

# Accelerating an SPH-FEM Solver using Heterogeneous Computing for use in Fluid-Structure Interaction Problems

John N. Gilbert

Dissertation submitted to the Faculty of the  
Virginia Polytechnic Institute and State University  
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

in

Aerospace Engineering

Leigh S. McCue-Weil, Chair

Owen F. Hughes

Mayuresh J. Patil

Heng Xiao

May 4, 2015

Blacksburg, Virginia

Keywords: Computational Fluid Dynamics, Smoothed Particle Hydrodynamics,  
Fluid-Structure Interaction

Copyright 2015, John N. Gilbert

# Accelerating an SPH-FEM Solver using Heterogeneous Computing for use in Fluid-Structure Interaction Problems

John N. Gilbert

## ABSTRACT

This work presents a partitioned approach to simulating free-surface flow interaction with hyper-elastic structures in which a smoothed particle hydrodynamics (SPH) solver is coupled with a finite-element (FEM) solver. SPH is a mesh-free, Lagrangian numerical technique frequently employed to study physical phenomena involving large deformations, such as fragmentation or breaking waves. As a mesh-free Lagrangian method, SPH makes an attractive alternative to traditional grid-based methods for modeling free-surface flows and/or problems with rapid deformations where frequent re-meshing and additional free-surface tracking algorithms are non-trivial. This work continues and extends the earlier coupled 2D SPH-FEM approach of Yang et al. [1, 2] by linking a double-precision GPU implementation of a 3D weakly compressible SPH formulation [3] with the open source finite element software Code\_Aster [4]. Using this approach, the fluid domain is evolved on the GPU, while the CPU updates the structural domain. Finally, the partitioned solutions are coupled using a traditional staggered algorithm.

*For my parents, Nick and Donna*

# Acknowledgments

First, I would like to express my sincere gratitude to my advisor, Professor Leigh McCue, for her guidance, encouragement, steadfast support, and above all patience, throughout this process. Thank you for the opportunity to study at Virginia Tech, for being an amazing mentor and friend, and for always pushing me out of my comfort zone. Next, I would like to thank my committee members, Professors Owen Hughes, Mayuresh Patil, and Heng Xiao for their advice and guidance over the course of this work. Professor Hughes, in particular, has served as a mentor and a friend since I first arrived in Blacksburg and my experience here at Virginia Tech has certainly been enriched because of it.

I would also like to acknowledge the efforts of Steve Edwards and Jonathan Spence for their tireless support of all things IT related. Also, a word of thanks must go out to the excellent staff of the Advanced Research and Computing center at Virginia Tech for providing access and technical support for the HokieSpeed cluster.

I also like to acknowledge the support of Kelly Cooper and the Office of Naval Research under grant number N00014-07-1-0833, and the related support of Dr. Patrick Purtell and Dr. Ki-Han Kim at the Office of Naval Research under grant number N00014-08-1-0695,

N00014-10-1-0398 for financially supporting this work.

Finally, to my family and friends. Your unwaivering love and support throughout this mad journey humbles me. Thank you, and know that I will continually strive to be worthy of such devotion.

# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Meshless Methods: An Overview . . . . .	3
1.2	SPH: Current State-of-the-Art . . . . .	6
1.2.1	SPH for Incompressible Flows . . . . .	7
1.2.2	Multiphase SPH . . . . .	9
1.2.3	Turbulence Models in SPH . . . . .	9
1.2.4	Imposing Boundary Conditions in SPH . . . . .	11
1.3	Coupled SPH-FEM Solvers . . . . .	12
1.4	Dissertation Structure . . . . .	14

<b>I Computational Approach</b>	<b>15</b>
<b>2 SPH for Dynamic Fluid Flows</b>	<b>16</b>
2.1 Kernel Approximation of a Function . . . . .	17
2.2 Kernel Approximation of Spatial Derivatives . . . . .	19
2.3 Particle Approximations of Functions and Derivatives . . . . .	20
2.4 The Kernel Function . . . . .	23
2.5 Governing Equations for Compressible Fluids . . . . .	25
2.6 Discrete Equations of Motion. . . . .	26
2.6.1 Conservation of Mass . . . . .	26
2.6.2 Conservation of Momentum . . . . .	28
2.6.3 Equations of State . . . . .	29
2.6.4 Boundary Conditions. . . . .	30
2.6.5 Numerical Stability and Time Integration. . . . .	33
2.6.6 Density Re-initialization and Numerical Dissipation . . . . .	34
2.6.7 GPU Implementation of SPH Solver . . . . .	36
<b>3 Fluid-Structure Coupling</b>	<b>39</b>

3.1	Fluid-Structure Coupling . . . . .	39
3.1.1	Boundary Sampling. . . . .	41
3.1.2	Interface Coupling. . . . .	41
<b>II</b>	<b>Verification and Validation</b>	<b>44</b>
<b>4</b>	<b>Code Verification and Validation</b>	<b>45</b>
4.1	Validation of SPH Model . . . . .	45
4.1.1	Couette Flow . . . . .	45
4.1.2	3D Dambreaking . . . . .	47
4.1.3	3D Sloshing Tank . . . . .	54
4.2	Validation of SPH-FEM Model . . . . .	62
<b>III</b>	<b>SPH-FEM Applications</b>	<b>70</b>
<b>5</b>	<b>Application of SPH-FEM model to SES Skirt Seal Dynamics</b>	<b>71</b>
5.1	Initial Simulation Configuration . . . . .	72
5.2	Modeling the SES Air Cushion . . . . .	73
5.3	SES Bow Seal Details . . . . .	76

5.4	Inlet-Outlet Boundary Conditions . . . . .	79
5.5	Simulation Results . . . . .	80
<b>6</b>	<b>Conclusions</b>	<b>89</b>
	<b>Bibliography</b>	<b>92</b>

# List of Figures

2.1	SPH particle approximation . . . . .	21
2.2	Solid boundary modeling in . . . . .	31
2.3	GPU implementation schematic. . . . .	38
3.1	SPH-FEM Coupling Scheme Representation . . . . .	40
3.2	Boundary Surface Sampling . . . . .	43
4.1	Couette flow geometry and results . . . . .	46
4.2	Initial set-up for 3D dambreaking . . . . .	50
4.3	Pressure time history for dambreaking: Probes P1-P4. . . . .	51
4.4	Pressure time history for dambreaking: Probes P5-P8. . . . .	52
4.5	3D dambreaking flow evolution. . . . .	53
4.6	Initial set-up for sloshing tank case. . . . .	55

4.7	Sloshing tank validation case: Comparision with experimental observations . . . . .	58
4.8	Sloshing tank validation case: Comparision with experimental observations . . . . .	59
4.9	Pressure time history for 3D sloshing tank case. . . . .	61
4.10	Elastic gate: Initial condition and results . . . . .	64
4.11	Initial particle setup for elastic membrane in sloshing tank . . . . .	66
4.12	Elastic membrane in sloshing tank: flow field samples . . . . .	68
4.13	Membrane tip displacement: comparison to experiments . . . . .	69
5.1	Photo and CAD rendering of experimental SES apparatus . . . . .	72
5.2	SES Simulation Configuration (1) . . . . .	73
5.3	SES Simulation Configuration (1) . . . . .	75
5.4	Sample of FEM mesh used to model SES bow seal . . . . .	78
5.5	Schematic showing inlet and outlet regions for SES simulations . . . . .	80
5.6	Time history of bow seal tip displacement . . . . .	85
5.7	Steady-state SES simulation results . . . . .	86
5.8	Close-up of bow seal position in Run 1013(b) . . . . .	87
5.9	Comparison of bow seal position with UM experiment. . . . .	88

# List of Tables

2.1	Relative computation time spent in each phase of SPH solver. . . . .	37
4.1	3D dambreaking validation parameters. . . . .	49
4.2	3D sloshing tank validation parameters. . . . .	57
4.3	2D elastic gate simulation parameters . . . . .	63
4.4	3D elastic membrane in sloshing tank parameters . . . . .	67
5.1	Dimensions of SES Simulation. . . . .	74
5.2	SES bow seal details. . . . .	77
5.3	SES simulation details. . . . .	81

# Chapter 1

## Introduction

In 2005, the Office of Naval Research (ONR) created the Innovative Naval Prototype (INP) Sea Base Connector Transformational Craft (T-Craft) program in response to repeated demands from the Navy and Marine Corps for a high-speed, high-capacity landing craft to compliment on-going and future sea-base assets [5]. ONR identified several performance goals for the program, including high-speed shallow-water transit in full load condition, good sea-keeping during cargo transfer at-sea, amphibious operation, and a sea state 6 operating capability [6, 7]. Three hybrid vessel prototypes, which blended together characteristics of surface effect ships (SES), catamarans, and air cushioned vehicles (ACV), were selected by ONR for further study and development. Under the proposed designs, the T-Craft was expected to operate in a high-speed SES mode for well over 95% of a typical mission scenario, then transition into a slower ACV mode for amphibious operations [8]. Recognizing that the T-Craft project would require advancements in technology and design, a portion of the

program funding was invested in a tool development program toward advanced phenomenological models and numerical algorithms to incorporate into future modeling and simulation tools[9]. One key technological gap the program was tasked with addressing was the lack of accurate tools for predicting the dynamics and estimating the loads seen by the SES/ACV skirt and its attachments when encountering steep or breaking waves during surf zone transit.

Early numerical work on ACVs focused on the resistance, sea-keeping performance, and dynamics and control of simplified geometries. Doctors (1975) [10] proposed a reduced order model which accounted for skirt drag to study the non-linear motions of an ACV in pitch and heave in regular head seas. Sullivan et al. (1994) [11] investigated the dynamics of the bag-and finger skirt system using a Lagrangian technique to derive the equations of motion of the skirt and vehicle in two-dimensions. Thiagarajan and Morris-Thomas (2006) [12] considered heave and pitch response of an air-cushioned open bottom box using linear wave theory. Additionally, the use of potential flow solvers coupled with air cushion models [13, 14] were quite useful for exploring large design spaces and obtaining response amplitude operators for sea-keeping analysis. However, accurate prediction of the behavior and resistance of the seals as they interact with the surrounding fluid requires the use more advanced CFD methods which have only become available within the last decade. Presently, a variety of advanced CFD approaches have been, or are currently being used, to tackle these problems, ranging from PFEM ([15]), to RANS ([16]), to SPH ([1, 2], [17],[18]).

Previously, Yang et al. [1, 2] demonstrated that a 2D SPH-FEM model is capable of sim-

ulating violent free-surface flow interacting with structural deformation in the time domain [1, 2]. In this thesis, the coupled SPH-FEM approach is accelerated via GPU implementation of the SPH fluid solver and extended to 3D to simulate a three-dimensional free-stream flow interacting with a deformable, pressurized surface such as an ACV/SES bow seal in order to characterize the deflection and internal stress of a skirt/seal system given a set of free-stream flow and cushion/seal parameters. However, while ACVs provided the initial motivation for this work, the tools being developed are intended for a broad class of flexible FSI problems, from sloshing to other flexible structure vessels potentially including the WAM-V (MAR).

## 1.1 Meshless Methods: An Overview

Continuum mechanics can be divided into two alternative descriptions: the Eulerian description and the Lagrangian description. The Eulerian description focuses on what happens at a fixed spatial location as time progresses, while the Lagrangian description follows the history of individual particles as they move in space and time. The Finite Volume (FVM) and Finite Difference (FDM) methods are two examples of numerical methods derived using the Eulerian perspective, which is far and away the most common description used in Computational Fluid Dynamics (CFD); however, in Eulerian formulations it is computationally expensive to accurately resolve free surfaces and track moving interfaces and boundaries. The Lagrangian description, on the other hand, is often used with problems involving moving boundaries or material interfaces because the time-history of the material is readily available. The Finite

Element Method (FEM) is one example of a numerical method derived from the Lagrangian perspective.

Numerical methods can be further divided into mesh or cell-based methods and meshfree or particle-based methods. Mesh-based methods, such as FVM or FEM, provide good predictive accuracy over a broad range of engineering problems and remain the dominant numerical tools for fluid and structural dynamics analysis. However, these traditional mesh-based approaches do have their limitations. Modeling phenomena that involve large relative motions, multiple interacting rigid structures, multiple phases and continua, complex free surface flows, large deformations, and fragmentation of solids can be a complex and computationally costly undertaking for traditional mesh-based approaches due to frequent re-meshing and the need for interface tracking algorithms. Meshfree numerical methods, on the other hand, do not rely on an underlying mesh or expressed nodal connectivity, which greatly reduces preprocessing time and eliminates the need for free-surface tracking, contact algorithms, or re-meshing, making meshfree methods an attractive alternative to mesh-based methods for these types of problems.

The fundamental concept behind meshfree methods is to obtain numerical solutions to integral equations or partial differential equations (PDEs) using a set of arbitrarily distributed nodes (or particles) without relying on any connectivity between these nodes or particles. The earliest particle models are the Marker And Cell (MAC) and Particle In Cell (PIC) methods developed by Harlow and Harlow and Welch [19, 20, 21]. These formulations were the first to incorporate a set of Lagrangian marker particles which convect with the fluid, but

still use an Eulerian mesh for the solution of the Navier-Stokes equations. Belytschko et al. [22] introduced the Element Free Galerkin (EFG) method, which uses a moving least squares interpolant to construct a set of basis functions to solve the variational form of the governing equation. The Finite Volume Particle Method (FVPM) was first introduced by Hietel et al. [23] and later extended by Keck and Hietel [24], Nestor et al. [25] and Teleaga [26]. In FVPMs, as with other particle-based methods, the fluid is represented by a set of particles. The particles are viewed as discrete volumes to which the integral form of the governing equations applies. The interaction between particles is calculated from the flux rate between neighboring particles. The Moving Particle Semi-implicit (MPS) method was developed by Koshizuka et al. [27] to simulate incompressible free-surface viscous flows particularly those with fluid fragmentation. In Particle Vortex Methods (see e.g. [28]), the Navier-Stokes equations are expressed in vorticity formulation and the vorticity field becomes the primary variable of interest for computation and eliminates the need to solve for pressure explicitly, as it is eliminated by the curl operator.

The Material Point Method (MPM), introduced by Sulusky et al. [29] is a particle method, but not a meshfree method in the strict sense of the word. MPM is an extension of the PIC method and requires material points (aka particles) as well as a background mesh and is primarily used in the study of solid mechanics. In MPM, material point information is mapped back to an Eulerian grid at the beginning each time step and used to compute velocity gradients and fluid accelerations. The velocity gradients and accelerations are then mapped back to the material points to update stress/strain, velocity, and position of

the material points. MPM is well suited to problems where stress is history dependent, as Lagrangian particles allow easy implementation of complex constitutive equations. MPM has an advantage over pure meshfree methods because the use of an Eulerian grid makes the calculation of gradients trivial. However, MPM uses both particles and a grid, which makes it more computationally expensive, in terms of memory storage. In addition, MPM suffers from interpolation noise when material points cross grid boundaries, also known as cell-crossing noise. This drawback was significantly reduced with the introduction of the Generalized Interpolation Material Point (GIMP) method, presented by Bardenhagen and Kober (2004) [30], which benefits from smoother interpolation functions.

Another truly meshfree method is the Meshless Local Petrov-Galerkin Method (MLPG) [31], originally developed for convection-diffusion problems. In the MLPG method, node trial and test functions can be from different function spaces, unlike conventional Galerkin methods, which make the MLPG method very flexible. Additional information on meshfree methods can be found in the review paper of Belytschko et al. [32] and the books of Liu and Liu [3] and Li and Liu [33]. This work focuses on the particle-based method known as Smoothed Particle Hydrodynamics (SPH), which is discussed in greater detail in the following section.

## 1.2 SPH: Current State-of-the-Art

Smoothed particle hydrodynamics is a true meshfree particle-based method originally developed in 1977 independently by Lucy [34] and Gingold and Monaghan [35] to study astro-

physical problems. Over the last several decades it has matured considerably and has since been extended to a variety of engineering problems including, but are not limited to, general free-surface flows [36], ship dynamics [37], landslide simulations [38], and flood simulations [39]. In-depth introductions into SPH can be found in the books by Liu and Liu [3] and Violeau [40].

### 1.2.1 SPH for Incompressible Flows

Monaghan (1994) [36] was the first to apply the SPH method to simulate incompressible free-surface fluid flows. In his work, the fluid was considered weakly compressible and fluid pressures were obtained from the density field using an appropriate equation of state. The majority of SPH formulations are explicit in time, therefore weakly compressible SPH (WC-SPH) simulations are usually performed at a reduced Mach number to limit the size of the required time step. The WCSPH method has been used to simulate a variety of free-surface fluid flows, such as run-up and run-down of waves on beaches (Monaghan and Kos, 1999) [41], breaking waves (Landrini et al., 2007 [42]) and wave breaking and post-breaking on beaches (Dalrymple and Rogers, 2006 [43]). Cummins and Rudman (1999) [44] were the first to propose strict Incompressible SPH (ISPH) for flows without a free-surface based on the projection method presented in [45], in which an intermediate velocity field is projected onto a divergence free space by solving a pressure Poisson equation. Shao and Lo (2003) [46] extended the ISPH method to allow for the simulation of free-surface flows and successfully simulated several free-surface fluid flow problems such as breaking waves (Shao et al., 2006

[47]) and wave-structure interaction (Gotoh et al., 2004 [48]; Shao and Gotoh, 2004 [49]).

When ISPH is used to model flows with a free-surface, a free-surface detection algorithm is required to correctly enforce the kinematic and dynamic boundary conditions. This enforcement represents an disadvantage of ISPH with respect to WCSPH, in particular for 3D free surface. Conversely, to satisfying the fluid incompressibility, an advantage of ISPH to WCSPH is its independent from the speed of sound; thus, according to the required stability conditions (such as Courant condition) a larger computational time step would be allowed in ISPH calculations. Vila (1999) [50] introduced an SPH model in an Arbitrary Lagrangian Eulerian (ALE) framework with Riemann solvers, rewriting the SPH formalism in a way inspired by the Finite Volumes Method. In SPH-ALE, interaction between two particles can be seen as the result of a flux acting at the middle of the distance between the two particles. A diffusive weakly-compressible SPH scheme was introduced by Antuono et al. (2010) [51] and subsequently called  $\delta$ -SPH. It is characterized by a novel system of equations which contains diffusive terms in both the continuity and energy equations and, at the leading order, coincides with a standard weakly-compressible SPH scheme with artificial viscosity. A proper state equation is used to associate the internal energy variation to the pressure field and to increase the speed of sound when strong deformations/compressions of the fluid occur.

### 1.2.2 Multiphase SPH

Monaghan and Kocharyan (1995) [52] presented a general SPH formulation for the simulation of multi-phase flows and successfully demonstrated the capability of SPH to simulate weakly compressible multiphase flows with small density ratios. However, flows with large density ratios, such as air-water, cause numerical instabilities to develop along the interface due to the discontinuous density field. Colagrossi and Landrini (2003) [53] presented modified gradient and divergence operators and incorporated a moving least-squares (MLS) filter to avoid the instabilities arising at the interface of two fluids. Hu and Adams (2006) [54] proposed a particle-averaged spatial derivative approximation in which neighboring particles only contribute to the specific volume, which is continuous across the interface. Hu and Adams (2009) [55] also introduced a constant-density approach for multiphase incompressible flows, which corrects intermediate density errors by adjusting the half-time-step velocity with exact projection. Grenier et al. (2009) [56] extended the works of Colagrossi and Landrini (2003) [53] and Hu and Adams (2006) [54] to include multi-fluid flows in the presence of a free-surface.

### 1.2.3 Turbulence Models in SPH

Although this work does not make use of a turbulence model, a brief summary of previous efforts to introduce and expand turbulence modeling in SPH is provided here for completeness.

Welton (1998) [57] applied the SPH method within a Monte-Carlo probability density formulation to simulate turbulent flows. Wagner and Liu (2000) [58] applied the Reproducing Kernel Particle Method (RKPM) as a filter in their Large Eddy Simulation (LES) of turbulent flows. By applying the same concept of Sub-Grid-Scale turbulence modeling in grid-based methods to particle methods, Gotoh et al. (2001) [59] developed a Sub-Particle-Scale turbulence model for particle-based simulations of turbulent flows. By employing the SPS turbulence model, Gotoh et al. (2004) [48] performed an ISPH-LES simulation to study wave transmission and reflection by a half-immersed curtain breakwater. In 2002, Monaghan [60] proposed a SPH version of the alpha turbulence model for compressible flow with a resolution that varies in space and time. The alpha model involves two velocity fields. One velocity field is obtained from the momentum equation, the other by averaging this velocity field as in the version of SPH called XSPH. Furthermore, Monaghan (2009) [61] presented a model similar to the Lagrangian Averaged Navier Stokes alpha (LANS-alpha) turbulence model that satisfied different scaling laws while conserving energy, angular and linear momentum and circulation, though the latter is only conserved approximately. Violeau and Issa (2007) [62] presented a review of developed turbulence models adapted to the SPH method, from the simplistic point of view of a one-equation model involving mixing length to more sophisticated (and thus realistic) models like Explicit Algebraic Reynolds Stress Models (EARSM) or Large Eddy Simulation (LES).

#### 1.2.4 Imposing Boundary Conditions in SPH

The imposition of boundary conditions, solid boundaries in particular, remains a challenge in SPH. Monaghan (1994) [36] proposed a Lennard-Jones type repulsive force to prevent fluid penetration of solid boundaries of arbitrary shape. This early approach is quite easy to implement, however it causes pressure-wave disturbances which severely limit the time step and result in poor predictions of the pressure field near solid boundaries. With the ghost particle method (Colagrossi and Landrini, 2003) [53], particles close to walls are mirrored about the solid boundary and field properties of the ghost particles are then computed from fluid particles inside the domain. This approach results in a more accurate pressure field near solid boundaries, but is challenging to implement when dealing with the complex geometries. Dalrymple and Knio (2001) [63] used several layers of fixed dummy particles placed in a staggered configuration to model solid walls. These dummy particles share some properties with fluid particles, but their positions are fixed relative to some reference configuration and their velocity is prescribed. This method, also known as the Dynamic Boundary Particle (DBP) approach, works well and is more computationally efficient than the ghost particle method when modeling rigid bodies, but does not guarantee that the fluid will not penetrate the boundary and is not readily applicable to modeling structures that undergo large deformations. Liu et al. [64] presented a hybrid method that incorporates a single layer of repulsive particles together with dynamic boundary particles. The repulsive particles are placed along solid boundaries and exert a “soft” repulsive force to prevent fluid penetration, while fixed boundary particles located outside solid boundaries provide

full particle support for fluid particles near walls. Ferrand et al. [65] presented a method based on a geometrical parameter measuring the missing area in the kernel support when a particle is in the vicinity of a solid boundary.

The enforcement of upstream/downstream boundary conditions in SPH is not trivial. De Leffe et al. (2009) [66] proposed a new boundary treatment for the SPH model in ALE formalism with Riemann solvers that can handle a variety of boundary conditions (inflow, outflow, non-reflecting, free slip, etc.). Lastiwka et al. (2008) [67] presented a boundary condition implementation that enables the simulation of flow through permeable boundaries. Each permeable boundary is associated with an inflow or outflow zone outside the domain, in which particles are created or removed as required. The analytic boundary condition is applied by prescribing the appropriate variables for particles in an inflow or outflow zone, and extrapolating other variables from within the domain.

### 1.3 Coupled SPH-FEM Solvers

A number of works have been published on coupled SPH-FEM solvers. The earliest attempts were by Attaway et al. [68], Johnson [69] and Johnson and Beissel [70]. These efforts all utilized master-slave type algorithms to account for the contact between SPH particles and FE elements. Attaway et al. [68] use the master-slave algorithm to couple an SPH solver to PRONTO, a hydrocode developed at Sandia National Laboratories. Johnson [69] and Johnson and Beissel [70] use a master-slave algorithm to enforce the conservation of momentum

across the fluid-structure interface and ensure the normal velocity components of the particle and the master element are equal. Sauer et al. [71] also proposed a master-slave algorithm based on a gap function which applies a contact force on a fluid particle before it penetrates the master segment. Vuyst et al. [72] coupled an SPH solver with LLNL-DYNA3D, a 3D FEM code developed at Lawrence Livermore National Laboratory. In their work, the contact force vector at the fluid structure interface is estimated by treating the FE nodes as SPH particles. Unlike the previous approaches, the coupling method proped by Vuyst et al. handles the interaction between the particle and element segment naturally without the need for an additional contact algorithm; however, it does require similar refinement of both the fluid and structural domains. Fourey et al. [73] coupled an SPH-ALE formulation to the open-source Finite Element software *Code\_Aster* [4] using the ghost particle approach to model solid boundaries. In this approach, the hydrodynamic loads on the structure are obtained by averaging the pressure of particles located in a narrow region near element faces. Yang et al. [1, 2] successfully simulated fluid flow interaction with a Surface Effect Ship (SES) bow seal in 2D using a coupled SPH-FEM solver. In those works, repulsive boundary particles co-located at finite element nodes along the fluid-structure interface are used to couple the fluid and structural domains. Like the approach of Vuyst et al. [72], the method of Yang et al. requires similar refinement in each domain. The fluid-structure coupling presented in this thesis is unique in that it uses a surface sampling algorithm to populate the structural surface with repulsive boundary particles, which are then used to transfer loads to the structure. This allows for different refinement levels to exist in the structure and fluid

and improve computational efficiency.

## 1.4 Dissertation Structure

The remainder of the dissertation is organized as follows. Chapter 2 expands on the theoretical and numerical aspects of the SPH interpolation technique, introduces the discretized fluid equations of motion, and presents several adaptations used to improve stability and accuracy of the model predictions. Chapter 3 highlights the key features necessary for efficient implementation of the SPH method on the GPU and describes the method used to couple the SPH solver with the FEM solver. Chapter 4 presents results of verification and validation study on the current FSI solver. Chapter 5 presents a comparison between the proposed FSI solver and experimental results of an SES skirt seal test run in a tow tank at the University of Michigan. Chapter 6 offers concluding remarks and suggests future work.

# **Part I**

## **Computational Approach**

# Chapter 2

## SPH for Dynamic Fluid Flows

Smoothed particle hydrodynamics (SPH) is a meshfree, Lagrangian particle method in which the state of a physical system is represented by a set of arbitrarily distributed particles. Each particle possesses its own individual physical properties and moves according to one or more governing equations. In this chapter, the foundation behind the SPH method is introduced, beginning with the continuous integral approximations for a function and its spatial derivatives, and followed by a derivation of the discrete particle approximations to these integral representations. Several kernel functions frequently encountered in the SPH literature are also presented.

Following the introduction of these essential formulations, the SPH method is used to discretize the governing equations describing general dynamic fluid flows. The Navier-Stokes equations in Lagrangian form are briefly described and SPH approximations are then used to derive the corresponding discrete equations of motion. Finally, several numerical aspects

of SPH, specifically the imposition of essential boundary conditions, weak compressibility, time integration and stability, and density re-initialization, are introduced and discussed.

## 2.1 Kernel Approximation of a Function

Per Liu and Liu (2003) [3], the classical SPH formulation is typically divided into two key steps: (1) the kernel approximation, and (2) the particle approximation. The concept behind the kernel approximation of a function  $f(\mathbf{x})$  used in SPH begins from the identity

$$f(\mathbf{x}) = \int_{\Omega} f(\mathbf{x}') \delta(\mathbf{x} - \mathbf{x}') d\mathbf{x}' \quad (2.1)$$

where  $f$  is a function of the position vector  $\mathbf{x}$ , and  $\delta(\mathbf{x} - \mathbf{x}')$  is the Dirac delta function, given by

$$\delta(\mathbf{x} - \mathbf{x}') = \begin{cases} +\infty & \mathbf{x} = \mathbf{x}' \\ 0 & \mathbf{x} \neq \mathbf{x}' \end{cases} \quad (2.2)$$

If the Dirac delta function is replaced by a kernel function  $W(\mathbf{x} - \mathbf{x}', h)$ , the kernel approximation of  $f(\mathbf{x})$  can be given by

$$\tilde{f}(\mathbf{x}) = \int_{\Omega} f(\mathbf{x}') W(\mathbf{x} - \mathbf{x}', h) d\mathbf{x}'. \quad (2.3)$$

where  $h$  is a parameter known as the *smoothing length* which determines the width of the kernel function. The kernel function is usually “bell-shaped” and is typically required to possess symmetry, positivity, have compact support, satisfy the unity condition, and mimic

the Dirac delta function as the smoothing length,  $h$ , approaches zero [3, 32]. That is

$$W(\mathbf{x} - \mathbf{x}', h) = W(\mathbf{x}' - \mathbf{x}, h), \quad (2.4)$$

$$W(\mathbf{x} - \mathbf{x}', h) \geq 0, \quad (2.5)$$

$$W(\mathbf{x} - \mathbf{x}', h) = 0, \quad \text{for } |\mathbf{x} - \mathbf{x}'| \geq kh, \quad (2.6)$$

$$\int_{\Omega} W(\mathbf{x} - \mathbf{x}', h) d\mathbf{x}' = 1, \quad (2.7)$$

$$\lim_{h \rightarrow 0} W(\mathbf{x} - \mathbf{x}', h) = \delta(\mathbf{x} - \mathbf{x}'). \quad (2.8)$$

where  $k$  is a constant related to the kernel function for a point at  $\mathbf{x}$ , and defines the effective area of the kernel function. This effective area,  $|\mathbf{x} - \mathbf{x}'| \geq kh$ , is referred to as the *support domain* for the kernel function at point  $\mathbf{x}$  in SPH literature. Due to the compact support of the kernel function, the integration in Eq. (2.3) only needs to be carried out over the support domain of the kernel function.

Assuming  $f(\mathbf{x})$  is sufficiently smooth and differentiable, the truncation error in the SPH kernel approximation can be estimated using the Taylor series expansion of  $f(\mathbf{x}')$  around  $\mathbf{x}$ . Substituting the expansion into Eq. (2.3), and noting from Eq. (2.6) that the support domain of the kernel function is  $|\mathbf{x} - \mathbf{x}'| \geq kh$ , Eq. (2.3) can be expressed as

$$\tilde{f}(\mathbf{x}) = \int_{\Omega} [f(\mathbf{x}) + f'(\mathbf{x}' - \mathbf{x}) + \mathcal{O}((\mathbf{x}' - \mathbf{x})^2)] W(\mathbf{x} - \mathbf{x}', h) d\mathbf{x}' \quad (2.9)$$

$$\begin{aligned} &= f(\mathbf{x}) \int_{\Omega} W(\mathbf{x} - \mathbf{x}', h) d\mathbf{x}' \\ &\quad + f' \int_{\Omega} (\mathbf{x}' - \mathbf{x}) W(\mathbf{x} - \mathbf{x}', h) d\mathbf{x}' + \mathcal{O}(h^2) \end{aligned} \quad (2.10)$$

Recall that  $W$  is an even function with respect to  $x$ , therefore  $(\mathbf{x}' - \mathbf{x})W(\mathbf{x} - \mathbf{x}', h)$  should be an odd function, thus

$$\int_{\Omega} (\mathbf{x}' - \mathbf{x})W(\mathbf{x} - \mathbf{x}', h) d\mathbf{x}' = 0. \quad (2.11)$$

Using Eqs. (2.7) and (2.11), Eq. (2.10) becomes

$$\tilde{f}(\mathbf{x}) = f(\mathbf{x}) + \mathcal{O}(h^2) \quad (2.12)$$

From Eq. (2.12), it can be seen that the SPH kernel approximation of a function is of second order accuracy; however, this approximation is not necessarily of second order if the kernel function is not even, or if the unity condition of Eq. (2.7) is not satisfied.

## 2.2 Kernel Approximation of Spatial Derivatives

Following Liu and Liu [3], the concept of a kernel approximation of a function can be extended to its spatial derivative by simply substituting  $\nabla \cdot f(\mathbf{x})$  in for  $f(\mathbf{x})$  in Eq. (2.3)

$$\nabla \cdot \tilde{f}(\mathbf{x}) = \int_{\Omega} \nabla_{x'} \cdot f(\mathbf{x}') W(\mathbf{x} - \mathbf{x}', h) d\mathbf{x}', \quad (2.13)$$

where the divergence in the integral is with respect to  $\mathbf{x}'$ . Using the chain rule,

$$\nabla \cdot f(\mathbf{x}') W(\mathbf{x} - \mathbf{x}', h) = \nabla \cdot [f(\mathbf{x}') W(\mathbf{x} - \mathbf{x}', h)] - f(\mathbf{x}') \cdot \nabla W(\mathbf{x} - \mathbf{x}', h), \quad (2.14)$$

Equation (2.13) can be expressed in the form

$$\nabla \cdot \tilde{f}(\mathbf{x}) = \int_{\Omega} \nabla_{x'} \cdot [f(\mathbf{x}') W(\mathbf{x} - \mathbf{x}', h)] d\mathbf{x}' - \int_{\Omega} f(\mathbf{x}') \cdot \nabla_{x'} W(\mathbf{x} - \mathbf{x}', h) d\mathbf{x}' \quad (2.15)$$

The first integral on the right hand side (RHS) of Eq. (2.15) can be converted into a surface integral using the divergence theorem, which gives

$$\nabla \cdot \tilde{f}(\mathbf{x}) = \int_S f(\mathbf{x}') W(\mathbf{x} - \mathbf{x}', h) \cdot \vec{\mathbf{n}} dS - \int_{\Omega} f(\mathbf{x}') \cdot \nabla_{x'} W(\mathbf{x} - \mathbf{x}', h) d\mathbf{x}' \quad (2.16)$$

where  $\vec{\mathbf{n}}$  is the outward unit normal to the bounding surface  $S$ . Since the kernel function  $W$  has compact support the first term on the RHS of Eq. (2.16) is zero, provided the support domain does not intersect the problem domain. The expression in Eq. (2.16) then simplifies to

$$\nabla \cdot \tilde{f}(\mathbf{x}) = \int_{\Omega} f(\mathbf{x}') \cdot \nabla_x W(\mathbf{x} - \mathbf{x}', h) d\mathbf{x}' \quad (2.17)$$

where  $\nabla_{x'} W = -\nabla_x W$  has been used.

Note from Eq. (2.17), that differentiation of the function is transferred to the smoothing function. If the support domain intersects with the problem domain (i.e. boundaries), then the surface integral in Eq. (2.16) is no longer zero and the accuracy of the approximation given in Eq. (2.17) will suffer. This *boundary deficiency* necessitates modifications to Eq. (2.17) to maintain accurate derivative approximations near boundaries.

## 2.3 Particle Approximations of Functions and Derivatives

In SPH formalism [3], the particle approximation begins by dividing the computational domain into particles of finite volume, as illustrated in Fig. 2.1. For a given particle,  $i$ ,

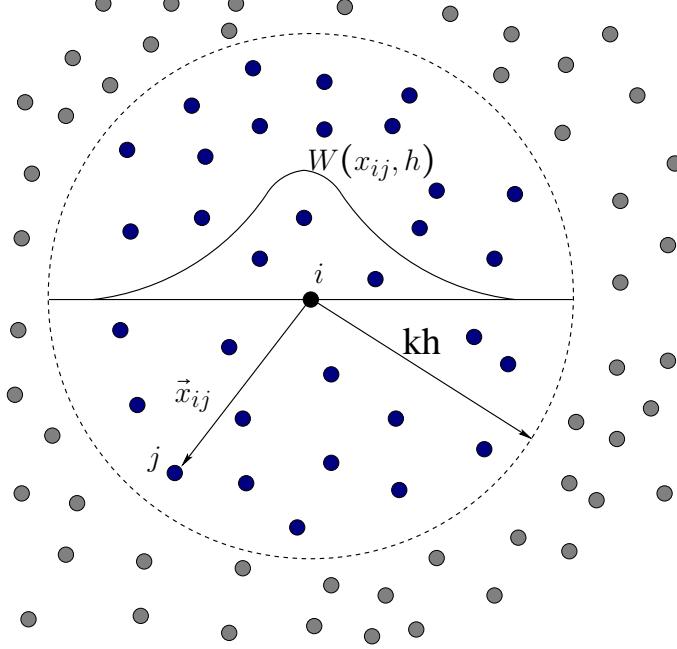


Figure 2.1: Only particles within the cut-off distance,  $|\vec{x}_{ij}| \leq kh$ , influence the state of particle  $i$ .

the integrals in Eqs. (2.3) and (2.17) can be replaced with summations over neighboring particles that lie within the compact support of the kernel function

$$\tilde{f}(\mathbf{x}_i) = \sum_{j=1}^N f(\mathbf{x}_j) W(\mathbf{x}_{ij}, h) \Delta V_j, \quad (2.18)$$

$$\nabla \cdot \tilde{f}(\mathbf{x}_i) = \sum_{j=1}^N f(\mathbf{x}_j) \nabla_i W(\mathbf{x}_{ij}, h) \Delta V_j, \quad (2.19)$$

where  $\mathbf{x}_{ij} = \mathbf{x}_i - \mathbf{x}_j$  has been used for compactness. Here,  $N$  is the total number of particles within the cut-off distance,  $|\mathbf{x}_{ij}| \leq kh$ ,  $\Delta V_j$  is the volume of particle  $j$ , and  $\nabla_i W$  is the gradient of the kernel function relative to particle  $i$ . The volume of a particle can be related to mass and density via

$$m_j = \Delta V_j \rho_j \quad (2.20)$$

where  $\rho_j$  is the density of particle  $j$ . Substituting this expression into Eqs. (2.18) and (2.19) yields

$$\tilde{f}(\mathbf{x}_i) = \sum_{j=1}^N \frac{m_j}{\rho_j} f(\mathbf{x}_j) W(\mathbf{x}_{ij}, h), \quad (2.21)$$

$$\nabla \cdot \tilde{f}(\mathbf{x}_i) = \sum_{j=1}^N \frac{m_j}{\rho_j} f(\mathbf{x}_j) \nabla_i W(\mathbf{x}_{ij}, h). \quad (2.22)$$

The particle approximations for a function and its spatial derivatives at a point  $\mathbf{x}_i$  can be seen to be a weighted average of the function values of all particles in the support domain of particle  $i$ .

In practice, however, the particle approximation given in Eq. (2.22) leads inaccurate and unstable hydrodynamic simulations because interactions between particles are not necessarily symmetric. Two alternative formulations for the spatial derivative of a function have been presented by Monaghan which provide more stable and accurate solutions [74]. Consider the following two identities:

$$\nabla \cdot f(\mathbf{x}) = \frac{1}{\rho} [\nabla \cdot (\rho f(\mathbf{x})) - f(\mathbf{x}) \cdot \nabla \rho] \quad (2.23)$$

$$\nabla \cdot f(\mathbf{x}) = \rho \left[ \nabla \cdot \left( \frac{f(\mathbf{x})}{\rho} \right) + \frac{f(\mathbf{x})}{\rho^2} \cdot \nabla \rho \right] \quad (2.24)$$

Substituting the identities above into Eq. (2.17), the same procedure to obtain the particle approximation in Eq. (2.22) is applied to each gradient term on the right hand side of Eqs. (2.23) and (2.24). This results in the following two particle approximations for the divergence of  $f(\mathbf{x})$

$$\nabla \cdot \tilde{f}(\mathbf{x}_i) = \frac{1}{\rho_i} \left\{ \sum_{j=1}^N m_j [f(\mathbf{x}_j) - f(\mathbf{x}_i)] \cdot \nabla_i W_{ij} \right\} \quad (2.25)$$

$$\nabla \cdot \tilde{f}(\mathbf{x}_i) = \rho_i \left\{ \sum_{j=1}^N m_j \left[ \frac{f(\mathbf{x}_j)}{\rho_j^2} + \frac{f(\mathbf{x}_i)}{\rho_i^2} \right] \cdot \nabla_i W_{ij} \right\} \quad (2.26)$$

where  $W_{ij} = W(\mathbf{x}_{ij}, h)$  has been used for brevity, and terms outside the gradient operator are evaluated at particle  $i$ . Note that particles appear in pairs in Eqs. (2.25) and (2.26), unlike in the original discretization. The anti-symmetric form of Eq. (2.25) ensures that the gradient of a constant field can be reproduced exactly, and is frequently used to discretize the divergence of velocity in the continuity equation. The symmetric form of Eq. (2.26) is a commonly used discretization of the pressure gradient in the momentum equation and ensures that the momentum of the system is conserved.

## 2.4 The Kernel Function

The performance of an SPH model is critically dependent on the choice of the kernel function. As mentioned in Section 2.1, the kernel function should satisfy several conditions, such as positivity, compact support, and unity. Also, the kernel function must be monotonically decreasing with increasing distance from particle  $i$ , and behave like the delta function as the smoothing length,  $h$ , tends to zero. In this section, four of the more commonly used kernel functions are presented.

Gingold and Monaghan [35], in their original paper, selected the following Gaussian kernel to simulate non-spherical stars,

$$W(R, h) = \alpha_D e^{-R^2} \quad (2.27)$$

where  $R = |\mathbf{x}_{ij}|/h$  is the normalized distance between two particles and  $\alpha_D$  is  $1/\pi^{1/2}h$ ,  $1/\pi h^2$ , and  $1/\pi^{3/2}h^3$  in one-, two-, and three-dimensional space, respectively, and is chosen such that

$W(R, h)$  satisfies the unity condition.

Johnson et al. [70] used the following quadratic kernel function to simulate high velocity impact problems

$$W(R, h) = \alpha_D \left[ \frac{3}{16} R^2 - \frac{3}{4} R + \frac{3}{4} \right] \quad 0 \leq R \leq 2 \quad (2.28)$$

where  $\alpha_D$  is  $1/h$ ,  $2/\pi h^2$ , and  $5/4\pi h^3$  in one-, two-, and three-dimensions, respectively.

Monaghan and Lattanzio [75] devised the following kernel function based on the cubic spline function known as the B-spline function.

$$W(R, h) = \alpha_D \times \begin{cases} \frac{2}{3} - R^2 + \frac{1}{2}R^3 & 0 \leq R < 1 \\ \frac{1}{6}(2-R)^3 & 1 \leq R < 2 \\ 0 & R \geq 2 \end{cases} \quad (2.29)$$

The B-spline function (often referred to as the cubic spline function [3, 76]) is one of the more widely used kernel functions in the SPH literature.

Another popular choice for the kernel function in SPH fluid simulations is the quintic Wendland kernel [77]. The quintic Wendland kernel is given by

$$W(R, h) = \alpha_D (1 + 2R) \left(1 - \frac{1}{2}R\right)^4 \quad 0 \leq R < 2 \quad (2.30)$$

The constant  $\alpha_D$  is  $7/4\pi h^2$  and  $21/16\pi h^3$  in two- and three-dimensional space, respectively. Morris et al. [78] and Monaghan and Kajtar [79] previously demonstrated the superior accuracy and stability of the Wendland kernel function relative to the B-spline function given in Eq. (2.29). For this reason, the Wendland kernel function is used for all simulations in this work.

## 2.5 Governing Equations for Compressible Fluids

The governing equations for compressible dynamic fluid flows are described by the continuity and Navier-Stokes equations, which, in Lagrangian formalism are given in indicial notation by

$$\frac{D\rho}{Dt} = -\rho \frac{\partial v^\beta}{\partial x^\beta}, \quad (2.31)$$

$$\frac{Dv^\alpha}{Dt} = \frac{1}{\rho} \frac{\partial \sigma^{\alpha\beta}}{\partial x^\beta} + f_b^\alpha, \quad (2.32)$$

These equations are coupled through an equation of state,  $p = p(\rho)$  [80]. In the above equations,  $f_b^\alpha$  represents external body forces and  $\sigma^{\alpha\beta}$  is the total stress tensor, given by

$$\sigma^{\alpha\beta} = -p \delta^{\alpha\beta} + \tau^{\alpha\beta}. \quad (2.33)$$

Here,  $p$  is the pressure and  $\tau$  is the viscous stress tensor. For Newtonian fluids, the viscous shear stress is proportional to the shear strain rate,  $\epsilon$ , through the dynamic viscosity  $\mu$ .

$$\tau^{\alpha\beta} = \mu \epsilon^{\alpha\beta} \quad (2.34)$$

where

$$\epsilon^{\alpha\beta} = \frac{\partial v^\beta}{\partial x^\alpha} + \frac{\partial v^\alpha}{\partial x^\beta} - \frac{2}{3} (\nabla \cdot \mathbf{v}) \delta^{\alpha\beta} \quad (2.35)$$

If the fluid is incompressible ( $\nabla \cdot \mathbf{v} = 0$ ) then the governing equations reduce to

$$\frac{\partial v^\beta}{\partial x^\beta} = 0, \quad (2.36)$$

$$\frac{Dv^\alpha}{Dt} = -\frac{1}{\rho} \frac{\partial p \delta^{\alpha\beta}}{\partial x^\beta} + \nu \frac{\partial^2 v^\alpha}{\partial x^\beta \partial x^\beta} + f_b^\alpha, \quad (2.37)$$

Finally, if viscous effects are negligible, the governing equations simplify to the well-known Euler equations

$$\frac{\partial v^\beta}{\partial x^\beta} = 0, \quad (2.38)$$

$$\frac{Dv^\alpha}{Dt} = -\frac{1}{\rho} \frac{\partial p \delta^{\alpha\beta}}{\partial x^\beta} + f_b^\alpha, \quad (2.39)$$

## 2.6 Discrete Equations of Motion.

### 2.6.1 Conservation of Mass

Several expressions can be used to model the continuity equation. Two of the most commonly encountered in SPH literature are the *summation density* and *continuity density* approximations, given by

$$\rho_i = \sum_j m_j W_{ij}, \quad (2.40)$$

$$\frac{d\rho_i}{dt} = \sum_j m_j \mathbf{v}_{ij} \cdot \nabla W_{ij}, \quad (2.41)$$

respectively. Here, the subscript  $i$  refers to the particle of interest,  $j \in N$  are its neighboring particles,  $m$  is the particle mass,  $\mathbf{v}_{ij} = \mathbf{v}_i - \mathbf{v}_j$ , and  $\nabla W_{ij}$  is the gradient of the kernel function. For flows involving more than one fluid, Monaghan and Rafiee (2012) [81] suggests the following discretization

$$\frac{d\rho_i}{dt} = -\rho_i \sum_j \frac{m_j}{\rho_j} \mathbf{v}_{ij} \cdot \nabla_i W_{ij} \quad (2.42)$$

because the  $m_j/\rho_j$  term is invariant to the changing mass of neighboring particles, thus the SPH estimate of  $\nabla \cdot \mathbf{v}$  will depend only on the local velocity field.

The advantage of the summation density approach is that it preserves mass exactly, however, Eq. (2.40) suffers from edge effects which smooth out the density field near boundaries and material interfaces. The continuity density approximations given in Eqs. (2.41) and (2.42) yield better estimates when features such as free-surfaces are present, but they do not preserve mass exactly (see [35, 78]).

Alternatively, Vila (1999) [50] has shown that the conservation of mass is exactly equivalent to

$$\frac{d\rho_i}{dt} = \frac{d}{dt} \left( \sum_j m_j W_{ij} \right) \quad (2.43)$$

if time is considered a continuous variable. Integrating Eq. (2.43) gives

$$\rho_i^{n+1} - \rho_i^n = \sum_j m_j (W_{ij}^{n+1} - W_{ij}^n). \quad (2.44)$$

Following Vila [50] and Ferrand et al. [65], the density field can be updated in a completely position dependent way according to

$$\rho_i^{n+1} = \rho_i^n + \sum_j m_j (W_{ij}^{n+1} - W_{ij}^n). \quad (2.45)$$

Equation (2.45) significantly reduces the edge effects present in the traditional summation density approach and prevents systematic errors time integration errors, which can become significant for duration simulations. For this reason, Eq. (2.43) is used to update particle density in this work.

### 2.6.2 Conservation of Momentum

Like the continuity equation, several discrete forms of the momentum are possible depending on how the pressure gradient and viscous stress tensor are discretized. The two most common discretizations for the pressure gradient are given by

$$\nabla p_i = \sum_j m_j \left( \frac{p_i}{\rho_i^2} + \frac{p_j}{\rho_j^2} \right) \nabla W_{ij} \quad (2.46)$$

$$\nabla p_i = \sum_j m_j \left( \frac{P_i + P_j}{\rho_i \rho_j} \right) \nabla W_{ij} \quad (2.47)$$

where  $p_i$  is the pressure of particle  $i$ . Equation (2.46) is perhaps the most frequently used approximation for the pressure gradient [3] for single phase flow. The advantage is that using a symmetric form leads to local momentum conservation [82]. Equation (2.47) is best suited to multiphase flows because the  $\rho_i \rho_j$  term in the denominator is continuous across material interfaces [52]. Equation (2.46) is used for all simulations in this study.

There are also different forms to approximate the viscous term,  $\nabla^2 \mathbf{v}$ , in the momentum equation. It is possible to use one of the discretization schemes presented in Section 2.3 for the second derivative. This approach leads to a doubled summation form and has been used to include physical viscosity in astrophysical problems [83] and to study low-Reynolds number incompressible flows [84]. However, this approach requires multiple summations, which adds additional computational expense, and has been shown to lead to oscillatory solutions in the presence of discontinuous initial conditions [85].

Alternatively, Monaghan (1994) [36] proposed an artificial viscosity model, which has been used to applied to several astrophysical and gas dynamics problems [86], but for problems

involving fluid-structure interaction a physical viscosity model is preferred. In this work, the physical viscosity model of Cleary (1998) [87] is used to approximate the viscous diffusion and is given by

$$\Pi_{ij} = -\frac{\xi}{\rho_i \rho_j} \frac{4\mu_i \mu_j}{\mu_i + \mu_j} \frac{\mathbf{v}_{ij} \cdot \mathbf{x}_{ij}}{|\mathbf{x}_{ij}|^2} \quad (2.48)$$

Here,  $\mu_i$  is the dynamic viscosity of particle  $i$  and  $\xi$  is a tuning parameter chosen to be  $\xi = 4.9633$  through numerical calibration [87]. Note that the value of  $\xi$  is independent of fluid viscosity. The Cleary viscosity model was chosen for this work because it preserves angular momentum. This viscosity model is validated in Chapter 4 through a Couette flow simulation.

To summarize, the SPH approximation of the momentum equation used in this study is given by

$$\frac{d\mathbf{v}_i}{dt} = -\sum_j m_j \left( \frac{p_i}{\rho_i^2} + \frac{p_j}{\rho_j^2} + \Pi_{ij} \right) \nabla W_{ij} \quad (2.49)$$

### 2.6.3 Equations of State

Modeling incompressible flows with the SPH method is typically done in one of two ways. Incompressible SPH (ISPH) [44, 88] involves solving pressure Poisson equation and comparatively computationally expensive. A more common approach in SPH [3], and the approach used in this work, is to consider an inviscid, weakly-compressible fluid. Since the fluid is assumed compressible, an equation of state is needed in order to close the system of equations.

Here, the Tait equation of state [3] is used

$$P = \frac{\rho_o c_o^2}{\gamma} \left[ \left( \frac{\rho}{\rho_o} \right)^\gamma - 1 \right], \quad (2.50)$$

in which  $\gamma$  is the ratio of specific heats, and  $\rho_o$  and  $c_o$  denote the reference density and sound speed of the fluid, respectively. In this work, the working fluid is assumed to be water ( $\gamma = 7.15$  and  $\rho_o = 1000 \text{ kg/m}^3$ ), or oil ( $\gamma = 7.0$  and  $\rho_o = 915 \text{ kg/m}^3$ ). For quasi-incompressible flows, like those under consideration, an artificial speed of sound is chosen to allow a reasonable time step while limiting the density variations to within 1%. Following [78], the square of the speed of sound is chosen by considering a balance of pressure, viscous, and body forces, and is given by

$$c_o^2 = \max \left( \frac{V_b^2}{\delta}, \frac{\nu V_b}{\delta L}, \frac{FL}{\delta} \right) \quad (2.51)$$

where  $\delta$  is the magnitude of the density variations (in this work  $\delta = 0.01$ ),  $V_b$  is the bulk fluid velocity,  $\nu$  is the kinematic viscosity,  $F$  is the magnitude of the body force, and  $L$  is the characteristic length scale of the flow.

#### 2.6.4 Boundary Conditions.

Accurately enforcing essential boundary conditions remains a challenging problem within the SPH community. In this work, the Dynamic Boundary Condition (DBC) [89] is used to model static solid walls, and a hybrid approach which combines the DBC and “soft” repulsive force [64] is used to model deformable boundaries and couple fluid and structural domains during fluid-structure interaction simulations.

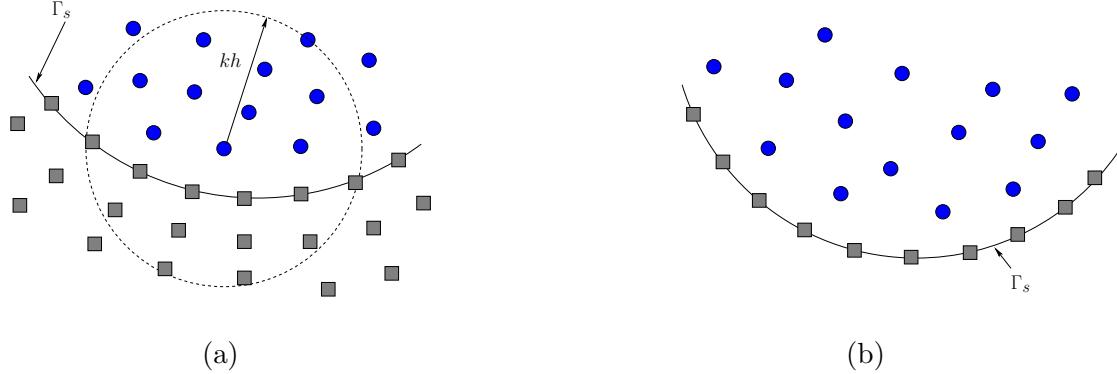


Figure 2.2: (a) A pure DBC is used to model static boundaries. (b) “Soft” repulsive particles placed along deformable boundaries are used to transfer fluid loads to structural nodes. In the figures above, fluid particles are denoted by ●, and boundary particles are denoted by ■.

### Dynamic Boundary Condition

With the DBC method, solid boundaries are constructed from two or more layers of boundary particles, as shown in Fig. 2.2(a). These boundary particles are made to satisfy Eq. (2.41) like a fluid particle, but remain static or move according to some predefined motion. When a fluid particle approaches a boundary, the density of the neighboring boundary particles increases, causing a pressure increase, which results in a repulsive force being exerted on the fluid particle.

### Repulsive Particles

As was mentioned in Chapter 1, Monaghan (1994) [36] first introduced repulsive particles as a way to model solid boundaries in SPH. These particles were placed in a single layer

along a solid boundary, as shown in Fig. 2.2(b), and exerted a Lennard-Jones type force on approaching fluid particles. Though they are effective at preventing fluid penetration, Monaghan's initial boundary condition and subsequent variations [41, 79] did nothing to correct for the lack of particle support near boundaries and can lead to a noisy pressure field near solid boundaries.

Liu et al. (2012) [64] presented a coupled dynamic Solid Boundary Treatment (SBT) algorithm that combined the dynamic boundary condition approach [89] with a “soft” repulsive particle layer. In this approach, ghost particles are located outside solid boundaries and provide interior fluid particles with a full support domain. In addition, a layer of particles placed along the solid boundary exerts a “soft” repulsive force on nearby fluid particles to ensure the fluid does not penetrate the wall. This modified repulsive force can prevent fluid penetration without the obvious pressure disturbances present in the many of the repulsive particle approaches. The repulsive force is distance-dependent and is given by

$$\mathbf{F}_{ij} = 0.01c^2 \cdot \chi \cdot f(\eta) \cdot \frac{\mathbf{x}_{ij}}{|\mathbf{x}_{ij}|^2}, \quad (2.52)$$

$$\chi = \begin{cases} 1 - |\mathbf{x}_{ij}|/1.5\Delta d, & 0 < |\mathbf{x}_{ij}| < 1.5\Delta d, \\ 0, & \text{otherwise,} \end{cases} \quad (2.53)$$

$$\eta = |\mathbf{x}_{ij}| / (0.75h_{ij}), \quad (2.54)$$

$$f(\eta) = \begin{cases} 2/3, & 0 < \eta \leq 2/3, \\ (2\eta - 1.5\eta^2), & 2/3 < \eta \leq 1, \\ \frac{1}{2}(2 - \eta)^2, & 1 < \eta < 2, \\ 0, & \text{otherwise,} \end{cases} \quad (2.55)$$

where  $\Delta d$  is the initial particle spacing,  $h_{ij} = 0.5(h_i + h_j)$  is the average smoothing length of the interacting particles, and  $c$  is the sound speed. In the coupled dynamic SBT algorithm, the field variables of the ghost particles and repulsive particles are dynamically evolved from SPH approximations, similar to the DBC method.

In this thesis, when modeling deformable boundaries, only the repulsive particle layer is used. This approach avoids the complexity and computational cost associated with regenerating ghost particle layers inside a deforming body after each time step and still benefits from the modified repulsive force while preserving momentum conservation across the fluid-structure interface.

### 2.6.5 Numerical Stability and Time Integration.

An explicit Predictor-Corrector scheme, which is second order accurate in time, is employed for temporal integration of the SPH solver. The time step is limited by a combination of the Courant-Friedrichs-Lowy (CFL) condition, viscous diffusion, and a boundary force condition. A variable time step  $\Delta t$  is calculated according to [79] using

$$\Delta t = 0.3 \cdot \min(\Delta t_{bf}, \Delta t_{cv}), \quad (2.56)$$

where

$$\Delta t_{cv} = \min_{\Omega_f} \left( \frac{h}{c + \max \left| \frac{h\vec{v}_{ij} \cdot \vec{x}_{ij}}{(x_{ij}^2 + \eta^2)} \right|} \right). \quad (2.57)$$

and

$$\Delta t_{bf} = \min_{\Omega_f} \left( \frac{|x_{ik} - d|}{\sqrt{K}} \right), \quad (2.58)$$

### 2.6.6 Density Re-initialization and Numerical Dissipation

By using the continuity density approach, the consistency between mass, density, and volume cannot be enforced exactly, as is possible with the summation density approach. This leads to an accumulation of systematic time integration errors which can result in numerical instability. To restore this consistency the density field must be periodically re-initialized by applying either a Sheppard or moving-least-squares (MLS) filter.

Additionally, while the dynamics from SPH simulations are generally realistic, the pressure field of the particles exhibits large pressure oscillations. Efforts to overcome this problem have concentrated on several approaches including correcting the kernel (for an overview see Bonet & Lok, 1999) and developing an incompressible solver. One of the most straight forward and computationally least expensive is to perform a filter over the density of the particles and the re-assign a density to each particle (Colagrossi and Landrini, 2003). There are two orders of correction, zeroth order and first order.

The Sheppard filter is a simple correction to the density field, given by

$$\tilde{\rho}_i = \sum_j m_j W_{ij}^{SH} \quad (2.59)$$

$$W_{ij}^{SH} = \frac{W_{ij}}{\sum_j \frac{m_j}{\rho_j} W_{ij}} \quad (2.60)$$

where the kernel has been corrected using a zeroth-order correction.

The MLS approach was developed by Dilts (1999) and applied by Colagrossi and Landrini (2003) and by Panizzo (2004). The MLS filter is given by

$$\tilde{\rho}_i = \sum_j m_j W_{ij}^{MLS} \quad (2.61)$$

where the corrected kernel is evaluated by

$$W_{ij}^{MLS} = \boldsymbol{\beta}_i \cdot \mathbf{x}_{ij} W_{ij} \quad (2.62)$$

The correction vector,  $\boldsymbol{\beta}$  is given by

$$\boldsymbol{\beta}_i = \mathbb{A}_i^{-1} \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \text{ where } \mathbb{A} = \sum_j W_{ij} \tilde{\mathbb{A}}_{ij} V_j \quad (2.63)$$

The matrix  $\tilde{\mathbb{A}}$  is defined as

$$\tilde{\mathbb{A}}_{ij} = \begin{bmatrix} 1 & (x_i - x_j) & (y_i - y_j) & (z_i - z_j) \\ (x_i - x_j) & (x_i - x_j)^2 & (x_i - x_j)(y_i - y_j) & (x_i - x_j)(z_i - z_j) \\ (y_i - y_j) & (x_i - x_j)(y_i - y_j) & (y_i - y_j)^2 & (y_i - y_j)(z_i - z_j) \\ (z_i - z_j) & (x_i - x_j)(z_i - z_j) & (y_i - y_j)(z_i - z_j) & (z_i - z_j)^2 \end{bmatrix} \quad (2.64)$$

This is a first-order correction so that linear variation of the density field can be exactly reproduced, however it is more computationally expensive than the Shepard filter due to the inversion of  $\mathbb{A}$  for each particle. Both approaches restore the consistency between particle mass, density, and volume, and have been shown to be effective at suppressing numerical instabilities and result in smoother pressure fields.

### 2.6.7 GPU Implementation of SPH Solver

In recent years, several SPH solvers have been implemented on GPUs [76, 90, 91, 92, 93].

Our own implementation extends these works by interfacing with a separate Finite-Element solver for fluid-structure interaction problems, allowing for independent resolution of the fluid and structural domains, and providing support for both single- and double-precision floating-point operations. The optimization strategies described in [94] and [95] have been used in this work, the key features of which will be summarized here for completeness.

In this implementation, as in many other GPU implementations, evolution of the SPH equations of motion in time is executed in three phases: i) cell-linked list creation , ii) accumulation of particle interactions in continuity and momentum equations, and iii) time integration of the equations of motion, the detail of which are summarized in Fig. 2.3. Particle interactions are restricted to neighbors located a distance  $kh$  away and the cell-linked list method [96], in which the fluid domain is divided into a rectilinear grid with cells of size ( $kh \times kh \times kh$ ), is used to improve the efficiency of the neighbor search. Particles are sorted according to the cell they are located in using the *radixsort* algorithm [95] to improve performance and simplify the cell-linked list data structure. During phase (ii), each particle searches adjacent cells for neighbors, and equations for conservation of mass and momentum are computed for each particle-particle interaction.

To avoid memory read-write conflicts, execution on the GPU does not take advantage of pair-wise symmetry when computing interactions, unlike CPU implementations. Additionally, interactions between each type of particle (i.e. Fluid-Boundary, Boundary-Fluid) are

carried out separately to minimize thread divergence [76, 93].

Table 2.1 lists the percentage of computational time spent on each phase described above. In 3D, each particle interacts with 180 neighbors or more, depending on its smoothing length. Thus, the computational stencil (number of interactions) in SPH much larger than mesh-based methods. The current GPU implementation spends approximately 99% of the execution time computing the particle-particle interactions, therefore future optimizations should be focused on this phase of execution.

Table 2.1: Relative computational time spent in each phase of SPH solver based on the 3D Dambreak case presented in Section 4.1.2.

Phase	% of time spent
CLL creation	0.1
Particle Interactions	98.7
Time Integration	0.2
Other (I/O)	1

During FSI simulations, boundary information is exchanged between the SPH and structural solvers. Once the structural solver updates the structure, interface position and velocity need to be transferred back to the fluid solver to update the boundary particle states. This involves memory transfer between the GPU and the CPU, but the size of the data transferred is small compared to the full fluid domain size, so the computational cost per time step is minimal.

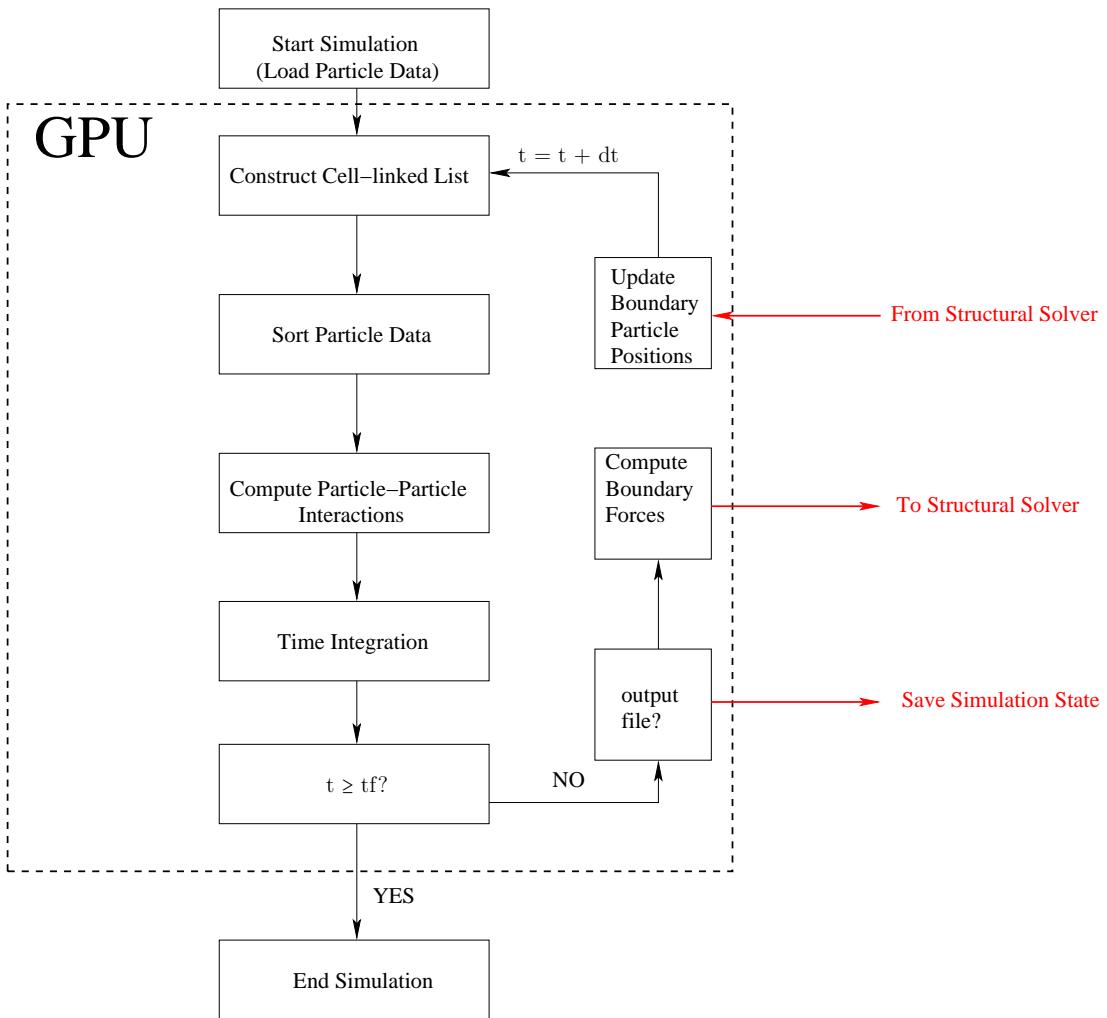


Figure 2.3: GPU implementation schematic.

# **Chapter 3**

## **Fluid-Structure Coupling**

### **3.1 Fluid-Structure Coupling**

Yang et al. [1, 2] successfully demonstrated an SPH-FEM coupling in two-dimensions via repulsive boundary particles which were used to transfer interface loads and displacements between the fluid and the structural domains. The method is particularly well suited to problems with complex structural geometries because only a single layer of boundary particles is required to model the interface. One drawback, however, was that the method required the repulsive particles to be co-located with the finite-element nodes located along the fluid-structure interface, as illustrated in Fig. 3.1. This forces the fluid and structural domains to have roughly equal resolution. The approach introduced in this thesis extends the fluid-structure coupling method of Yang et al. [1] to three-dimensions and allow independent

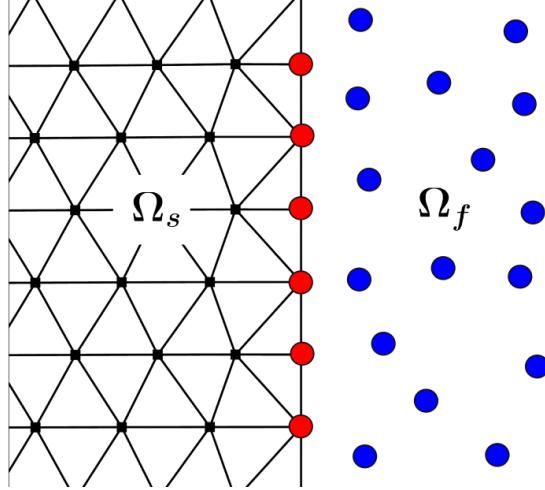


Figure 3.1: Previous SPH-FEM coupling scheme with boundary particles co-located with interface nodes. Boundary particles are denoted with ●, Fluid particles are denoted with ●, and FEM nodes are denoted with ■

refinement of the fluid and structure through the use of boundary sampling. The open-source solver *Code\_Aster*, developed by Electricité De France (EDF) is used as the structural solver and the coupling algorithm used here is a traditional staggered algorithm [97].

Previously, Fourey et al. (2010) [73] introduced a coupled SPH-FEM solver which also made use of the FEM solver *Code\_Aster*. The approach presented in this thesis differs from that of Fourey et al. in two key ways: (1) Fluid-structure interaction in this work is accomplished via a single layer of repulsive boundary particles. The coupled solver presented by Fourey et al. used the ghost particle approach [53] to represent structural boundaries, and fluid pressures along the fluid-structure interface were evaluated using a spatial averaging technique and applied to interface element faces. As was mentioned in Chapter 1, the

ghost particle approach does well to eliminate boundary deficiencies, but is challenging to implement for complex geometries. (2) The SPH solver in this thesis has been implemented on the GPU to maximize computational resources.

### **3.1.1 Boundary Sampling.**

Several algorithms exist for sampling a surface with a set of particles. Here, the sampling algorithm introduced by Akinci et al. [98] is used for generating a set of uniformly distributed particles from an underlying triangular surface mesh. The algorithm generates boundary particles in three steps: i) particles are first placed at each vertex, ii) particles are placed along the edges of each triangle, and iii) particles are generated in the interior of each triangle. The results from this algorithm are shown in Figs. 3.2(a)-3.2(c). Once created, these boundary particles are used during the simulation to apply boundary particle forces to fluid particles to simulate a solid wall boundary condition.

### **3.1.2 Interface Coupling.**

To couple the fluid and structural interfaces, each of the newly created boundary particles is first assigned a “master” element in which they are located. The barycentric coordinates of a boundary particle,  $P$ , relative to the positions of the corresponding “master” element

vertices is given by

$$N_1 = \frac{A_{P23}}{A}, \quad (3.1)$$

$$N_2 = \frac{A_{P13}}{A}, \quad (3.2)$$

$$N_3 = \frac{A_{P12}}{A}, \quad (3.3)$$

where  $A = A_{P12} + A_{P23} + A_{P13}$  is the total area of the triangular surface element. The barycentric coordinates are then used to transfer boundary forces from particle  $P$  to the structure according to

$$\vec{f}_i = N_i \vec{f}_P \quad (i = 1, 2, 3), \quad (3.4)$$

and update boundary particle positions at the end of each time step by

$$\vec{x}_P = \sum_{i=1}^3 N_i \vec{x}_i \quad (3.5)$$

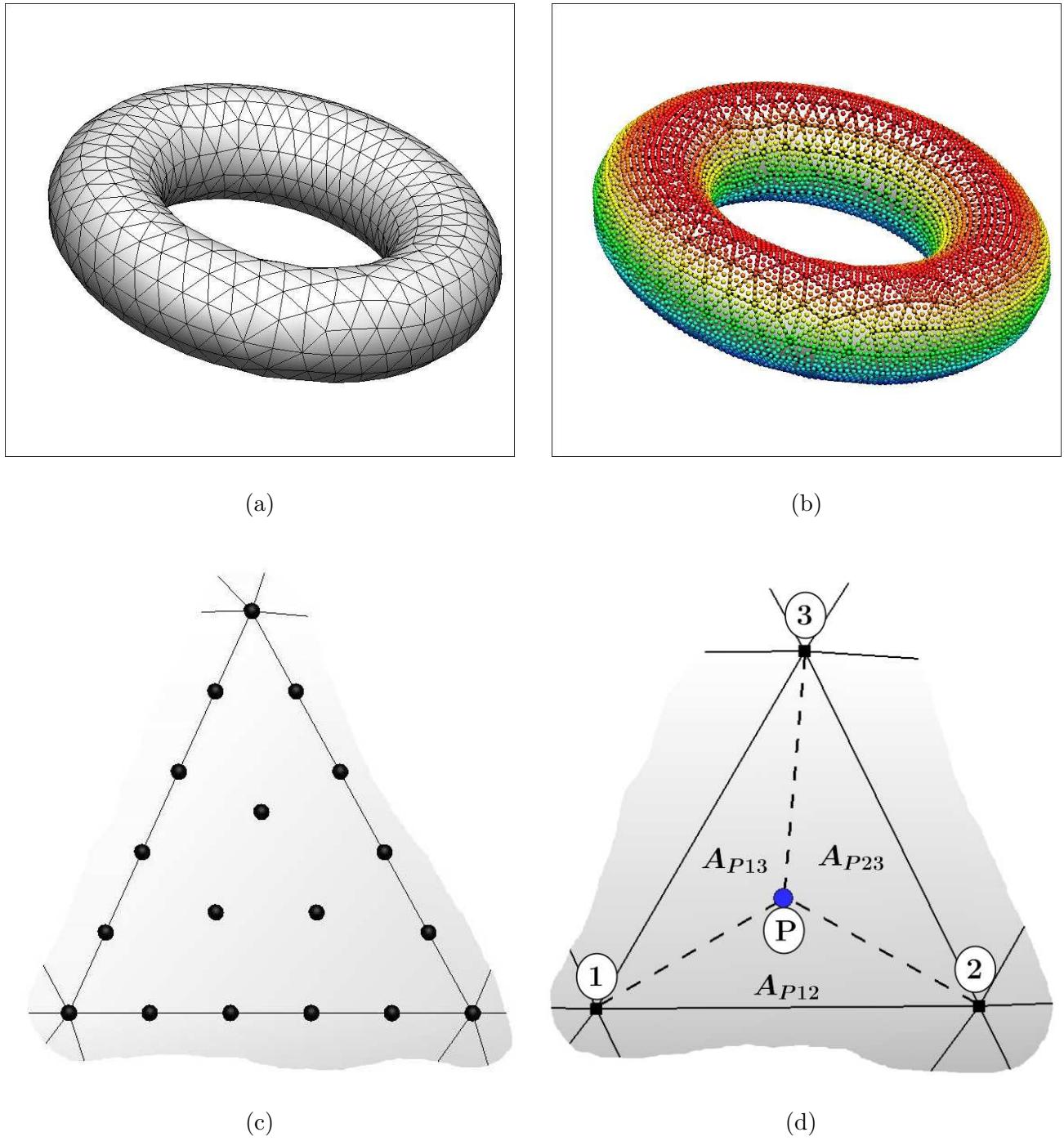


Figure 3.2: (a) Initial triangular surface mesh to be sampled. (b) Boundary particle distribution resulting from boundary sampling algorithm. (c) Close-up view of a single element shows uniform distribution of particles across the element. (d) Barycentric coordinates of an interior particle found from a ratio of triangular areas.

## **Part II**

# **Verification and Validation**

# Chapter 4

## Code Verification and Validation

### 4.1 Validation of SPH Model

#### 4.1.1 Couette Flow

Couette flow involves flow between two initially stationary infinite parallel plates. The flow is generated after the upper plate is impulsively moved at a constant velocity,  $v_o$ . Batchelor (1967) [99] presented an infinite series solution for the time history of the Couette flow given by

$$v_x(z, t) = \frac{v_o}{l}z + \sum_{n=1}^{\infty} \frac{2v_o}{n\pi}(-1)^n \sin\left(\frac{n\pi}{l}z\right) \exp\left(-\nu \frac{n^2\pi^2}{l^2}t\right) \quad (4.1)$$

In this work, the Couette case is simulated to validate Cleary's viscosity model [87] for  $\nu = 10^{-6} \text{ m}^2\text{s}^{-1}$ ,  $l = 10^{-3}\text{m}$ ,  $\rho = 10^3\text{kgm}^{-3}$ , and  $v_o = 1.25 \times 10^{-5}\text{ms}^{-1}$ . The problem geometry is shown in Fig. 4.1(a) and is modeled with three rows of boundary particles [89] along the

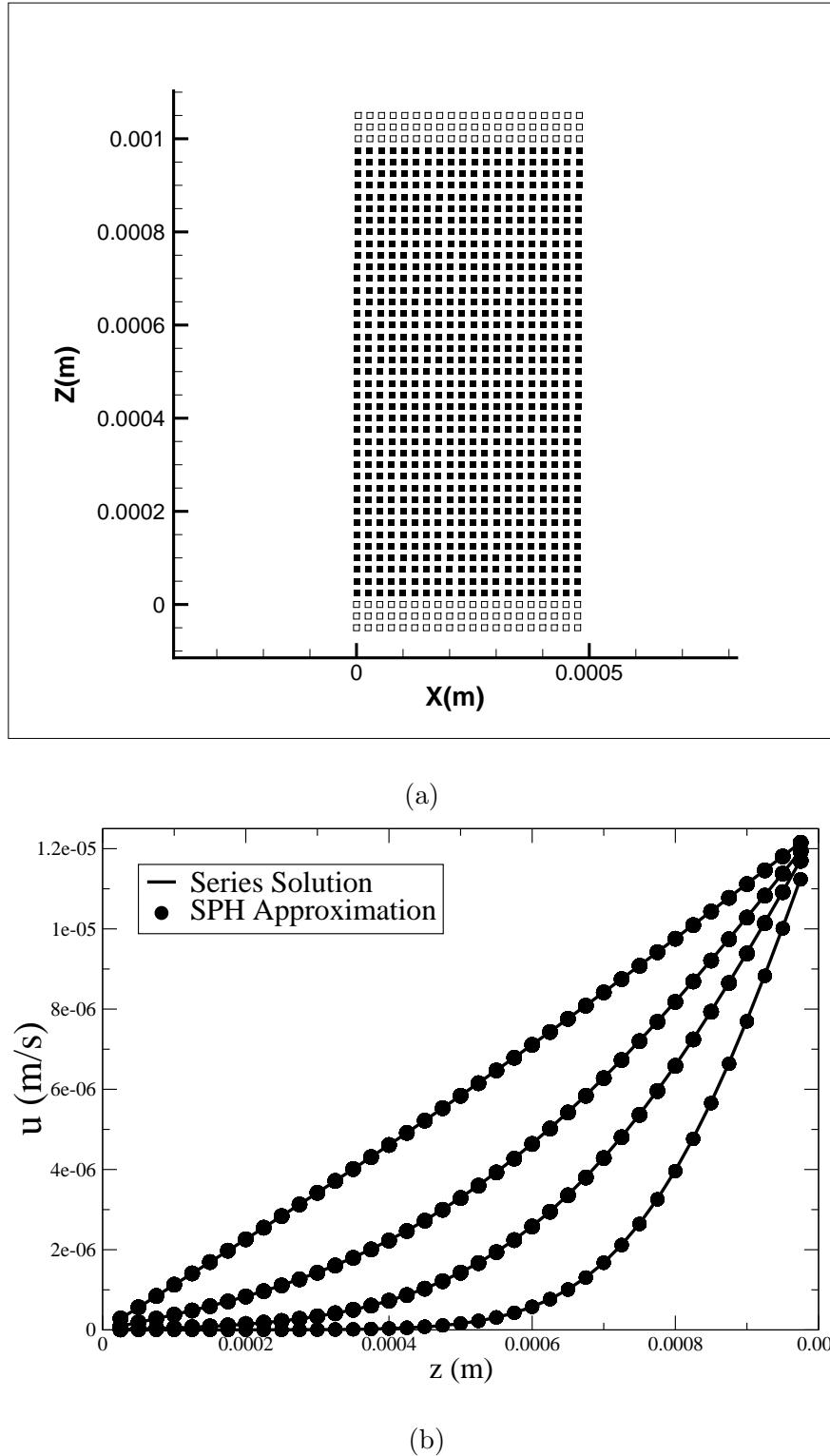


Figure 4.1: (a) Initial set-up for the Couette validation case. (b) Comparison between the series solution and SPH predictions

upper and lower walls, and 780 fluid particles uniformly distributed in the interior of the domain. The initial particle spacing is  $\Delta p = 2.5 \text{e-}5 \text{ ms}^{-1}$  and the smoothing length used for the simulation is set to 1.3 times the initial particle spacing, which is consistent with that used in previous works [78, 3]. Periodic boundary conditions are enforced along the flow direction by allowing particles located within the support domain from a periodic boundary to interact with particles near the opposite boundary (see Gómez-Gesteira et al. (2010) [100] for details).

For this problem, the time step is set to  $10^{-4}\text{s}$ . After approximately 6000 time steps, the flow reaches a steady-state solution. Figure 4.1(b) shows the comparison between the velocity profile obtained using the Cleary viscosity model and the series solution at  $t = 0.02 \text{ s}, 0.05\text{s}, 0.1 \text{ s}$ , and the final steady-state solution. The predictions from the SPH model agree to within 1% of the infinite series solution.

### 4.1.2 3D Dambreaking

The dambreaking case is a popular validation case within the SPH community because the geometry is quite simple to set-up and requires no special in- or out-flow boundary conditions. Here, the 3D dambreaking case presented by Kleefsman et al. (2005) [101] is simulated using the proposed SPH model. The geometry of the dambreak case in [101] is shown in Fig. 4.2 and consists of a 0.55 m high water column and  $0.403\text{m} \times 0.161 \text{ m} \times 0.161 \text{ m}$  box placed inside an open roof tank. The box, which represents a scale model of a container on the deck

of a ship, was instrumented with eight pressure sensors, four on the side facing the water column and four along the top. In this case, quantitative comparisons are made between the SPH predictions and experimental observations for the pressure time histories at the eight pressure sensor locations published in [101]. In the case of the SPH model, the pressure at the sensor locations is calculated using a normalized SPH interpolation, given by

$$f(\mathbf{x}_i) = \sum_j^N f(\mathbf{x}_j) \tilde{W}_{ij} V_j \quad (4.2)$$

where  $\tilde{W}_{ij} = \sum_j W_{ij} V_j$ . A traditional SPH sum at the sensor locations would suffer from a lack of particle support and lead to a poor estimation of the pressure. Normalizing the kernel function near the boundary attempts to correct for this lack of particle support. The parameters used in the SPH model for this validation case are listed in Table 4.1. Figure 4.5 shows the fluid flow evolution as several instances in time, while Figs. 4.3 and 4.4 show comparisons of pressure time history between the SPH model and the experiment at sensor locations P1 through P8. The SPH prediction for the time of the initial wave impact agrees quite well with the experiment; however, the impact creates a significant amount of noise in the SPH pressure signal. The model is able to capture the general trend in the pressure time history for probes P1 - P3, though the prediction becomes less accurate after about 2s. The SPH model over-predicts the peak pressures recorded as sensors P1 - P4, and the agreement between the SPH simulation and the experiment becomes poorer as one moves from the bottom to the top of the box. SPH pressure measurements taken along the top of the box contain a significant amount of high frequency content during the initial wave

Table 4.1: Simulation parameters for the 3D dambreak validation case.

Parameter	Value
Fluid Spacing (m)	0.009
$h/dp$	1.5
No. of Fluid Particles	912560
No. of Boundary Particles	147342
Sound Speed (m/s)	23
CFL No.	0.3
$\rho_o$ (kgm $^{-3}$ )	1000
$\gamma$	7.1
$\nu$ (m $^2$ s $^{-1}$ )	$10^{-6}$

impact which is not present in the experiment. A similar discrepancy in the pressure history was observed in the VOF simulations presented by Kleefsman et al. [101]. For probes P5

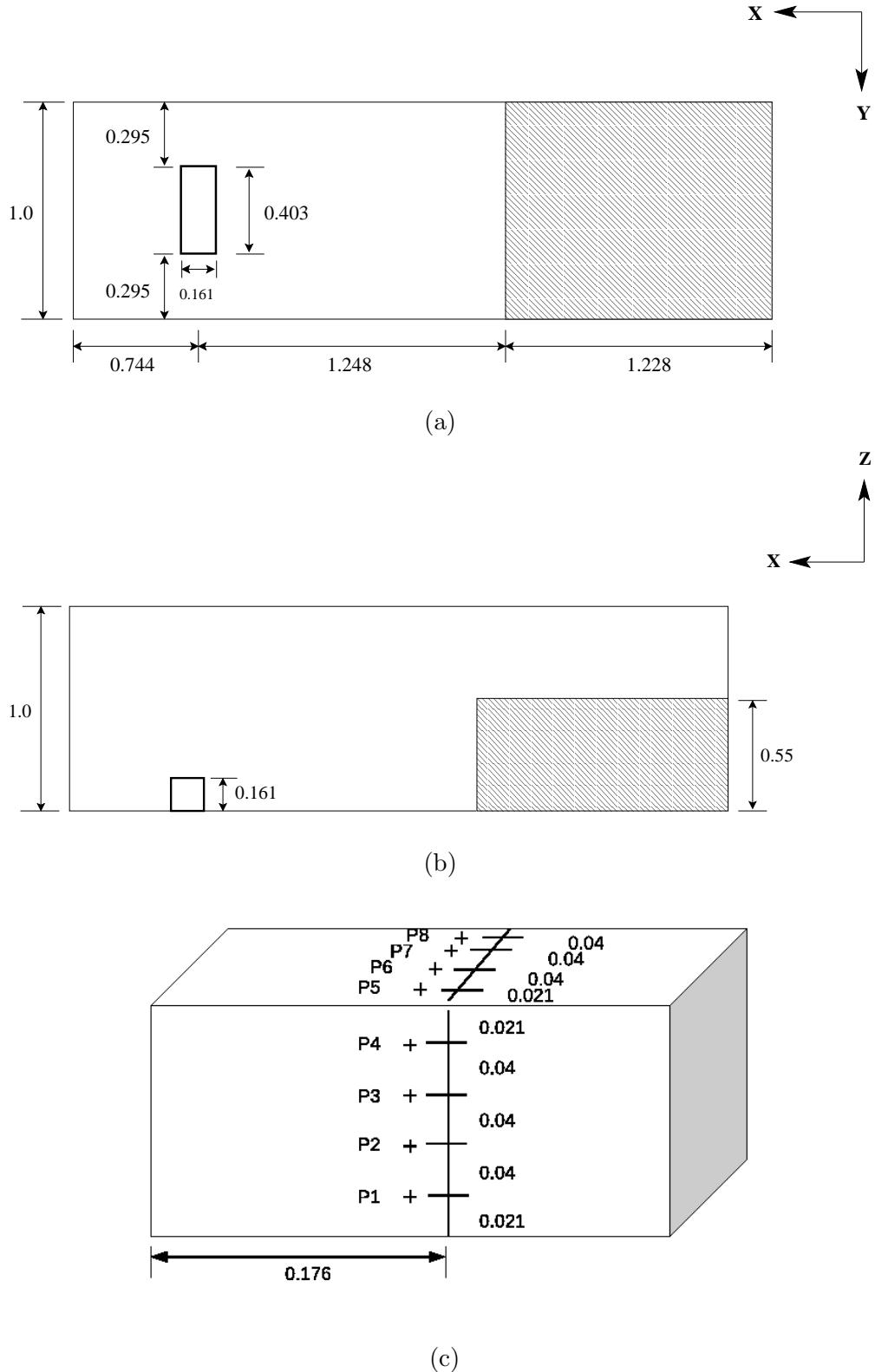


Figure 4.2: (a)-(b) Dimensions and initial configuration for the dam-break validation case.

(c) Pressure sensor locations and box dimensions.

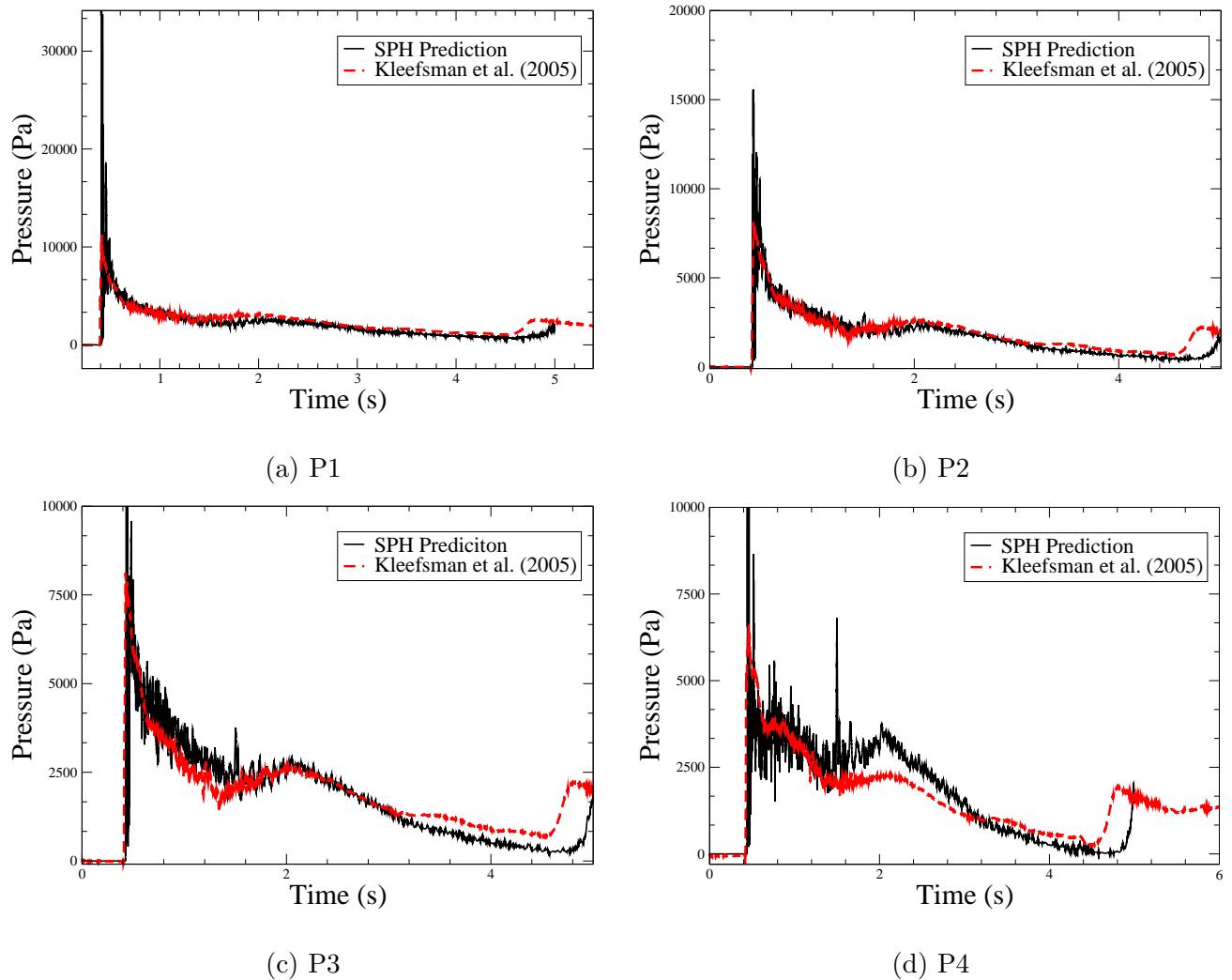


Figure 4.3: Comparision between SPH pressure predictions and experimental observations for sensors P1 - P4.

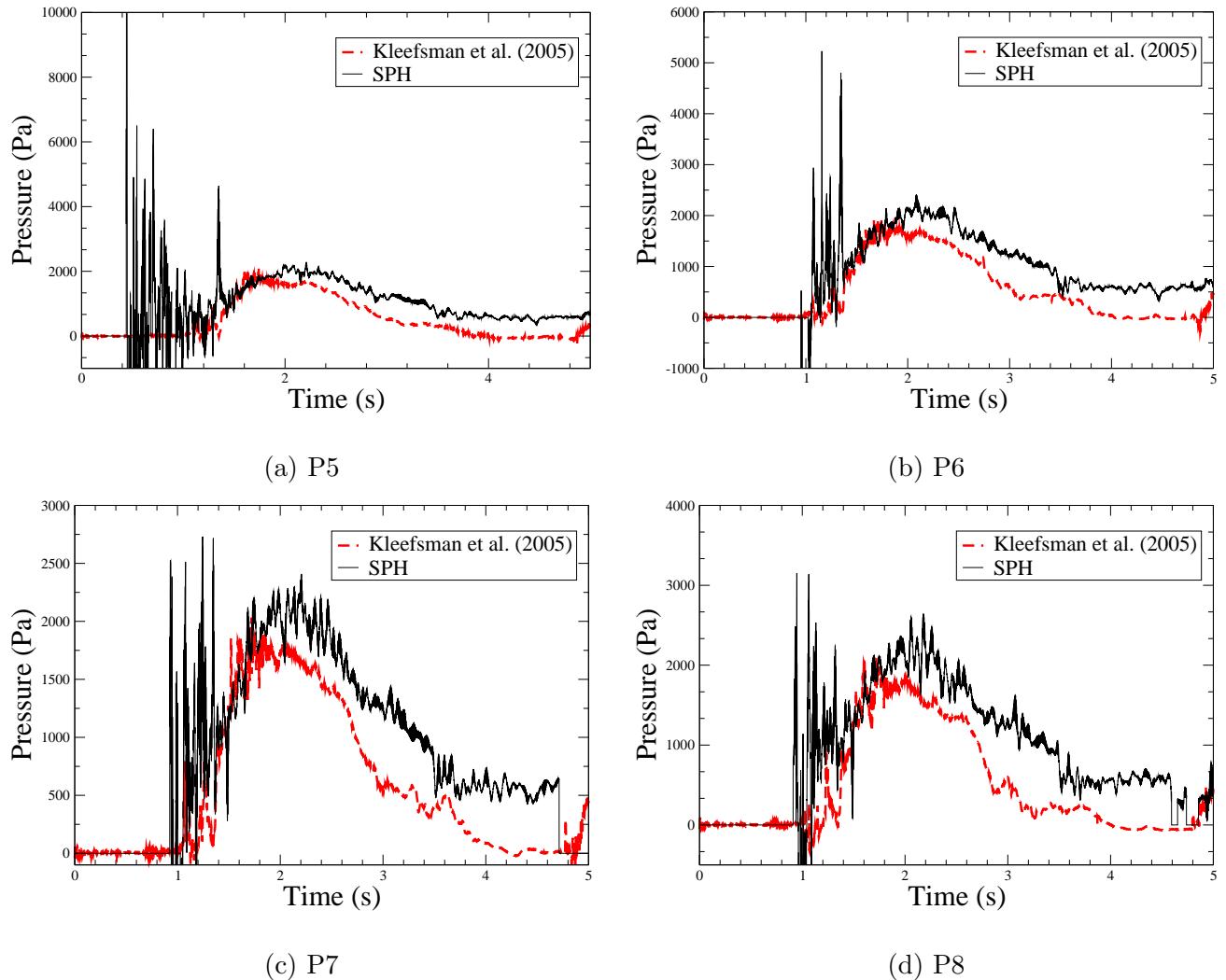


Figure 4.4: Comparision between SPH pressure predictions and experimental observations for sensors P5 - P8.

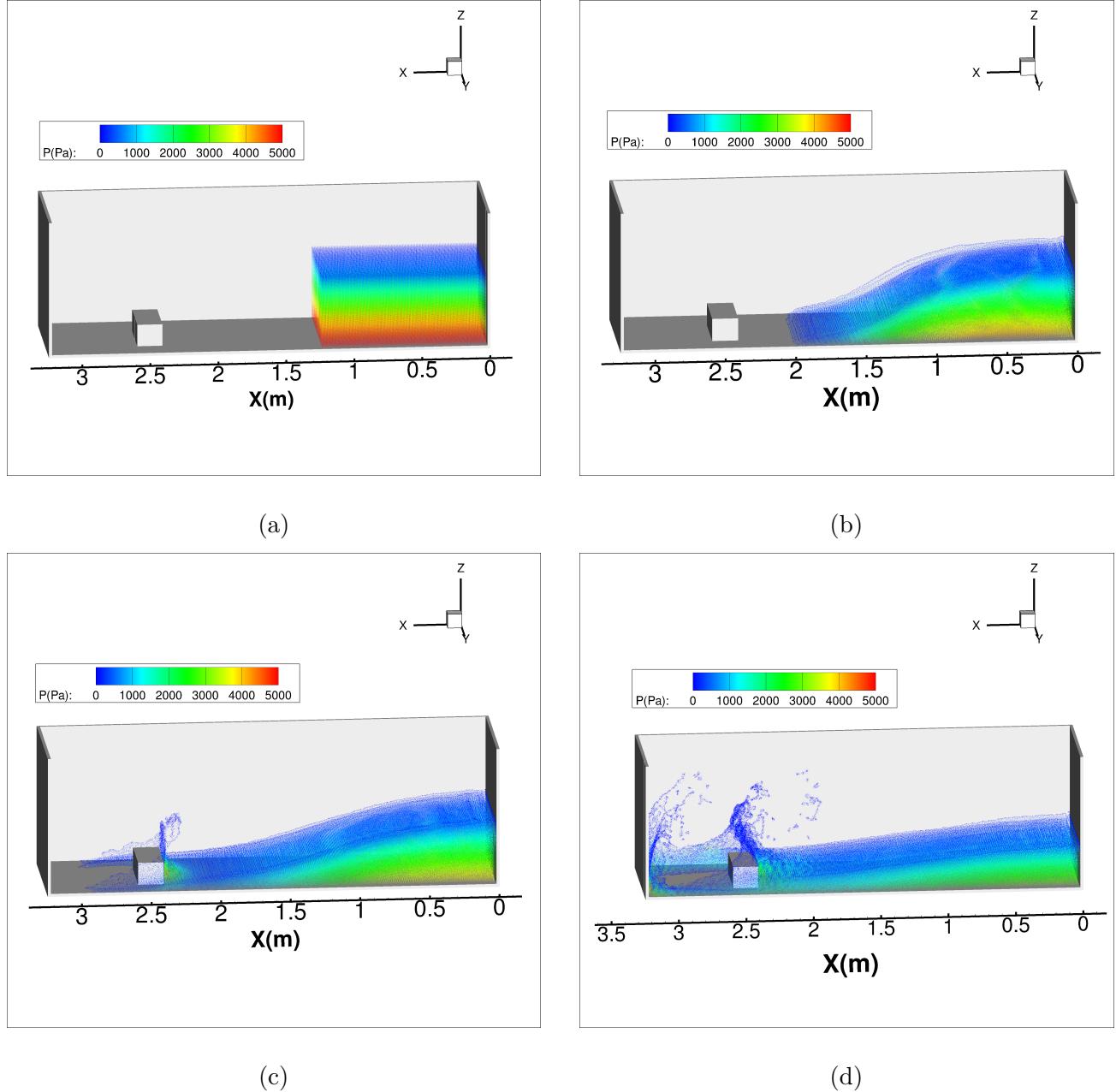


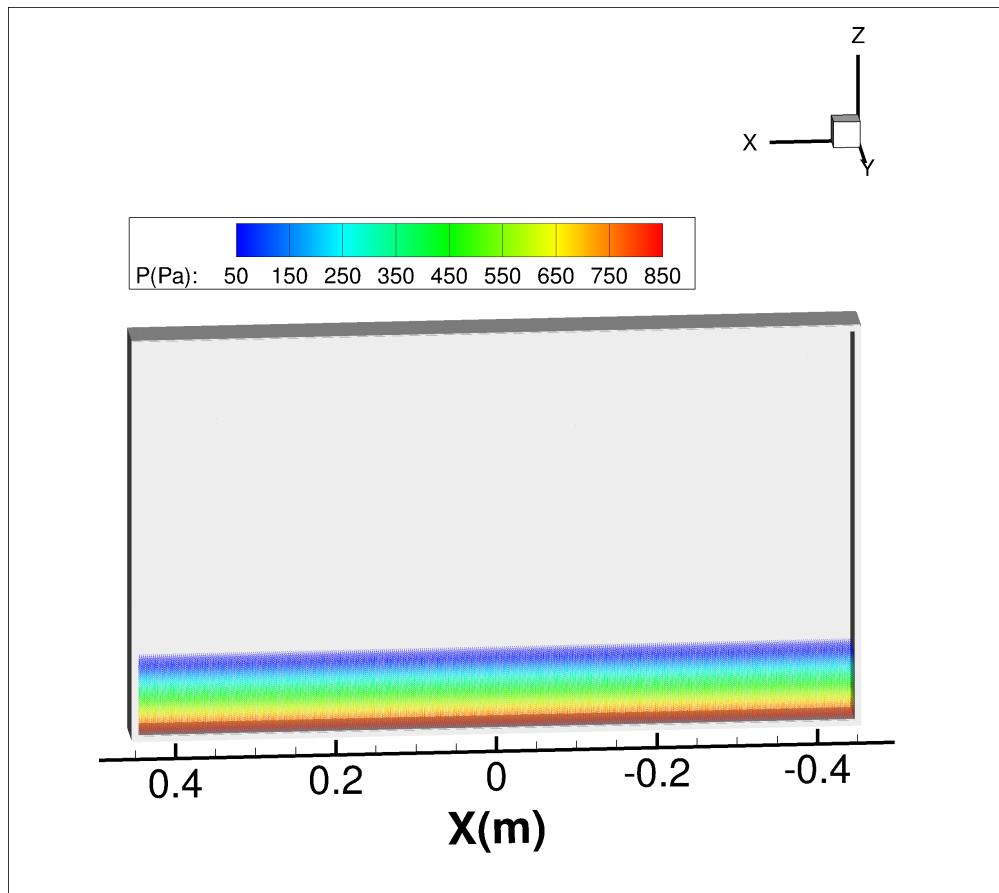
Figure 4.5: (a) Initial particle configuration for the dambreak validation case. (b) - (d) Free-surface evolution at times 0.32 s, 0.56 s, and 0.96 s, respectively.

through P8, the SPH model captures the initial rise and general behavior of the pressures seen in the experiment at those locations, but does not accurately capture the peak pressure rise, which occurs around 2s.

The high frequency content in the SPH pressure signals results from a combination of weak compressibility and the Lagrangian nature of SPH. One can imagine that, as the wave front approaches a sensor location, the normalized SPH interpolation used to compute pressure will first only include one fluid particle, then two, then three, and so on and so forth. This can result in large oscillations in the pressure estimation. As the number of particles surrounding the sensor location increases, the noise in the pressure estimation decreases, which is what one sees in Figs. 4.3 and 4.4 after approximately 2s. Alternative approaches to estimating pressure for comparison to experiments are currently being investigated.

#### 4.1.3 3D Sloshing Tank

Understanding and accurately predicting the loads caused by violent sloshing in cargo tanks is a large concern for the oil and gas industry due to increasing interest in offshore production and storage [102], and several experimental studies have been conducted to better understand these sloshing loads [103, 104, 105, 102]. Here, flow inside a sloshing tank is simulated using the SPH model and a comparison is made with the forced-roll experiment performed by Delorme et al. (2009) [106] to further validate the current numerical model. The forced-roll experiment of Delorme et al., shown in Fig. 4.6 consist of a longitudinal tank filled with



(a)

Figure 4.6: Configuration for the sloshing tank validation case.

either water or

sunflower oil, excited into a sinusoidal rocking motion via an electric motor. The tank is a 1/50<sup>th</sup> scale model tank from a 138000 m<sup>3</sup> LNG membrane tanker.

For this validation case, the SPH model is used to simulate the forced-roll of a tank filled with sunflower oil to a depth of 93cm. The simulation is run for two initial particle spacings,  $dp = 5.0e-3$  and  $dp = 2.5e-3$ , and predictions for the pressure time history at sensor 1 (low filling level) are compared with experimental observations. Like in the previous validation case, the pressure at sensor 1 in the SPH simulation is computed through the normalized sum given by Eq. (4.2). The relevant parameters for the sloshing tank validation case are listed in Table 4.2.

The initial configuration for the simulation is comprised of several layers of boundary particles to model the solid walls of the tank and a uniform distribution of fluid particles. At the start of the simulation, the particle motion is purposely damped to allow the system to settle into a more equilibrium position, as described in [61]. This settling period lasts for approximately 5000 time steps. Once the settling period is complete, the instantaneous roll-angle of the tank is interpolated from the spline-smoothed roll-angle time history data provided by Souto-Iglesias et al. [107] for this validation case. At the start of the experiment there is a short ramp up in roll angle due to the physical limitations of the driving motors. Shortly thereafter, the tank settles into a periodic roll motion.

Figures 4.8(c) - 4.8(d) show the evolution of the fluid at several instances in time as the tank oscillates about its center of rotation for one period of oscillation. Qualitatively, the free-surface deformations predicted by the SPH simulation agree well with those seen in the

Table 4.2: Simulation parameters for the 3D sloshing tank validation case.

Parameter	Value
Fluid Spacing (m)	5e-3, 2.5e-3
$h/dp$	1.5
No. of Fluid Particles	37411, 318792
No. of Boundary Particles	86980, 348624
Sound Speed (m/s)	19.85
CFL No.	0.3
$\rho_o$ (kgm <sup>-3</sup> )	900
$\gamma$	7.0
$\nu$ (m <sup>2</sup> s <sup>-1</sup> )	5 × 10 <sup>-5</sup>

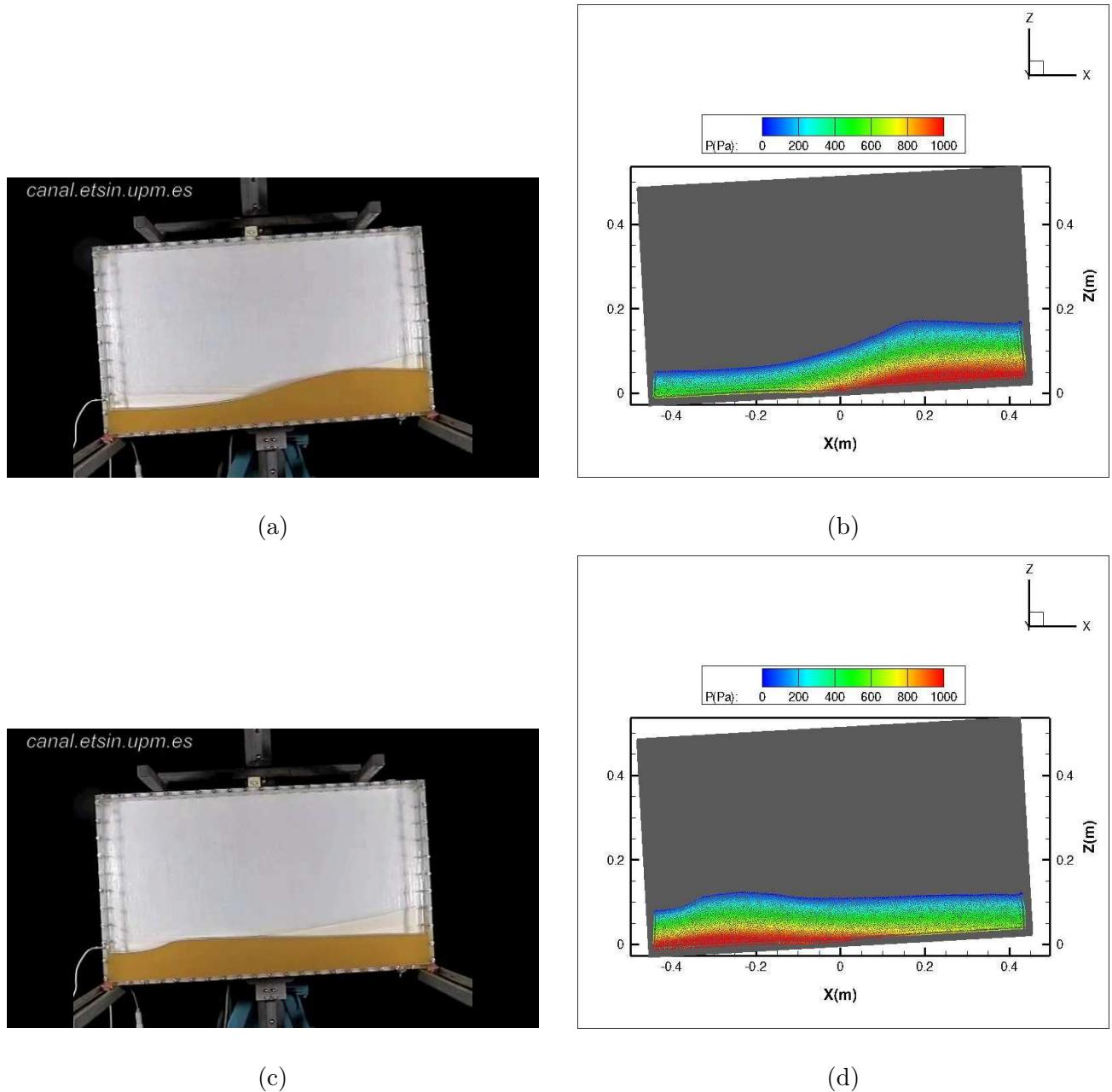


Figure 4.7: Comparision of the free-surface deformation with experimental observations (a,c) with SPH predictions (b,d) at (a,b)  $t = 2\text{s}$  and (c,d)  $t = 2.3\text{s}$ .

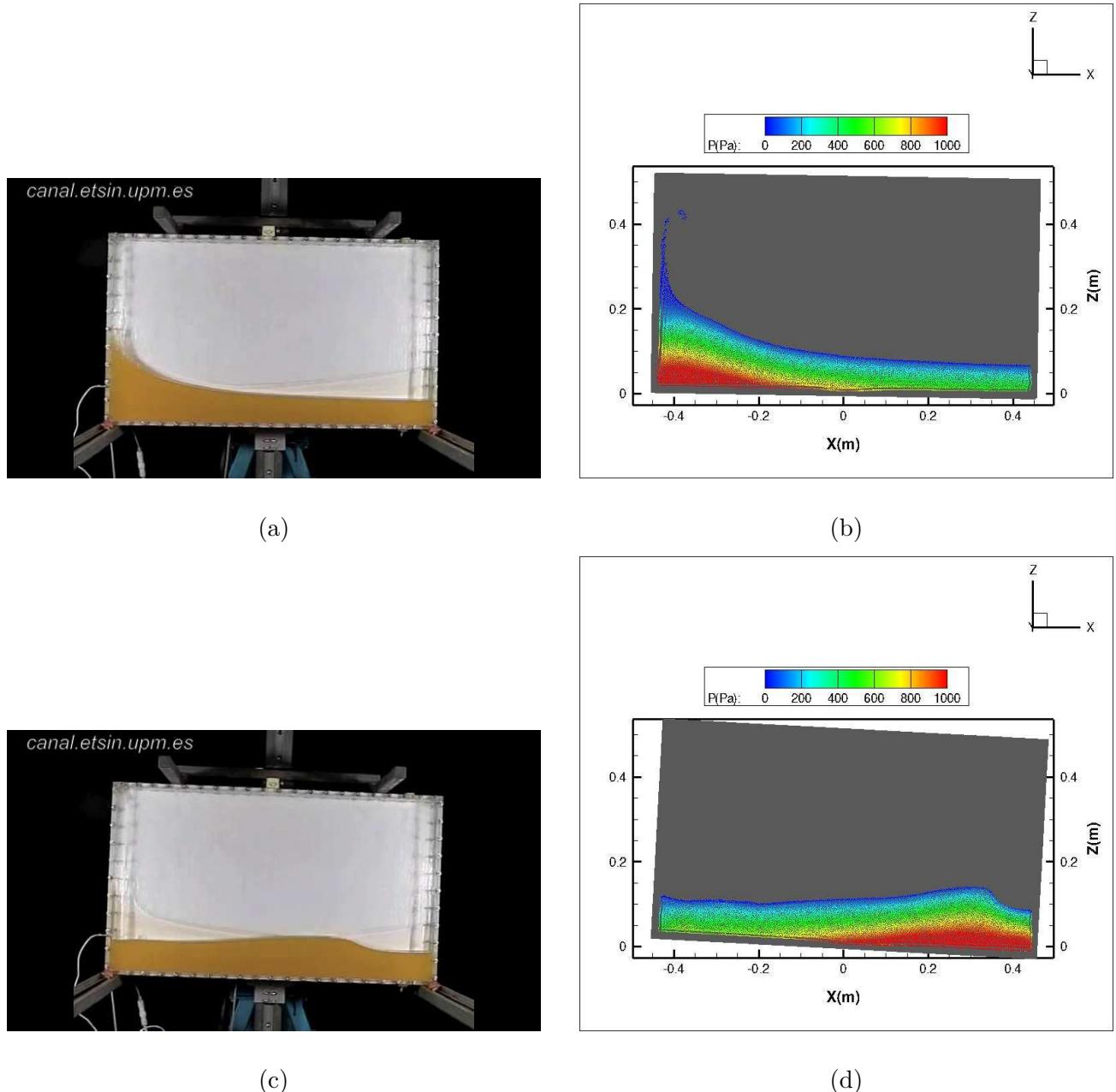
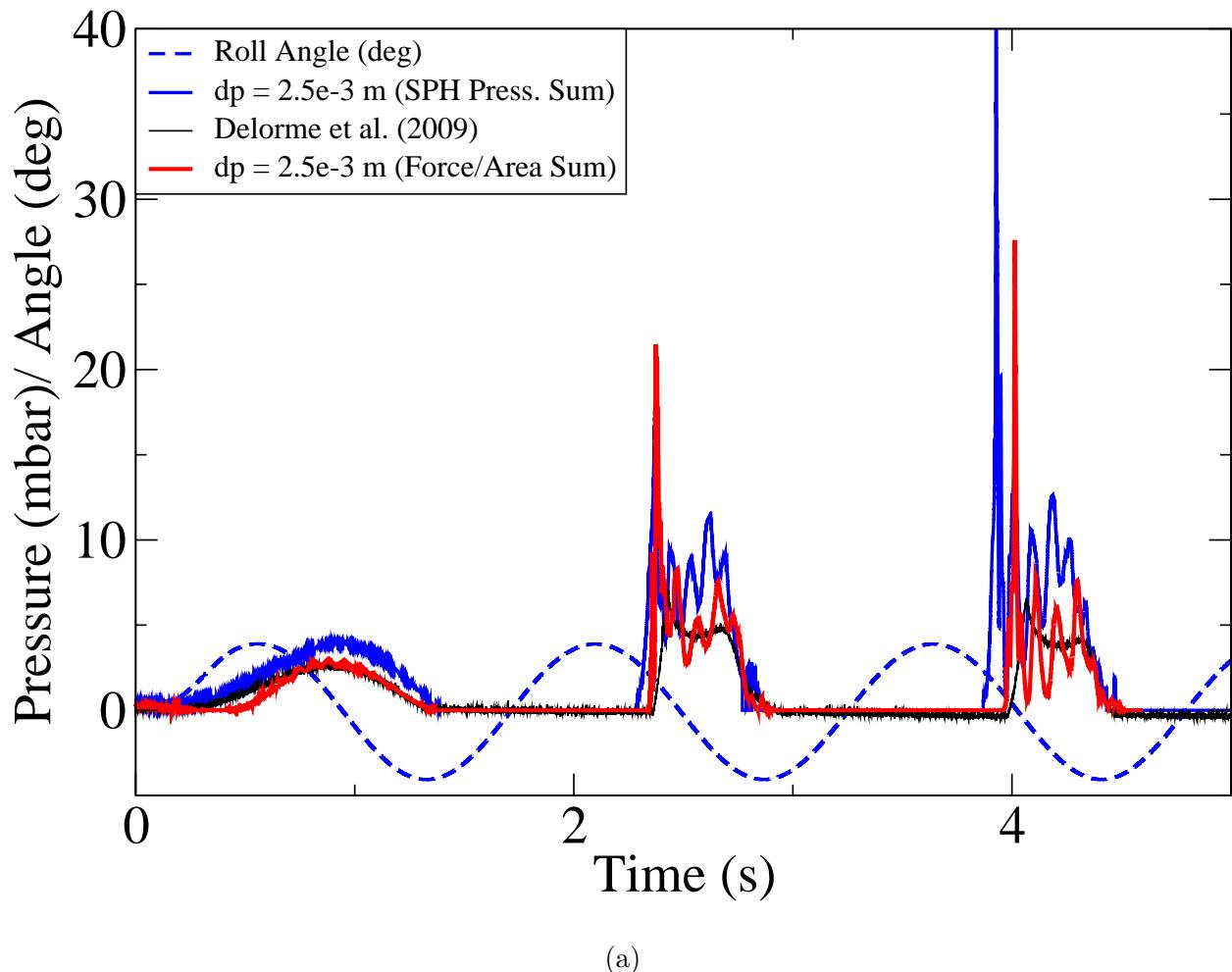


Figure 4.8: Comparision of the free-surface deformation with experimental observations (a,c) with SPH predictions (b,d) at (a,b)  $t = 2.6\text{s}$  and (c,d)  $t = 3.1\text{s}$ .

experiment. A quantitative comparison between the pressure observed in the experiment and simulation predictions for each particle discretization are shown in Fig. 4.9, along with the roll-angle history. It is clear from Fig. 4.9 that the SPH simulation is unable to accurately predict the pressure loads on the tank wall during the forced roll, though the general trend is captured. The small rise in pressure that occurs around 1s during start-up is over-predicted in the SPH simulation. Additionally, subsequent waves arrive at the pressure location sooner in the SPH simulation than in the experiment and the duration of the pressure pulse is longer, though the decay period seems to agree well with the experiment. Increasing the particle resolution (i.e. decreasing the initial particle spacing) results in only slight improvements to the predictions.

One thing to note is that the hydrostatic pressure in the SPH simulation appears to have been approximately 2 mbars higher than the theoretical value at the bottom of the tank prior to beginning the forced roll. Shifting the SPH pressure predictions down by 2 mbars results in a much better agreement with the experiment. Therefore, the discrepancies observed between the simulation and experiment may be a result of using too stiff an equation of state, or perhaps the density formulation given in Eq. (2.43) is not as accurate in the presence of moving boundaries because the velocity difference is not taken into account, as it is when the traditional continuity density of Eq. (2.41) is used to update the particle density.



(a)

Figure 4.9: Pressure time history at sensor 1 for simulations run at  $dp = 5.0\text{e-}3 \text{ m}$  and  $dp = 2.5\text{e-}3 \text{ m}$ .

## 4.2 Validation of SPH-FEM Model

### 2D Elastic Dambreaking

The first simulation used to validate the SPH-FEM model is a two-dimensional validation case comparing flow field predictions from the SPH-FEM solver for a dam-break flow through an elastic gate with the experimental results of Antoci et al. [108]. In this case, a flexible rubber gate, which is fixed in place along the top edge and free along the bottom edge, is initially clamped in place to hold back a column of water. At time  $t = 0$ , the clamp is released and the rubber gate is free to deform under the fluid pressure. The geometry for this case is illustrated in Fig. 4.10(a), while the details of the simulation are listed Table 4.3. A Mooney-Rivlin constitutive model is used to describe the elastic gate material, as it has been shown to yield better agreement with the experimental results [1]. Figures 4.10(c) and 4.10(d) show the initial condition and the system state at time = 0.33s, respectively. A comparison between experimental results and simulation predictions for the time history of the vertical and horizontal gate tip displacements are shown in Fig. 4.10(b). The simulation over predicts both the vertical and horizontal displacements, though the trends are similar to those observed during the experiment. As previously noted by [108] and [1], the over prediction in tip displacement may be attributed to three-dimensional effects not captured in the simulation, such as water leakage around the gate sides which would cause a pressure drop in the fluid behind the gate. Uncertainty in the material properties of the rubber gate may also be a source of additional error in the simulation.

Table 4.3: Dimensions and physical characteristics of the elastic gate validation case.

Parameters	Values
<i>Fluid Details</i>	
$\rho_o$	1000 (kg/m <sup>3</sup> )
Fluid Spacing	8.33e-4 (m)
No. of Fluid Particles	20111
No. of DBC Particles	2520
No. of Interface Particles	253
<i>Elastic Gate Details</i>	
$\rho$	1100 (kg/m <sup>3</sup> )
E	$\approx 10$ (MPa)

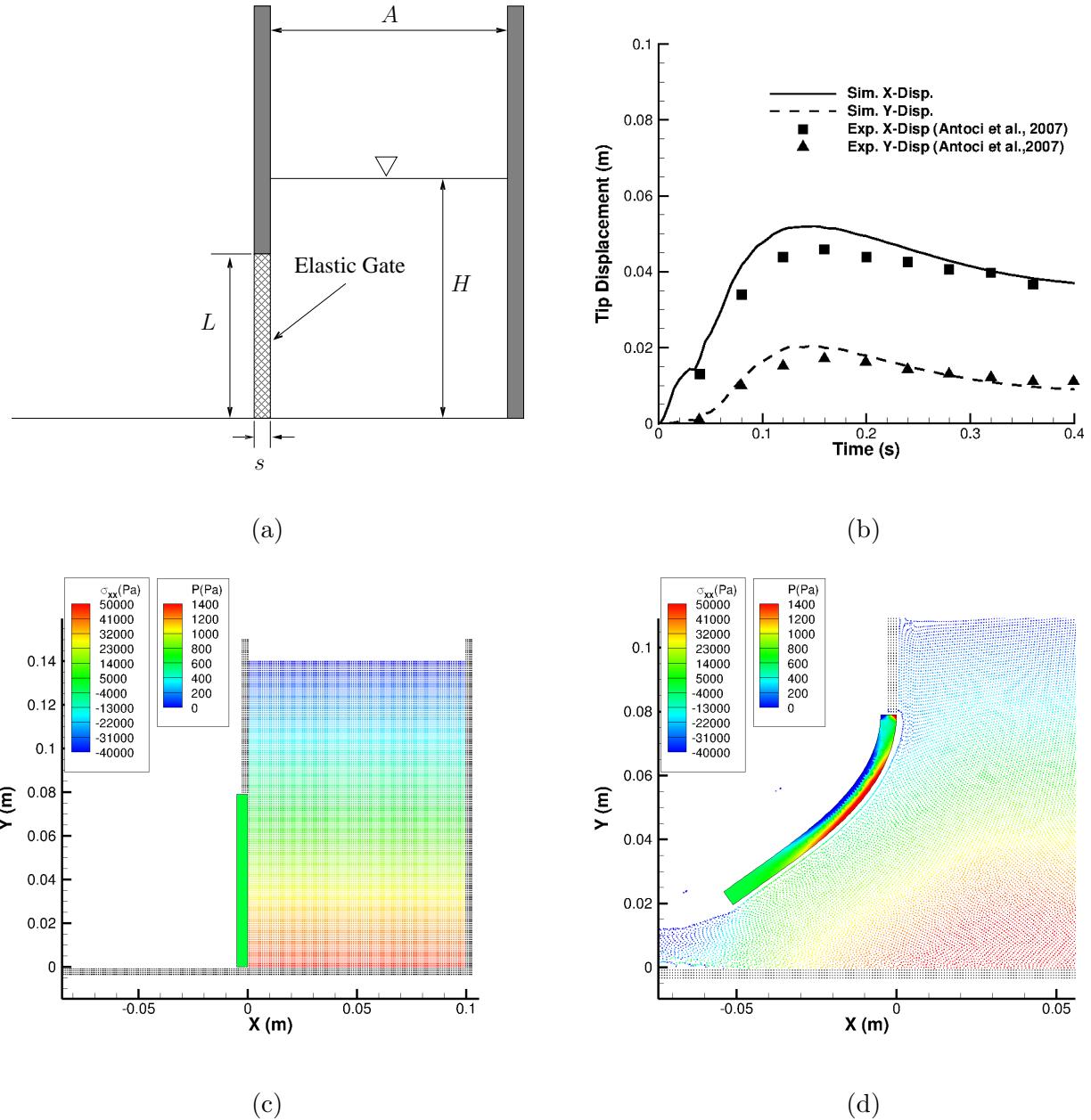


Figure 4.10: (a) Geometry for the dam-breaking with elastic gate validation case. (b) Tip displacement of elastic gate in horizontal and vertical directions. (c) Initial particle distribution for the dambreak with elastic gate. (b) Simulation predictions at time = 0.33 s.

### 3D Elastic Body in Sloshing Tank

The final validation test for the 3D SPH-FEM model involves flow in a sloshing tank interacting with an easily deformable elastic body and is one of the few canonical validation cases which combines fluid-structure interactions and a free-surface. For this validation case, a quantitative comparison is made between the SPH model predictions for the tip displacement of the elastic membrane and the experimentally recorded displacements presented in Souto-Iglesias et al. (2008) [109]. The initial configuration is shown in Fig 4.11(a) and consists of a prismatic tank with dimensions  $609.0\text{mm} \times 344.5\text{mm} \times 39.0\text{mm}$ . The tank's rotational center is at the midpoint of the bottom of the tank. Mounted to the bottom of the tank is a 4mm thick elastic membrane made from commercial neoprene rubber that interacts with the fluid. In this validation case, the working fluid is sunflower oil with a density of  $917 \text{ kgm}^{-3}$ . The liquid depth is set to 114.8mm and the tank is excited with a forced roll amplitude of 4 degrees. The elastic membrane is discretized using both linear and quadratic tetrahedral elements. Table 4.4 lists the relevant parameters for this simulation.

Figure 4.11 shows the initial particle distribution, as well as the distribution of interface particles used to transfer fluid loads to the structural mesh. Figure 4.12 shows the fluid evolution at several instances in time, along with a close-up view of the flow field in the region surrounding the membrane. Interestingly, the pressure field in this SPH simulation was not as smooth as that observed in the previous sloshing tank simulation. Pockets of low pressure appear to develop at several points during the simulation, though the reason for this is not yet known. However, Fig. 4.13(b) shows results for the validation case run by updating

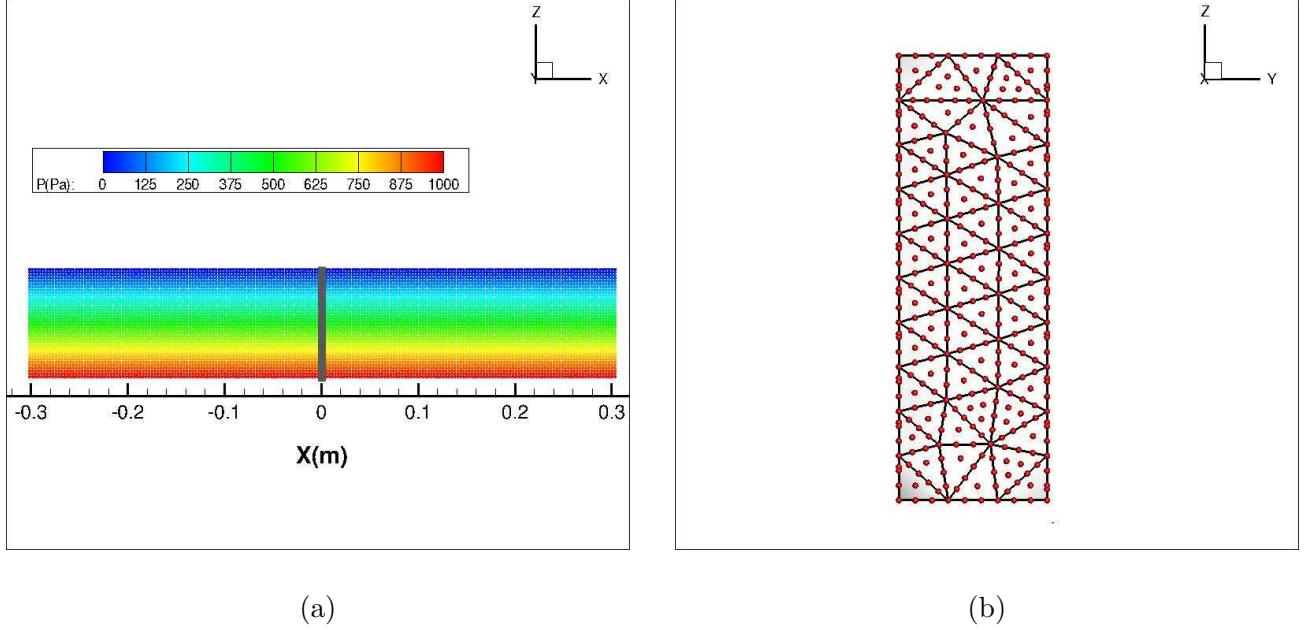


Figure 4.11: (a) Initial particle distribution for validation case (b) Elastic membrane mesh shown with surface interface particles.

particle density using the traditional continuity density approach, rather than the approach of Vila (1999) [50], results in a smooth pressure field, as expected. This suggests that the approach suggested by Vila for updating density may not be suitable for FSI problems, though further study is needed to determine the validity of this hypothesis.

The time history of the local X-displacement at the tip of the elastic body is shown in Fig. 4.13(a) for both 2D and 3D simulation results. From the figure it is clear that the use of linear tetrahedral finite elements results in an overly stiff structure as a result of shear locking and provides poor agreement with the experiment, while the use of quadratic reduced integration tetrahedral elements results in a far more accurate prediction of the membrane deflections.

Table 4.4: Dimensions and physical characteristics of the elastic membrane in sloshing tank validation case.

Parameter	Value
<i>Fluid Details</i>	
Fluid Spacing (m)	0.003
$h/d_p$	1.7
No. of Fluid Particles	91200
No. of Boundary Particles	54936
No. of Interface Particles	791
Sound Speed (m/s)	22.27
CFL No.	0.3
$\rho_o$ (kgm $^{-3}$ )	917
$\gamma$	7.0
$\nu$ (m $^2$ s $^{-1}$ )	$5 \times 10^{-5}$
<i>Elastic Membrane Details</i>	
$\rho$ (kg/m $^3$ )	1100
E (GPa)	0.006

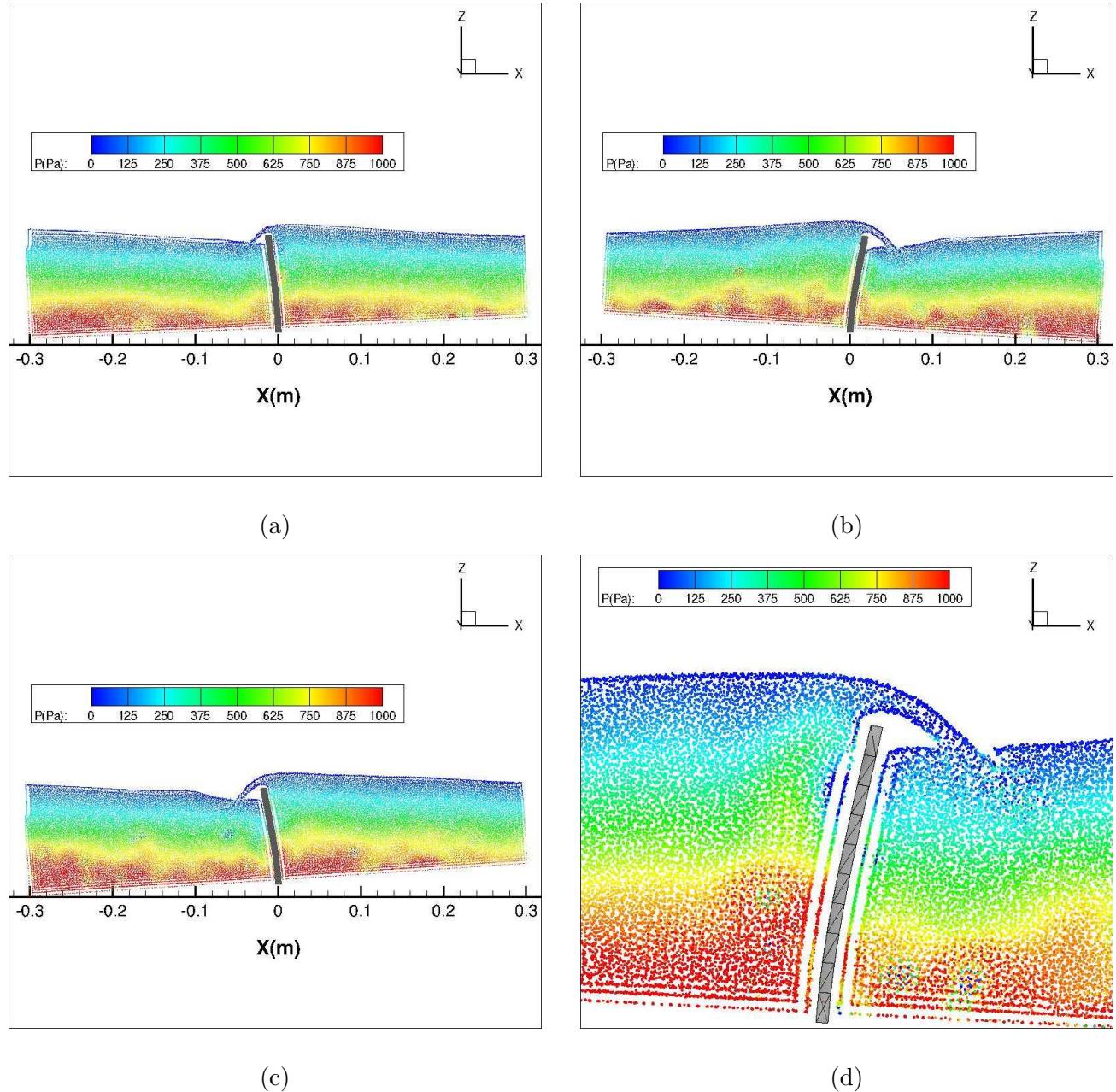


Figure 4.12: SPH pressure predictions at times (a)  $t = 1.35s$ , (b)  $t = 2.45s$ , and (c)  $t = 3.65s$ . (d) Close up of the flow field surrounding the elastic membrane.

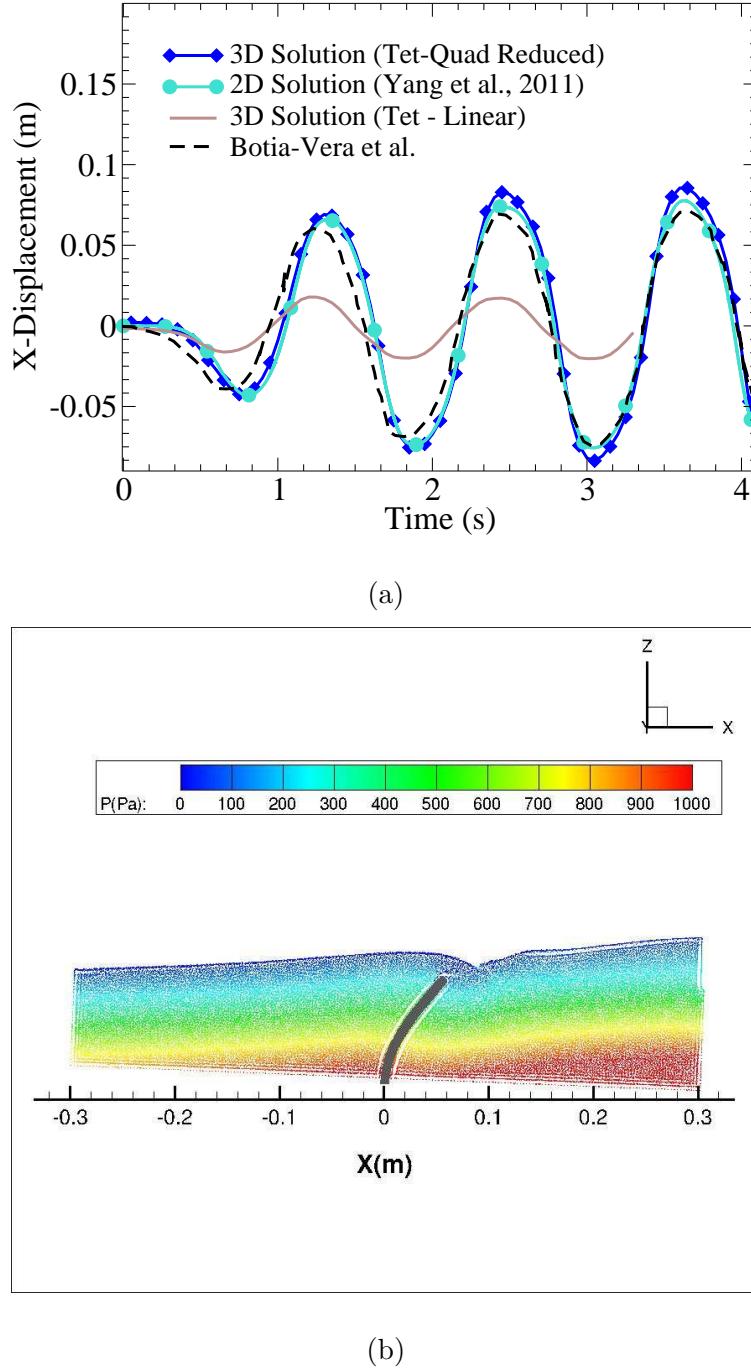


Figure 4.13: (a) Tip displacement results for the SPH-FEM model using both linear and quadratic finite elements. (b) Simulation run with continuity density formulation resulted in smoother pressure field compared with the formulation of Vila (1999) [50].

# Part III

## SPH-FEM Applications

# **Chapter 5**

## **Application of SPH-FEM model to SES Skirt Seal Dynamics**

Zalek and Doctors (2010) presented results from a sophisticated experimental study of the behavior of a surface-effect ship bow seal over a range of forward speeds, cushion pressures, and immersion depths [110]. The experiment, which was performed by the Marine Hydrodynamics Laboratory at the University of Michigan, focused on characterizing the water rise ahead of the seal (bow wave height), wetted seal length, and hydrodynamics resistance under a variety of test conditions. Additionally, experimental measurements of the deflected shape of the bow seal were made at three locations along the centerline of the seal. The experimental apparatus used in the experiment is shown in Fig. 5.1. Interested readers are referred to [110] for further details on the experimental setup. In this chapter, the SPH-FEM model is used to simulate the bow seal dynamics of the SES apparatus described in [110].

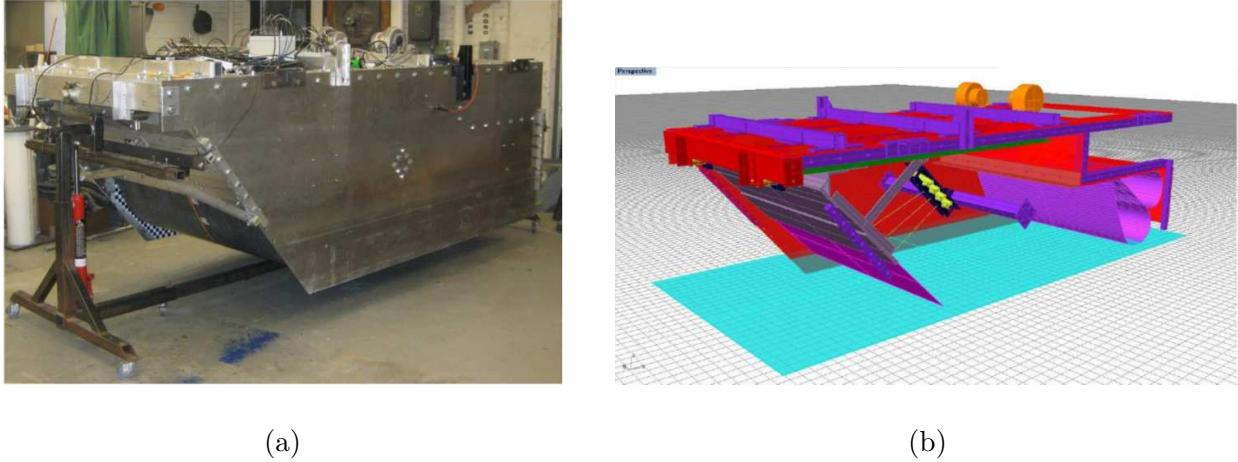


Figure 5.1: (a) Photo of SES apparatus. (b) CAD rendering of test apparatus. Figures reproduced from Zalek and Doctors (2010) [110] with permission.

and the numerical predictions of bow seal displacement are compared to the experimental results provided by Dr. Zalek at the University of Michigan.

## 5.1 Initial Simulation Configuration

Two geometrical configurations were used for the SES simulations. Configuration (1), shown in Fig. 5.2, considers only the SES bow seal and seal mounting plate, which are positioned inside a channel of uniform width. In configuration (1), the bow seal mounting plate is modeled using the DBC method (see section 2.6.4) and SES side walls and stern seal are not included. Configuration (2), shown in Fig. 5.3, attempts to more closely mimic the tow tank conditions of the original experiment. In configuration (2), the overall channel width enlarged to 3 times the width of the ACV, similar to the approach in [16] and additional

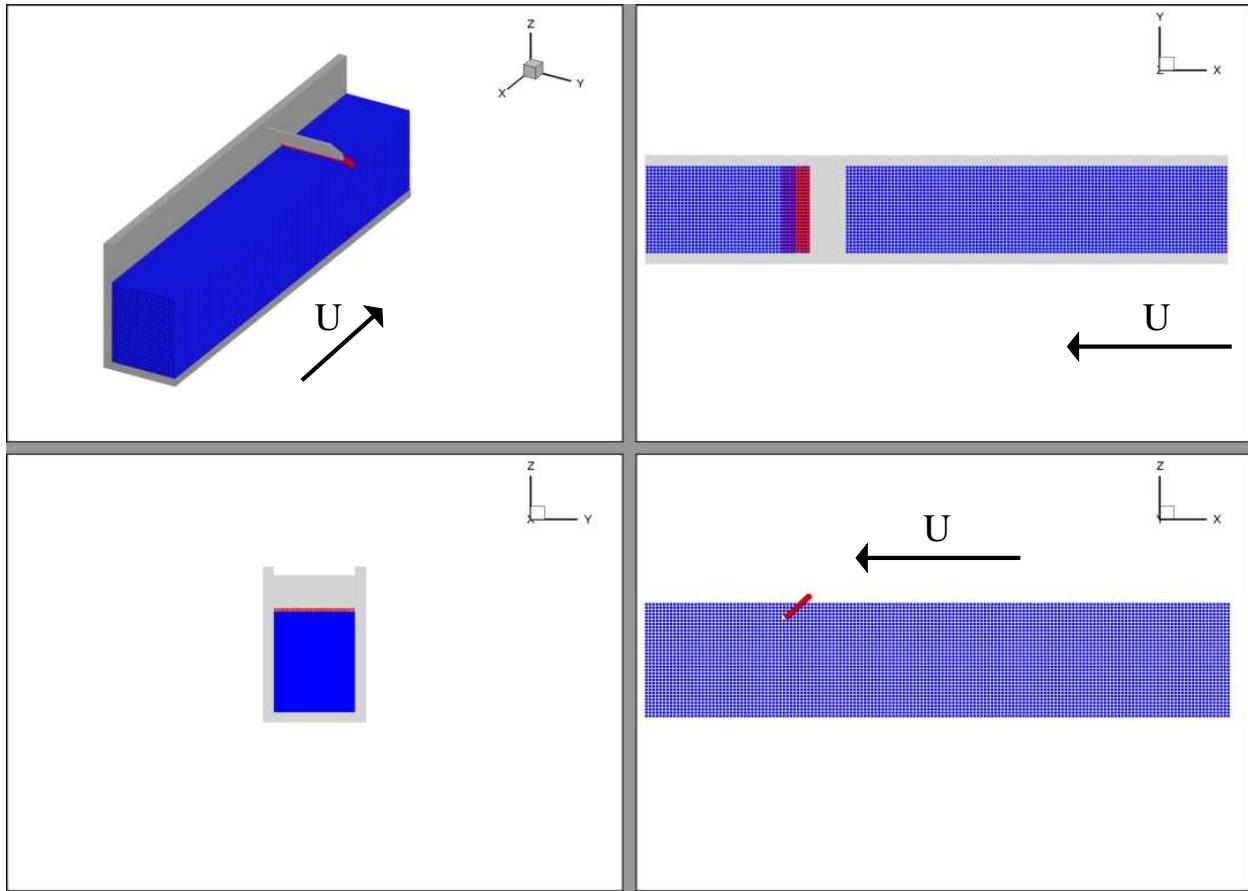


Figure 5.2: Geometry for Configuration (1) used in SES simulations.

DBC particles are used to model the SES side walls. Details of each configuration are listed in Table 5.1.

## 5.2 Modeling the SES Air Cushion

Previously, Yang et al. (2011) [1] used an SPH-FEM approach to simulate the dynamics of an SES bow seal in 2D using two approaches to model the SES air cushion: 1) a specified

Table 5.1: Simulation dimensions and parameters for each configuration used in the SES simulations.

	Configuration (1)	Configuration (2)
Domain Width (m)	1.524	4.572
Domain Length (m)	10	10
Fluid Depth (m)	2	2
SES Beam (m)	1.524	1.542
dp (m)	0.06	0.06
No. of Particles	226,161	544,302
No. of Fluid Particles	156,600	446,232
No. of Boundary Particles	67,684	96,198
No. of FSI Particles	1872	1872

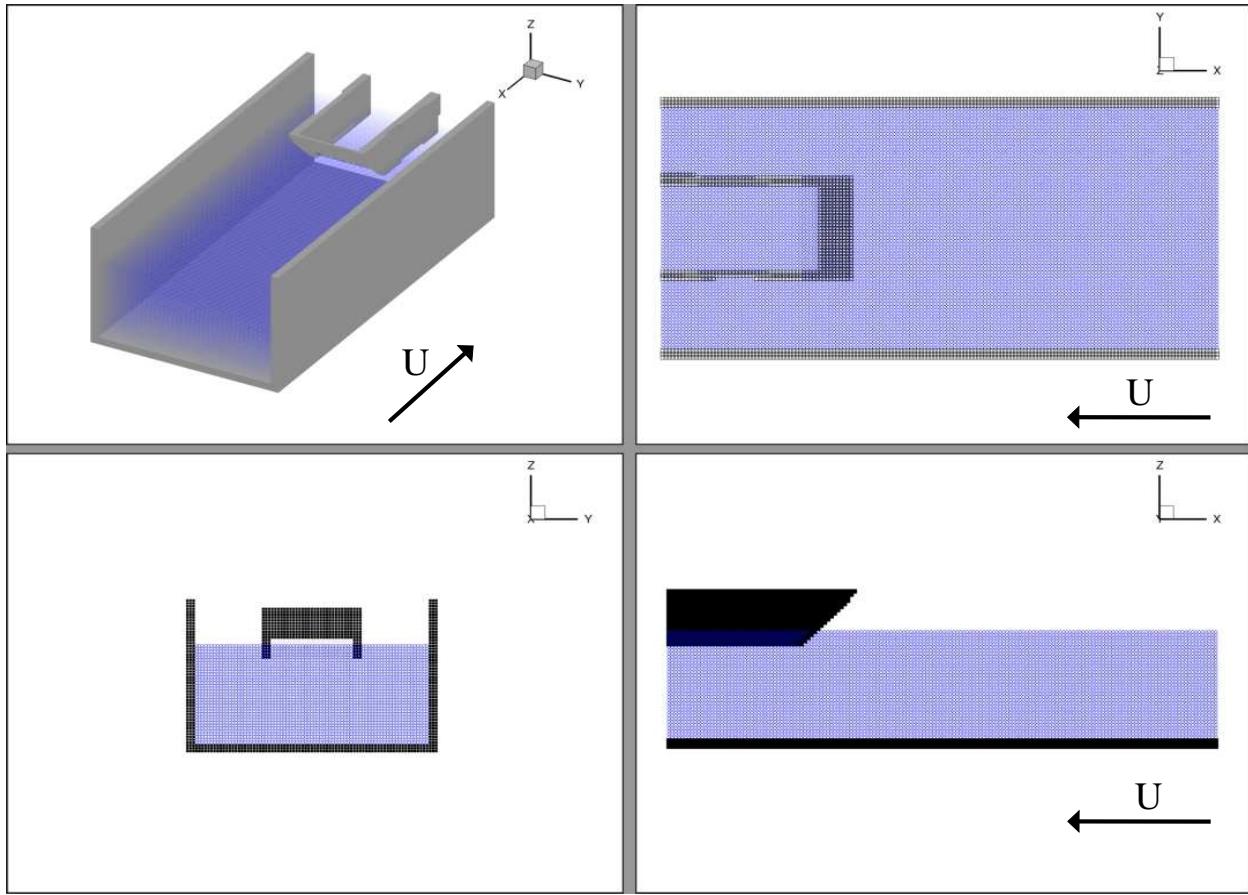


Figure 5.3: Geometry for Configuration (2) used in SES simulations.

cushion pressure applied directly to the cushion side of the SES bow seal, 2) a two-phase (air-water) simulation which applied cushion pressure to the bow seal via pressurized SPH air particles. Both approaches yielded reasonable agreement with experiments, however, the specified seal pressure approach achieved slightly more accurate results and is significantly less computationally expensive than a multi-phase simulation. Based on its success in previous 2D simulations, the specified seal pressure approach was chosen to model the SES cushion pressure in all simulations presented in this chapter.

### 5.3 SES Bow Seal Details

The flexible bow seal used in the UM experiments was constructed from 3.175 mm thick vulcanized neoprene rubber surrounding two nylon fabric layers, which were incorporated to minimize in-plane extension. The flexible bow seal was 0.508 m in length and the upper portion of the seal was attached to a stiffened aluminum plate mounted at a 45 degree angle to the main deck. Three fiberglass transverse stiffeners were mounted onto the seal to minimize three-dimensional effects. The beam of the SES apparatus is 1.524 m and the bow seal was trimmed to fit the SES side walls to minimize air cushion pressure loss. For the simulations presented in this chapter, the bow seal is modeled using a single layer of 3D continuum stress elements and a linear elastic material model. To mimic the presence of the fiberglass stiffeners, a rigid connection is enforced between three transverse rows of FEM nodes along the pressure side of the SES bow seal model. A sample of the FEM mesh used in the SES simulations is shown in Fig. 5.4. Table 5.2 lists the dimensions and material properties of the SES bow seal.

Table 5.2: Dimensions and material properties of SES bow seal. Material constants per Yang (2011) [2].

Parameter	Value
Bow Seal Length (m)	0.508
Bow Seal Width (m)	1.524
Bow Seal Thickness (m)	0.003175
Bow Seal Density (kg/m <sup>3</sup> )	1107
Young's Modulus (MPa)	12.71
Poisson's Ratio, $\nu$	0.25

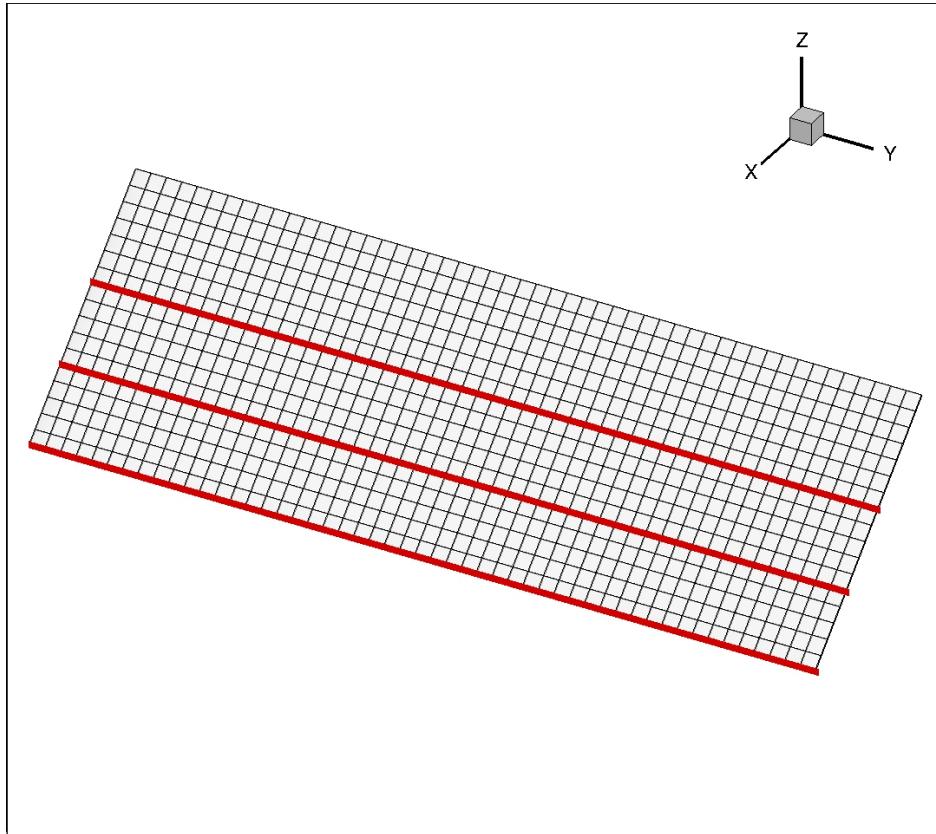


Figure 5.4: Sample of FEM mesh used to model SES bow seal. Here, the bow of the SES is to the right upper right of the figure, and the stern is towards the lower left. A rigid connection is enforced between the transverse nodes highlighted by the red lines to mimic the fiberglass stiffeners used in the experiment.

## 5.4 Inlet-Outlet Boundary Conditions

Simulating the UM SES experiments requires the use of inlet and outlet boundary conditions, which remain a challenge to implement within the SPH formalism. To enforce inflow/outflow boundary conditions in SPH requires fluid particles to be inserted or removed from the domain, respectively, which can cause pressure disturbances to propagate into the computational domain if not handled carefully. For these simulations the approach presented by Marrone [111] is used to enforce inlet and outlet boundary conditions through buffer zones located at the inflow and outflow planes. This is illustrated in Fig. 5.5. In this approach, particles are designated as either inlet, outlet, or fluid particles. Inlet particles are assigned an inlet velocity, vertical height, and hydrostatic pressure and convect across the inlet region according to their prescribed velocity, but do not update their density. Fluid particles interact with both inlet and outlet particles, and when an inlet particle leaves the inlet region it becomes a fluid particles. Likewise, when a fluid particle enters the outlet region, it becomes an outlet particle. An outlet particle is assigned an outlet velocity, but does not update density. Once an outlet particle exits the outlet region it is placed in a queue, which is used to generate new inlet particles. The use of inlet and outlet buffer zones is intended to prevent pressure disturbances from entering into the fluid region during particle creation and removal.

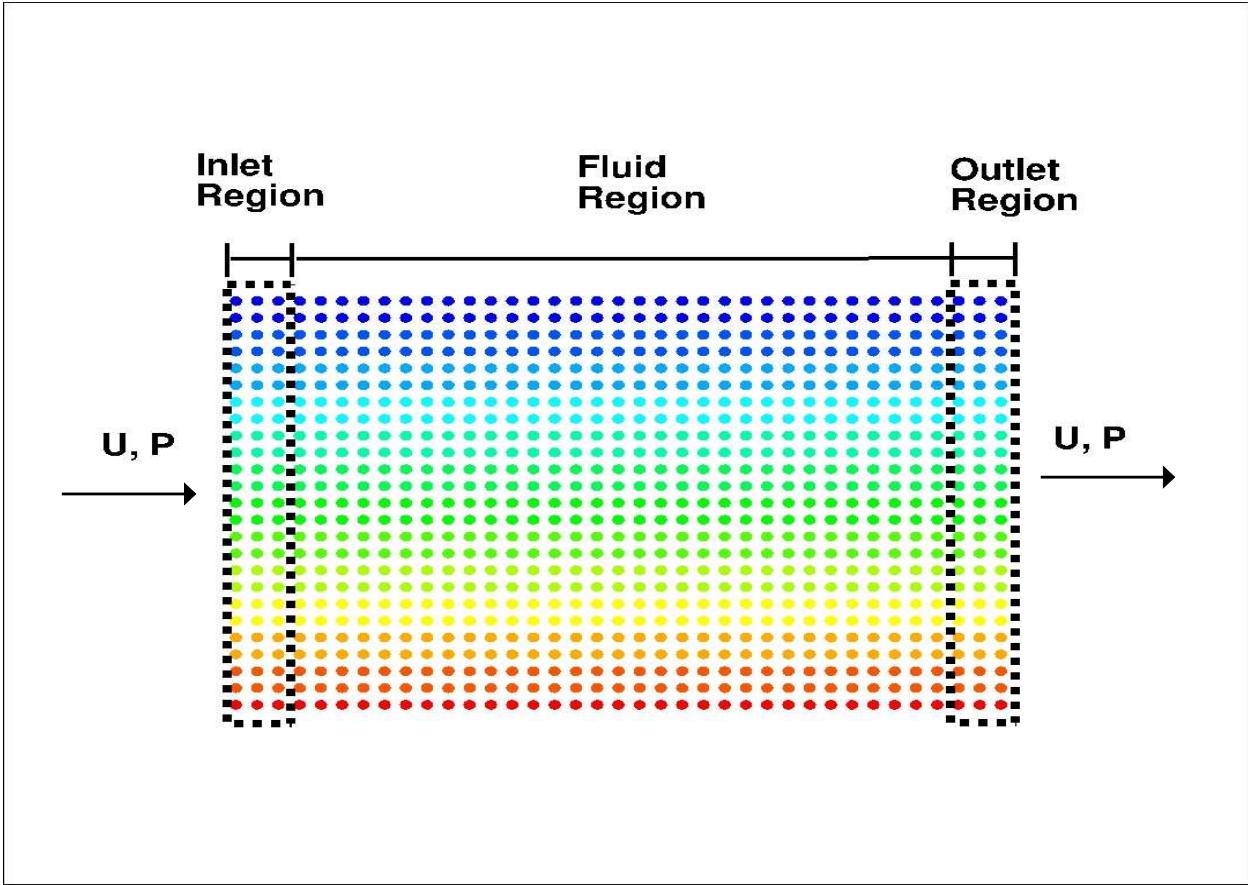


Figure 5.5: Schematic showing inlet and outlet regions for SES simulations. Both inlet and outlet regions are assigned a velocity.

## 5.5 Simulation Results

A total of four SES simulations were run using the 3D SPH-FEM model, the details of which are listed in Table 5.3. Runs 0948, 1013(a), and 1306 were run using Configuration (1), while Run 1013(b) was run using Configuration (2). The simulation parameters for Runs 1013(a) and 1013(b) were identical, save for the SES configuration employed. This was done to

Table 5.3: Details of each SES simulation.

	Run 0948	Run 1013(a)	Run 1306	Run 1013(b)
Configuration	1	1	1	2
Flow Speed, U (m/s)	1.83	2.44	2.74	2.44
Initial Seal Depth, $\delta_s$ (m)	0.191	0.191	0.191	0.191
Cushion Pressure, $P_c$ (Pa)	909.17	418.47	998.85	418.47
Static seal Froude number,				
$Fr = U/\sqrt{g\delta_s}$	1.336	1.7814	2.004	1.7814

determine to what degree the exterior flow field surrounding the SES influences the behavior of the bow seal. The parameters chosen for these simulations allows for a direct comparison with the UM experimental results. Each of the simulations took approximately 6-7 days for 9 seconds of simulation time on the HokieSpeed cluster of Virginia Tech's Advanced Research Computing center. Figures 5.6(a) and 5.6(b) show the time history of the x- and z-displacement of the trailing edge of the bow seal, measured at the centerline of the seal. No seal deflection is observed during the first 0.6 seconds of the simulation because the bow seal is held fixed during this interval as part of the start-up procedure to allow the fluid to accelerate to the appropriate flow speed. From these figures it appears that Runs 0948 and 1306 have reached a quasi-steady state after approximately 7 seconds, however, Runs 1013(a) and 1013(b) do not reach a quasi-steady state before the end of the simulations. It is also

interesting to note the differences in bow seal behavior between Runs 1013(a) and 1013(b). In Run 1013(b), the presence of the SES side walls have altered the flow field behind the bow seal in such a way as to cause the bow seal to behave much more dynamically than the behavior observed in Run 1013(a). One can see that the wetted length of the cushion side of the bow seal is much greater in Run 1013(b) than in 1013(a) due to a lower fluid velocity in the region near the bow seal trailing edge. Clearly, the 3D effects associated with flow around the SES is a more important feature to capture when trying to model the bow seal of an SES in 3D than previous 2D simulations had suggested.

Figures 5.7(a)-5.7(d) show a cross-section of the flow field and bow seal position predicted by the SPH-FEM model at the end of each simulation, while Fig. 5.8 shows a close-up view of the flow field surrounding the bow seal during Run 1013(b). In all the runs, save Run 1306, fluid in the SES cushion can be seen flowing back onto the pressure side of the bow seal, rather than flowing smoothly towards the outlet region, as one might expect. The most extreme case of this can be seen in Fig. 5.7(a), where the bow seal appears to be completely immersed in the fluid, instead of riding along the water's surface. In the case of Runs 1013(a) and 1013(b), the fluid was observed to periodically splash up against the air cushion side of the bow seal causing it to “flutter”, which can be seen clearly in the behavior of the seal tip shown in Fig. 5.6(b). This behavior is inconsistent with that observed in the SES experiments and may be largely attributed to an inability of the current inlet-outlet boundary condition to correctly reproduce tow tank flow conditions. This is partly because the flow velocity set at the outlet is not consistent with the tow tank outflow condition and

causes a standing wave to develop behind the bow seal.

A quantitative comparison between the quasi-steady state bow seal positions predicted by the SPH-FEM model and the experiment are shown in Figs. 5.9(a)-5.9(c). In these figures, the bow seal configuration shown represents the mean value of the seal position along the centerline over the last second of simulation time. In all cases, the SPH-FEM model is unable to correctly predict the quasi-steady state bow seal position. Rather than riding on the free surface of the water, the bow seal appears to be pushed deeper into the water due to the constant cushion pressure, which is what one sees in Figs. 5.7(a)-5.7(d) as well.

There are several plausible reasons why the 3D SPH-FEM model predictions show sub-optimal agreement with the experimental observations. As was previously mentioned, the current inlet-outlet boundary condition may be unsuitable for modeling tow tank scenarios, and accurately enforcing inflow/outflow conditions remains an active area of research within the SPH community. One approach may be to create a numerical tow tank within which an SES model can be towed. This proposition would be computationally expensive, but not impossible, as previous multi-GPU SPH codes have demonstrated [112]. An alternative approach would be to replace the outlet buffer region with several layers of ghost particles set to hydrostatic pressure and density. These ghost particles would maintain their hydrostatic properties throughout the simulation, but take on the velocities of the fluid particles with which they interact. Put simply, a fluid particle near the outlet would see a ghost particle moving at its same velocity, but at the free stream pressure and density. This is somewhat analogous to a pressure outlet condition frequently employed in FVM simulations.

Additionally, the influence of the pressurized air cushion on the fluid inside the cushion was neglected in these simulations, as was the presence of stern seals on the SES model. This was done to reduce the complexity of the simulations by eliminating the need to identify free-surface fluid particles inside the SES cushion and the need for a second, more complex, structural model. Moreover, based on the 2D results of Yang et al. [2], it was reasoned that these features would have a minimal effect on the behavior of the bow seal. However, based on the results from the 3D SPH-FEM model, including the SES stern seals and applying the cushion pressure to the fluid surface may alter the profile of the free surface behind the bow seal, which can influence its behavior. Furthermore, the fluid domain is quite small and under-resolved, in comparison to similar FVM simulations performed previously [16]. This is due to computational limitations and increasing the particle count and domain size may lead to improved estimates in the bow seal behavior. However, a rigorous convergence study is needed to verify this and is outside the scope of this work.

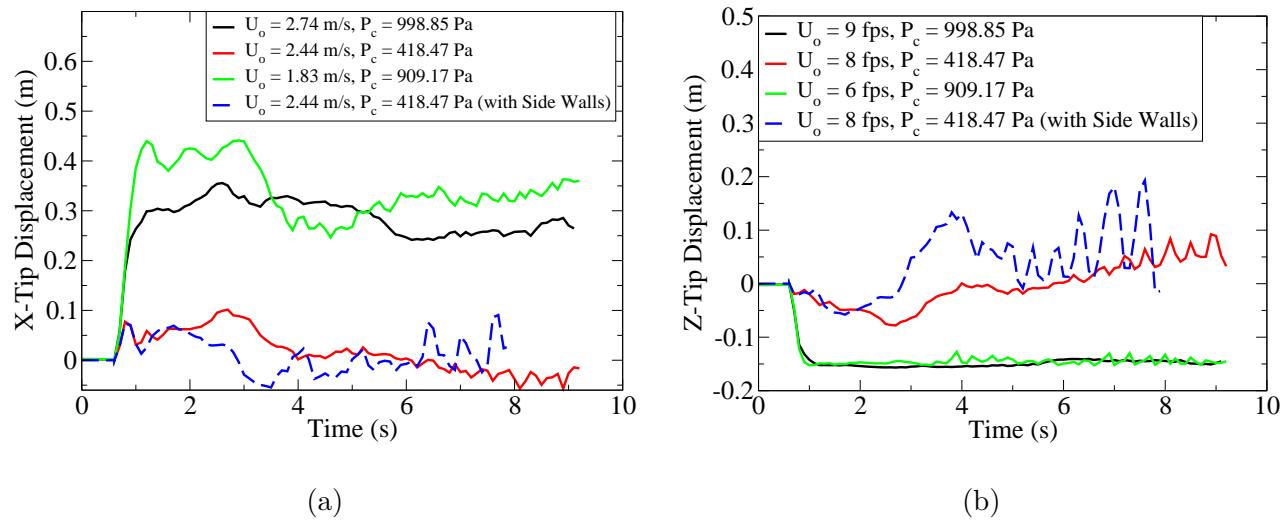


Figure 5.6: (a) Time history of the x-displacement measured from the trailing edge of the bow seal. (b) Time history of the z-displacement measured from the trailing edge of the bow seal.

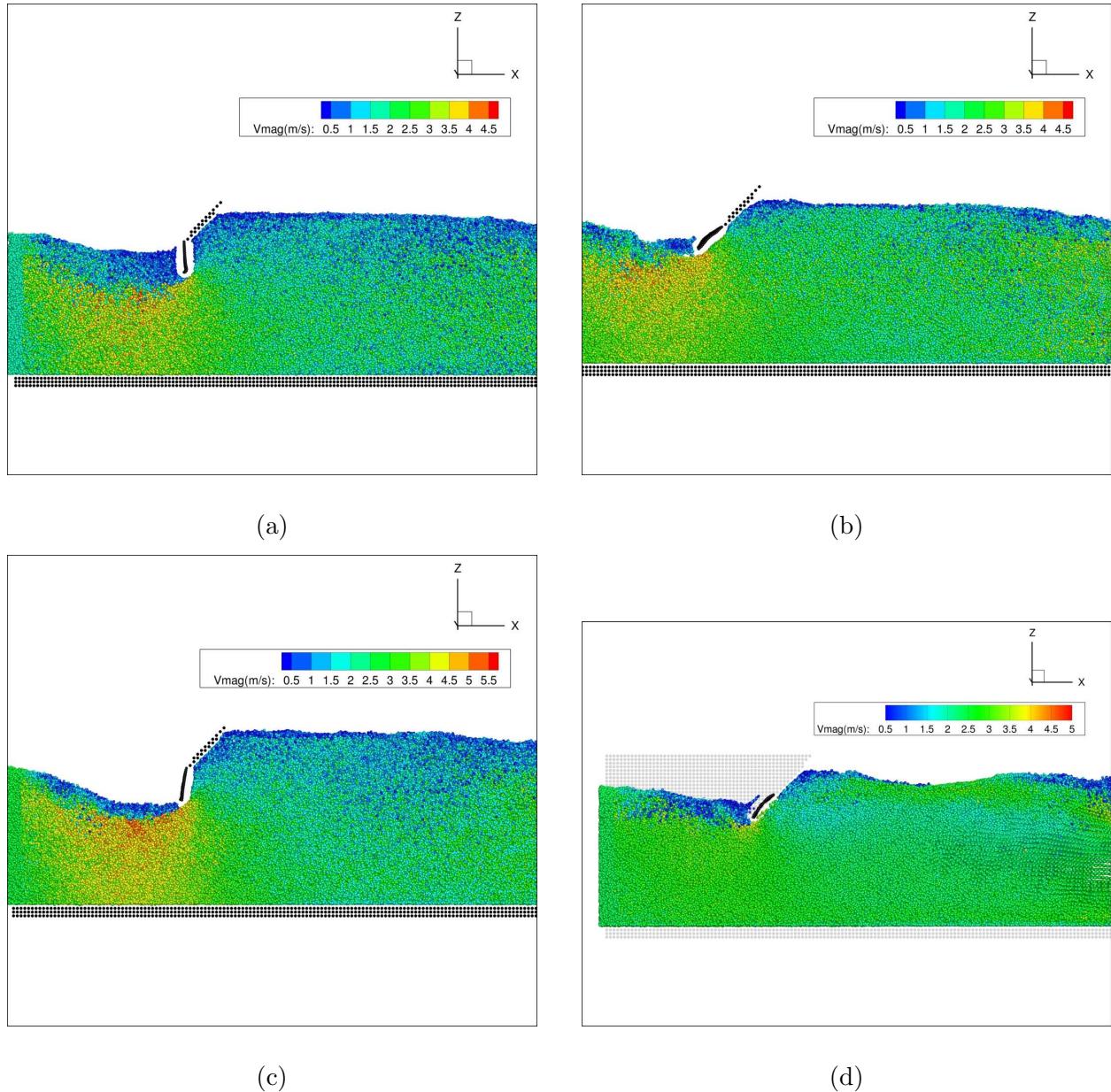


Figure 5.7: Cross-section of the flow surrounding the bow seal after 9 seconds of simulation time for (a) Run 0948, (b) Run 1013(a), (c) Run 1306, and (d) Run 1013(b).

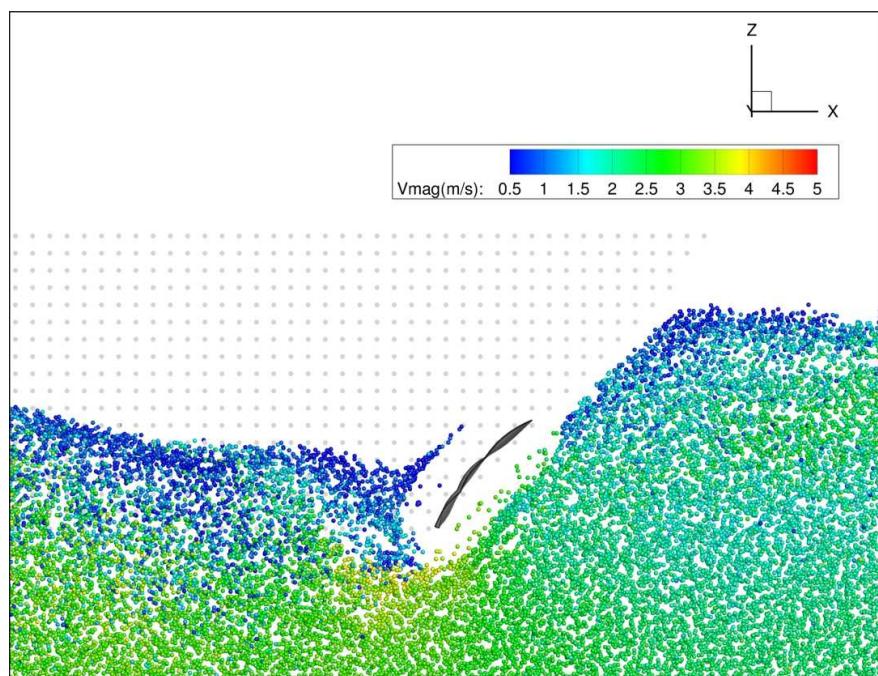


Figure 5.8: Close-up of the flow field surrounding the bow seal during Run 1013(b).

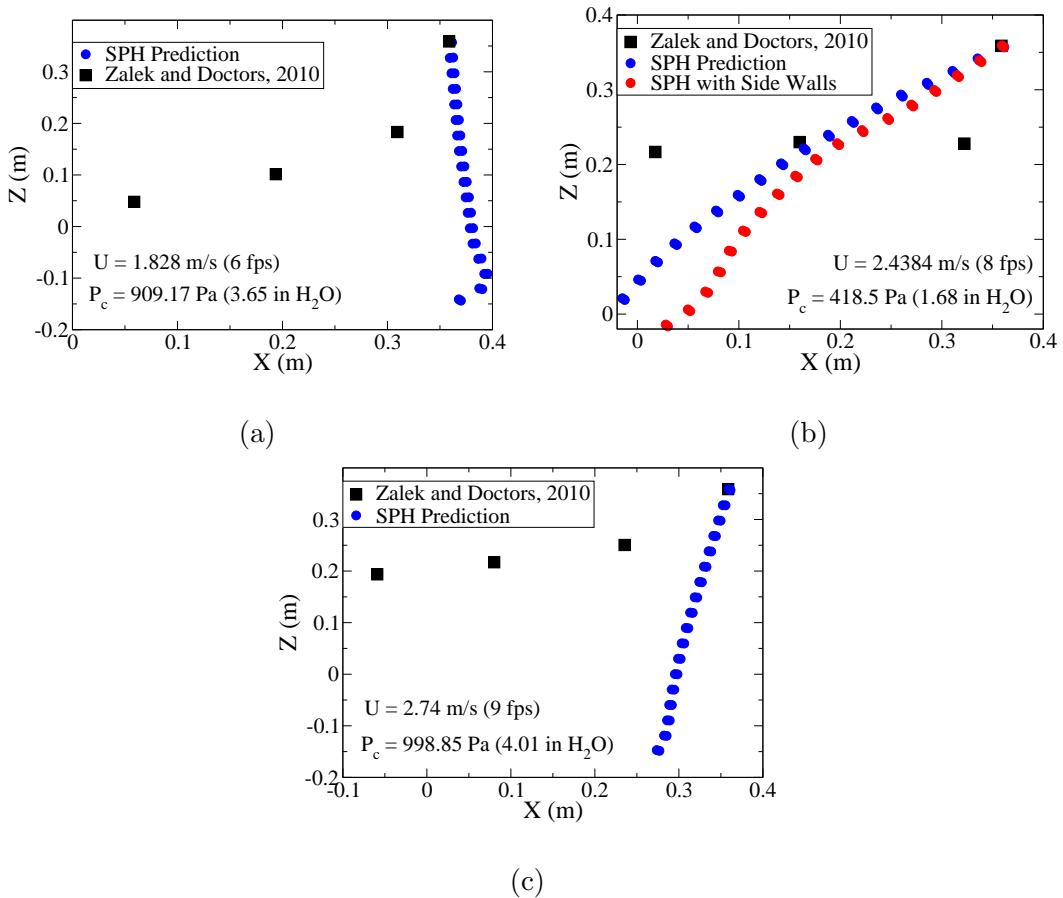


Figure 5.9: A comparison of the SPH-FEM and experimental results for steady-state bow seal position for (a) Run 0948, (b) Run 1013(a) and 1013(b), and (c) Run 1306.

# Chapter 6

## Conclusions

A GPU accelerated SPH-FEM model has been developed with the intention of analyzing a broad class of fluid-structure interaction (FSI) problems. The computational tool uses a partitioned approach to simulate FSI phenomena and incorporates an in-house 3D SPH code implemented on the GPU and the open-source general FEM software *Code\_Aster*. Chapter 2 introduced the basic concepts behind the SPH method, its application to fluid flows, and discussed the details of implementing SPH on the GPU. Chapter 3 presented the fluid-structure coupling approach through the use of repulsive boundary particles. Chapter 4 presented results from several validation cases, which included a 2D Couette flow, a 3D dambreaking, and a 3D sloshing tank case. In addition, two FSI validation cases, namely a 2D elastic gate problem [108] and the 3D interactions between a sloshing fluid and an elastic membrane [107], demonstrated the ability of the 3D SPH-FEM solver to accurately predict the behavior of flexible structures interacting with complex free-surface flows.

In Chapter 5, the 3D SPH-FEM model was used to simulate the dynamics of an SES bow seal. The quasi-steady state position of the bow seal predicted by the simulation was compared with experimental data collected at the University of Michigan [110]. Noteworthy weakness were identified when attempting to utilize the 3D GPU-CPU coupled SPH-FEM model for simulating the bow seal of an air cushion vehicle. These weaknesses include inconsistencies between the inlet/outlet boundary conditions used in the simulation and the conditions actually present during the tow tank experiments, an overly simplified SES configuration (i.e. lack of a stern seal), omitting the influence of the SES air cushion on the fluid, and poor resolution of the fluid domain. However, these challenges are specific to the 3D SES simulations. It should be noted that the objective of this work was the development of a general tool for analyzing fluid-structure interaction problems. That the 3D solver predictions agreed quite well with the FSI validation cases presented in Chapter 4 suggests that the SPH-FEM method is capable of accurately simulating FSI problems.

This suggests many avenues for future work. Increasing the available computational resources, primarily through the use of multiple GPUs and variable particle resolution [113] is a fairly straightforward and necessary extension of the current work. Improvements to the current fluid-structure coupling approach also need to be investigated. While the use of repulsive boundary particles is straightforward, conserves momentum, and is well suited to modeling complex deformable geometries, it can limit the time step significantly and results in poor estimations of pressure near the boundary due to particle deficiencies. It is also limited in that the force applied to the structure is based solely on the distance between a fluid

particle and a boundary particle, independent of what pressure exists within the fluid. This would mean, for instance, that suction cannot be accurately modeled using this approach. Additionally, as the scale and complexity of these simulations increases, so too does the difficulty of assessing the predictive accuracy of the simulation. Therefore, better approaches to code verification and validation of SPH solvers, such as Method of Manufactured Solutions (MMS) [114], should be explored.

# Bibliography

- [1] Q. Yang, V. Jones, and L. McCue, “Free-surface flow interactions with deformable structures using an SPH-FEM model,” *Ocean Engineering*, vol. 55, pp. 136–147, 2012.
- [2] Q. Yang, V. Jones, and L. McCue, “Validation of a SPH-FEM model for seal dynamics of surface effect ships,” *Naval Engineers Journal*, vol. 125, no. 4, 2013.
- [3] G. Liu and M. Liu, *Smoothed Particle Hydrodynamics: A Meshfree Particle Method*. World Scientific Publishing Co., 2003.
- [4] “*Code\_Aster*: An open source general FEA software.” URL <http://www.code-aster.org>.
- [5] Naval Studies Board, “Naval expeditionary logistics enabling operational maneuver from the sea,” 1999. National Academy Press Washington, D.C.
- [6] A. Skolnick and R. A. Wilson, “T-Craft-what, why & how?,” 11th International Conference on Fast Sea Transportation, (Honolulu, Hawaii), 2011.

- [7] Office of Naval Research, “Sea base connector transformable-craft (t-craft) prototype demonstrator,” 2005. Broad Agency Announcement (BAA) 05-020.
- [8] J. C. Ryan, M. Redmond, H. Weimin, M. Badal, and W. Latham, “T-Craft-a transformable sea base connector,” 11th International Conference on Fast Sea Transportation, (Honolulu, Hawaii), 2011.
- [9] K. Cooper, “Sea base enabler innovative naval prototype transformable craft (t-craft),” 2009. SD-5 Panel Brief, Office of Naval Research.
- [10] L. J. Doctors, “Nonlinear motion of an air-cushion vehicle over waves,” *Journal of Hydraulics*, vol. 9, no. 2, pp. 44–57, 1975.
- [11] P. A. Sullivan, P. A. Charest, and T. Ma, “Heave stiffness of an air cushion vehicle bag and finger skirt,” *Journal of Ship Research*, vol. 38, no. 4, pp. 302–307, 1994.
- [12] K. P. Thiagarajan and M. T. Morris-Thomas, “Wave-induced motions of an air cushion structure in shallow water,” *Ocean Engineering*, vol. 33, pp. 1143–1160, 2006.
- [13] B. Milewski, B. Connell, J. Wilson, and D. Kring, “Dynamics of air cushion vehicles operating in a seaway,” 9th International Conference on Numerical Ship Hydrodynamics, 2007.
- [14] B. Milewski, B. Connell, J. Wilson, and D. Kring, “Initial validation of the acvsim model for dynamics of air cushion vehicles,” 27th Symposium on Naval Hydrodynamics, 2008.

- [15] J. Garcia-Espinosa, B. Servan, D. D. Capua, P. Ubach, and E. Onate., “Advances in the development of a fem model for evaluation of a surface effect ship (ses) including skirt dynamics,” SNAMES 34th Annual Journal, 2012/2013.
- [16] A. Bloxom, *Numerical Simulation of the Fluid-Structure Interaction of a Surface Effect Ship Bow Seal*. PhD Thesis, Virginia Polytechnic Institute and State University, 2014.
- [17] A. Bloxom, A. Medeling, C. Vince, and C. S. Yim, “Modeling and testing of inflatable structures for rapidly deployable port infrastructures,” tech. rep., Technical report NSWCCD-CISD-2010/010-Ship System Integration and Design Department, Naval Surface Warfare Center, Carderock Division, West Bethesda, MD 20817-5700, 2010.
- [18] R. Paredes and L. Imas, “Application of sph in fluid-structure interaction problems involving free-surface hydrodynamics,” 11th International Conference on Fast Sea Transportation, (Honolulu, Hawaii), 2011.
- [19] F. H. Harlow, “The particle-in-cell computing method for fluid dynamics,” *Methods in Computational Physics*, vol. 3, pp. 319–343, 1964.
- [20] F. H. Harlow and J. E. Welch, “Numerical calculation of time-dependent viscous incompressible flow of fluid with free surface,” *Physics of Fluids*, vol. 8, pp. 2182–2189, 1965.
- [21] F. H. Harlow and J. E. Welch, “Numerical study of large-amplitude free-surface motions,” *Physics of Fluids*, vol. 9, pp. 842–851, 1966.

- [22] T. Belytschko, Y. Y. Lu, and L. Gu, “Element-free galerkin methods,” *International Journal for Numerical Methods in Engineering*, vol. 37, pp. 229–256, 1994.
- [23] D. Hietel, K. Steiner, and J. Struckmeier, “A finite volume particle method for compressible flows,” *Mathematical Models and Methods in Applied Science*, vol. 10, pp. 1363–1382, 2000.
- [24] R. Keck, “A projection technique for incompressible flow in the meshless finite volume particle method,” *Advances in Computational Mathematics*, vol. 23, pp. 143–169, 2005.
- [25] R. M. Nestor, M. Basa, M. Lastiwka, and N. J. Quinlan, “Extension of the finite volume particle method to viscous flow,” *Journal of Computational Physics*, vol. 228, pp. 1733–1749, 2008.
- [26] D. Teleaga, *A finite volume particle method for conservation laws*. PhD Thesis, University of Kaiserslautern, 2005.
- [27] S. Koshizuka, H. Tamako, and Y. Oka, “A particle method for incompressible viscous flow with fluid fragmentation,” *Computational Fluid Dynamics Journal*, vol. 4, no. 1, pp. 29–46, 1995.
- [28] G. Cottet and P. Koumoutsakos, *Vortex methods: theory and practice*. Cambridge University Press, 2000.

- [29] D. Sulusky, Z. Chen, and H. L. Schreyer, “A particle method for historydependent materials,” *Computational Methods and Applications in Mechanics and Engineering*, vol. 118, pp. 179–195, 1994.
- [30] S. G. Bardenhagen and E. M. Kober, “The generalized interpolation material point method,” *Tech Science Press*, vol. 5, no. 6, pp. 477–495, 2004.
- [31] H. Lin and S. N. Atluri, “Meshless local petrov galerkin (mlpg) method for convection-diffusion problems,” *Computer Modeling in Engineering and Sciences*, vol. 1, no. 2, pp. 45–60, 2000.
- [32] T. Belytschko, Y. Krongauz, D. Organ, M. Fleming, and P. Krysl, “Meshless methods: An overview and recent developments,” *Computer Methods in Applied Mechanics and Engineering*, vol. 139, pp. 3–47, 1996.
- [33] S. Li and W. K. Liu, *Meshfree Particle Methods*. Springer, 2004.
- [34] L. B. Lucy, “A numerical approach to the testing of the fission hypothesis,” *Astronomical Journal*, vol. 82, pp. 1013–1024, 1977.
- [35] 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.
- [36] J. J. Monaghan, “Simulating free surface flows with sph,” *Journal of Computational Physics*, vol. 110, pp. 399–406, 1994.

- [37] 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.
- [38] M. Pastor, B. Haddad, G. Sorbino, S. Cuome, and V. Drempetic, “A depth-integrated, coupled sph model for flow-like landslides and related phenomena,” *International Journal of Numerical and Analytical Methods in Geomechanics*, vol. 33, pp. 143–172, 2009.
- [39] R. Vacondio, B. D. Rogers, P. K. Stansby, and P. Mignosa, “Sph modeling of shallow flow with open boundaries for practical flood simulation,” *Journal of Hydraulic Engineering*, vol. 138, pp. 530–541, 2012.
- [40] D. Violeau, *Fluid Mechanics and the Sph Method: Theory and Applications*. Oxford University Press, 2012.
- [41] J. J. Monaghan and A. Kos, “Solitary waves on a cretan beach,” *Journal of Waterway, Port, Coastal, and Ocean Engineering*, pp. 145–154, 1999.
- [42] M. Landrini, A. Colagrossi, M. Greco, and M. P. Tulin, “Gridless simulations of splashing processes and near-shore bore propagation,” *Journal of Fluid Mechanics*, vol. 591, pp. 183–213, 2007.
- [43] R. A. Dalrymple and B. D. Rogers, “Numerical modeling of water waves with the sph method, coastal engineering,” *Coastal Engineering*, vol. 5, no. 2/3, pp. 141–147, 2006.

- [44] S. J. Cummins and M. Rudman, “An SPH projection method,” *Journal of Computational Physics*, vol. 152, pp. 584–607, 1999.
- [45] A. J. Chorin, “Numerical solutions of the navierstokes equations,” *Mathematics of Computation*, vol. 22, no. 104, pp. 745–762, 2006.
- [46] S. Shao and E. Y. M. Lo, “Incompressible sph method for simulating new-tonian and non-newtonian flows with a free surface,” *Advances in Water Resources*, vol. 26, no. 7, pp. 787–800, 2003.
- [47] S. D. Shao, C. Ji, D. Graham, D. Reeve, P. James, and A. Chadwick, “Simulation of wave overtopping by an incompressible sph model,” *Coastal Engineering*, vol. 53, no. 9, pp. 723–735, 2006.
- [48] 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. 1, pp. 39–63, 2004.
- [49] S. D. Shao and H. Gotoh, “Simulation coupled motion of progressive wave and floating curtain wall by sph-les model,” *Coastal Engineering Journal*, vol. 46, no. 2, pp. 171–202, 2004.
- [50] J. Vila, “On particle weighted methods and smooth particle hydrodynamics,” *Mathematical Models and Methods in Applied Sciences*, vol. 9, no. 2, pp. 161–210, 1999.

- [51] 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, pp. 532–549, 2010.
- [52] J. J. Monaghan and A. Kocharyan, “Sph simulation of multi-phase flow,” *Computer Physics Commununication*, vol. 87, p. 225, 1995.
- [53] A. Colagrossi and M. Landrini, “Numerical simulation of interfacial flows by smoothed particle hydrodynamics,” *Journal of Computational Physics*, vol. 191, pp. 448–475, 2003.
- [54] X. Y. Hu and N. A. Adams, “A multi-phase sph method for macroscopic and mesoscopic flows,” *Journal of Computational Physics*, vol. 213, no. 2, pp. 844–861, 2006.
- [55] X. Y. Hu and N. A. Adams, “A constant-density approach for incompressible multi-phase sph,” *Journal of Computational Physics*, vol. 228, no. 6, pp. 2082–2091, 2009.
- [56] N. Grenier, M. Antuono, A. Colagrossi, D. Le Touzè, and B. Alessandrini, “An hamiltonian interface sph formulation for multi-fluid and free surface flows,” *Journal of Computational Physics*, vol. 228, pp. 8380–8393, 2009.
- [57] W. C. Welton, “Two-dimensional pdf/sph simulations of compressible turbulent flows,” *Journal of Computational Physics*, vol. 139, pp. 410–443, 1998.
- [58] W. C. Welton, “Turbulence simulation and multiple scale subgrid models,” *Computational Mechanics*, vol. 25, pp. 117–136, 2000.

- [59] H. Gotoh, T. Shibahara, and T. Sakai, “Sub-particle-scale turbulence model for the mps method -lagrangian flow model for hydraulic engineering,” *Computational Fluid Dynamics Journal*, vol. 9, no. 4, pp. 339–347, 2001.
- [60] J. J. Monaghan, “Sph compressible turbulence,” *Monthly Notices of the Royal Astronomical Society*, vol. 335, no. 3, pp. 843–852, 2002.
- [61] J. J. Monaghan, “An sph turbulence model,” Proceeding of 4th SPHERIC International workshop, (Nantes, France), 2009.
- [62] D. Violeau and R. Issa, “Numerical modelling of complex turbulent free surface flows with the sph lagrangian method: an overview,” *International Journal for Numerical Methods in Fluids*, vol. 53, no. 2, pp. 277–304, 2007.
- [63] R. A. Dalrymple and O. Knio, “Sph modelling of water waves,” Proceedings of Coastal Dynamics, (Lund, Sweden), pp. 779–787, 2001.
- [64] M. Liu, J. Shao, and J. Chang, “On the treatment of solid boundary in smoothed particle hydrodynamics,” *Science China Technological Sciences*, vol. 55, no. 1, pp. 244–254, 2012.
- [65] M. Ferrand, D. R. Laurence, B. D. Rogers, D. Violeau, and C. Kassiotis, “Unified semi-analytical wall boundary conditions for inviscid, laminar or turbulent flows in the meshless SPH method,” *International Journal for Numerical Methods in Fluids*, vol. 71, pp. 446–472, 2013.

- [66] M. De Leffe, D. Le Touzé, and B. Alessandrini, “Normal flux method at the boundary for sph,” Proceeding of 4th SPHERIC International workshop, (Nantes, France), pp. 150–156, 2009.
- [67] M. Lastiwka, M. Basa, and N. Quinlan, “Permeable and non-reflecting boundary conditions in SPH,” *International Journal for Numerical Methods in Fluids*, vol. 61, no. 7, pp. 709–724, 2008.
- [68] S. Attaway, M. W. Heinstein, and J. Swegle, “Coupling of smooth particle hydrodynamics with the finite element method,” *Nuclear Engineering Design*, vol. 150, pp. 199–205, 2010.
- [69] G. R. Johnson, “Linking of lagrangian particle methods to standard finite element methods for high velocity impact computations,” *Nuclear Engineering and Design*, vol. 150, no. 2-3, pp. 265–274, 1994.
- [70] G. R. Johnson, R. A. Stryk, and S. R. Beissel, “SPH for high velocity impact computations,” *Computer Methods in Applied Mechanics and Engineering*, vol. 139, pp. 347–373, 1996.
- [71] M. Sauer, “Simulation of high velocity impact in fluid-filled containers using finite elements with adaptive coupling to smoothed particle hydrodynamics,” *International Journal of Impact Engineering*, vol. 38, pp. 511–520, 2011.

- [72] T. D. Vuyst, R. Vignjevic, and J. C. Campbell, “Coupling between meshless and finite element methods,” *International Journal of Impact Engineering*, vol. 31, pp. 1054–1064, 2005.
- [73] G. Fourey, G. Oger, D. L. Touzé, and B. Alessandrini, “Violent fluid-structure interaction simulations using a coupled SPH/FEM model,” *IOP Conference Series: Materials Science and Engineering*, vol. 10, 2010.
- [74] J. J. Monaghan, “Smoothed particle hydrodynamics,” *Annual Review of Astronomy and Astrophysics*, vol. 30, pp. 543–574, 1992.
- [75] J. J. Monaghan and J. C. Lattanzio, “A refined particle method for astrophysical problems,” *Astronomy and Astrophysics*, vol. 149, pp. 135–143, 1985.
- [76] A. J. Crespo, J. M. Domínguez, B. D. Rogers, M. Gómez-Gesteira, S. Longshaw, R. Canelas, R. Vacondio, A. Barreiro, and O. García-Feal, “Dualsphysics: open-source parallel CFD solver on Smoothed Particle Hydrodynamics (SPH),” *Computer Physics Communications*, vol. 187, pp. 204–216, 2015.
- [77] H. Wendland, “Piecewise polynomial, positive definite and compactly supported radial functions of minimal degree,” *Advances in Computational Mathematics*, vol. 4, pp. 389–396, 1995.
- [78] J. P. Morris, P. J. Fox, and Y. Zhu, “Modeling low reynolds number incompressible flows using SPH,” *Journal of Computational Physics*, vol. 136, pp. 214–226, 1997.

- [79] J. J. Monaghan and J. B. Kajtar, “SPH particle boundary forces for arbitrary boundaries,” *Computer Physics Communications*, vol. 180, pp. 1811–1820, 2009.
- [80] P. K. Kundu and I. M. Cohen, *Fluid Mechanics*. Academic Press, 4th ed., 2008.
- [81] J. J. Monaghan and A. Rafiee, “A simple SPH algorithm for multi-fluid flow with high density ratios,” *International Journal for Numerical Methods in Fluids*, vol. 71, pp. 537–561, 2013.
- [82] R. Fatehi and M. T. Manzari, “A consistent and fast weakly compressible Smoothed Particle Hydrodynamics with a new wall boundary condition,” *International Journal for Numerical Methods in Fluids*, vol. 68, no. 7, pp. 905–921, 2012.
- [83] S. Watkins, A. Bhattacharjee, N. Francis, J. Turner, and A. Whitworth, “A new prescription for viscosity in smoothed particle hydrodynamics,” *Astronomy and Astrophysics*, vol. 119, pp. 177–187, 1996.
- [84] J. Bonet and T. Lok, “Variational and momentum preservation aspects of smooth particle hydrodynamic formulations,” *Computer Methods in Applied Mechanics and Engineering*, vol. 1, pp. 97–115, 1999.
- [85] R. Fatehi, M. Fayazbakhsh, and M. T. Manzari, “On discretization of second-order derivatives in smoothed particle hydrodynamics,” *International Journal Natural and Applied Sciences*, vol. 3, no. 1, pp. 50–53, 2009.

- [86] J. J. Monaghan, “Smoothed Particle Hydrodynamics and Its Diverse Applications,” *Annual Review of Fluid Mechanics*, vol. 44, pp. 323–346, 2012.
- [87] P. W. Cleary, “Modeling confined multi-material heat and mass flows using sph,” *Applied Mathematical Modelling*, vol. 2, pp. 981–993, 1998.
- [88] 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*, pp. 8417–8436, 1999.
- [89] A. J. Crespo, M. Gómez-Gesteira, and R. A. Dalrymple, “Boundary conditions generated by dynamic particles in SPH methods,” *CMC: Computers, Materials, & Continua*, vol. 5, no. 3, pp. 173–184, 2007.
- [90] A. Héault, G. Bilotta, and R. A. Dalrymple, “SPH on GPU with CUDA,” *Journal of Hydraulic Research*, vol. 48, pp. 74–79, 2010.
- [91] “Isph - cross-platform library for mesh-free particle based simulation and visualization of incompressible flows using SPH methods..” URL <http://isph.sourceforge.net/>.
- [92] “Aquagpusph - free SPH software accelerated with OpenCL..” URL <http://canal.etsin.upm.es/aquagpusph/>.
- [93] J. C. Marongiu, J. Leduc, and M. Schaller, “GPU implementation of a SPH-ALE fluid dynamics solver,” Proceedings of the 6<sup>th</sup> international SPHERIC workshop, (Hamburg, Germany), June 2011.

- [94] 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, pp. 617–627, 2013.
- [95] NVIDIA, “NVIDIA CUDA Toolkit Documentation - v6.5,” URL <https://developer.nvidia.com/resources>, August 2011.
- [96] J. M. Domínguez, A. J. Crespo, M. Gómez-Gesteira, and J. C. Marongiu, “Neighbor lists in smoothed particle hydrodynamics,” *International Journal for Numerical Methods in Fluids*, vol. 67, pp. 2026–2042, 2011.
- [97] C. Farhat, A. Rallu, K. Wang, and T. Belytschko, “Robust and provably second-order explicit-explicit and implicit-explicit staggered time-integrators for highly nonlinear compressible fluid-structure interaction problems,” *International Journal for Numerical Methods in Engineering*, vol. 84, pp. 73–107, 2010.
- [98] N. Akinci, J. Cornelis, G. Akinci, and M. Teschner, “Coupling elastic solids with smoothed particle hydrodynamics fluids,” *Computer Animation and Virtual Worlds*, vol. 24, pp. 195–203, 2013.
- [99] G. K. Batchelor, *An introduction to fluid dynamics*. Cambridge University Press, 1967.
- [100] M. Gómez-Gesteira, B. Rogers, R. Dalrymple, A. Crespo, and M. Narayanaswamy, “User guide for the sphysics code v2.0,” 2010.

- [101] K. M. T. Kleefsman, G. Fekken, A. E. P. Veldman, B. Iwanowski, and B. Buchner, “A volume-of-fluid based simulation method for wave impact problems,” *Journal of Computational Physics*, vol. 206, pp. 363–393, 2005.
- [102] W. Pastoor, T. Klungseth-Ostvold, E. Byklum, and S. Valsgard, “Sloshing load and response in LNG carriers for new designs, new operations and new trades,” tech. rep., Det Norske Veritas Technical Paper, 2005.
- [103] D. Peregrine, “Water-wave impact on walls,” *Annual Review of Fluid Mechanics*, vol. 35, pp. 23–43, 2003.
- [104] M. Hattori, A. Arami, and T. Yui, “Wave impact pressure on vertical walls under breaking waves of various types,” *Coastal Engineering*, vol. 22, pp. 79–114, 1994.
- [105] R. Bass, E. Bowles, R. Trudell, J. Navickas, J. Peck, N. Endo, and B. Pots, “Modeling criteria for scaled lng sloshing experiments,” *Transactions of the ASME*, vol. 107, pp. 272–280, 1985.
- [106] L. Delorme, A. Colagrossi, A. Souto-Iglesias, R. Zamora-Rodríguez, and E. Botía-Vera, “A set of canonical problems in sloshing, part i: Pressure field in forced roll-comparision between experimental results and sph,” *Ocean Engineering*, vol. 36, pp. 168–178, 2009.
- [107] A. Souto-Iglesias and E. Botía-Vera, “Wave impact problem: SPHERIC benchmark test case,” tech. rep., Model Basin Research Group (CEHINAV), Naval architecture department (ETSIN), Technical University of Madrid (UPM), Madrid Spain, 1975.

- [108] C. Antoci, M. Gallati, and S. Sibilla, “Numerical simulation of fluid-structure interaction by SPH,” *Computers and Structures*, vol. 85, pp. 879–890, 2007.
- [109] A. Souto-Iglesias, S. R. Idelsohn, J. Martì, R. Zamora-Rodríguez, and E. Oñate, “Modeling of free surface flows with elastic bodies interactions,” 27th Symposium on Naval Hydrodynamics, 2008.
- [110] S. Z. Zalek and L. J. Doctors, “Experimental study of the resistance of surface-effect-ship seals,” 28th Symposium on Naval Hydrodynamics, (Pasadena, California), 2010.
- [111] S. Marrone, *Enhanced SPH modeling of free-surface flows with large deformations*. PhD Thesis, Università Di Roma, 2011.
- [112] J. M. Domínguez, A. J. Crespo, D. Valdez-Balderas, and B. C. Rogers, “New multi-GPU implementation for smoothed particle hydrodynamics on heterogeneous clusters,” *Computer Physics Communications*, vol. 184, pp. 1848–1860, 2013.
- [113] R. Vacondio, B. D. Rogers, P. K. Stansby, P. Mignosa, and J. Feldman, “Variable resolution for sph: A dynamic particle coalescing and splitting scheme,” *Computational Methods in Applied Mechanics and Engineering*, vol. 256, pp. 132–148, 2013.
- [114] W. L. Oberkampf and C. J. Roy, *Verification and Validation in Scientific Computing*. Cambridge Univeristy Press, 2010.