of violent sloshing flows,” *Applied Ocean Research*, vol. 46, pp. 104–115, 2014.

- [97] J. Ren, T. Jiang, W. Lu, and G. Li, “An improved parallel SPH approach to solve 3D transient generalized Newtonian free surface flows,” *Computer Physics Communications*, vol. 205, pp. 87 – 105, 2016.
- [98] D. Violeau and A. Leroy, “Optimal time step for incompressible SPH,” *Journal of Computational Physics*, vol. 288, pp. 119–130, 2015.
- [99] H. Gotoh and A. Khayyer, “Current achievements and future perspectives for projection-based particle methods with applications in ocean engineering,” *Journal of Ocean Engineering and Marine Energy*, pp. 1–28, 2016.
- [100] S. Lind, P. Stansby, and B. Rogers, “Fixed and moored bodies in steep and breaking waves using SPH with the Froude–Krylov approximation,” *Journal of Ocean Engineering and Marine Energy*, pp. 1–24, 2016.
- [101] J. M. Domínguez, A. J. Crespo, D. Valdez-Balderas, B. D. Rogers, and M. Gómez-Gesteira, “New multi-GPU implementation for Smoothed Particle Hydrodynamics on heterogeneous clusters,” *Computer Physics Communications*, vol. 184, no. 8, pp. 1848–1860, 2013.
- [102] R. Vacondio, P. Mignosa, and S. Pagani, “3D SPH numerical simulation of the wave generated by the Vajont rockslide,” *Advances in Water Resources*, vol. 59, pp. 146–156, 2013.
- [103] A. Barreiro, A. J. Crespo, J. M. Domínguez, and M. Gómez-Gesteira, “Smoothed Particle Hydrodynamics for coastal engineering problems,” *Computers & Structures*, vol. 120, pp. 96–106, 2013.
- [104] J. M. Domínguez, A. J. Crespo, and M. Gómez-Gesteira, “Optimization strategies for CPU and GPU implementations of a Smoothed Particle Hydrodynamics method,” *Computer Physics Communications*, vol. 184, no. 3, pp. 617–627, 2013.
- [105] J. Cherfils, G. Pinon, and E. Rivoalen, “Josephine: A parallel SPH code for free-surface flows,” *Computer Physics Communications*, vol. 183, no. 7, pp. 1468–1480, 2012.

- [106] A. Mokos, B. D. Rogers, P. K. Stansby, and J. M. Domínguez, “Multi-phase SPH modelling of violent hydrodynamics on GPUs,” *Computer Physics Communications*, vol. 196, pp. 304–316, 2015.
- [107] J. L. Cercos-Pita, “AQUAghusph, a new free 3D SPH solver accelerated with OpenCL,” *Computer Physics Communications*, vol. 192, pp. 295–312, 2015.
- [108] D. Valdez-Balderas, J. M. Domínguez, B. D. Rogers, and A. J. Crespo, “Towards accelerating Smoothed Particle Hydrodynamics simulations for free-surface flows on multi-GPU clusters,” *Journal of Parallel and Distributed Computing*, vol. 73, no. 11, pp. 1483–1493, 2013.
- [109] H. Gotoh, T. Shibahara, and T. Sakai, “Sub-particle-scale turbulence model for the MPS method-Lagrangian flow model for hydraulic engineering,” *Advanced Methods for Computational Fluid Dynamics*, vol. 9, pp. 339–347, 2001.
- [110] J. Smagorinsky, “General circulation experiments with the primitive equations: I. the basic experiment*,” *Monthly weather review*, vol. 91, no. 3, pp. 99–164, 1963.
- [111] A. Khayyer and H. Gotoh, “A 3D higher order Laplacian model for enhancement and stabilization of pressure calculation in 3D MPS-based simulations,” *Applied Ocean Research*, vol. 37, pp. 120–126, 2012.
- [112] A. Khayyer, H. Gotoh, Y. Shimizu, and K. Gotoh, “Comparative study on accuracy and conservation properties of particle regularization schemes and proposal of an improved particle shifting scheme,” in *Proceedings of the 11th Int. SPHERIC workshop*, pp. 416–423, 2016.
- [113] D. J. Miller, *Giant waves in lituya bay, alaska*. US Government Printing Office Washington, DC, 1960.
- [114] L. Müller-Salzburg, *The rock slide in the Vajont Valley*. Springer-Verlag, 1964.
- [115] D. Tappin, P. Watts, G. McMurtry, Y. Lafoy, and T. Matsumoto, “The sisano, papua new guinea tsunami of july 1998—offshore evidence on the source mechanism,” *Marine Geology*, vol. 175, no. 1, pp. 1–23, 2001.

- [116] B. Ataie-Ashtiani and A. Nik-Khah, “Impulsive waves caused by subaerial landslides,” *Environmental Fluid Mechanics*, vol. 8, no. 3, pp. 263–280, 2008.
- [117] H. Fritz, W. Hager, and H.-E. Minor, “Landslide generated impulse waves. 2. hydrodynamic impact craters,” *Experiments in Fluids*, vol. 35, no. 6, pp. 520–532, 2003.
- [118] H. Fritz, W. Hager, and H.-E. Minor, “Landslide generated impulse waves. 1,” *Experiments in Fluids*, vol. 35, no. 6, pp. 505–519, 2003.
- [119] J. Kamphuis and R. Bowering, “Impulse waves generated by landslides,” in *Proc. 12th Coastal Engin. Conf. ASCE*, vol. 1, pp. 575–588, 1970.
- [120] F. Enet and S. T. Grilli, “Experimental study of tsunami generation by three-dimensional rigid underwater landslides,” *Journal of waterway, port, coastal, and ocean engineering*, vol. 133, no. 6, pp. 442–454, 2007.
- [121] S. Yavari-Ramshe and B. Ataie-Ashtiani, “Numerical modeling of subaerial and submarine landslide-generated tsunami waves-recent advances and future challenges,” *Landslides*, pp. 1–44, 2016.
- [122] P.-F. Liu, T.-R. Wu, F. Raichlen, C. Synolakis, and J. Borrero, “Runup and rundown generated by three-dimensional sliding masses,” *Journal of fluid Mechanics*, vol. 536, pp. 107–144, 2005.
- [123] A. Serrano-Pacheco, J. Murillo, and P. García-Navarro, “A finite volume method for the simulation of the waves generated by landslides,” *Journal of hydrology*, vol. 373, no. 3, pp. 273–289, 2009.
- [124] B. Ataie-Ashtiani and G. Shobeyri, “Numerical simulation of landslide impulsive waves by incompressible Smoothed Particle Hydrodynamics,” *International Journal for numerical methods in fluids*, vol. 56, no. 2, pp. 209–232, 2008.
- [125] T. Capone, A. Panizzo, and J. J. Monaghan, “SPH modelling of water waves generated by submarine landslides,” *Journal of Hydraulic Research*, vol. 48, no. S1, pp. 80–84, 2010.
- [126] S. Viroulet, D. Cébron, O. Kimmoun, and C. Kharif, “Shallow water waves generated by subaerial solid landslides,” *Geophysical Journal International*, vol. 193, no. 2, pp. 747–762, 2013.

- [127] C. Shi, Y. An, Q. Wu, Q. Liu, and Z. Cao, “Numerical simulation of landslide-generated waves using a soil–water coupling Smoothed Particle Hydrodynamics model,” *Advances in Water Resources*, vol. 92, pp. 130–141, 2016.
- [128] D. Yuk, S. Yim, and P.-F. Liu, “Numerical modeling of submarine mass-movement generated waves using RANS model,” *Computers & geosciences*, vol. 32, no. 7, pp. 927–935, 2006.
- [129] J. S. Russell, “On waves,” in *Report of fourteenth meeting of the British Association for the Advancement of Science, York*, pp. 311–390, 1844.
- [130] H. A. Barnes, J. F. Hutton, and K. Walters, *An introduction to rheology*, vol. 3. Elsevier, 1989.