

APPLIED COMPUTATIONAL GEOMETRY FOR SMOOTHED PARTICLE HYDRODYNAMICS

by

Van S. Jones
A Dissertation
Submitted to the
Graduate Faculty
of
George Mason University
in Partial Fulfillment of
The Requirements for the Degree
of
Doctor of Philosophy
Information Technology

Committee:

_____ Dr. Leigh McCue, Dissertation Director
_____ Dr. Chi Yang, Committee Member
_____ Dr. Jeffrey Moran, Committee Member
_____ Dr. Celso Ferreira, Committee Member
_____ Dr. Deborah Goodings, Associate Dean

Date: _____ Summer Semester 2022
George Mason University
Fairfax, VA

Applied Computational Geometry for Smoothed Particle Hydrodynamics

A dissertation submitted in partial fulfillment of the requirements for the degree of
Doctor of Philosophy at George Mason University

By

Van S. Jones
Bachelor of Science
Virginia Polytechnic Institute and State University, 2008

Director: Dr. Leigh McCue, Associate Professor
Department of Mechanical Engineering

Summer Semester 2022
George Mason University
Fairfax, VA

Copyright © 2022 by Van S. Jones
All Rights Reserved

Dedication

I dedicate this dissertation to the many people who have made this work and my journey to this point possible. My thanks to Jonathan Bender, my high school math teacher who shared with me the foundations of math and geometry on which this thesis is built. I want to thank my advisor Leigh McCue for guiding me through the path of graduate study, and for believing in me even when my faith in my own ability faltered. And above all, I want to thank my mother, for encouraging and supporting me throughout my education and life.

Acknowledgments

I would like to thank the following people and institutions who made this work possible. My gratitude to The Office of Naval Research, Kelly Cooper, Pat Purtell, Ki-Han Kim, George Mason University, and the numerous professors and educators who have aided me on this path.

Table of Contents

	Page
List of Figures	vii
List of Abbreviations and Symbols	ix
Abstract	x
1 Introduction	1
1.1 Motivation	1
1.2 Thesis Structure	1
1.3 Smoothed Particle Hydrodynamics Overview	2
1.3.1 Smoothed Particle Hydrodynamics for Liquid Simulation	6
1.3.2 Boundary Conditions for SPH	9
1.3.3 Weaknesses of Smoothed Particle Hydrodynamics	10
1.3.4 SPH Validation Cases	13
2 Underwater Deflagration Experiment	18
2.1 Experiment Overview	18
2.2 Materials and Methods	20
2.3 Results	23
2.3.1 Rigid Plate Target	24
2.3.2 Flexible Membrane Target	28
2.4 Discussion	29
2.5 Summary	30
3 Underwater Deflagration Simulation	38
3.1 Simulation Results	39
4 Boundary Deficiency Correction	45
4.1 SPH Boundary Behavior	45
4.2 Acceleration Boundary Deficiency Correction	48
4.3 Summary	53
5 Spatial Filtering for Variable Spatial Resolution	54
5.1 Overview of Variable Spatial Resolution	54
5.2 Spatial Filtering	55

5.3	Expanded Circle Filter	57
5.4	Additional Notes and Future Work	65
6	Interdomain Property Transfer for Variable Resolution	67
6.1	Domain Discretization	67
6.2	Dynamic Domain Adaptation	69
6.3	Iterative Conserved Property Transfer	70
7	Voronoi Tessellation	75
7.1	Voronoi Tessellation Overview	75
7.2	Finite Radius Voronoi	76
7.3	Future Work	78
8	Conclusions	81
	Bibliography	83

List of Figures

Figure	Page
1.1 Cubic Spline Kernel Function	4
1.2 SPH Particle Support Domain	5
1.3 SPH Acceleration - Pressure Term	10
1.4 Boundary Deficiency	11
1.5 Couette Flow SPH Simulation	14
1.6 Density Driven Flow: Rising-Bubble - Detail	15
1.7 Rising-Bubble Simulation (current work left, Yang right)	16
2.1 Deflagration Targets	21
2.2 Experiment Assembly	22
2.3 Flexible Membrane	23
2.4 Stereoscopic Camera View of Flexible Membrane Target	24
2.5 Deflagration and Bubble History (Run 4)	31
2.6 Bubble Radius	32
2.7 Bubble Radius (Aligned at Maximum Diameter) and Deflagration Stages, I: Combustion in Cylinder, II: Combustion in the Bubble, III: Inertial Expansion, IV: Collapse, V: Post-Collapse	32
2.8 Deflagration Stage Transitions	33
2.9 Cylinder Pressure	34
2.10 Bottom Plate Pressure	34
2.11 Bubble Shape at Maximum Expansion	35
2.12 Flame-Front Distance From Origin	35
2.13 Deflagration Impinging on Flexible Membrane Target	36
2.14 Membrane Displacement & Rigid Plate Pressure	36
2.15 Qualitative DIC Visualization of 2D Planar Membrane Displacement	37
2.16 Combustion Detail	37
3.1 Deflagration Simulation Results (Left: Pressure, Right: Density)	40
3.1 Deflagration Simulation Results (Left: Pressure, Right: Density)	41

3.2	Simulation and Experiment Cylinder Pressure Plotted Relative to Peak Cylinder Pressure	42
3.3	Simulation and Experiment Bottom Plate Pressure Plotted Relative to Peak Cylinder Pressure	42
4.1	Boundary Deficiency in Acceleration (influence radius=1, length is unitless)	46
4.2	Clumping of Virtual Particles due to Deformation	47
4.3	Clumping of Virtual Particles due to Deformation (length is unitless)	48
4.4	2D Boundary Correction Weight Blending Visualization (influence radius=0.775, boundary spacing=0.25, boundary radius of curvature=1)	50
4.5	Rectangular Tank Boundary Correction Test Case (influence radius=0.5 spacing= $r_{influence}/10$)	51
4.6	Elliptical Tank Boundary Correction Test Case (influence radius=0.5 spacing= $r_{influence}/10$)	52
5.1	Ordered and Disordered Particles and Clumping Due to Tensile Instability .	56
5.2	Expanded Circle Filter Example	59
5.3	Expanded Circle Filter Spatial Conditioning	60
5.4	Expanded Circle Filter Spatial Conditioning - Gradient of Particle Number Density	61
5.5	Expanded Circle Filter Detailed Iterations	62
6.1	Example SPH Domain Discretizations	67
6.2	Particles Generated using Uniform Rectangular Distributions	69
6.3	Particle Generation from a Mesh	69
6.4	Initial and Blank Domain Discretizations	72
6.5	Continuous Property Target U Interpolated to Domain B	72
6.6	Expanded Circle Filter Spatial Conditioning	73
6.7	L_2 Norm of Density Target Error	74
7.1	Voronoi Tessellation of Space from Points	76
7.2	Finite Radius Voronoi Algorithm	77
7.3	Finite Radius Voronoi Tessellation from Points	78
7.4	Finite Radius Voronoi with Boundaries (section centroids shown)	79

List of Abbreviations and Symbols

Smoothed Particle Hydrodynamics	SPH
Computational Fluid Dynamics	CFD
Support Domain	Ω
Dirac Delta Function	δ
Kernel Function	W
Smoothing Length	h
Number of Particles in Support Domain	N
Vector Distance from Particle j to Particle i	\vec{x}_{ij}
Kernel Gradient	$\frac{\partial W_{ij}}{\partial \vec{x}_{ij}}$
Fluid Density	ρ
Stress Tensor	σ
Shear Stress Tensor	τ
Dynamic Viscosity	μ
Pressure	p
Velocity	v
Mass	m
Volume	V

Abstract

APPLIED COMPUTATIONAL GEOMETRY FOR SMOOTHED PARTICLE HYDRODYNAMICS

Van S. Jones, PhD

George Mason University, 2022

Dissertation Director: Dr. Leigh McCue

Smoothed Particle Hydrodynamics (SPH) is a Computational Fluid Dynamics method that has significantly increased in capability in recent years. The advent of general-purpose computing on graphics processing units (GPGPU) has helped to enable large-scale SPH simulation outside of supercomputer environments. However variable resolution methods for free-surface and multiphase SPH simulations remains limited. The lack of robust and efficient variable resolution methods for free-surface and multiphase simulations limits the potential computational efficiency and accuracy of SPH when used to model these flows. Recent work has made steps towards solving these issues and has improved the capability of SPH in these areas. This thesis seeks to add to these growing capabilities by presenting new computational geometry algorithms to improve the accuracy and efficiency of SPH simulations. These methods include a boundary condition for flexible walls, a spatial filter to reduce particle disorder, and a modified Voronoi tessellation method for determining particle volumes. These methods provide a basis for future work to increase the computational efficiency of SPH and will facilitate further improvements to the accuracy and scalability of SPH simulations involving free-surface and multiphase flows.

Chapter 1: Introduction

1.1 Motivation

Smoothed particle hydrodynamics (SPH) is a Lagrangian computational fluid dynamics (CFD) method. It was first developed by Gingold, Monaghan, and Lucy for the simulation of astrophysics phenomena [1] [2]. It was later extended to applications in fluid dynamics by Monaghan [3] and was adapted to include structural simulation capability by Libersky and Petschek [4] and others. Since its development, SPH has gained widespread use in astrophysics simulations [5]. It is utilized in high velocity impact (HVI) simulations [6] [7] and high explosive simulations [8]. The method is widely used for computer-generated imagery (CGI) and in realtime simulation of fluids [9]. It is also well suited to mixed media simulations such as sediment transfer [10].

However, despite Smoothed Particle Hydrodynamics' use for many specialized engineering and scientific applications, the utilization of SPH for general purpose single-phase and multiphase CFD simulations remains limited compared to Finite Volume Methods (FVM) and Finite Element Methods (FEM) [11]. With the primary goal of improving SPH for multiphase simulations, this thesis seeks to identify areas for potential improvement to existing SPH methods when applied to multiphase flows. Then new methods to enhance SPH capabilities in these identified areas are proposed.

1.2 Thesis Structure

The primary SPH issues addressed in this thesis are thin and flexible boundaries, particle disorder and clumping, and domain property transfer to facilitate dynamic resolution adaptation. These issues were selected through work done to identify areas for potential

improvement to SPH. This was done by analyzing a multiphase underwater deflagration test case both experimentally and numerically. This test case was designed to include multiple flow features which are difficult to simulate. These features include large pressure variations, a multiphase interface, large discontinuities in bulk deformation across interfaces, and fluid-structure interaction. By analyzing such a complicated fluid flow, specific deficiencies in the SPH method become apparent. With deficiencies in the base SPH method identified, new computational geometry algorithms are proposed to address the identified deficiencies.

This thesis is broken into several distinct sections. Chapter 1 provides an overview of Smoothed Particle Hydrodynamics and presents validation case results using an SPH code developed for this work. Chapter 2 details the experimental design and results of the multiphase test case. Chapter 3 outlines numerical simulation results of the multiphase test case and discusses identified deficiencies in the SPH methods used. Chapter 4 details a new numerical method to address thin or deformable boundaries without the utilization of ghost particles. Chapter 5 presents a spatial filtering method that reduces particle disorder and clumping and has potential application to domain resolution adaptation. Chapter 6 presents an iterative scheme to transfer conserved properties between two domain discretizations. Chapter 7 outlines a new finite-radius Voronoi algorithm that supports the iterative scheme introduced in chapter 6 by determining particle volumes for arbitrary particle distributions.

1.3 Smoothed Particle Hydrodynamics Overview

The core numerical mechanism of SPH is the approximation of a field value through a finite weighted integration of the surrounding field. This is based on the exact reproduction of a field value by the integration of the product of the Dirac delta function and a field variable (eq. 1.1). In this exact formulation, the Dirac delta function evaluates to unity at $x = x'$ and is zero for all other points. This yields an integration that exactly reproduces the value of a function at any point in space.

$$f(x) = \int_{-\infty}^{\infty} \delta(x - x') f(x') dx' \quad (1.1)$$

This equivalency by itself is of little use. However, if an approximation for the discontinuous Dirac delta function is introduced, then a useful approximation of a function may be obtained (eq. 1.2).

$$f(x) \approx \int_{\Omega} W(x - x') f(x') dx' \quad (1.2)$$

Here the Dirac delta function is replaced by a kernel function (W) which possesses similar properties to the delta function but is non-zero over a finite volume Ω . A kernel function has units of inverse volume and must be chosen such that in the limit as Ω approaches zero the kernel function approaches the Dirac delta function. This ensures that the SPH approximation will, in the limiting sense, regain the exact function equivalency of eq. 1.1. One of the initial kernel functions developed by Monaghan and Lattanzio is the Piecewise Cubic Spline kernel [12] (Figure 1.1). The Cubic Spline smoothing function displays Gaussian behavior which is a common characteristic, but not a requirement, of smoothing functions.

Finally, in order to transform the continuous definition of eq. 1.2 into a discrete form, the integration is discretized into a sum of volume-weighted function values, yielding the final form of the standard SPH particle representation of a field function (eq. 1.3). This reduces to eq. 1.4 when solving for field properties at a particle's position.

$$f(x) \approx \sum_{\Omega} W(x_j - x) f(x_j) V_j \quad (1.3)$$

$$f_i \approx \sum_{j=1}^N W(x_j - x_i) f_j V_j \quad (1.4)$$

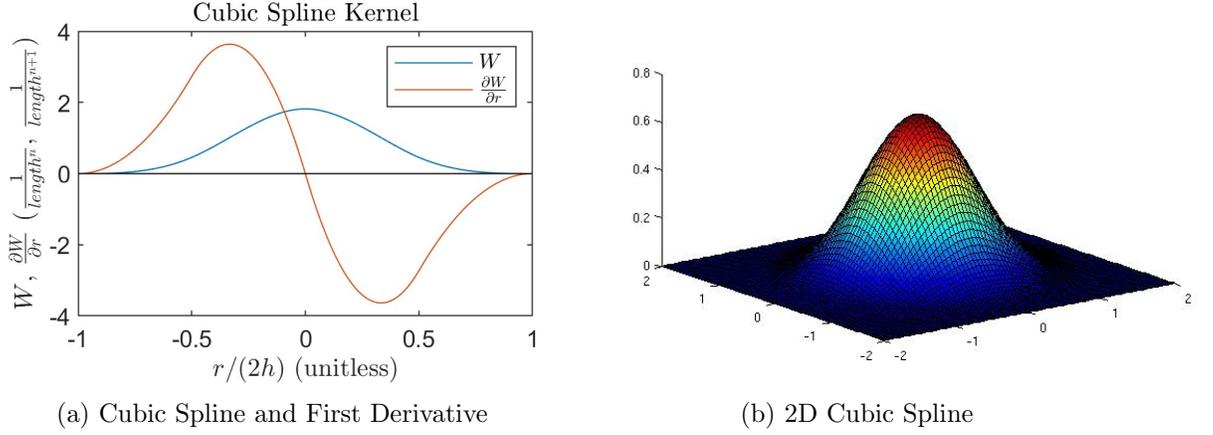


Figure 1.1: Cubic Spline Kernel Function

In both eq. 1.3 and eq. 1.4 the support domain Ω is the set of all particles j which lie within the non-zero region of the kernel function of a subject particle i and V_j is the volume of particle j . The scale of the non-zero support region about a particle is a tunable parameter and is typically presented in terms of a smoothing length h . The radius of the support region is then defined as a multiple of the length, where this multiple is dependent upon the selected kernel function. In this thesis, the radius which defines a particle's support domain will be referred to as the influence radius, which is defined as $2h$ for the kernel used in all simulation results. Figure 1.2 illustrates an example support domain for a particle in a random particle field. In this thesis, a particle that is referenced for property calculations is referred to as a *subject particle* and interacting particles are designated as *support particles*.

To obtain spatial derivatives of field functions using SPH, a gradient may be substituted into the base SPH formulation (eq. 1.2). This leads to equation 1.5. Then, using a chain rule identity (eq. 1.6), the integrand may be modified to obtain equation 1.7.

$$\nabla \cdot f(x) \approx \int_{\Omega} [\nabla \cdot f(x')] W(x - x') dx' \quad (1.5)$$

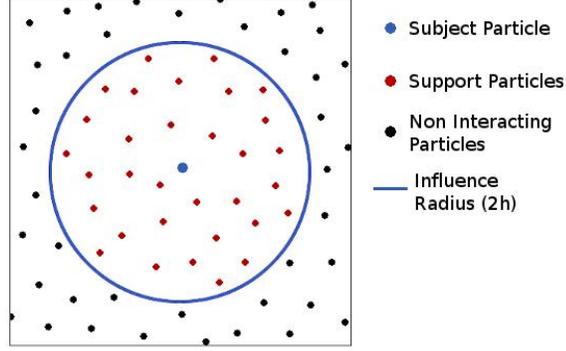


Figure 1.2: SPH Particle Support Domain

$$[\nabla \cdot f(x')] W(x - x') = \nabla \cdot [f(x')W(x - x')] - f(x') \cdot \nabla W(x - x') \quad (1.6)$$

$$\nabla \cdot f(x) \approx \int_{\Omega} \nabla \cdot [f(x')W(x - x')] dx' - \int_{\Omega} f(x') \cdot \nabla W(x - x') dx' \quad (1.7)$$

By the divergence theorem (eq. 1.8), the first term of eq. 1.7 can be transformed into a surface integral over the integrated volume in order to obtain equation 1.9. For integral volumes which are defined such that $\|x - x'\| \geq 2h$ the kernel function in the first term of eq. 1.9 will be zero over the surface. It follows that in all cases where an integral volume extends to at least the limit of the kernel function eq. 1.9 will reduce to equation 1.10.

$$\int (\nabla \cdot f(x)) dx = \oint (f(x) \cdot \hat{n}) ds \quad (1.8)$$

$$\nabla \cdot f(x) \approx \oint_{\Omega} \nabla \cdot [f(x')W(x - x')] dx' - \int_{\Omega} f(x') \cdot \nabla W(x - x') dx' \quad (1.9)$$

$$\nabla \cdot f(x) \approx - \int_{\Omega} f(x') \cdot \nabla W(x - x') dx' \quad (1.10)$$

However for cases in which the volume integral does not extend to the extent of the kernel function the simplification used to obtain eq. 1.10 may not be performed. Cases in which the integral is truncated in this manner may occur at fluid surfaces or boundaries. These cases are said to be integral deficient and corrections must be made to approximate the non-zero additional term in eq. 1.9. Integral deficiency is further discussed in section 1.3.3.

To obtain the SPH formulation for the gradient of a field function, the continuous integral of eq. 1.10 must be discretized into finite form. Replacing the continuous volume integral with a finite summation and defining $\vec{x}_{ij} = \vec{x}_i - \vec{x}_j$ and $\frac{\partial W_{ij}}{\partial \vec{x}_{ij}} = \frac{\partial W}{\partial r} (||\vec{x}_i - \vec{x}_j||) \frac{\vec{x}_{ij}}{||\vec{x}_{ij}||}$ leads to equation 1.11. This is the standard, non-symmetrized, SPH form for the gradient of a field function.

$$(\nabla \cdot f(x))_i = - \sum_{j=1}^N f_j \cdot \frac{\partial W_{ij}}{\partial \vec{x}_{ij}} V_j \quad (1.11)$$

1.3.1 Smoothed Particle Hydrodynamics for Liquid Simulation

The SPH method can be adapted to fluid dynamics equations of motion in a number of ways. The most common approach for applying SPH to fluid dynamics is the Weakly Compressible Smoothed Particle Hydrodynamics (WCSPH) method. In WCSPH fluids typically considered nearly incompressible are allowed a small degree of compressibility. This yields a finite speed of sound for the fluid and changes the characteristic of the governing equations to allow for uncoupled solving of particle properties. Particle pressures for water are derived from an empirical equation of state. Parameters of the equation of state determine the numerical speed of sound in that medium. The Cole Equation of State (eq. 1.12) is used

in this thesis. In 1.12 β is a compressibility constant defined as $\beta = \frac{c\rho_0}{\gamma}$, c is the speed of sound in the medium, ρ_0 is a reference density, and γ is an empirically determined constant set to 7 for water in simulations used in this thesis.

$$P = \beta\left(\frac{\rho}{\rho_0} - 1\right)^\gamma + P_0 \quad (1.12)$$

This and similar equations of state are often erroneously referred to as the Tait Equation of State. Tait proposed an empirical model for related water density and pressure in 1888 [13]. However, work by Murnaghan [14], Cole [15], MacDonald [16], and others have since added to and refined Tait’s original work on the subject.

Methods of calculating SPH particle density vary, but most techniques fall under two principle approaches. These are the summation-density and continuity-density formulations. In the summation density formulation, density is directly determined by summing the kernel-weighted densities of all support domain particles (eq. 1.13), where $W_{ij} = W(\|\vec{x}_i - \vec{x}_j\|)$ and V_j is the volume of particle j . Substituting the definition of density (eq. 1.14) into eq. 1.13 yields eq. 1.15 and transforms the formula into a mass summation over a volume (as the kernel function has units of inverse volume).

$$\rho_i = \sum_{\Omega} W_{ij}\rho_j V_j \quad (1.13)$$

$$\rho \equiv \frac{m}{V} \quad (1.14)$$

$$\rho_i = \sum_{\Omega} W_{ij}m_j \quad (1.15)$$

The continuity density formulation is obtained by discretizing the continuity equation (eq. 1.16). The SPH discretized form of the divergence operator is shown in eq. 1.17. Substituting eq. 1.17 into eq. 1.16 yields eq. 1.18. This can be simplified by eq. 1.14 and

the definition of $\vec{v}_{ij} := \vec{v}_i - \vec{v}_j$ to obtain equation 1.19, the SPH discretized form of the continuity equation [8].

$$\frac{\partial \rho}{\partial t} = -\nabla \cdot (\rho \vec{v}) \quad (1.16)$$

$$\nabla \cdot \vec{u}_i = \sum_{j=1}^N (\vec{u}_i - \vec{u}_j) \frac{\partial W_{ij}}{\partial \vec{x}_{ij}} V_j \quad (1.17)$$

$$\frac{D\rho_i}{Dt} = -\sum_{j=1}^N \rho_j (\vec{v}_j - \vec{v}_i) \frac{\partial W_{ij}}{\partial \vec{x}_{ij}} V_j \quad (1.18)$$

$$\frac{D\rho_i}{Dt} = \sum_{j=1}^N m_j \vec{v}_{ij} \frac{\partial W_{ij}}{\partial \vec{x}_{ij}} \quad (1.19)$$

The SPH momentum equation can be obtained from the Lagrangian form of the Cauchy Momentum Equation (eq. 1.20). The general stress tensor in the Cauchy Momentum Equation can then be separated into pressure forces, viscous forces, and body forces to obtain Equation 1.21. Equation 1.22 shows a symmetrized form of the SPH gradient formulation. For particle interactions involving conserved fluid properties (such as momentum), it is desirable that particle interactions are symmetrized. By enforcing symmetric particle interactions in the momentum equation any loss of momentum from one particle in a particle-particle interaction will be accompanied by an equivalent gain of momentum for the other particle in the interaction. This ensures momentum transfer between particles is precisely conserved. Applying eq. 1.22 to the pressure term in eq. 1.21 yields the SPH formulation for the pressure term for the discretized momentum equation [17]. To complete the momentum equation an approximation of viscous force must be determined. Multiple viscous stress models exist for determining shear forces for the momentum equation.

Cleary's viscosity model (eq. 1.23) has been selected for use in this thesis as the model ensures continuous viscous stress across dissimilar fluid interfaces [18]. Cleary's viscosity method approximates viscous stress as a function of particle relative velocity modified by a calibrated empirical factor ζ . The kernel function used in this thesis is the Wendland C_2 kernel [19]. For this kernel function Cleary suggests $\zeta = 4.96333$. This empirical factor is independent of viscosity. The value η is a small, non-zero value to prevent the denominator from approaching zero and causing near singular behavior for very short-range interactions. Equation 1.24 shows the SPH discretized form of the momentum equation with both pressure terms and Cleary's viscosity model.

$$\rho \frac{D\vec{v}_i}{Dt} = \nabla \cdot \boldsymbol{\sigma}^{\alpha\beta} + \rho \vec{g} \quad (1.20)$$

$$\rho \frac{D\vec{v}}{Dt} = -\nabla p + \nabla \cdot \boldsymbol{\tau} + \rho \vec{g} \quad (1.21)$$

$$(\nabla u)_i = \sum_{j=1}^N m_j \left(\frac{u_i}{\rho_i^2} + \frac{u_j}{\rho_j^2} \right) \frac{\partial W_{ij}}{\partial \vec{x}_{ij}} \quad (1.22)$$

$$\left(\frac{\nabla \cdot \boldsymbol{\tau}}{\rho} \right)_{ij} \approx \frac{\zeta}{\rho_i \rho_j} \frac{4\mu_i \mu_j}{\mu_i + \mu_j} \frac{\vec{v}_{ij} \cdot \vec{r}_{ij}}{r_{ij}^2 + \eta^2} \frac{\partial W_{ij}}{\partial \vec{x}_{ij}} \quad (1.23)$$

$$\frac{D\vec{v}_i}{Dt} = - \sum_{j=1}^N m_j \left(\frac{p_i}{\rho_i^2} + \frac{p_j}{\rho_j^2} \right) \frac{\partial W_{ij}}{\partial \vec{x}_{ij}} - \frac{\zeta}{\rho_i \rho_j} \frac{4\mu_i \mu_j}{\mu_i + \mu_j} \frac{\vec{v}_{ij} \cdot \vec{r}_{ij}}{r_{ij}^2 + \eta^2} \frac{\partial W_{ij}}{\partial \vec{x}_{ij}} + \vec{g} \quad (1.24)$$

1.3.2 Boundary Conditions for SPH

A variety of solutions for modeling boundaries in SPH exist. Chapter 4 provides a detailed analysis of SPH boundary conditions and presents a new boundary deficiency correction

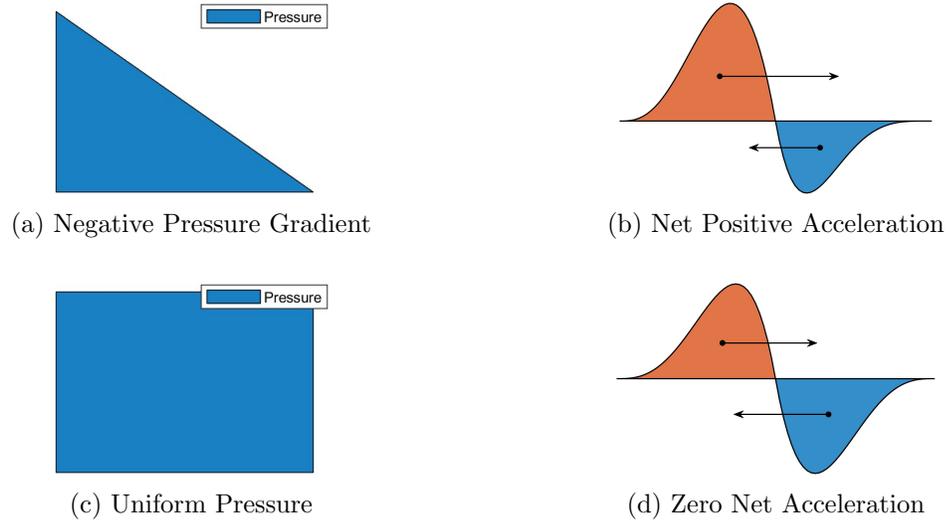


Figure 1.3: SPH Acceleration - Pressure Term

method.

For the simulations shown in this thesis, fluid boundaries are primarily modeled as spatially locked fluid particles (ghost particles), as proposed by Monaghan [3]. Ghost particles are allowed to evolve fluid properties but are fixed to an object or in space. Several rows of ghost particles are typically required to emulate a solid surface and prevent particle penetration into the boundary.

1.3.3 Weaknesses of Smoothed Particle Hydrodynamics

Integral Deficiency

The basic SPH method described by eq. 1.2 requires the assumption that a continuous fluid domain exists within the bounds of a particle's support domain. When this condition is not met, the domain is integral deficient and SPH formulations may break down, leading to nonphysical results. Because fluid interfaces and boundaries can break this key assumption, it is often necessary for SPH simulations to address integral deficiency.

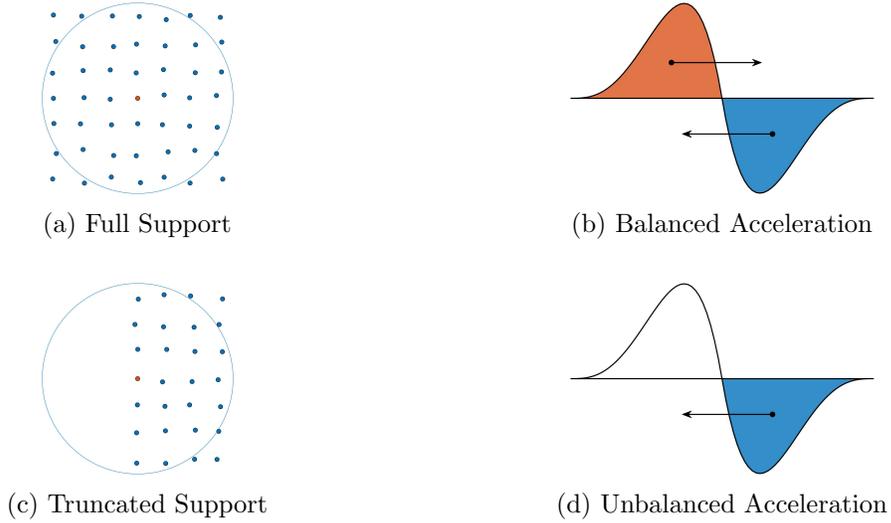


Figure 1.4: Boundary Deficiency

An integral deficiency in any gradient-based SPH formulation results in a vector rather than scalar imbalance. Because the pressure terms of the momentum equation predominantly create a repulsive inter-particle force based on pressure, any particle near an interface will experience a net force towards the area of integral deficiency, and subsequently towards the interface. Similar corrections exist for the kernel gradient but are beyond the scope of this thesis. For an in-depth analysis of kernel correction methods see Bonet and Lok [20].

The most basic integral deficiency occurs in the summation density approach (eq. 1.15) and other summed properties. An integral deficiency for summed SPH values results in an error in the total kernel weight in the formulations. Because Kernel function weights are normalized to sum to unity for a full support domain, any integral deficiency will fall short of this sum and cause a proportional drop in the summed value. This can cause (eq. 1.15) to produce nonphysical reductions in density near a boundary or free surface. A simple solution to errors of this nature was proposed by Monaghan [3]. In this solution the kernel is renormalized by the sum of the kernel weights, effectively adjusting the kernel normalization factor to the volume of each particle's true support domain. The renormalized formulation

for eq. 1.15 is shown in eq. 1.25.

$$\rho_i = \frac{\sum_{\Omega} m_j W_{ij}}{\sum_{\Omega} V_j W_{ij}} \quad (1.25)$$

Clumping

Just as in FVM and FEM formulations, discretization geometry can have a large effect on the consistency of SPH formulations. Asymmetric fluid deformation can cause particles to elongate on a single axis while being condensed on a transverse axis. This clumping of SPH particles can lead to anisotropy in the fullness of particle support domains. This can lead to an effective reduction in resolution along any sparsely populated axis, and in severe cases complete breakdown of the SPH formulations. Several methods to preemptively prevent or correct particle clumping exist. An artificial viscosity force proposed by Monaghan [21] is a popular choice for preventing clumping. This method is useful in that it can also prevent non-physical particle-particle penetration and provide numerical damping of shockwaves when no energy model is present.

Fluid Interfaces

Fluid interfaces also break the underlying SPH assumption of a continuous property field. Densities and other properties are often discontinuous across a fluid interface. This can result in a breakdown of some forms of the continuity equation. The continuity formulation used in this thesis (eq. 1.19) is restricted to similar fluids, with cross-interface particle interactions ignored in density calculations.

Similarly, the momentum equation suffers a breakdown of the expected pressure-based force balance used to distribute momentum. The primary solution for correcting the momentum equation for dissimilar fluid interfaces is to introduce a pairwise repulsive force between any such particles. This prevents non-physical particle-particle penetration and allows the fluids to interact across an interface through this dominant repulsive force. In all cases, the repulsive force must be strong enough to prevent fluid penetration across the

interface by particles on either side. Monaghan suggests a cross-interface repulsive term R_{ij} [22]. This repulsive term (eq. 1.26) is a function of fluid pressure and density and is added to the momentum equation 1.24. In the repulsion term, ρ_{0i} and ρ_{0j} are the reference densities of any two interacting particles, P_i and P_j are particle pressures, and ρ_i and ρ_j are particle densities. The simulations in this thesis utilize this formulation to correct the momentum-equation interfacial error.

$$R_{ij} = 0.08 \left| \frac{\rho_{0i} - \rho_{0j}}{\rho_{0i} + \rho_{0j}} \right| \cdot \left| \frac{P_i + P_j}{\rho_i \rho_j} \right| \quad (1.26)$$

1.3.4 SPH Validation Cases

Couette Flow

To validate the viscosity model used in this thesis, a Couette flow simulation was run. Here a periodic domain of fluid with height 1e-3m is started at rest. The domain is discretized by 40 particles in the x-direction and 80 particles across the vertical span of the fluid (not including boundary particles). The smoothing length was set to $h = 1.4\Delta x$, where Δx is the initial particle spacing in the uniform rectangular particle distribution. At time $t = 0$, the upper boundary of the domain is prescribed a velocity of 1.25e-5m/s while the bottom boundary remains fixed. The flow is allowed to evolve and velocity profiles for times 0.01s, 0.1s and 1.0s are plotted against a series solution by Morris et al. [23]. Figure 1.5 shows the resultant velocity profiles. SPH velocity profiles are plotted by points with theoretical velocity profiles shown with dotted lines. Boundary particles for the upper and lower surface are not shown. It is notable that for larger values of h the SPH results slightly overpredict viscous forces. It is possible that making the empirical factor ζ a function of smoothing length rather than a constant might be able more accurately reproduce physical viscosity values over a range of smoothing lengths.

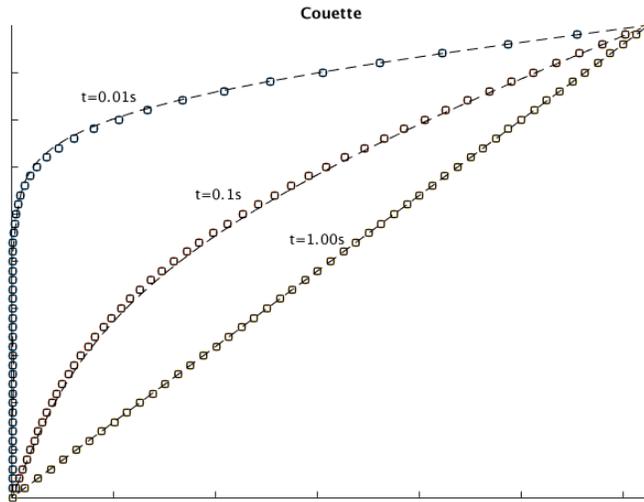


Figure 1.5: Couette Flow SPH Simulation

Rising Bubble

To test basic multiphase flow simulation capability, a viscous rising-bubble test case was run. Following the work of Colagrossi and Landrini [24] and Meister and Rauch [25] a bubble in a domain measuring $6R$ by $10R$ was generated. The fluid density was set to $\rho_w = 1000 \frac{kg}{m^3}$ with the gas density as $\rho_a = 1 \frac{kg}{m^3}$. The bubble diameter was set to $2.3e - 3m$ and gravity set to $9.8 \frac{m}{s^2}$. The speed of sound in the two fluids was set to $c_w = 2.1 \frac{m}{s}$ and $c_a = 14.1c_w$. This non-physical inversion of the speed of sound ratio in the gas and liquid phases is used to aid in the stabilization of the gas-liquid interface following recommendations of Monaghan and Colagrossi and Landrini [22] [24]. Kinematic viscosities were set to $\nu_w = 1 \times 10^{-6} \frac{m^2}{s}$ and $\nu_a = 128 \times 10^{-6} \frac{m^2}{s}$ respectively.

Figure 1.6 shows simulation results for non-dimensional time values ($t\sqrt{\frac{g}{R}}$) of 0, 3.1, and 4.7. The pressure difference due across the bubble causes the fluid at the bottom of the bubble to bow upwards. As the bubble rises, a strong velocity gradient between the

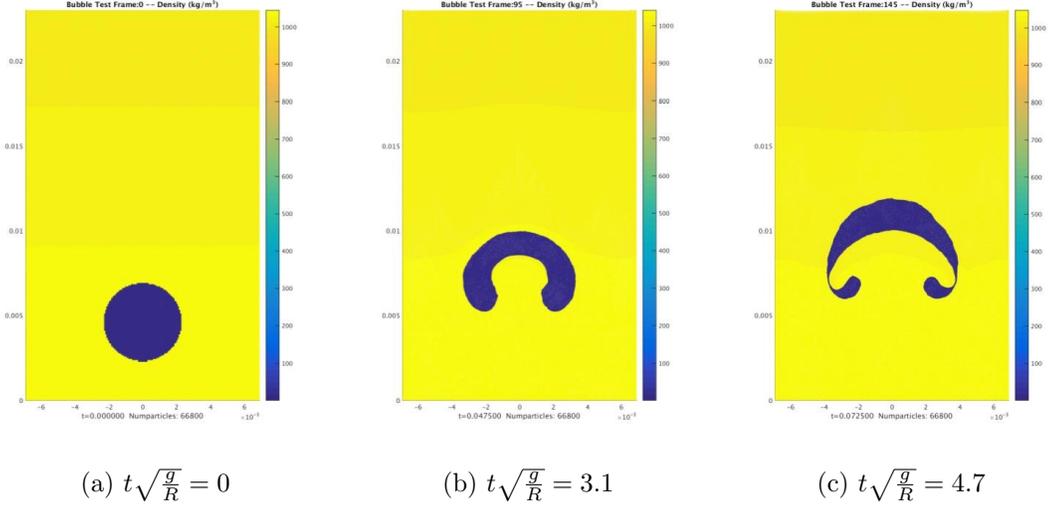


Figure 1.6: Density Driven Flow: Rising-Bubble - Detail

rising column of liquid behind the bubble and downward moving liquid on the sides of the bubble causes viscous forces which lead to strong circulatory flow at the edges of the bubble. This circulation causes the bubble to elongate and eventually leads to the separation of two smaller bubbles.

Figure 1.7 shows a comparison of simulation results (left subfigures) with a simulation performed by Yang [26] (right subfigures). Non dimensional times $t\sqrt{\frac{g}{R}}$ of 0 through 0.9 are shown. Water in the simulation has a density of $\rho_w = 1000kg/m^3$, air has a density of $\rho_a = 1kg/m^3$. The numerical speed of sound for water is set to $c_{H_2O} = 28.28m/s$. While this speed is significantly lower than the true speed of sound for water, so long as the numerical speed of sound in the simulation is significantly higher than the maximum flow-velocity, simulation results will be insensitive to the numerical speed of sound. Liu [8] recommends a numerical speed of sound 20 times that of the largest expected fluid velocity in a flow to minimize non-physical compressibility effects on simulation results.

Both simulations show an asymmetry that forms despite symmetric initial conditions. This suggests the possibility of sensitivity to perturbations inherent to the bubble-rise case.

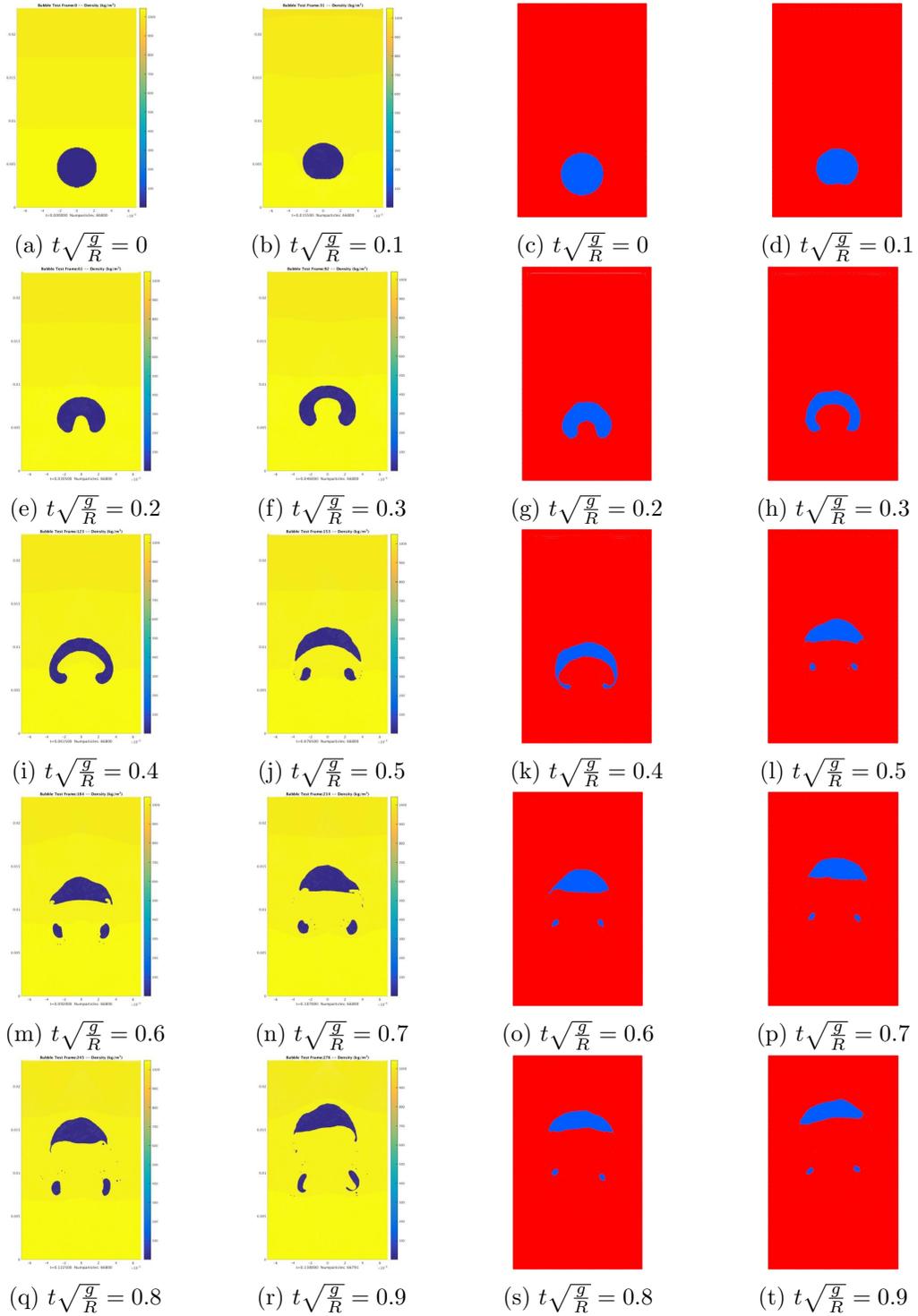


Figure 1.7: Rising-Bubble Simulation (current work left, Yang right)

If there is a high sensitivity to initial conditions, even the small rounding errors accumulated from the non-symmetric traversal pattern of particle-particle interactions may be the source of this observed oscillation. Density-driven bubble simulations of this type are sometimes observed to be performed with symmetric half-plane conditions. Doing this would obfuscate the chaotic nature of the flow, however, to ensure repeatability this may be desirable.

Chapter 2: Underwater Deflagration Experiment¹

Experimental datasets are a critical component for the validation of Computational Fluid Dynamics (CFD) codes. CFD is increasingly being utilized as a primary analysis tool for cases in which experimental analysis is cost-prohibitive or otherwise impractical. The analysis of underwater explosions (UNDEX) is a case that often meets these parameters. To provide a dataset to add to the library of validation data, an experiment consisting of a simple underwater deflagration impinging on fixed and flexible targets is presented. While deflagration is a fundamentally distinct combustion mechanism from detonation, the resultant multiphase flow of underwater deflagration shares the central characteristic trait of high-pressure gas interacting with fluid. The experiment presented in this thesis is designed to provide a validation dataset for CFD codes simulating similar flows. Photographic data and pressure time histories for the experiment are presented in this thesis.

2.1 Experiment Overview

The study of underwater explosions (UNDEX) is critical for damage survivability and many other topics in the field of naval engineering [27, 28]. It is also an important subject for underwater construction, demolition, harbor maintenance, and studies of the resultant environmental impacts of these activities [29–31]. Increasingly, Computational Fluid Dynamics (CFD) is augmenting or replacing experiments for applied UNDEX analysis. A recent survey of state-of-the-art UNDEX simulation and experiments by de Camargo concludes that "[...] numerical simulation software is a well-established time-saving tool for yielding precise UNDEX results, leaving aside the need for resource costly experimental tests" [32].

¹Work from this chapter has been submitted to the journal *Ocean Engineering* and is currently pending revisions.

This increased utilization of CFD as a primary prediction tool makes the validation of the numerical methods used in CFD software ever more critical. Validation of CFD codes is most often performed by choosing representative test cases and comparing numerical and experimental results to those cases [33–40]. This process allows developers to establish approximate bounds on the expected accuracy and robustness of numerical results. While a large library of experimental UNDEX data exists in the context of naval vessels [41–44], the data from these experiments may be difficult to utilize for applications unrelated to naval vessels. More generalized experimental UNDEX datasets such as those generated by Cuio et al [45], De et al [46], Yang et al, [47] expand on the available library of UNDEX data. These and similar datasets form a basis for experimental work in UNDEX. This work seeks to add to the existing library of generalized experimental UNDEX data by providing a dataset for a low-intensity underwater deflagration with fluid-structure interaction (FSI). Some years ago, the original experiment design and initial results were published in [48, 49]. In the years since, additional video data has been published [50]. This thesis provides further information on the experiment design and provides additional quantitative and qualitative data for those in the simulation community wishing to use it for validation purposes.

To obtain quantitative data for a small-scale underwater deflagration, an experiment consisting principally of a submerged, low-aspect-ratio, vertically-oriented cylinder with one open end was devised. Premixed propane and air were selected for the combustible reactants. Earlier work identified a slightly fuel-rich mixture as desirable to reduce uncertainties in the limiting reactant in the combustion process [49]. In this low-pressure, unconstrained configuration, the combustion of air and propane propagates via thermal conduction (a process known as deflagration). Combustion in a deflagration reaction differs substantially from combustion in detonation. In detonation, the combustion reaction propagates by a compression wave moving through the reacting material. The high-speed reaction of detonation typically results in much higher gas pressure in a post-combustion state. Furthermore, the compression wave of a detonation after exiting the reactants continues to propagate into the surrounding fluid as a strong shock. However, even with these significant

differences, post-combustion underwater deflagration and detonation both share the central characteristic of a high-pressure gas bubble surrounded by water. Studying deflagration instead of detonation provides several key advantages. Most notably, the lack of a shockwave in underwater deflagration facilitates data gathering in several ways. The lack of a shock allows for experiments to be performed housed in an un-reinforced tank. This simplifies the acquisition of visual data. Another advantage is the longer timescale and lower peak pressures inherent to deflagration reactions. Detonation reactions progress extremely rapidly and generate strong shockwaves and very large pressure spikes. Deflagration reactions in contrast progress at timescales orders of magnitude slower than that of detonations. The lower peak pressures, slower reaction rate, and longer timescales of a deflagration simplify the acquisition of high fidelity pressure data. This is because the reduced peak pressures and longer timescales of a deflagration do not necessitate high-frequency hardened pressure sensors -such as piezoelectric pressure transducers- to accurately resolve pressure changes throughout the reaction. While deflagration is a fundamentally different mechanism of fuel-oxidizer reaction compared to detonation, the high-pressure gas generated in deflagration is characteristically similar to those generated in detonation. Considering this and the advantages deflagration presents over detonation for ease of data acquisition, it is proposed here as a method for obtaining high-pressure gas dynamics which may be useful for validation of CFD simulations focused on the non-shock-related effects of underwater explosions. The data generated in this study is designed to focus in detail on the bubble dynamics and pressure of underwater deflagration and is intended to provide data for validation of CFD codes focused on these effects.

2.2 Materials and Methods

To provide data on the forces and displacements experienced by a flexible membrane under the influence of a low-intensity underwater deflagration, a two-part experiment was performed. Both stages of the experiment used a vertically oriented acrylic cylinder, sealed on the upper end to contain a mixture of air and propane. In the first stage a deflagration

from the cylinder was allowed to impinge on a rigid plate target (Fig. 2.1a). In the second stage, the rigid plate was replaced with a flexible rubber membrane (Fig. 2.1b).

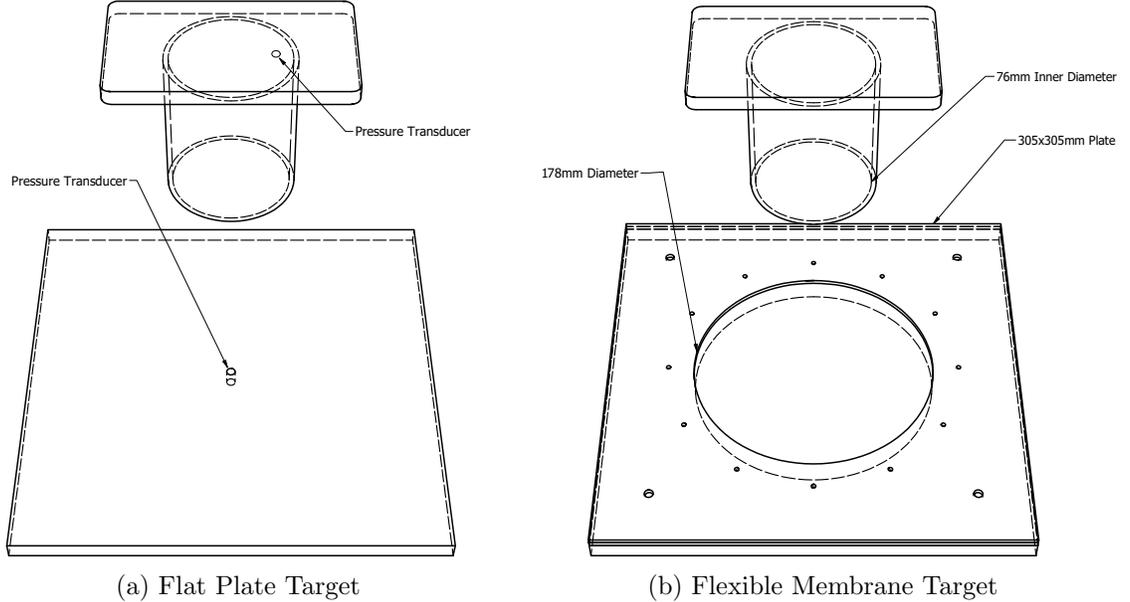


Figure 2.1: Deflagration Targets

The acrylic cylinder measured 100 ± 1 mm in length with an inner diameter of 76 ± 1 mm (Fig. 2.2a). It was mounted in a rigid structure 24 ± 1 cm above the rigid plate target (Fig. 2.2b). The assembly was placed in a $1.2 \text{ m} \times 1.2 \text{ m} \times 1.2 \text{ m}$ water tank. The tank was filled with 67.5 ± 2 cm of water, submerging the upper surface of the rigid plate target to a depth of 53 ± 2 cm. Two electrodes were embedded in the top-center of the cylinder to provide spark-ignition (Fig. 2.2a). An OMEGA PX26-030DV pressure transducer was mounted off-center within the top of the cylinder to measure cylinder pressure. To minimize variation in the total energy output of the combusting gases, a target fuel-air equivalence ratio (ϕ) of 1.21 ± 0.07 was selected. This ratio yields a rich air-fuel mixture with oxygen as the limiting reactant. Experimental work performed by Ebaid et al. [51] suggests that propane combustion at this equivalence ratio will have a laminar flame speed of approximately 26

cm/sec.

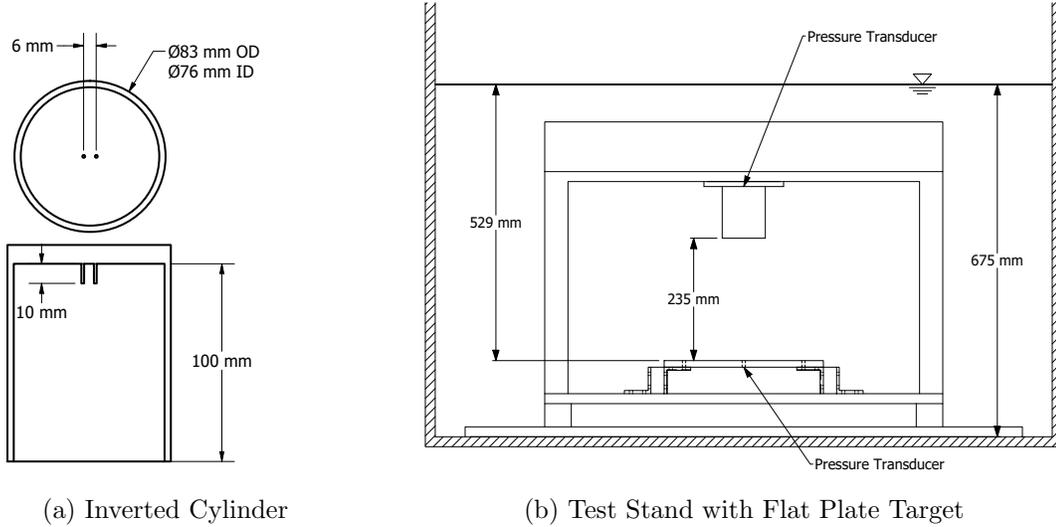


Figure 2.2: Experiment Assembly

To prepare the system for ignition, air and propane volumes were measured at atmospheric pressure and were mixed before being added to the cylinder. This was done by adding 434 ± 5 ml air and 22 ± 1 ml propane gas (an air-fuel ratio of 20:1) to a balloon. Then a small amount of water was introduced to the balloon reservoir and the system was agitated to mix the gasses. A positive displacement pump was then used to move the fuel-air mixture from the reservoir into the inverted combustion cylinder. Water was then added to the inlet line to displace any remaining fuel-air mixture and prepare the system for ignition.

In the first stage of the experiment, a deflagration impinging on a flat plate was analyzed. The rigid plate target was fixed $235 \text{ mm} \pm 5 \text{ mm}$ below the bottom of the cylinder and was instrumented with an OMEGA PX26-005DV pressure transducer. The pressure transducer was fixed in the center of the plate in line with the cylinder axis. The gas mixture was then ignited with a piezoelectric spark generator. This procedure was repeated five times to obtain a set of video and pressure time histories for the experimental results.

In the second configuration of the experiment, the rigid plate target was replaced with

a target that included a circular 178 ± 0.5 mm diameter, 2.4 ± 0.1 mm thick, 50 DURO flexible silicon membrane (Fig. 2.3). To integrate the membrane into the target plate, a circular cutout was made through a 12.7 mm thick acrylic plate. The membrane was then placed on top of this base plate and secured by affixing a 1.6 mm thick acrylic plate with a matching circular cutout on top of the flexible membrane. The top plate was affixed to the base plate by twelve evenly spaced bolts in a 216 mm diameter circular pattern (Fig. 2.1b). To obtain a stereoscopic view of the membrane a pair of Edgertronic SC1 high-speed cameras were placed approximately two meters from the target with a one-half meter spacing (Fig. 2.4). The gas mixing and ignition procedure was repeated and video data of the membrane response was recorded. The membrane's center-point vertical displacement history was calculated by pixel tracking and strain data was obtained through Digital Image Correlation (DIC) using LaVision DaVis software [52].

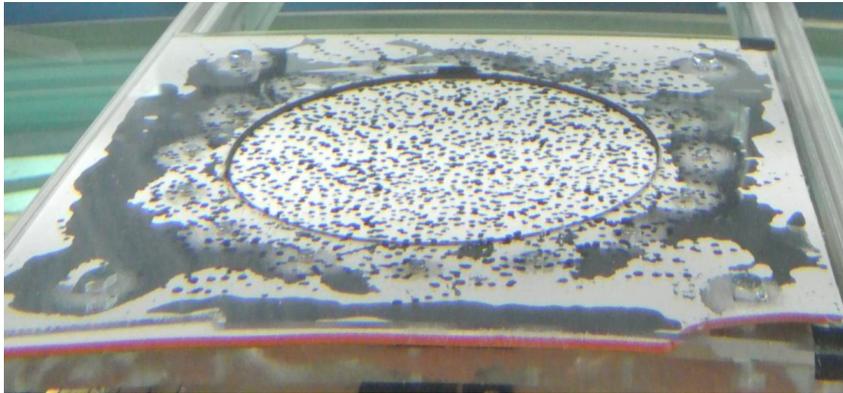


Figure 2.3: Flexible Membrane

2.3 Results

As described above, the experiment was broken into two parts. Section 2.3.1 presents the results from the experiment using an instrumented rigid plate target and Section 2.3.2

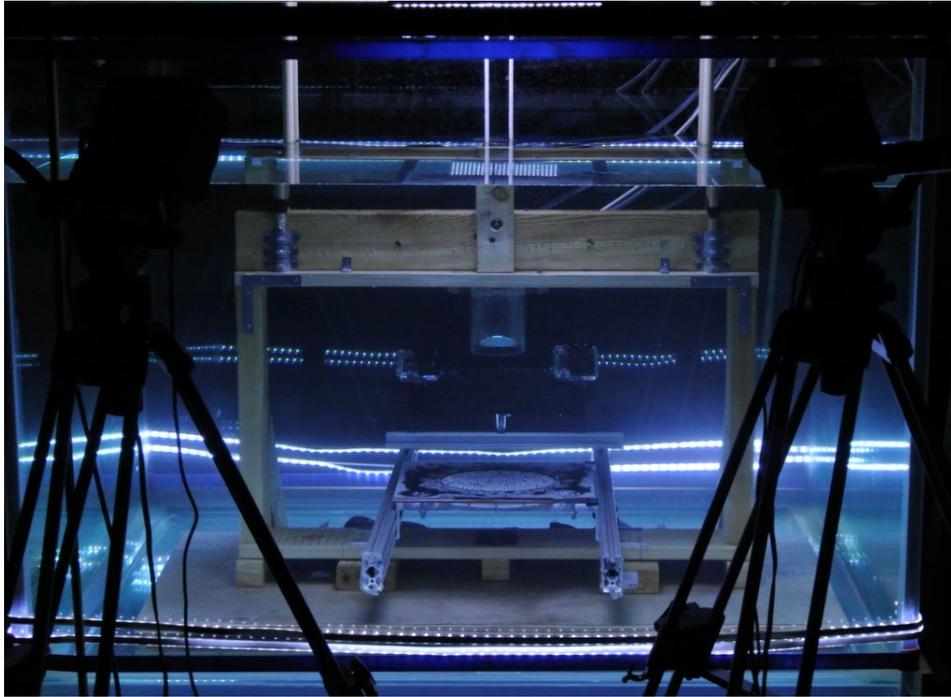


Figure 2.4: Stereoscopic Camera View of Flexible Membrane Target

presents results using a flexible-membrane target.

2.3.1 Rigid Plate Target

For the rigid plate case, all five experiment runs yielded characteristically similar behavior, though with noticeable variability in the resultant combustion rate due to variations in the composition of the fuel-air mixture. However, the behavior of the system can be broken into a sequence of characteristically distinct stages, with the data from post-combustion stages exhibiting very little variability. A time history of a single run of the rigid plate experiment is shown in Fig. 2.5.

For the selected test (Run 4) of the rigid plate target case, the following behavior was observed. After ignition, a flame-front propagates radially outwards from the ignition electrodes. As the fuel-air mixture combusts, pressure in the cylinder increases slowly. (Fig.

2.8a). As the flame-front propagates through the cylinder, a Poiseuille-like velocity profile causes the flame-front to transition from its initial spherical shell to a progressively sharper parabolic form. This behavior is expected in a premixed flame propagating in a tube prior to Shelkin acceleration [53][54]. Fourteen to twenty milliseconds after ignition, the air-water interface begins to form an oblate bubble centered with the plane of the cylinder outlet. The bubble maintains approximate radial symmetry about the vertical axis and expands approximately equally above and below the plane of the cylinder outlet. Between 22 and 36 ms after ignition, the flame-front reaches the lower outlet of the cylinder (2.8b). As the flame-front travels past the cylinder outlet and into the primary bubble, its surface area expands greatly. Because the fuel combustion rate is proportional to the surface area of the flame front, this expansion leads to a rapid increase in the combustion rate of the fuel-air mixture. This causes a rapid increase in the cylinder and bubble pressure (visible as the first pressure spike in Fig. 2.9). Maximum bubble diameter occurs 20 ± 2 ms after the flame front reaches the cylinder outlet (Fig. 2.8d). At its maximum diameter, inertial forces have caused the bubble to over-expand. In this state, the gas pressure in the cylinder and bubble has dropped below the original hydrostatic pressure at the cylinder outlet. This pressure imbalance between the gas and surrounding fluid slows and then ultimately reverses the bubble's expansion, causing the bubble to collapse. Through the majority of the bubble's collapse, its shape remains approximately vertically symmetric. However, this vertical symmetry is broken as the air-water interface on the top half of the bubble nears the cylinder wall. This is likely a result of the presence of the combustion cylinder's walls, and their action to disrupt the gas flow in the upper half of the bubble. This loss of vertical symmetry occurs 10 ± 1 ms after the maximum observed bubble diameter (2.8e). Inwards moving fluid in the plane of the cylinder outlet converges on the cylinder impinging on its sides. This moving fluid separates the two vertical halves of the original bubble. Fluid striking the cylinder wall bifurcates, and fluid moving upwards along the cylinder outer wall pulls the upper bubble with it. The separated upper portion of the bubble becomes turbulent and rises upwards under the combined effects of this moving water on the cylinder

and buoyancy. Portions of the inwards moving fluid that are deflected downwards from the bifurcation on the cylinder are directed downwards at high velocity. This rapidly moving fluid leads to cavitation appearing on the outer edge of the cylinder visible in Fig. 2.8f. This downwards moving fluid is redirected into the center of the separated lower bubble imparting rotational motion to the fluid in the region. The separated lower bubble then undergoes a further collapse. This secondary bubble collapse leads to a high-pressure region in the fluid directly below the cylinder outlet. This leads to the formation of a pair of axial water jets (visible in Fig. 2.5k, 2.5l). The upper jet is unimpeded until it strikes the upper boundary of the cylinder at which point pressure data in the cylinder sensor becomes chaotic due to the highly turbulent flow.

Pixel tracking from video data was analyzed to obtain time histories of the resultant bubble radii. Figure 2.6 shows the bubble radii from each of the five test runs. While the maximum bubble radius observed in the cases varies only slightly (with a difference of 9 mm between the smallest and largest bubble radii), the bubble expansion rates vary significantly. The maximum bubble diameter occurs in the most rapid case within 34 ms of initial gas emergence from the cylinder while the slowest expansion case required 49 ms to reach maximum diameter.

Cylinder and bottom-plate pressure histories for the five experiment test-runs are shown in Figures 2.9 and 2.10 respectively. During the initial combustion phase, prior to the flame-front reaching the outlet, cylinder pressure rises nearly linearly. When the flame-front passes the cylinder outlet the surface area of the flame-front expands greatly. This leads to an increase in the combustion rate and causes a rapid increase in measured pressure. After combustion completes the bubble continues to expand. The measured pressure continually decreases in this stage, eventually decreasing below gauge pressure as inertial forces cause the bubble to over-expand. The over-expanded bubble then collapses, leading to a steady increase in cylinder pressure. Finally, in all cases, a rapid and chaotic pressure rise in the cylinder is observed after bubble collapse. This is most likely due to a jet of water impinging on the upper surface of the cylinder.

Figure 2.6 shows time histories of the radius of the gas bubble about the vertical axis. A significant degree of variability is observable in resultant behavior. This variability however appears primarily in the rates of bubble growth. There is an observable trend between the rate of bubble growth (initial slope of the bubble radius) and the maximum observed bubble diameter for each run. However, despite significant variance in the time between ignition and maximum bubble diameter, the largest observed bubble radius (Run 1) is only 14% larger than the minimum observed radius at maximum expansion (Run 3). Differences in air-fuel mixture homogeneity and water-vapor content are likely significant factors in this observed variability. However, it is notable that combustion occurs only early in the experiment. By observation, post-combustion bubble-radius exhibits similar characteristics. To further highlight this trend, it is useful to align the experiment data by the time of the maximum observed bubble diameter. To do this, the time from ignition to maximum-bubble-diameter for each case was tabulated and data was offset by these values. Ignition times for each run are shown. Figure 2.7 shows the bubble radius time-history aligned in this manner. The experiment can then be separated into five discrete stages. These are (I) combustion in the cylinder, (II) combustion in the bubble, (III) inertial expansion, (IV) collapse, and (V) post-collapse. The visual state of the initial and final stages and transitions between each of the stages is detailed in Figure 2.8.

Figures 2.9 and 2.10 show pressure time histories of the cylinder and lower-plate pressure sensors. The pressures have been aligned by the minimum observed cylinder pressure (to correspond with the maximum bubble diameter). After the first pressure peak, the pressure in the bubble rapidly decreases as its volume grows. After the bubble has reached pressure equilibrium with the surrounding fluid inertial forces continue to cause the bubble to expand, reaching a lower than hydro-static pressure at maximum bubble volume. The bubble then collapses, with pressure in the cylinder rising to a peak of similar magnitude to that of the initial expansion.

To facilitate validation for simulations without combustion dynamics, bubble geometry

for the maximum expansion state is presented. At maximum bubble expansion, fluid velocities in the tank are expected to be negligible. This should allow simulations to begin from this state, bypassing combustion and expansion phases. Figure 2.11a illustrates the oblate spheroid shape of the bubble at maximum expansion. Equation 2.1 with parameters from Table 2.11b provides a fit of the bubble’s surface in cylindrical coordinates for each run of the experiment at maximum bubble expansion. The bubble center is offset from the cylinder outlet by a distance h . Because buoyant forces cause the bubble to translate upwards through the duration of the experiment, this value is negative at maximum expansion (indicating the bubble center is above the plane of the cylinder outlet).

Figure 2.12 shows the mean as well as minimum and maximum envelope time-histories of the flame-front vertical distance from the closed upper end of the cylinder. Times are relative to the first visible presence of ignition. Significant variations in the flame-front propagation rate are visible. As noted earlier, this is likely due to both variations in the mixing of the fuel-air mixture as well as variability in the water vapor content of the gas mixture.

2.3.2 Flexible Membrane Target

Figure 2.13 shows the deflagration impinging on the flexible membrane target. A time history of vertical displacement for the center of the circular membrane was generated by pixel tracking. Figure 2.14 shows the membrane center-point track alongside the rigid plate pressure time histories. The center-point track shows the combined effect of multiple modes of membrane vibration. Because of this, the center point time history exhibits a somewhat complicated response behavior.

To better visualize the strain experienced by the membrane, a 2D planar strain visualization was generated using Digital Image Correlation (DIC) software and stereoscopic video data (Fig. 2.15). It is important to note that the two-dimensional strain analysis does not accurately depict true membrane strain (as the membrane oscillation includes significant out-of-plane deformation violating the planar assumption used by the analysis). However,

the visualization provides a useful qualitative approximation of the strain experienced by the membrane during deformation.

In Fig. 2.15e there is visible evidence of a secondary mode of membrane oscillation. This suggests imperfect radial symmetry in the experiment or a misalignment of the shared axis of the cylinder and membrane. However, because the asymmetry in the membrane behavior appears late in the membrane response, any alignment error is likely minor.

2.4 Discussion

As noted, significant variability in resultant combustion rates was observed. A possible source of the variability in the flame-front speed visible in Fig. 2.12 is the homogeneity of the fuel-air mixture. Figure 2.16 shows detailed images of the flame-front at equivalent states for two separate experiment runs. The flame-front visible in Figure 2.16a appears smooth, indicating the reacting gases are likely well mixed and largely homogeneous. This is contrasted by the flame-front visible in figure 2.16b which exhibits noticeable irregularities. These irregularities suggest the presence of localized variations in the fuel-air ratio of the mixture. Changes in the local equivalence ratio may increase or decrease the local combustion rate depending on the local mixture properties.

An additional factor leading to combustion rate variability is the method utilized to mix the combustion gases. It is likely that the method of mixing the gases by agitating the air and propane with a small volume of water increased the humidity of the fuel-air mixture. It is well documented that hydrocarbon combustion rates are reduced with increasing water vapor content [55].

Furthermore, the chosen fuel-equivalence ratio may have contributed to combustion-rate variability. A downside of the chosen equivalence ratio is that a small change in the fuel mixture at the chosen equivalence ratio results in a significant change in the expected linear flame speed. Combustion of a gas mixture nearer to a stoichiometric ratio would exhibit a lower flame-speed sensitivity to fuel mixture variation.

However despite the variations in flame-front propagation speed and bubble growth rate,

the low variations in the maximum bubble diameter of each experiment run visible in (Fig. 2.7) suggests that the choice of a rich fuel mixture achieved the desired low-variability in total energy released.

2.5 Summary

An experiment to generate a dataset for low-intensity underwater deflagration impinging on a rigid surface and flexible membrane was performed. Visual data and pressure time histories for the cylinder and rigid plate were recorded. Center-point displacement and qualitative strain data for the response of a flexible membrane were obtained using pixel tracking digital image correlation. Variations in the observed combustion rate caused variability in the initial deflagration stage in the cylinder. However post-combustion system behavior, including maximum bubble diameter, exhibited very little variability.

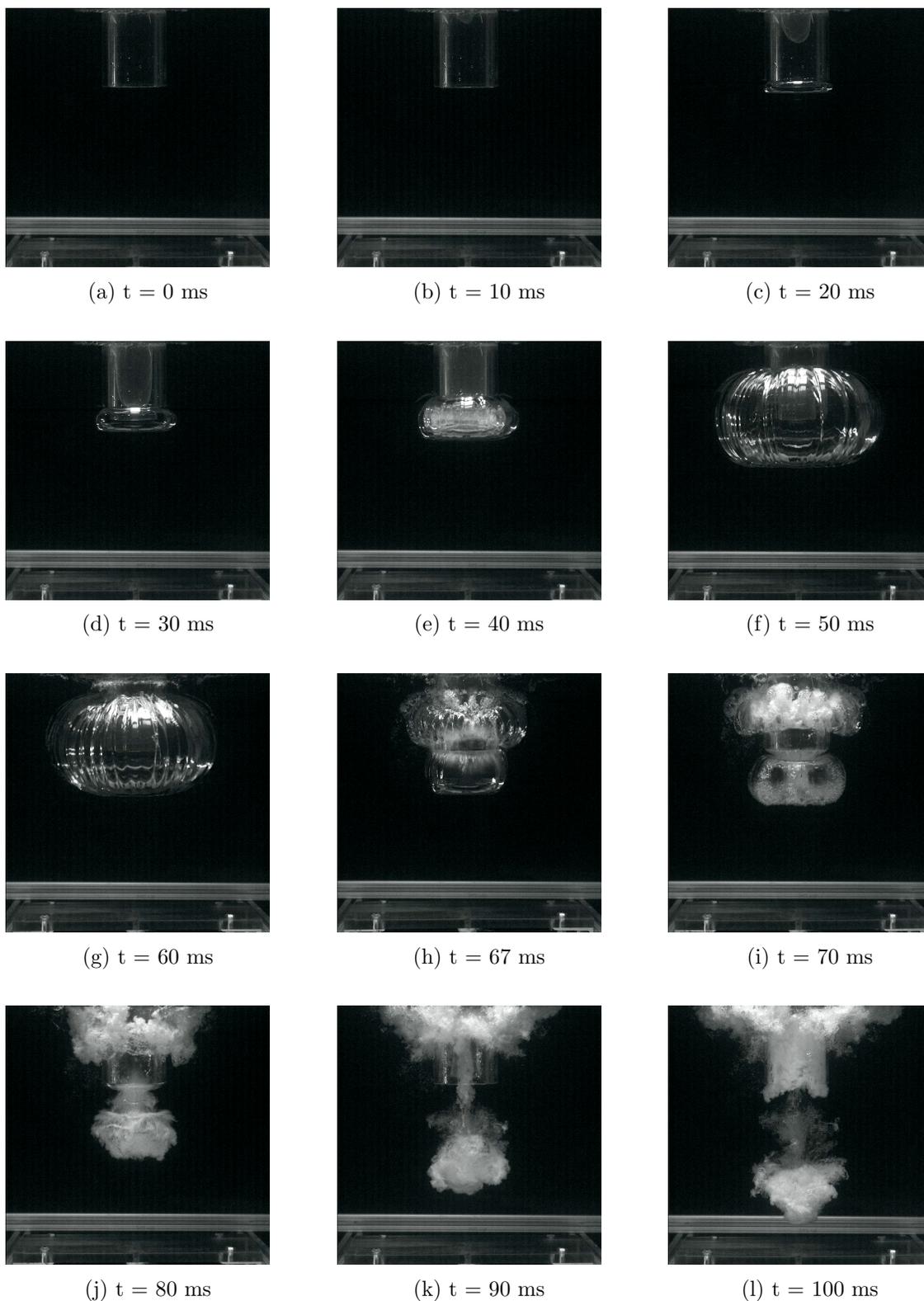


Figure 2.5: Deflagration and Bubble History (Run 4)

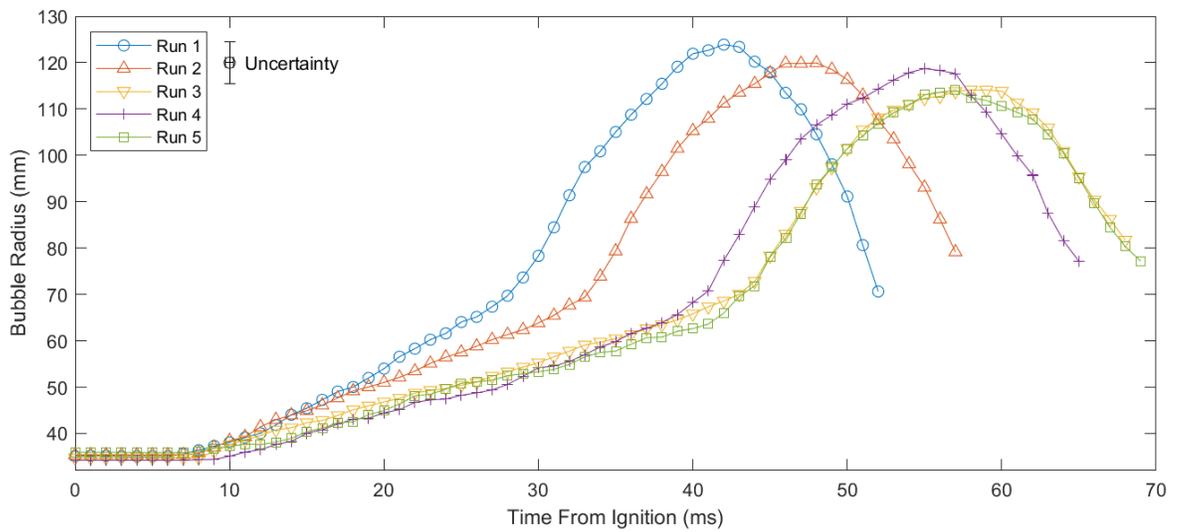


Figure 2.6: Bubble Radius

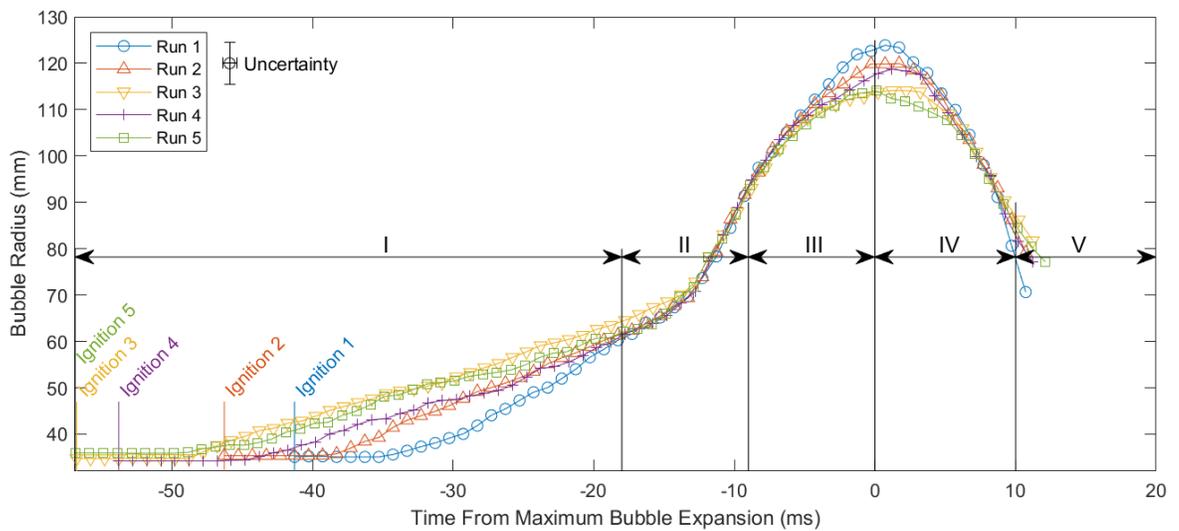
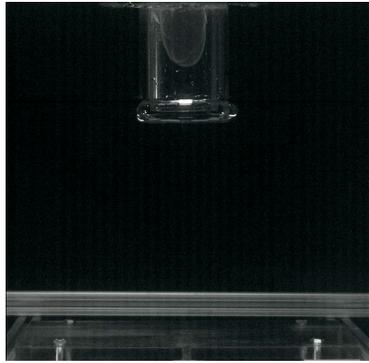
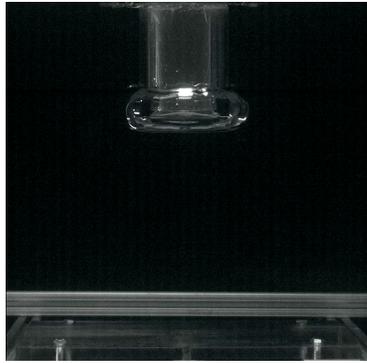


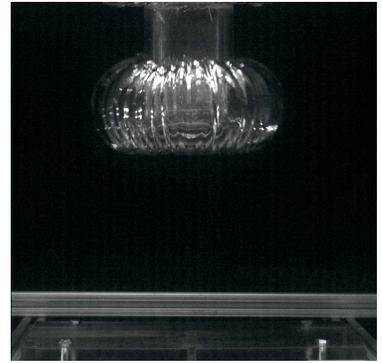
Figure 2.7: Bubble Radius (Aligned at Maximum Diameter) and Deflagration Stages, I: Combustion in Cylinder, II: Combustion in the Bubble, III: Inertial Expansion, IV: Collapse, V: Post-Collapse



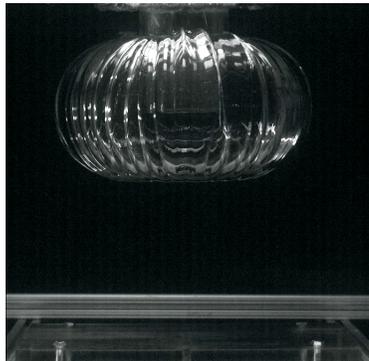
(a) Stage I: Ignition, Combustion in the Cylinder



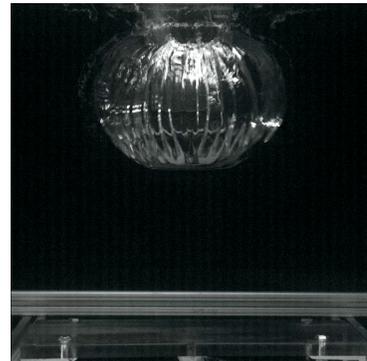
(b) Transition I→II: Flame-Front Reaches the Bubble



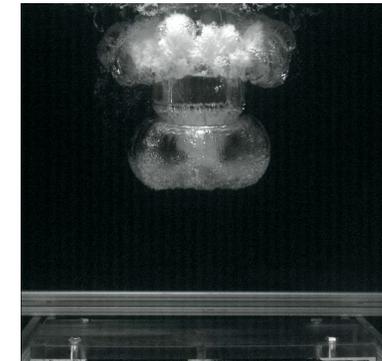
(c) Transition II→III: Combustion Completes



(d) Transition III→IV: Maximum Expansion



(e) Transition IV→V: Symmetry Breakdown



(f) Stage V: Post Bubble Collapse Behavior

Figure 2.8: Deflagration Stage Transitions

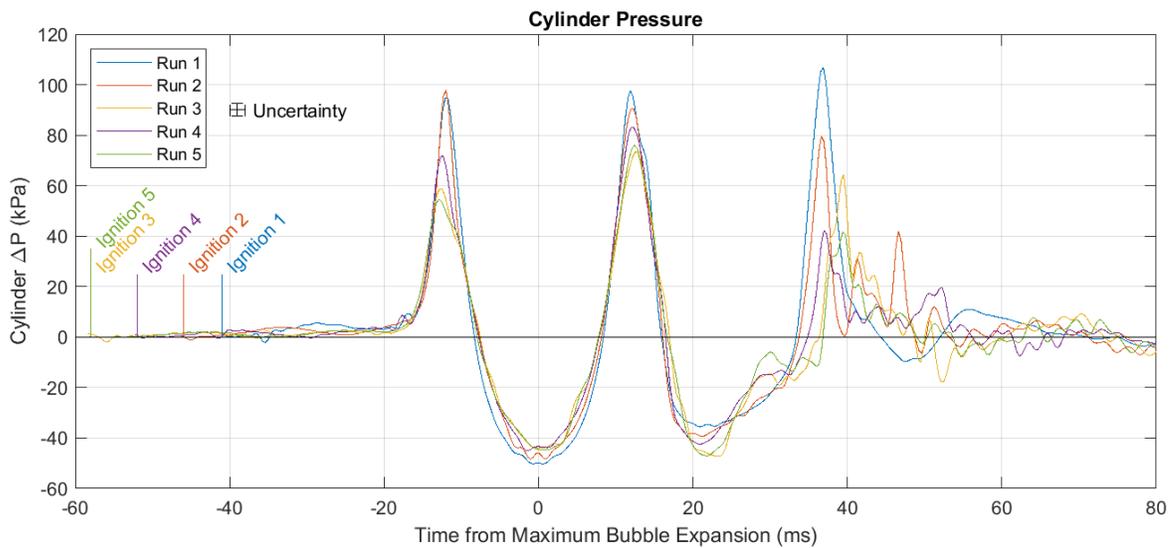


Figure 2.9: Cylinder Pressure

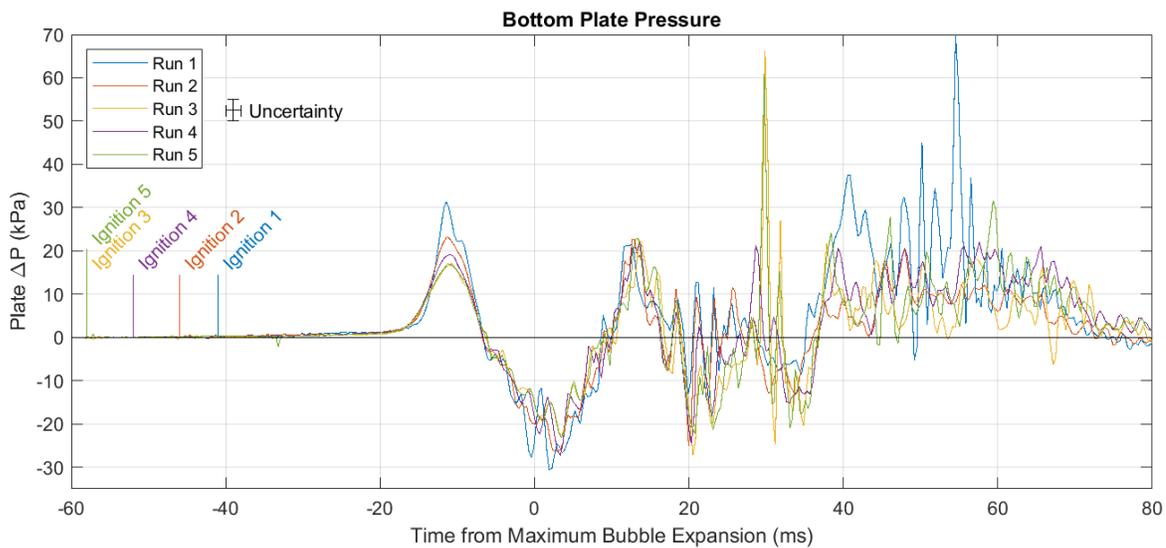
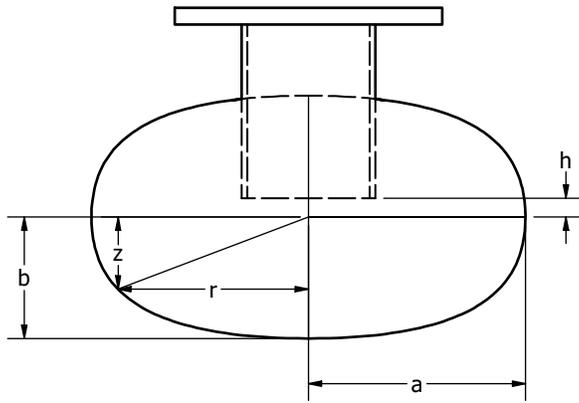


Figure 2.10: Bottom Plate Pressure



(a) Bubble Shape at Maximum Expansion

Run #	a (mm)	b (mm)	h (mm)
1	124	74	-4.5
2	120	72	-3.5
3	114	67	-4.5
4	119	69	-4.0
5	114	68	-6.0

(b) Shape Parameters at Maximum Expansion

Figure 2.11: Bubble Shape at Maximum Expansion

$$\frac{r^{3.7}}{a^{3.7}} + \frac{z^2}{b^2} = 1 \quad (2.1)$$

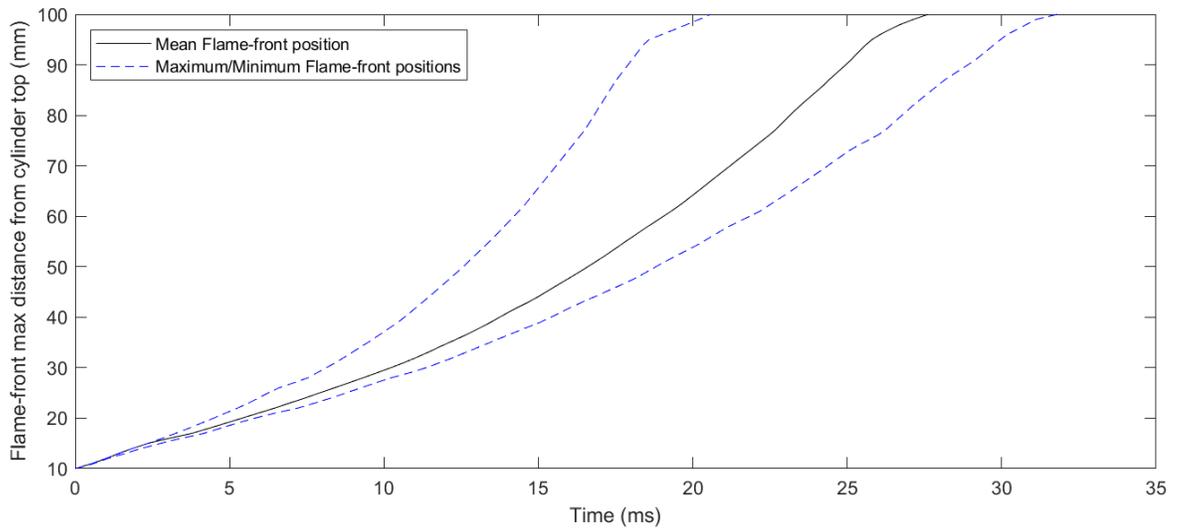


Figure 2.12: Flame-Front Distance From Origin

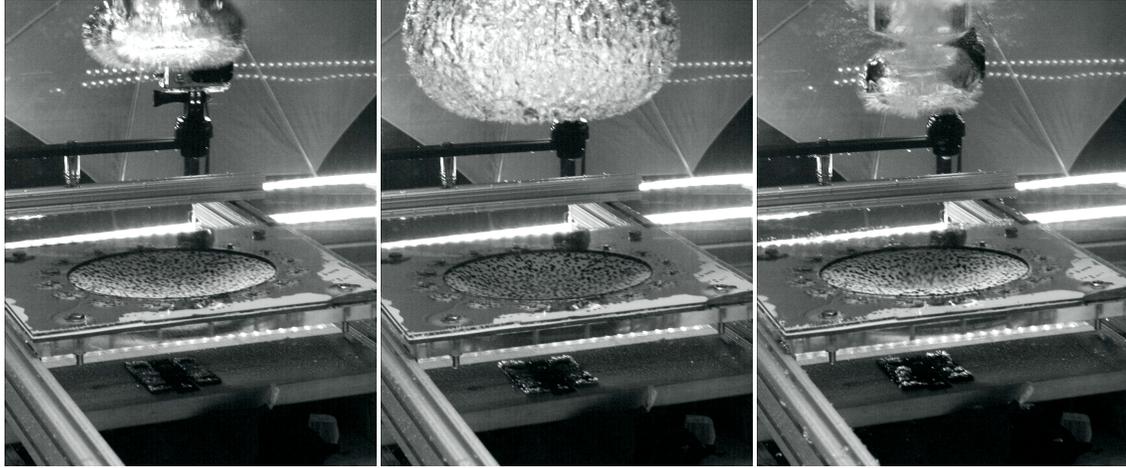


Figure 2.13: Deflagration Impinging on Flexible Membrane Target

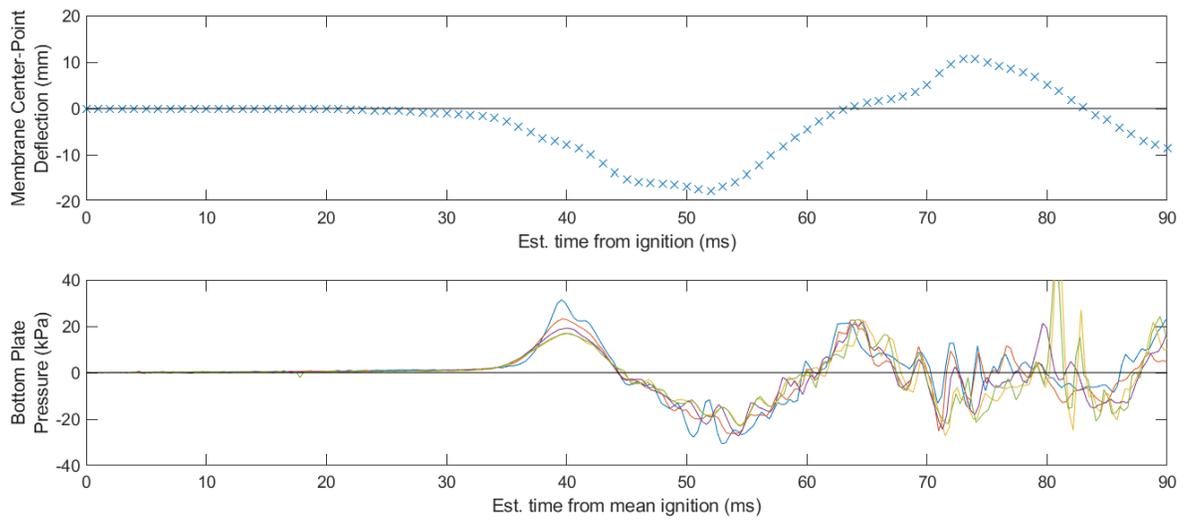


Figure 2.14: Membrane Displacement & Rigid Plate Pressure

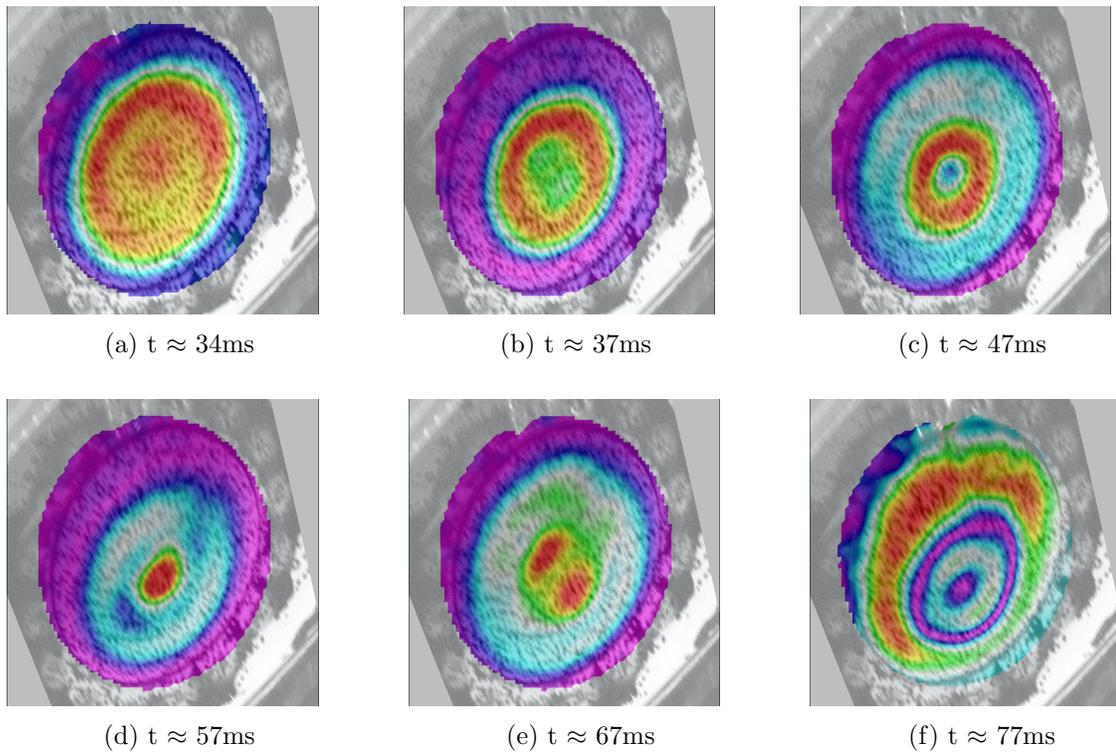


Figure 2.15: Qualitative DIC Visualization of 2D Planar Membrane Displacement

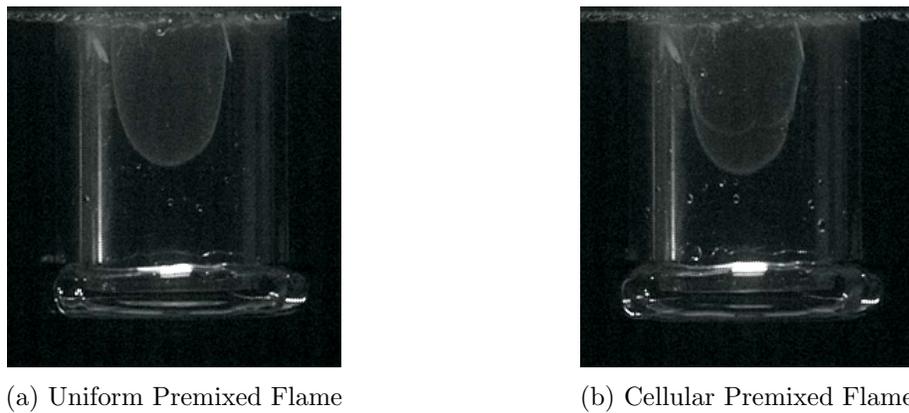


Figure 2.16: Combustion Detail

Chapter 3: Underwater Deflagration Simulation

In order to assess the ability of SPH to simulate an underwater deflagration using basic SPH formulations, a first approximation of the experimental cylinder test case was run. Several simplifications were made to the simulation, most centrally that the simulation is restricted to 2-dimensions. However, because the goal of this simulation is to identify deficiencies in stability and overall behavior, matching the experimental results is not necessary or expected. An additional simplification to the simulation is made by modeling the combustion process as a linear reference-density ramp over a timespan of 30ms. By altering the reference density in eq. 1.12 by a factor of k , the pressure of the combustible gas mixture in the cylinder can be indirectly influenced. Equation 3.1 shows this modified form of the equation of state for the combustible gas mixture. This approach can be used to loosely approximate the pressure rise due to combustion.

$$P = \beta \left(\frac{\rho}{k\rho_0} - 1 \right)^\gamma + P_0 \quad (3.1)$$

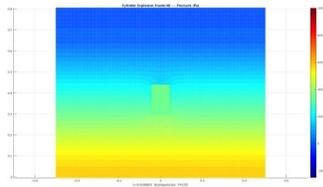
For the simulation, the reference density k was linearly changed from 1.0 to 0.5 over 30ms. The motivation for choosing this simplified reference-density target was to ensure that after the cylinder gas had expanded by a factor of two, the gas would be at the initial equilibrium pressure. By analysis of Equation 3.1, the theoretical increase in pressure from this density change in a static system starting at its reference density can be determined. In this case Equation 3.1 reduces to $\beta \left(\frac{1}{k} - 1 \right)^\gamma$. With $k = 0.5$, $\beta = 57715$ Pa, and $\gamma = 1.4$ this reduces to a pressure rise of β , or 57.7 kPa (22.1 psi).

However, because the gas in the simulation will expand during the pressure ramp, the true pressure rise will be significantly lower than this upper-bound estimate.

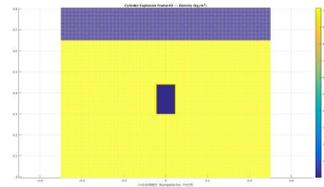
3.1 Simulation Results

Figure 3.1 shows a time history of the results of the SPH simulation of the underwater deflagration experiment. The left sub-figures are colored by pressure while the right sub-figures are colored by density. The simulation exhibits qualitative characteristics of the experiment. Pressure in the cylinder rises before the pressure in the fluid near the bottom plate increases. However, the simulation does not form a bubble in the shape of an oblate sphere. Instead, air is forced to the outside of the cylinder and begins rising due to buoyancy before additional expanding gas can form an oblate bubble shape. This is likely both a consequence of the inexact modeling and calibration of the combustion model as well as the fundamental difference between a 2-dimensional simulation and a full 3-dimensional flow.

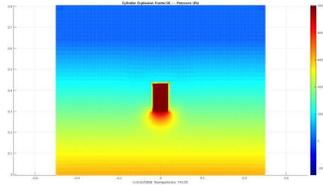
Figures 3.2 and 3.3 show pressure histories of the simulation compared to those of the experiment. While the total combustion period observed in experimental results varied from between 30ms and 50ms, the highly simplified 30ms pressure ramp used in the simulation is a poor analog to the true combustion process. Additionally, though combustion in the experiment occurs over a period ranging in duration from 35ms to 45ms, the observed combustion rate of the gas is highly variable throughout the deflagration. While combustion is confined within the cylinder the net combustion rate is relatively uniform for any single experiment run. However, once combustion reaches the bubble, the rate of combustion increases rapidly due to the large increase in the flame-front within the expanding gas bubble. This causes the majority of combustion for all experiment runs to occur during a narrow 10ms time period after this occurs. This non-linear combustion behavior is not modeled in the simulation. Future work to either fully model combustion or develop a more accurate low-order combustion model is planned.



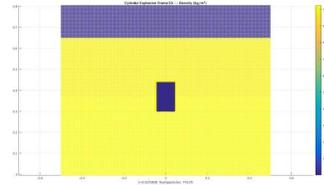
(a) $t=0\text{ms}$



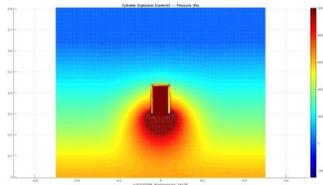
(b) $t=0\text{ms}$



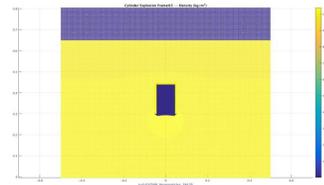
(c) $t=5\text{ms}$



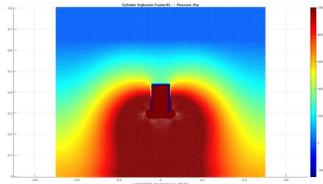
(d) $t=5\text{ms}$



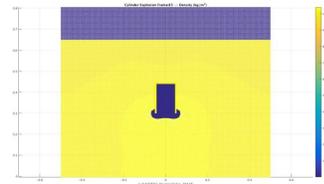
(e) $t=12.5\text{ms}$



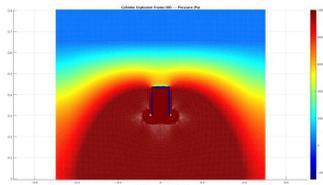
(f) $t=12.5\text{ms}$



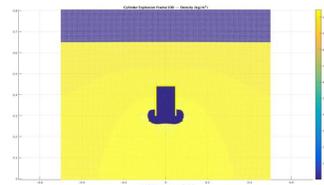
(g) $t=22.5\text{ms}$



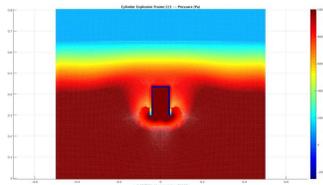
(h) $t=22.5\text{ms}$



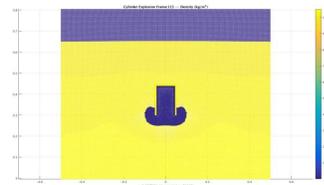
(i) $t=30\text{ms}$



(j) $t=30\text{ms}$

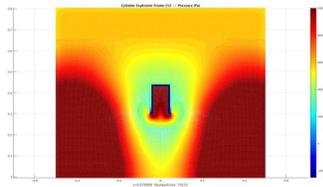


(k) $t=37.5\text{ms}$

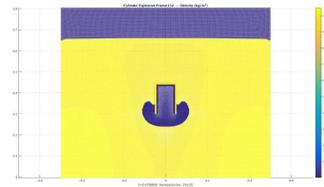


(l) $t=37.5\text{ms}$

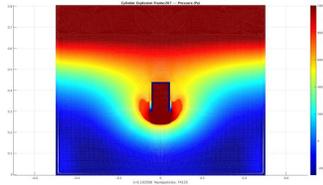
Figure 3.1: Deflagration Simulation Results (Left: Pressure, Right: Density)



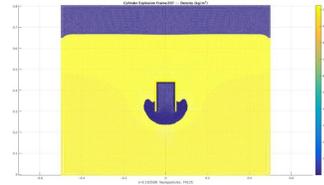
(m) $t=56\text{ms}$



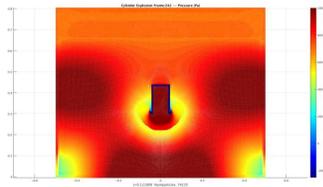
(n) $t=56\text{ms}$



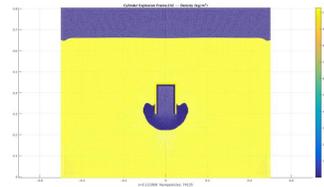
(o) $t=83\text{ms}$



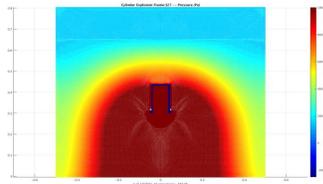
(p) $t=83\text{ms}$



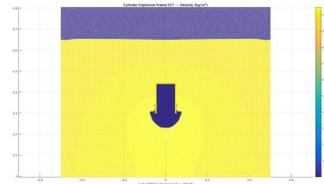
(q) $t=101\text{ms}$



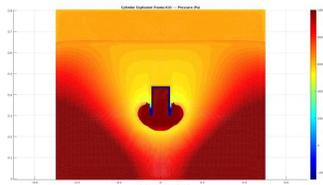
(r) $t=101\text{ms}$



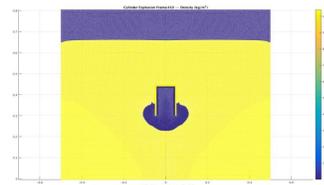
(s) $t=143.5\text{ms}$



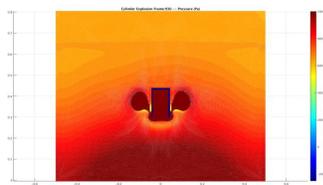
(t) $t=143.5\text{ms}$



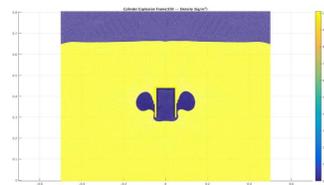
(u) $t=185\text{ms}$



(v) $t=185\text{ms}$



(w) $t=445\text{ms}$



(x) $t=445\text{ms}$

Figure 3.1: Deflagration Simulation Results (Left: Pressure, Right: Density)

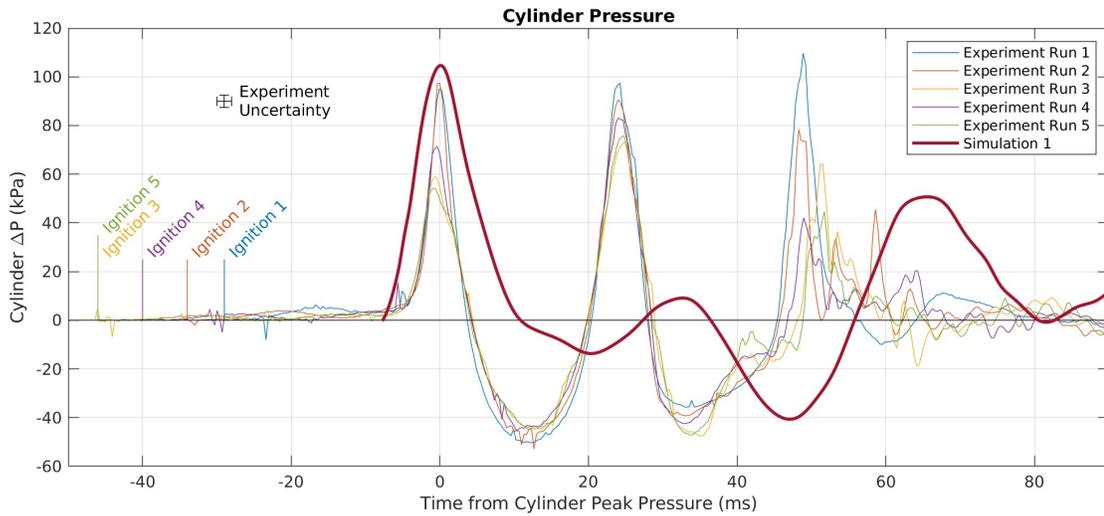


Figure 3.2: Simulation and Experiment Cylinder Pressure Plotted Relative to Peak Cylinder Pressure

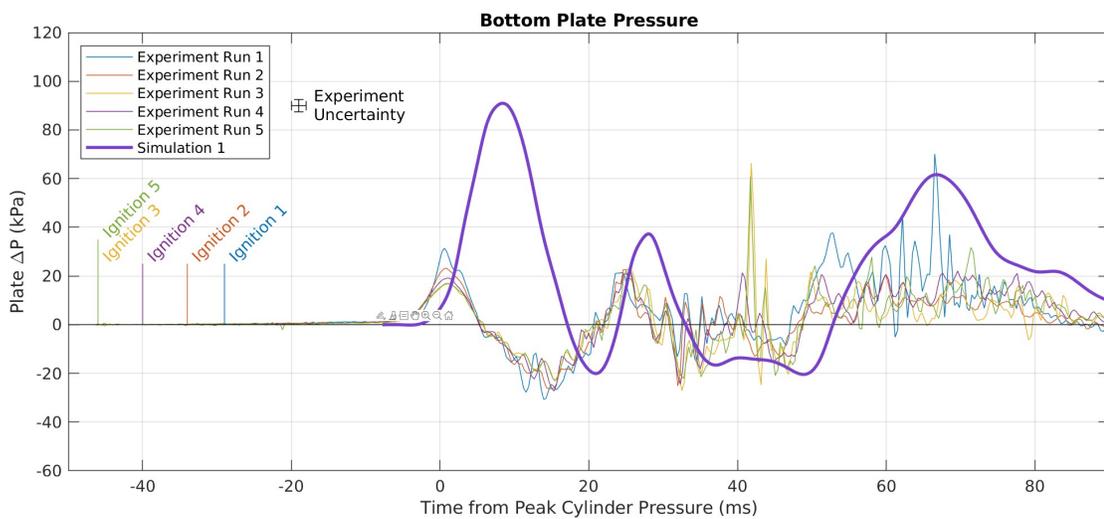


Figure 3.3: Simulation and Experiment Bottom Plate Pressure Plotted Relative to Peak Cylinder Pressure

Discussion

Because the fluid behavior is highly three-dimensional, no direct comparison between the 2-dimensional simulation and 3-dimensional experiment can be made. Additionally, while the model meant to approximate combustion functioned as intended, the uniform rate of reference density change did not match the true variable combustion rate of the experiment. Future work aims to implement a full 3-dimensional simulation of this flow. However, even with fundamental differences between the simulation and experiment, the original goal of analyzing the core capabilities of SPH to simulate the system was achieved.

During the development of the code, several areas of interest for further development of SPH became apparent. In cases where the local fluid support radius is larger than the thickness of a boundary two issues arise. The first is that in this situation, traditional ghost particles are insufficient to prevent boundary deficiency artifacts. Secondly, when the support radius is larger than the cylinder wall fluid on the outside of the cylinder is able to non-physically interact with gas on the interior of the cylinder. Careful control of particle smoothing length was performed to ensure no support radius would extend across a cylinder wall. This allowed for the simulation to proceed without non-physical fluid interaction across the cylinder. However, ideally, boundary methods should be developed which prevent cross-interface interaction and affect boundaries of any wall thickness. Work towards this goal is detailed in Chapter 4.

The next issue identified is the potential particle scale discontinuity at the fluid-gas interface which can emerge from the bulk expansion or contraction of the gas. As the gas expands, the initial particle representation of that gas rarifies, in extreme cases, this can cause the particle spacing in the gas to drop low enough to destabilize the simulation. Work done to simulate a pneumatic air-gun (a characteristically similar fluid flow) using SPH by de Graaf noted this instability for large bubble diameters [56]. A solution to this instability, proposed by Liu and Liu [17] and others, is to initialize a gas that is expected to expand with a significantly higher particle number-density than would otherwise be necessary or ideal. In this way when the gas expands, the particles will rarify and the local particle spacing

will reduce to a desirable level. This is the approach used in this simulation. However, this approach is inefficient in that it initially requires more particles than otherwise desirable to represent the gas discretization. A second drawback to this approach is that it requires advanced knowledge of the extent to which the gas will expand. A more ideal approach would be to dynamically add or remove particles to the deforming gas to enforce the local particle spacing in the gas is maintained at a prescribed level. This requires a method to vary resolution throughout a simulation. Chapters 5 and 6 discuss current methods of variable resolution using SPH and propose algorithms to expand this capability. Chapter 7 outlines a new algorithm to support the methods proposed in chapter 6.

Chapter 4: Boundary Deficiency Correction¹

4.1 SPH Boundary Behavior

In SPH simulations with uniformly distributed particles in an unbounded region, the SPH particle-discretized formulations approach the Navier Stokes governing equations as smoothing length approaches zero. The unbounded assumption is required to satisfy that fluid field values exist at all points within a particle's support domain. When a particle's support domain has an insufficient number of particles to adequately approximate the fluid field values within its support radius it is said to be integral or boundary deficient. One way in which this can occur is if the particle smoothing radius is set to a small value such that very few particles fall within a support radius, in this case, the SPH governing equations will yield poor approximations of field functions. Because of this, the smoothing radius is typically empirically related to particle mass and density such that an acceptable influence radius is maintained. A second more complicated manner in which integral deficiency can occur is observable when a particle is within close proximity to a fluid boundary. This situation is shown in Fig. 4.1, which plots particle acceleration of a finite 1D constant pressure fluid. While there exists no pressure gradient in the fluid, near-boundary particles exhibit a non-zero acceleration or boundary deficiency in acceleration.

This behavior is desirable at free surfaces for single-phase simulations as the boundary deficiency behaves identically to a zero gauge pressure fluid. However, boundary deficiency leads to difficulties at fluid-object boundaries. Relatively robust boundary conditions can be created by using repulsion forces based on spatial proximity (see [3][58][59]). However,

¹The work presented in this chapter was published in the journal *Ship Science and Technology : Ciencia y Tecnología de Buques* [57] in 2010

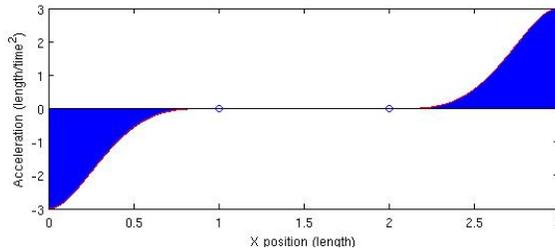


Figure 4.1: Boundary Deficiency in Acceleration (influence radius=1, length is unitless)

because of the lack of a pressure term to correct for boundary deficiency, proximity-based repulsion boundaries tend to yield a non-physical varying fluid particle to wall spacing at equilibrium. Typically this boundary deficiency is addressed by populating boundary regions with virtual particles whose properties are either fixed or derived from nearby fluid particles (see [60][61]). Robust boundary conditions which do not exhibit varying particle wall separation distance can be obtained by combining a spatial repulsive boundary with virtual particles [17][62][63]. However, for cases in which large boundary deformations can occur, virtual particle placement can become problematic due to virtual particle clumping. Clumping is a particle artifact in which particle spacing becomes skewed such that a directional spacing bias is present. This can result in an integral deficiency or surplus which increases the error of calculated field values. Fig. 4.2 shows an example of fixed virtual particle clumping due to boundary deformation. Visible in Fig. 4.2 is clumping resultant from deforming a boundary (shown in red). In the convex deformation case, near-boundary fluid particles experience a virtual particle integral surplus. Likewise, concave deformations lead to an integral deficiency.

It is therefore desirable to provide an alternative to virtual particles for boundary deficiency correction of highly deformable objects. While various correction methods exist which achieve similar results, they are often computationally expensive. Feldman and Bonet developed one such method to correct boundary deficiency for straight and corner boundaries by generating a curve-fit to boundary-deficiency accelerations [64]. While the goal is to achieve a boundary deficiency correction for arbitrary boundaries, it is advantageous to

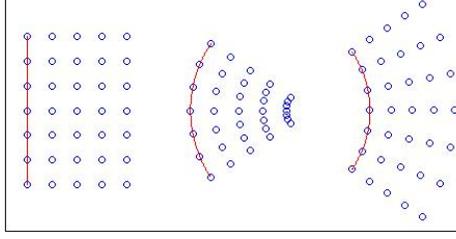


Figure 4.2: Clumping of Virtual Particles due to Deformation

first consider the simple case of a 1-dimensional boundary. By observation of the boundary deficiency of a one-dimensional fluid as in Figure 4.1, it is apparent that the boundary deficiency is similar in shape to the smoothing function. Analyzing equation 1.24 acting at a boundary and assuming constant pressure, density, and mass, the momentum equation can be reduced as shown in equation 4.1.

$$\frac{Dv_i^n}{Dt} = - \sum_{j=1}^N m_j \left(\frac{p_i}{\rho_i^2} + \frac{p_j}{\rho_j^2} \right) \frac{\partial W_{ij}}{\partial \vec{x}_{ij}} = - \frac{2mp}{\rho^2} \sum_{j=1}^N \frac{\partial W_{ij}}{\partial \vec{x}_{ij}} = - \frac{2Ap}{\rho} W_{ij} \propto CW_{ij} \quad (4.1)$$

Equation 4.1 shows that the acceleration boundary deficiency is proportional to the total area of the deficiency as well as fluid pressure. Because cases of interest typically involve non-constant pressures equation 4.1 is not strictly suitable for use as a boundary correction. However, proportionality to the kernel function for boundary deficiency can still be observed even in cases in which a pressure gradient is present. Figure 4.3 illustrates one such case. Shown in figure 4.3 is the vertical acceleration of a unity density fluid influenced by a downward body acceleration of unity magnitude.

The pressure at the top of the water column is zero (gauge pressure), with pressure in the fluid varying as $\frac{dp}{dz} = -\rho g$. The boundary deficiency in acceleration at the bottom of the water column still strongly correlates to the kernel function even in the presence of a

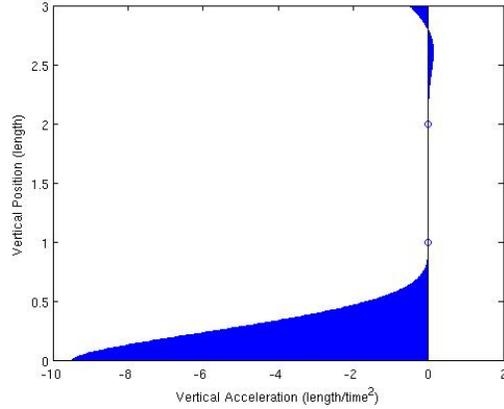


Figure 4.3: Clumping of Virtual Particles due to Deformation (length is unitless)

pressure gradient. It is interesting to note that a boundary deficiency exists not only at the bottom boundary but also at the free surface. However, the low magnitude boundary deficiency at the free surface does not introduce substantial error as it typically results in only minor particle clumping.

4.2 Acceleration Boundary Deficiency Correction

While an exact acceleration boundary deficiency correction could be gained by discerning the pressure gradient normal to a surface and analyzing the geometry of the deficiency, such an approach would increase computational complexity. Instead a simple -if inexact- correction is suggested in which the pressure is assumed to be nearly constant over the scale of the smoothing length. Then by assuming a known boundary acceleration, the relative fluid-boundary acceleration in the absence of boundary forces can be calculated. If this relative acceleration is assumed to be the result of a boundary deficiency then it can be corrected by applying a repulsive acceleration distributed as CW_{ij} away from the boundary. Where C is determined by choosing a sample particle, determining the relative particle-boundary normal acceleration, and dividing by W_{ij} (Eq. 4.3). To improve robustness it is advisable

to average C over a small sample of near-boundary particles. This reduces correction error due to spurious pressure fluctuations. Equation 4.2 shows this one dimensional acceleration boundary deficiency correction with j representing a boundary particle index.

$$\frac{Dv_i^n}{Dt \text{ correction}} = -C_j W_{ij} \quad (4.2)$$

$$C_j = \frac{1}{W_{ij}} \frac{Dv_i^n}{Dt \text{ uncorrected}} \quad (4.3)$$

The assumption that a boundary acceleration is known is suitable for fixed boundaries but requires an approximation when applied to freely moving boundaries. For objects with much greater density than the surrounding fluid, forward interpolation of acceleration is acceptable as object acceleration will change only slowly relative to the simulated timescales when acted on by fluid forces alone. However, as relative object-fluid density decreases, this approximation worsens. Further work is necessary to assess the impact of this assumption on low-density object dynamic behavior. While the correction presented in equation 4.2 is sufficient for one-dimensional cases where a single boundary particle governs a boundary deficient region, extension to higher dimensions requires a blending of corrections from multiple boundary particles. With Ω^* representing the set of all boundary particles within $r_{influence}$ of a boundary particle j . An empirical weighting function to blend correction values for two dimensional cases is shown in equation 4.4. Figure 4.4 shows a visualization of the resultant weights.

$$\frac{Dv_i^n}{Dt \text{ correction}} = -C_j W_{ij} weight_{j,normalized} \quad (4.4)$$

Where

$$weight_{j,normalized} = \frac{weight_{jk}}{\sum_{\Omega^*} weight_{jk} \hat{n}_j \hat{n}_k}$$

$$weight_{jk} = ActivationParameter_{jk}^2 + (GlancingParameter_{jk}^2 + \frac{1}{2})DistanceParameter_{jk}$$

$$ActivationParameter_{jk} = \frac{spacing}{\|\vec{r}_{ij}\|}$$

$$GlancingParameter_{jk} = \frac{\vec{r}_{ij}}{\|\vec{r}_{ij}\|} \cdot \hat{n}_{boundary}$$

$$DistanceParameter_{jk} = 1 - \frac{\|\vec{r}_{ij}\|}{r_{influence}}$$

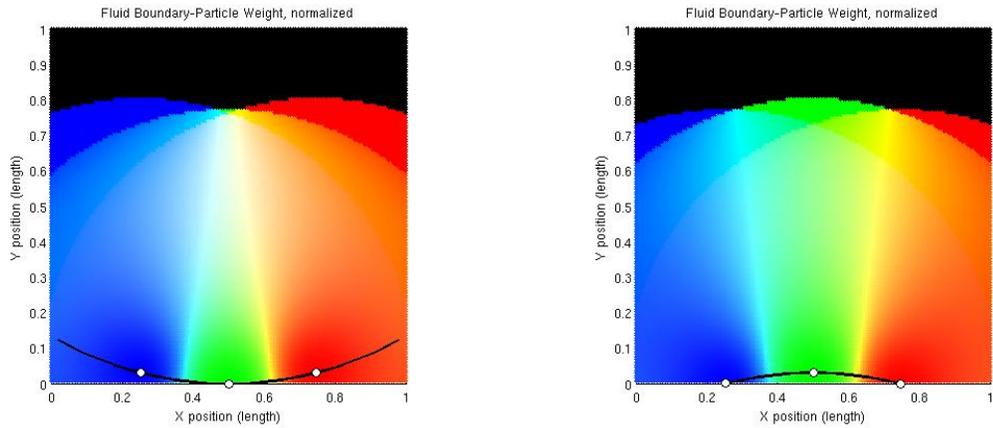


Figure 4.4: 2D Boundary Correction Weight Blending Visualization
(influence radius=0.775, boundary spacing=0.25, boundary radius of curvature=1)

Figure 4.4 illustrates the weighting scheme applied to simple concave and convex boundaries. The weights of the three boundary particles are shown by the blue, green, and red shades respectively. The boundary deficiency correction is applied to two test cases (one with a rectangular boundary, and one with an elliptical boundary) each with constant pressure fluid. Figure 4.5 shows the rectangular tank case. Fluid properties are set to representative values without units. Pressure and density are set to $p = 1.0$ (force per area) and $\rho = 1000$ (mass per volume, unitless). Particle influence radius set to 0.5 (length). The unmodified acceleration field is presented on the left and the corrected acceleration field is shown on the right. The normalized acceleration correction magnitude for each boundary particle is shown in blue and is plotted against boundary particle index (the zeroth boundary particle is located in the lower left with subsequent indexes moving counterclockwise about the rectangular tank). Figure 4.6 shows the elliptical tank case with identical fluid and simulation parameters.

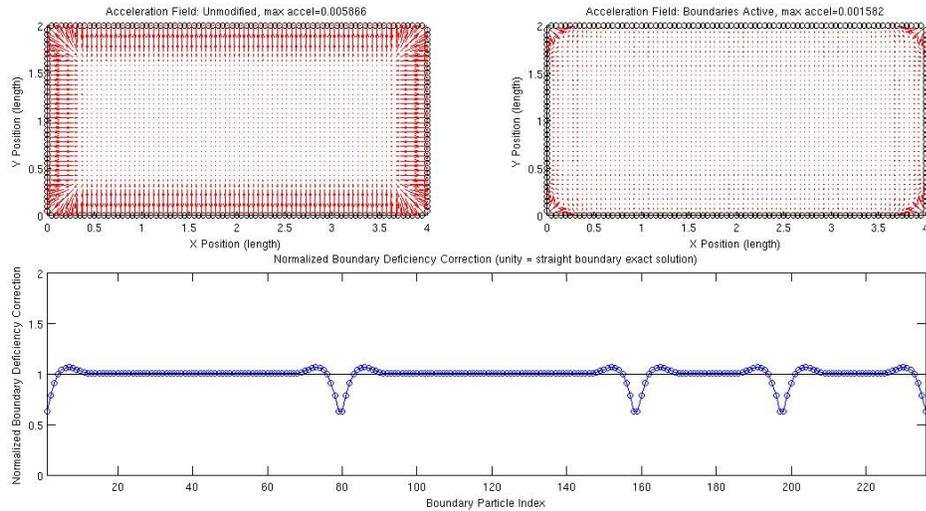


Figure 4.5: Rectangular Tank Boundary Correction Test Case (influence radius=0.5 spacing= $r_{influence}/10$)

The four dips in correction acceleration visible in figure 4.5 (rectangular case) are due to the change in boundary deficiency due to the reduced volume of fluid present near the

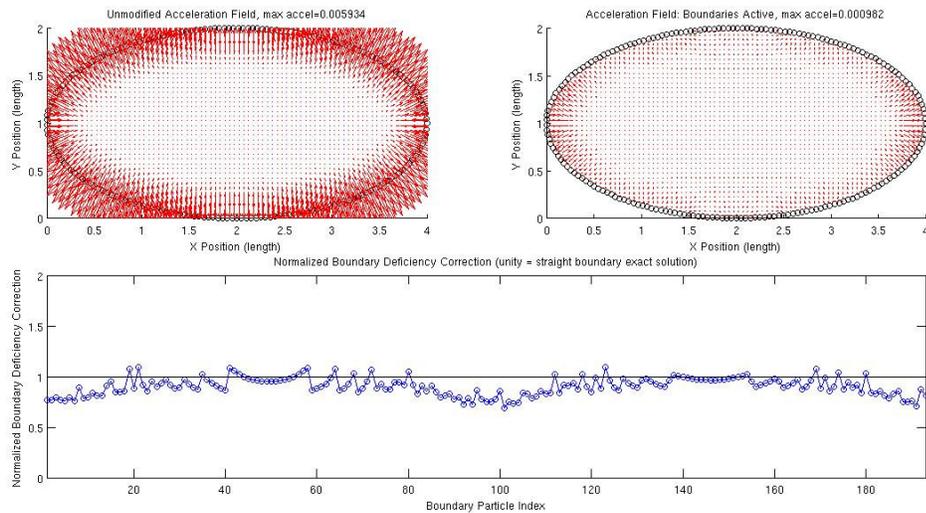


Figure 4.6: Elliptical Tank Boundary Correction Test Case
(influence radius=0.5 spacing= $r_{influence}/10$)

four tank corners. The correction slightly over corrects the boundary deficiency in the sharp corners of the rectangular case. For the elliptical tank case, the acceleration present after boundary deficiency correction near the high curvature sides of the elliptic tank represents an under correction of the boundary deficiency and is likely a result of the weighting function. Further refinement of the weighting function may reduce the error in the correction due to geometry. The low curvature and straight sections show good correction of acceleration boundary deficiency.

Because the correction presented emulates virtual particles, it alone is insufficient to act as a boundary condition. A secondary repulsion force such as the spatial repulsion force developed by Monaghan in 2009 [59] is necessary to prevent fluid-boundary penetration. The presence of virtual particles or a boundary acceleration deficiency corrective force can improve the behavior of spatial repulsive boundary forces. This is especially noticeable in the transient behavior present at the start of most simulations involving spatial repulsive forces alone. By correcting the boundary acceleration deficiency immediately, near-boundary fluid particles do not have to re-orient to allow for a spatial change to correct the boundary repulsion force.

Similar boundary acceleration correction work performed by Feldman and Bonet [64] does not require an assumption of a known boundary acceleration, but instead determines an acceleration correction by calculating the fluid pressure gradient and assuming a known boundary geometry (straight or straight-corner). Further work is required to compare the relative performance of the two methods when applied to various cases.

4.3 Summary

Virtual particles are normally used to correct errors due to integral deficiencies that appear in the governing equations near boundaries. Virtual particle behavior for deformable objects can be difficult to implement due to particle clumping after deformation. A simple repulsive correction that loosely emulates the presence of virtual particles in the momentum equation has been derived. An empirical weighting function to extend the theoretical boundary correction to higher dimension cases has been presented. Results obtained by applying the boundary correction to two constant pressure test cases were presented. The method yields good correction of acceleration boundary deficiency in regions of low curvature but tends to slightly overcorrect at sharp corners and undercorrect near regions with high curvature.

Since this work was published in 2010 [57] other researchers have developed alternate methods of addressing this issue. Notably work by Ferrand et al [65] in 2013 and Mayrhofer et al [66] in 2015 have provided highly adaptable algorithms for boundaries that correct for boundary deficiency errors.

Chapter 5: Spatial Filtering for Variable Spatial Resolution

5.1 Overview of Variable Spatial Resolution

As noted earlier, when a gas expands in a simulation in which there is a liquid-gas interface, the SPH particles representing the gas may rarefy. This can lead to lowering of local resolution, a loss of accuracy and, in extreme cases, instability. In order to prevent this from occurring, as well as improve computational efficiency throughout the domain, it is highly desirable to be able to modify fluid resolution throughout a simulation.

Modifying resolution in SPH requires several distinct steps. The first of these is to determine the desired local resolution throughout a domain. Once a target resolution is known, the task of generating a new discretization of domain volume can begin. While it is possible to generate a new discretization of a domain's volume using only information defining fluid boundaries, in many cases a resolution-adapted domain may be largely similar to a pre-existing discretization. Because of this, there exists a significant potential performance benefit to utilizing an existing domain discretization when generating a resolution adapted domain discretization. To capitalize upon the potential increase in performance, this approach is developed in this thesis.

When modifying an existing domain discretization to generate a discretization that adheres to the new target resolution it is necessary to determine and map the resolution of the existing domain discretization. Then, using the difference between the target and current resolutions, particles may be added or removed from the domain to yield the desired resolution.

Adding or removing particles however is a non-trivial task. If an additional particle is placed too close to an existing particle in a domain, then the information added by the new particle would largely be redundant and provide little effective increase in simulation

accuracy. Similarly, removing a particle in a given region without adjusting the positions of surrounding particles would leave a gap in an otherwise uniform distribution of datapoints describing the fluid. Both redundant particles or voids in particle distribution are undesirable. To address this, when adding or removing particles to a domain, it is desirable that the positions of the surrounding particles are adjusted in a manner to ensure similar local spacing between particles. In this thesis, spatial filtering is utilized to perform this task.

5.2 Spatial Filtering

Spatial filtering in SPH is the practice of changing the positions of particles to improve the numerical conditioning or physical consistency of a simulation. One of the most widespread spatial filtering methods is the XSPH algorithm. XSPH was developed by Monaghan to prevent non-physical particle-particle penetration [67]. It addresses cases where inadequate resolution or undamped numerical error causes a breakdown of continuity, leading to a domain state where an individual particle's velocity is such that it would overtake and pass through a neighboring particle. This non-physical behavior is undesirable. XSPH prevents particle penetration by updating particle locations by an averaged group velocity rather than from individual particle velocities.

Spatial filtering can also be used to prevent disordered or undesirable particle distributions. Disorder in particle spacing can decrease the computational efficiency, accuracy, and stability of a simulation [68] [69]. Ordered, but undesirable, particle distributions can similarly negatively affect simulation accuracy by overly packing particles along an axis and rarefying particles along a perpendicular axis. The mechanic which leads to this state is known as tensile instability. Tensile instability is a phenomenon in SPH which can lead to poor particle distributions. It describes a tendency for particles to directionally cluster together during a simulation. Figure 5.1 illustrates the directional clumping due to tensile instability in comparison to ordered and disordered particles.

Tensile instability (and particle penetration) primarily occur when using kernel functions that have a first derivative that decreases in magnitude at short range. This kernel

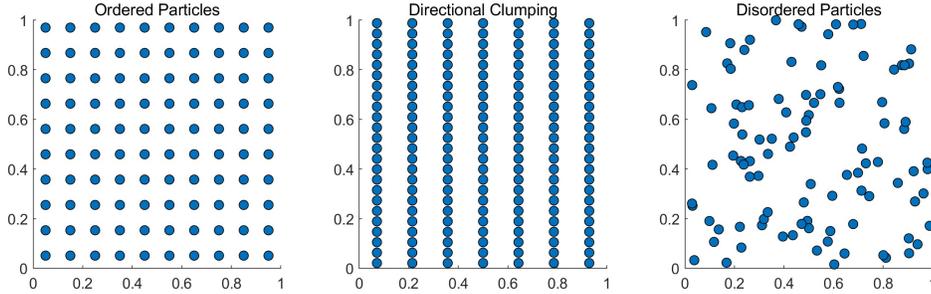


Figure 5.1: Ordered and Disordered Particles and Clumping Due to Tensile Instability

function characteristic is generally desirable in SPH as it prevents rapid fluctuations in interparticle forces as particles pass near one another. This can be seen by considering the Euler governing equation in discretized using SPH (eq. 5.1).

$$\frac{Dv_i^\alpha}{Dt} = - \sum_{j=1}^N m_j \left(\frac{p_i}{\rho_i^2} + \frac{p_j}{\rho_j^2} \right) \frac{\partial W_{ij}}{\partial \vec{x}_{ij}} \quad (5.1)$$

The acceleration from interparticle forces in this governing equation is weighted by the first derivative of the kernel function W_{ij} . Because Gaussian-like kernel functions are designed with a reducing first derivative as inter-particle distance approaches zero (see Fig. 1.1) the $\frac{\partial W_{ij}}{\partial \vec{x}_{ij}}$ term in eq. 5.1 will likewise approach zero for short-range particle interactions. While this design decision in the selection of the kernel function imparts desirable numerical characteristics for stability and continuum behavior, it allows for the undesirable emergence of particle penetration and tensile instability.

Tensile instability most commonly appears when ordered particles undergo non-symmetric deformation or shearing. Disordered particles do not exhibit tensile instability as tensile instability is itself an ordered particle topology. The directionally clustered particles which arise from tensile instability are undesirable as while they might provide high resolution along a clustered direction, resolution in the transverse direction may be greatly reduced.

In extreme cases, this local directional loss of resolution can lead to catastrophic instability.

Multiple mechanisms have been proposed to address tensile instability. Monaghan proposed a short-range inter-particle repulsive force to address this issue [70]. However, another approach is to perturb particle positions by a small fraction of their smoothing length without altering their simulated velocities. This approach has the advantage of not altering fluid compressibility but comes with disadvantages in that it could potentially introduce small variations in angular momentum.

5.3 Expanded Circle Filter

To address tensile instability and particle disorder while maintaining continuity, the Expanded Circle Filter algorithm was created by the author. However, because the addition or removal of particles to a domain can lead to the emergence of local regions of poorly distributed particles, the filter also has application to resolution adaptation. The method also provides free surface particle identification. This added capability is important to spatial filtering and resolution adaptation, as in both cases it is desirable to minimize any alteration of a fluid's free surface profile when perturbing particle positions. Many methods to identify free surface particles have been developed. Works devoted specifically to SPH include Marrone et al [71], Zheng et al [72], and others. Additional methods have been developed for free surface identification in the context of other particle methods, creating a large array of methods aimed at this task. However, the Expanded Circle Filter algorithm is novel in that it both identifies free surface SPH particles and determines updated positions for improved particle distribution in a single method.

The principle objective of the Expanded Circle Filter algorithm is to identify the local position at which the minimum distance of a subject particle to all of its neighbors is maximized. Moving a particle towards this position will equalize the spacing between it and its neighbors at a local level.

Because the filter is applied in an uncoupled manner, any update to a single particle will invalidate the solution for its neighboring particles. Because of this particle movements

must be made with a relaxation factor. Once the desired positions ($x_{c,i}$) for all particles in a domain have been determined, particles are moved towards these positions by Equation 5.2, where α is a relaxation factor. Future work is required to determine optimal relaxation values for convergence, however, values as high as 0.95 yield stable convergence. The algorithm is repeated throughout the domain until convergence is detected. In this way, particles will move towards the local positions in which the distance to any nearest neighbor is maximized.

$$\vec{x}_i^{k+1} = \vec{x}_i^k + \alpha(\vec{x}_{c,i} - \vec{x}_i^k) \quad (5.2)$$

An alternate description of the algorithm's objective of maximizing the distance to all particle neighbors is finding the maximum circumscribed circle which includes only the subject particle. Thus, the algorithm may be visualized by considering a circle that originates at a subject particle's position and then grows until it is tangent to the subject particle's neighbors and is constrained by those points such that no further increase in diameter is possible. Figure 5.2 illustrates this concept for a single particle surrounded by four neighbors. In each iteration, the circle expands either radially outward or along a vector perpendicular to a search plane. The algorithm concludes when the circle is constrained by surrounding points and cannot expand further without circumscribing at least one point other than the subject particle. The function's output is the center of this maximum circumscribed circle. This approach shares some characteristic similarities to Delaunay triangulation. However in Delaunay triangulation circumscribed circles contain no interior points, while in the Expanded Circle Filter precisely one point is required to be interior to the circumscribed circle.

Figure 5.3a demonstrates the Expanded Circle Filter when applied to a random particle distribution. In each subfigure, free surface particles are shown in red. In the initial domain state, the particles are highly disordered. Within five iterations of the filter the particles domain exhibit nearly uniform particle spacing, with only small variations in particle number density visible. However, it is notable that multiple regions show small-scale defects in

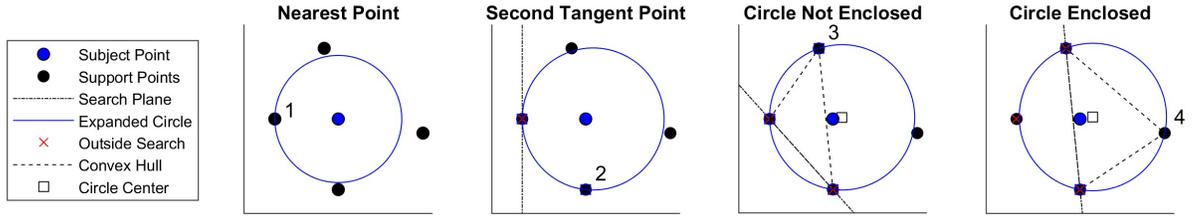
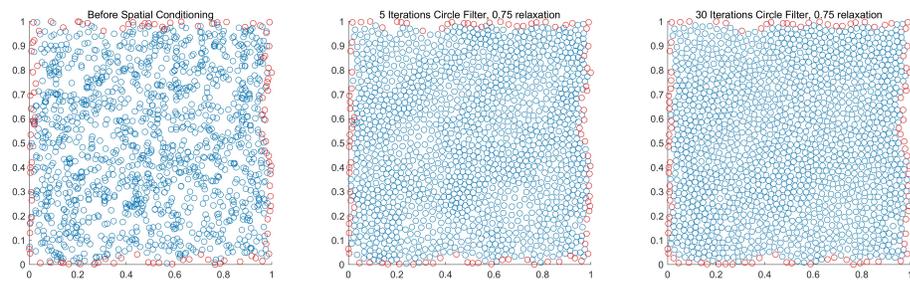


Figure 5.2: Expanded Circle Filter Example

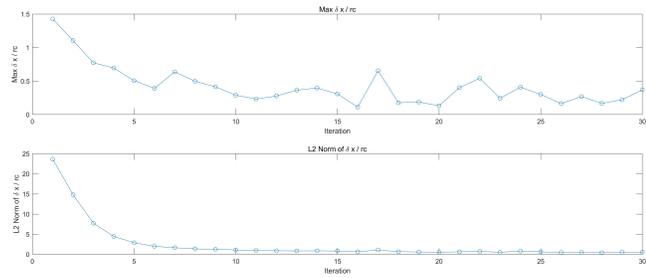
the otherwise ordered particle distribution. These defects are loosely analogous to vacancy defects in 2D crystal lattice structures and emerge as a consequence of the design of the filter. By thirty iterations nearly all local variance in particle number density has vanished, though the particle vacancy defects remain.

In order to track the filter’s convergence the magnitude of the distance between each particle and the center of that particle’s maximum circumscribed circle is considered. This distance is non-dimensionalized by the radius of the circumscribed circle determined by the Expanded Circle Filter algorithm. Figure 5.3b shows the maximum and L^2 norm of this non-dimensionalized particle movement for each iteration of the filter. While localized particle translations are observable even after multiple iterations of the filter, the domain-wide L^2 norm of particle movement exhibits rapid convergence.

The Expanded Circle Filter may also be applied to particle distributions with spatially varying resolution. This is illustrated in Figure 5.4. In Figure 5.4, a highly disordered set of particles is considered. However in this case the distribution is not random. But includes an increasing gradient of particle number density in the horizontal direction. Three iterations of the Expanded Circle Filter are shown. With each iteration, the local spacing between particles is maximized, providing a favorable distribution of particles. After three iterations the gradient in number density is visible, with an increasing number of particles per area on the rightmost edge of the domain. Despite the Circle Filter driving particles towards a local uniform spacing, the filter effect acts at a local level and does little to disrupt large-scale patterns in particle number density.



(a) Filter Iterations



(b) Convergence Behavior

Figure 5.3: Expanded Circle Filter Spatial Conditioning

In each iteration, particle positions are updated by eq. 5.2, where $x_{c,i}$ is the center of the expanded circle determined the filtering algorithm for particle i . Three iterations of the updated particle positions are shown. Free surface particles are plotted in red.

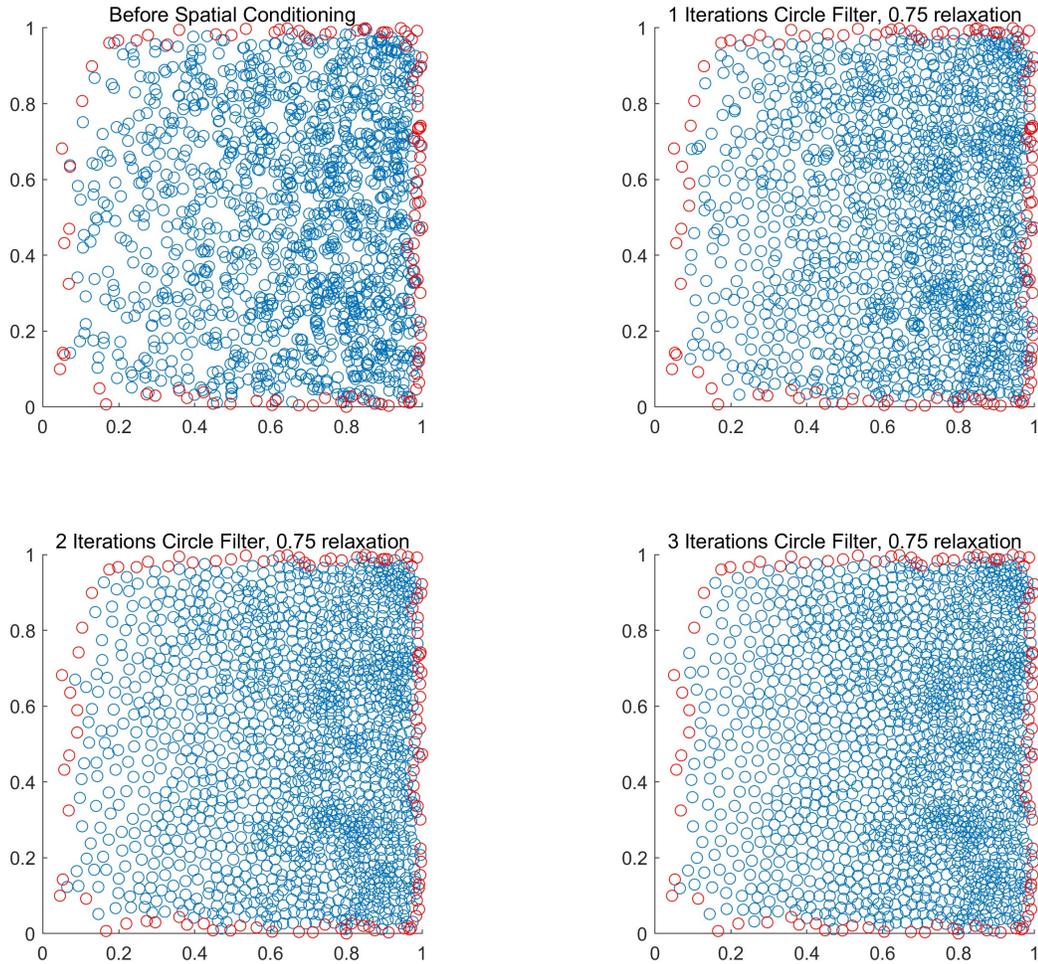


Figure 5.4: Expanded Circle Filter Spatial Conditioning - Gradient of Particle Number Density

Figure 5.2 provides a detailed analysis of the algorithm when applied to a single particle surrounded by four neighbors. In each step of the algorithm a new tangent point is identified and the position of the expanded circle is updated. The process continues until a maximum diameter circumscribed circle is found. This condition is reached when the center point of the circle is enclosed within a polygon constructed from the tangent points. In this state,

the angle between any two tangent points must be less than 90 degrees and no direction exists in which the circle could expand without a neighboring particle becoming interior to the circle.

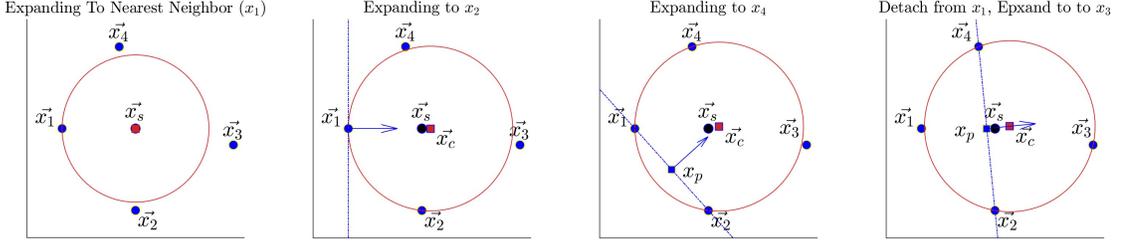


Figure 5.5: Expanded Circle Filter Detailed Iterations

To find the maximum circumscribed circle the Expanded Circle Filter employs an iterative scheme detailed in Algorithm 1. Figure 5.5 provides a detailed illustration of the algorithm iterations applied to an example case. In the example, the method begins by identifying the nearest neighbor to the subject particle by eq. 5.3. This nearest neighbor defines the first potential tangent point of the expanded circle and its position and index are stored as $x_{t,1}$ and $j_{t,1}$. The circle center is defined as x_c and is at this step coincident with the subject particle. Next, a plane is defined by the position of $x_{t,1}$ with a normal (\hat{n}) oriented towards the circle center. Algorithm 2 is then used to determine the minimum circle which is tangent to any neighboring point j and this search plane. Particles behind the search plane are ignored. This particle then defines the second tangent point of the expanded circle as $x_{t,2}$. With two potential tangent points $x_{t,1}$ and $x_{t,2}$ known, a new search plane is defined by these points with its origin at the midpoint of these points and a surface normal \hat{n} perpendicular to the surface and pointing towards the current circle center x_c . Algorithm 2 is again used to identify the smallest circle which is tangent to $x_{t,1}$ and a neighbor particle and lies along the line created by x_p and \hat{n} . The algorithm is continued until a triangle that encloses the subject particle is generated by the set of tangent points or no valid expansion point exists. A max iteration constraint is also included, however in most instances a particle will be declared unbounded or constrained within four to five

iterations. In the case that a subject particle is enclosed by a set of tangent points that particle is deemed to be interior to the fluid domain. Particles not enclosed are considered free surface particles.

Algorithm 1 Circle Filter

Require: N is the number of neighboring particles
Require: $[\vec{x}]$ is an array of the positions of neighboring particles
Require: \vec{x}_s is the position of the subject particle
Require: r_{search} is the search radius that contains all neighbors
if $N < 3$ **then**
 return: Subject Particle is Freesurface
end if
Find nearest neighbor and designate it as $x_{t,1}$ (eq. 5.3)
Set expansion plane position x_p to that of the nearest neighbor $x_{t,1}$
Set expansion plane normal vector to $x_{t,1} - \vec{x}_s$
while $iteration < \text{MAX_ALLOWED_ITERATIONS}$ **do**
 Find Minimum Tangent Circle From Points and Plane Origin (Alg. 2)
 if No Valid Tangent Point Exists **then**
 return: No Valid Circle, Subject Particle is Free Surface
 end if
 Store New Tangent Point $x_{t,k}$ and Radius r_k (where k is the number of tangent points)
 if The expanded circle extends beyond r_{search} **then**
 return: No Valid Circle, Subject Particle is Free Surface
 end if
 if Three tangent points have been found **then**
 if The circle center lies within triangle $[x_{t,1}, x_{t,2}, x_{t,3}]$ **then**
 return: Circle Coordinates, Subject Particle is Interior
 else
 Remove the tangent point on the far side of triangle $[x_{t,1}, x_{t,2}, x_{t,3}]$
 end if
 end if
 set \vec{x}_p to the midpoint between the two most recent tangent points $x_{t,k}, x_{t,k-1}$ and \hat{n} as the surface normal of the plane defined by these points
end while

$$r_1 = \min(\sqrt{(x_j - x_p)^2 + (y_j - y_s)^2}) \quad (5.3)$$

$$r_2 = \frac{(r_1 + \sqrt{(x_j - x_{t,1})^2 + (y_j - y_{t,1})^2})}{2} \quad (5.4)$$

$$(x_s - x_p)(x_j - x_p) + (y_s - y_p)(y_j - y_p) > 0 \quad (5.5)$$

Algorithm 2 Minimum Circle Tangent to a Point and Plane at a Point

Require: N is the number of neighboring particles

Require: \hat{n} is a vector normal of the search plane

Require: \vec{x}_p is the origin point of the search plane

Initialize:

$$r_{min} = 0.0$$

▷ Minimum radius of tangent circle

$$j_{min} = -1$$

▷ Index of minimum radius particle

for $j = 1 : N$ **do**

if Particle j lies in front of the search plane (eq. 5.5) and within circle \vec{x}_c, r_j (if it exists) **then**

 Find circle \vec{x}_c, r_j , tangent to plane origin \vec{x}_p and point j (eq. 5.6)

if $j_{min} == -1$ or $r_j < r_{min}$ **then**

$$r_{min} = r_j$$

$$j_{min} = j$$

end if

end if

end for

return: Minimum-radius and particle index

$$x_{c,j} = -\frac{-n_x x_{t,1}^2 + n_x x_j^2 - 2n_y x_p y_{t,1} - n_x y_{t,1}^2 + 2n_y x_p y_j + n_x y_j^2 + 2n_x y_{t,1} y_p - 2n_x y_j y_p}{2(n_x x_{t,1} - n_x x_j + n_y y_{t,1} - n_y y_j)} \quad (5.6a)$$

$$y_{c,j} = -\frac{-n_y x_{t,1}^2 + n_y x_j^2 + 2n_y x_{t,1} x_p - 2n_y x_j x_p - n_y y_{t,1}^2 + n_y y_j^2 - 2n_x x_{t,1} y_p + 2n_x x_j y_p}{2(n_x x_{t,1} - n_x x_j + n_y y_{t,1} - n_y y_j)} \quad (5.6b)$$

$$r_j = \sqrt{(x_j - x_{c,j})^2 + (y_j - y_{c,j})^2} \quad (5.6c)$$

5.4 Additional Notes and Future Work

It can be observed that the movement of interior particles near the free surface can occasionally cause particles near the free surface to change their classification between interior or free surface. This leads to very slight changes in the free surface profile in concave sections of the free surface. However, overall free surface particle positions remain fixed and the profile of the free surface is maintained.

Currently, an edge case exists for a uniform rectangular grid. In this configuration, any three surrounding particles define a polygon that does not enclose the subject particle. Instead in this edge case, the subject particle will lie directly on an edge of any triangle defined by three of its neighbors. This causes interior particles in this configuration to be misidentified as free surface particles. Future work will seek to handle this edge case, however, this edge case can be avoided by perturbing particles in a random direction with a magnitude near to the machine working precision.

This algorithm adds to the capability of SPH to simulate multiphase flows such as the underwater deflagration test case by removing local particle disorder and clumping due to tensile instability. It also has potential application to re-positioning particles after variable resolution algorithms have added or removed particles to a domain. Future work in this area will seek to utilize this functionality to both maintain a constant resolution while a gas rarefies as well as locally vary simulation resolution to match locally desired simulation accuracy.

The Expanded Circle Filter provides a robust mechanism for re-positioning disordered or clumped particles in a domain. The filter has applications both for addressing the loss of accuracy due to particle disorder or tensile instability and for the spatial filtering of domains after the addition or removal of particles. However, to be useful in engineering applications it must be expanded to three-dimensional space. This will expand the triangle which is circumscribed by an expanded circle to a tetrahedron circumscribed by a sphere. This expansion to three-dimensional space will require little modification to the algorithm itself, however, a new formulation for the sphere center will need to be developed to replace

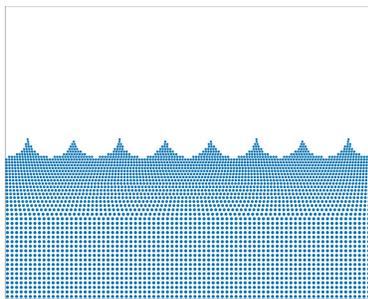
the formula presented in eq. 5.6.

Chapter 6: Interdomain Property Transfer for Variable Resolution

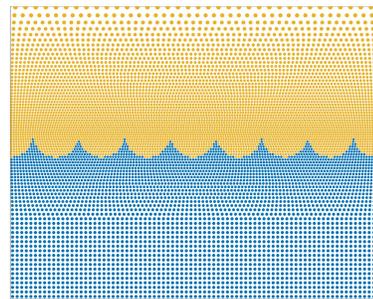
6.1 Domain Discretization

One approach to restoring lost resolution due to particle rarefaction in cases of expanding gas, such as the underwater deflagration simulation, is to reinitialize the entire domain. This section presents an overview of SPH domain discretization and presents a new method to transfer information from an initial to a resolution modified domain.

In order to apply numerical methods to any continuum, a finite representation of the continuum must be made. The first step in this process is to reduce an infinitely bounded region in space or time to a finite computational domain with known boundary conditions. This computational domain can then be discretized by subdividing it into a finite number of points, particles, volumes, or elements which together represent an approximation of the continuum in the computational domain.



(a) Single Phase Domain



(b) Multiphase Domain

Figure 6.1: Example SPH Domain Discretizations

Domain discretization schemes for CFD methods broadly fall into two categories, with each requiring a solver tailored to the approach. For single-phase CFD methods (Fig. 6.1a), only the movement of a single material is considered. Regions of the domain not filled with the tracked liquid or deformable solid are assumed to be filled with vacuum or a low-density gas at a constant reference pressure. The dynamics of any assumed gas in these regions is ignored. In multiphase domains (Fig. 6.1b), all fluids within a domain are discretized and modeled.

Single phase methods are well suited for astrophysics simulations, as treating far-field regions of the domain as empty is largely consistent with the near vacuum of space. When applied to terrestrial fluid flows, single-phase methods can often model fluid flows at a reduced computational cost compared to equivalent multi-phase methods. This computational cost reduction is a consequence of both the reduced complexity of single-phase solvers relative to an equivalent multi-phase method as well as the reduced size of the modeled domain.

However, if single-phase methods are used to simulate flows where gas dynamics have a significant factor in overall system behavior then any simulation results will poorly model to true behavior of the flow. Additionally, because single-phase solvers typically utilize an assumed reference pressure in empty regions, empty regions of the domain which might emerge due to cavitation may be non-physically modeled.

Particle-based domain discretizations for SPH can be performed using many different methods. One of the simplest approaches is to distribute particles in a region using an ordered pattern. With this method, each particle volume is determined by the known volume associated with the spacing used in the distribution. Figure 6.2 illustrates a volume filled with two sets of regularly spaced particles of varying volume. The boxes around each particle show the volume assigned to each particle in the region. Because SPH is a mesh-free method, the misalignment of the spacing used to distribute particles has no negative effect on any resultant simulation.

Another method of generating initial domain discretizations for SPH simulations is to

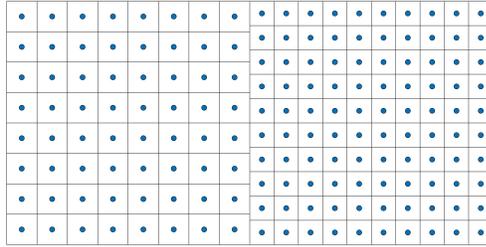


Figure 6.2: Particles Generated using Uniform Rectangular Distributions

use a Finite Volume mesh to discretize the computational domain and then transform that mesh into a particle discretization by generating particles at mesh nodes or element-centroids and using element volumes to determine particle volumes. Figure 6.3 shows an example this approach.

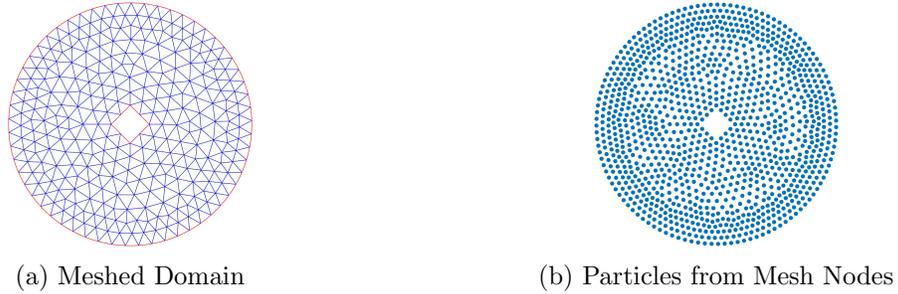


Figure 6.3: Particle Generation from a Mesh

While these and other methods of domain discretization are useful for generating initial particle distributions for an SPH simulation, they are not well suited for altering an existing fluid discretization.

6.2 Dynamic Domain Adaptation

Methods to dynamically modify an SPH domain discretization broadly follow one of two approaches. In the first, properties from an existing domain discretization are projected onto a modified discretization. Chaniotis et al first proposed a method to dynamically modify an SPH domain discretization in this manner [73]. In their Remeshed Smoothed

Particle Hydrodynamics (rSPH) method, particles are allowed to convect a short distance from their initial positions before domain properties are interpolated back onto a domain discretization using the original particle positions. In this way, properties can propagate and convect through the domain, but particle topology remains fixed. Koumoutsakos later adapted this approach to allow for a domain composed of regions of differing resolutions [74]. This approach is highly flexible, but may not strictly conserve fluid properties.

An alternate method to modify an existing domain is to discretely split or coalesce particles. Vacondio et al [75] and Xiong et al [76] developed and applied methods to discretely split particles when moving into a region of higher desired resolution. This approach has an advantage in that mass, linear momentum, and similar properties are intrinsically conserved as these properties are distributed or combined when particles are split or merged. Winchenbach et al developed a hybrid method that utilizes particle splitting as well as conserved property distribution to strictly conserve mass and momentum [77].

6.3 Iterative Conserved Property Transfer

In order to facilitate the transfer of information from an existing domain discretization to a resolution-adapted domain discretization while strictly conserving the sum of discrete fluid properties, a new iterative property transfer method was developed by the author. The method involves two stages. In the first stage, continuous fluid properties are interpolated onto the modified domain discretization yielding target continuous properties for that discretization. Then, discrete, conserved fluid properties are distributed such that the local error between calculated continuous particle properties and target properties is minimized. This algorithm has the same end objective of variable resolution with strictly conserved properties as demonstrated by Winchenbach et al, however, in Winchenbach’s work, no target continuous property is used to guide the distribution of conserved properties to neighboring resolution-adapted particles.

A continuous property U in the iterative conserved property transfer method must be related to two corresponding discrete fluid quantities u and d , where U_j for any discrete

particle j is defined by the ratio of the two conserved fluid properties (eq. 6.1). Here i will be used to indicate particles in the initial domain discretization (A) and j will indicate particles in the new discretization (B). Continuous properties U in this context describe fluid properties such as density, velocity, or temperature. For a single particle, density is the ratio of that particle's mass and volume. Likewise, velocity is the ratio of a particle's momentum and mass. And temperature is the ratio of internal energy and mass.

$$U_{derived,j} = \frac{u_j}{d_j} \quad (6.1)$$

The method requires an initial domain A and a blank domain discretization B with known particle volumes. Figure 6.4 illustrates an initial domain A with an approximately Gaussian distribution (eq. 6.4) of density and a higher resolution empty discretization B of the same domain.

The first property transfer using the iterative method must be mass/density, as the only known quantity in the blank discretization is particle volume. A target density $U_{b,i}$ is interpolated from the initial domain to the modified discretization by Equation 6.2 and 6.3. Here $\Omega_{A,i}$ is the support domain in discretization A of particle i (which resides in discretization B).

$$U_{B,target,i} = \frac{1}{W_{ij,sum,B,i}} \sum_{j \in \Omega_{A,i}} U_{A,j} W(\|\vec{x}_i - \vec{x}_j\|) V_{A,j} \quad (6.2)$$

$$W_{ij,sum,B,i} = \sum_{j \in \Omega_{A,i}} W(\|\vec{x}_i - \vec{x}_j\|) V_{A,j} \quad (6.3)$$

$$\rho(x, y) = e^{-(x^2+y^2)} \quad (6.4)$$

With the target density interpolated onto the new domain discretization, weight values for particles in the target domain are set to an initial value of unity. Particle weight sums

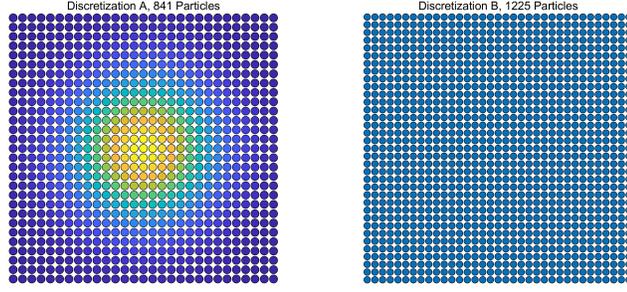


Figure 6.4: Initial and Blank Domain Discretizations

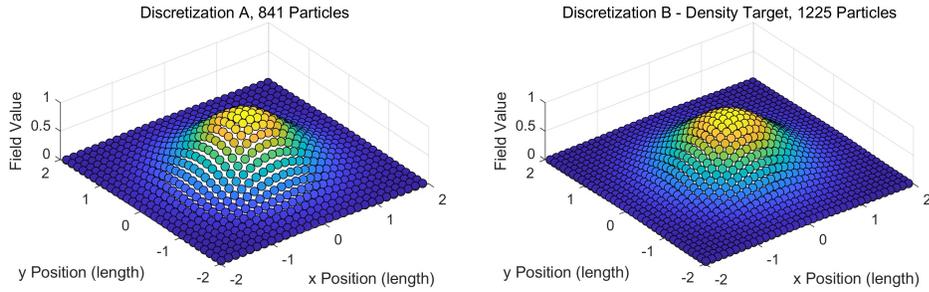


Figure 6.5: Continuous Property Target U Interpolated to Domain B

are determined by eq. 6.5. Mass is then distributed from particles in the original domain to the new domain using weights and weightsums by eq. 6.6. However, the mass transferred using these initial, uniform weight-values will result in a density field which does not match the desired target density.

$$XferWeightSum_{A,i} = \sum_{j \in \Omega_{B,j}} W(\|\vec{x}_i - \vec{x}_j\|) XferWeight_j \cdot V_{B,j} \quad (6.5)$$

$$u_{B,i} = \frac{1}{XferWeightSum_{A,i}} \sum_{j \in \Omega_{A,i}} u_{i,A} \cdot W(\|\vec{x}_i - \vec{x}_j\|) XferWeight_j \cdot V_{B,j} \quad (6.6)$$

It is therefor necessary to adjust the particle weights in discretization B such that mass transferred using the weights will result in a density field which more closely approximates the target density. Modifying particle weights by eq. 6.7 increases weights, and subsequently

mass-transfer, to particles which are at a lower derived density than their density target.

$$XferWeight_j^{k+1} = XferWeight_j^k \cdot \left(1 - \frac{U_{derived,j} - U_{target,j}}{U_{target,j}}\right) \quad (6.7)$$

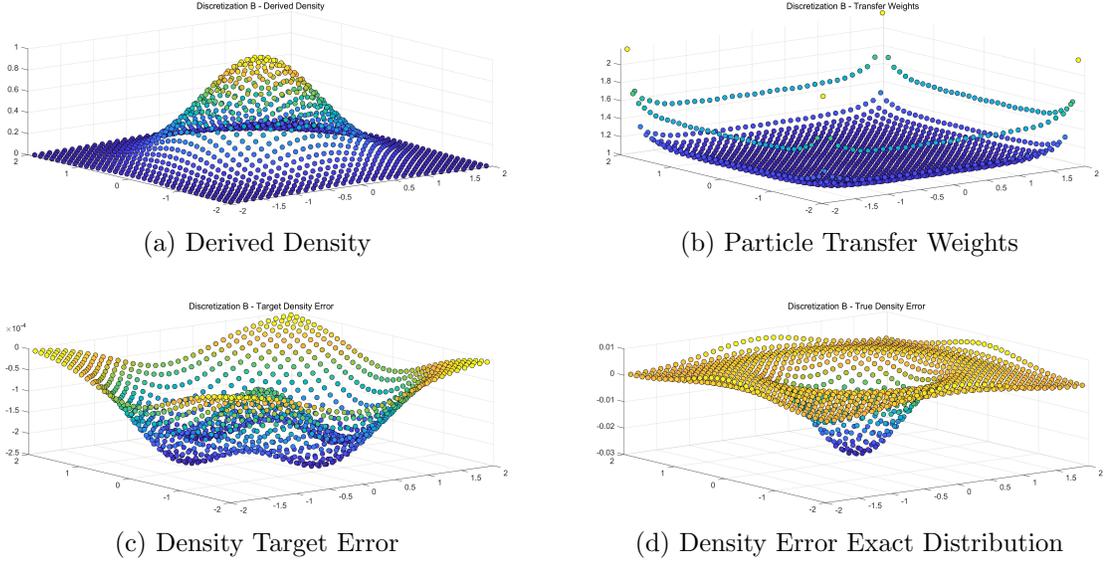


Figure 6.6: Expanded Circle Filter Spatial Conditioning

This process is repeated iteratively, until convergence is detected in the change to properties. Figure 6.6 shows the results of density transferred from discretization A to discretization B after 20 iterations. Particle weights (Fig. 6.6b) near boundaries converge to higher values than those of particles in the interior of the domain to attain a similar quantity of distributed conserved properties. Future work should utilize this known characteristic to generate a better initial particle weight estimate. Because the initial domain is a coarse discretization of the density distribution function (6.4) the total amount of mass added to the initial domain is an imperfect representation of the initial density function. This can be seen in 6.6d, where a domain-wide density deficiency is visible.

Transfer of further conserved properties follows the same process, though with the substitution of mass instead to relate velocity and momentum or temperature and internal

energy.

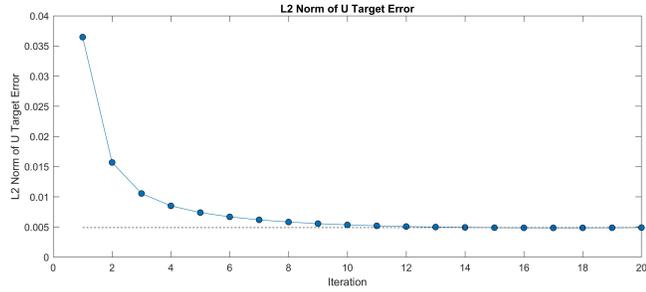


Figure 6.7: L_2 Norm of Density Target Error

Figure 6.7 shows the L_2 norm of the derived density error relative to the target density. Because the interpolated target density is not necessarily attainable with the total amount of mass in the domain, it is expected that L_2 target error will be impossible to drive to zero. The point iteration scheme used to alter local particle weights also does not guarantee an always reducing L_2 error. To apply this to practical flows, a methodology is required to infer particle volumes throughout the domain, thus providing the motivation for the algorithm detailed in chapter 7.

Chapter 7: Voronoi Tessellation

7.1 Voronoi Tessellation Overview

The Iterative Conserved Property Transfer algorithm of section 6.3 requires known particle volumes. However, not all methods of producing particle distributions throughout a domain include particle volume values. Additionally, while mass is strictly conserved in a weakly compressible SPH simulation, volume is not. Because of this, it is desirable to have a method of determining non-overlapping particle volumes for any distribution of points throughout a domain. This is a requirement for many numerical methods and researchers have developed multiple algorithms for this purpose. One of the oldest numerical methods for partitioning space between a set of points is Voronoi tessellation. Voronoi tessellation dates back to at least 1644 when Descartes used the method to describe regions of vortices. Many other researchers utilized functionally identical approaches, including Kepler, Snow, Dirichlet, and Gauss [78]. However, the method is named for Voronoi who formalized and expanded the method to n-dimensional space in 1908 [79].

In a classical Voronoi tessellation, space is partitioned by assigning points in a domain to a cell associated with the nearest particle to that point. The bounds of these emergent regions define a convex hull around each particle. Equation 7.1 shows the mathematical form of this definition. Where X is a defined space, $d(x, P_k)$ is the distance function of any point in space x from the point associated with region P_k .

$$R_k = \{x \in X | d(x, P_k) \leq d(x, P_j) \text{ for all } j \neq k\} \quad (7.1)$$

Figure 7.1 shows a classical Voronoi tessellation of a set of random points, as well as the centroids of each resultant Voronoi cell. While very useful for discretizing space in

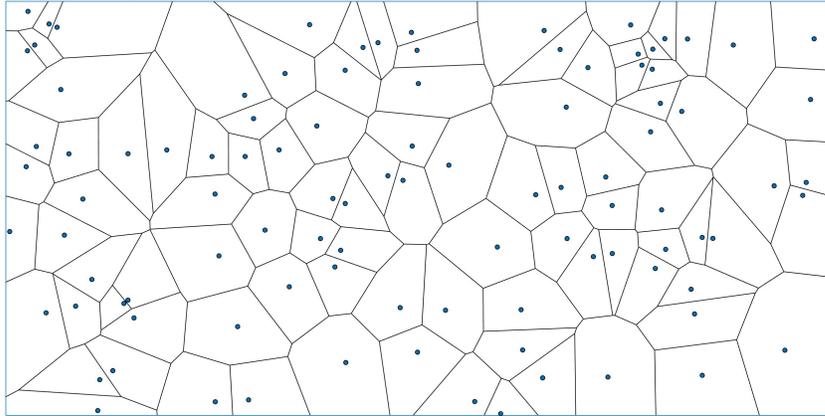


Figure 7.1: Voronoi Tessellation of Space from Points

a domain between a set of points, Voronoi discretizations are defined in a manner such that cells may be infinitely large. This has significant drawbacks when applied to SPH. When determining the Voronoi partition of any given particle any other particle in the domain may define a cell boundary. Because of this, unless special precautions are made to ensure closed Voronoi cells are formed, Voronoi methods have non-compact support and require the analysis of every point in the domain when generating a single cell. When parallelizing particle codes compact support is a highly desirable feature, as it guarantees consistency when considering only a limited set of points. To this end, it is desirable to create a modified Voronoi tessellation method that exhibits compact support behavior. If complete tessellation of space is not required -or desirable- then a simple modification of a Voronoi algorithm may be made to ensure compact support. This modification is to limit any particle's Voronoi region to a space within a finite radius.

7.2 Finite Radius Voronoi

The Finite Radius Voronoi algorithm is a new method developed by the author which performs this modified discretization of space. The method works by identifying bisection planes that separate the space in a domain between any two particles. Then, any regions

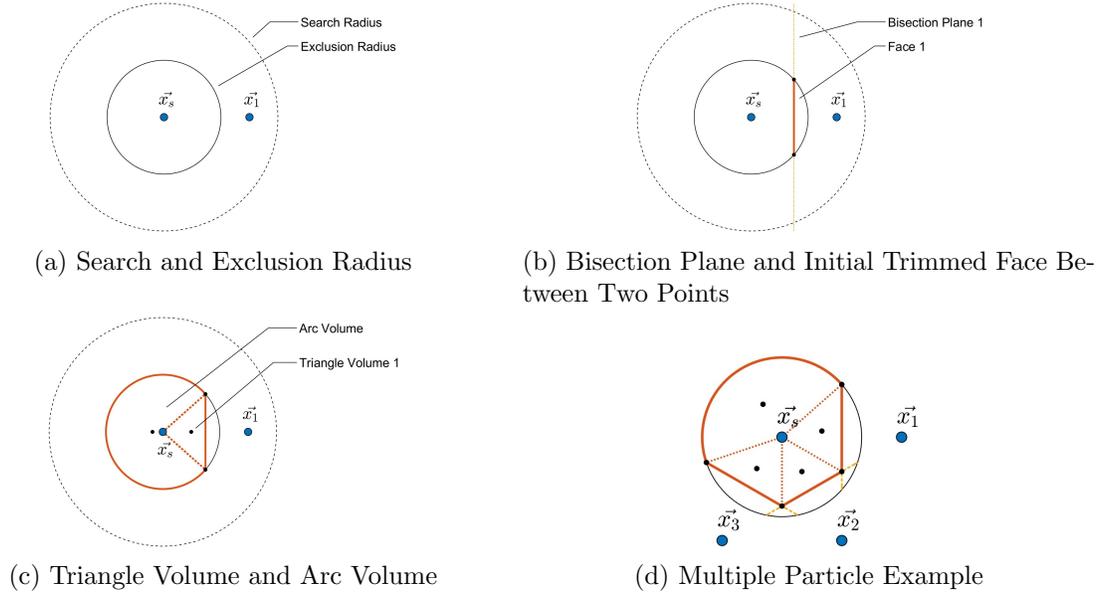


Figure 7.2: Finite Radius Voronoi Algorithm

of a cell not bounded by a bisection plane are limited to a volume defined by a subsection of the circle defined by the region's finite radius. Figure 7.2 illustrates this algorithm. Here the particle associated with the current cell is referred to as the *subject particle* (x_s). Because bisection planes are defined by all points in space equidistant to two particles, to ensure all particles which could generate a bisection plane within a finite exclusion radius, points within a distance of twice this radius must be considered (see Fig. 7.2a). This larger radius is referred to as the search radius and is related to the volume exclusion radius by $r_{search} = 2 \cdot r_{exclusion}$. Because any valid bisection plane intersects the exclusion radius, it may be reduced to a finite line segment or face. Figure 7.2b shows a bisection plane between two particles and the face generated by the intersection of the plane with the space interior to the finite radius. Area not enclosed by a triangle defined by a face and the subject particle is partitioned into *arc volumes* (see Fig. 7.2c). Face intersections with other bisection planes must also be considered, as the bisection plane of a separate point will occlude space that would otherwise be associated with an individual triangle volume. Figure

7.2d shows the generation of a Finite Radius Voronoi region between a subject particle and three neighboring points.

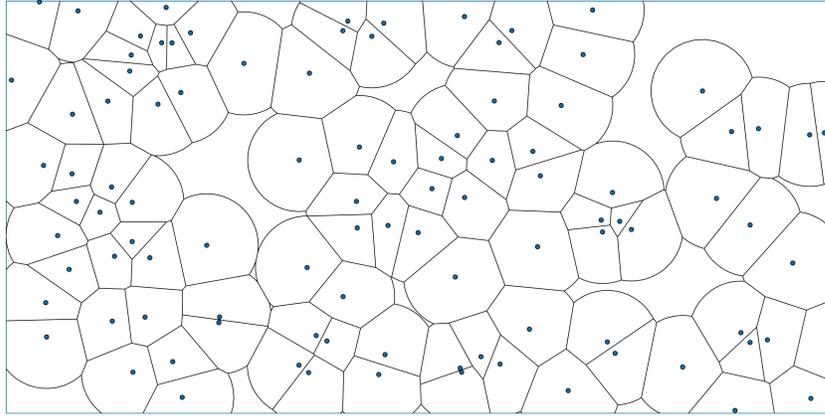


Figure 7.3: Finite Radius Voronoi Tessellation from Points

Figure 7.3 shows an example domain discretization using the Finite Radius Voronoi algorithm. A notable feature of the method is that it is not space-filling. This is intended behavior as in cases of single-phase domain discretizations volume far from free surface particles should not be associated with an SPH particle which is itself defined as existing within a region with finite bounds. The Finite Radius Voronoi algorithm is detailed in Algorithm 3.

7.3 Future Work

In order for this new Voronoi method to be applied to engineering fluid simulations, two advancements must be made. First, the method must be expanded to three-dimensional space. Moving to three-dimensional space will change the finite line segments generated from the intersection of bisection planes between points and the bounding radius to a convex polygon. This change will also complicate the tracking of individual arc sections of cell volumes. A possible solution to this issue is to no longer explicitly determine arc volumes and instead to determine the total volume removed from the potential full bounding

radius by each triangle volume. Then, by using boolean volume operations, subtract this removed volume from the total potential volume of the cell. In this way, the net arc section volume and moment can be determined.

Additional work will be done to explicitly include object boundaries into the cell generation algorithm. This is complicated by the potential non-convex nature of object boundaries. Early work towards this goal has utilized projected rays from object edges to transform non-convex cell features into convex cells. This adds a new requirement that space in a modified Voronoi cell must not lie on the opposite side of a boundary from the cell point. Figure 7.4 illustrates this feature. However, enforcing this "line of sight" requirement can result in an incomplete tessellation of the space within the bounded radius of domain points. Additional work is required to ensure that this volume is fully distributed to domain points, with no orphaned space.

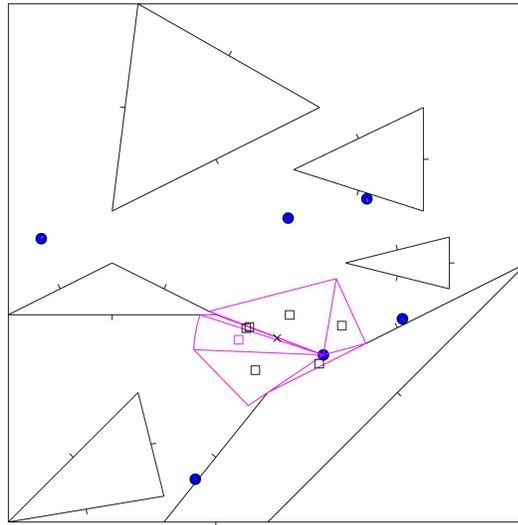


Figure 7.4: Finite Radius Voronoi with Boundaries (section centroids shown)

Algorithm 3 Finite Radius Voronoi

Require: N is the number of neighboring particles

Require: \vec{x}_s is the position of the subject particle

Require: r_{search} is the search radius which contains all neighbors

Require: Ω is the set of neighboring particles within r_{search} distance of \vec{x}_s

Require: $[\vec{x}]$ is an array of the positions of neighboring particles

$$r_{exclusion} = 0.5 \cdot r_{search}$$

if $N == 0$ **then**

$$\text{return: } \vec{x}_c = \vec{x}_s, V = \pi r_{exclusion}^2$$

end if

for Each particle i in Ω **do** ▷ Generate bisection planes for all particles

Generate bisection plane $plane_i$ with $plane_i \cdot \vec{x} = 0.5(\vec{x}_i + \vec{x}_s)$

Let $plane_i \cdot \hat{n} = \frac{\vec{x}_i - \vec{x}_s}{\|\vec{x}_i - \vec{x}_s\|}$ be the surface normal of the plane

Trim $plane_i$ by its intersection with the r_{search} to generate $face_i$

Set the face's endpoints to circle intersections

$$face_i \cdot \vec{x}_1 = Intersection_1$$

$$face_i \cdot \vec{x}_2 = Intersection_2$$

Set face's plane-intersections to none

$$face_i \cdot sharedplane_1 = None$$

$$face_i \cdot sharedplane_2 = None$$

end for

for each face $face_i$ **do** ▷ Trim faces by bisection planes

for each bisection plane $plane_j$ **do**

Trim $face_i$ by its intersection with $plane_j$

Update trimmed endpoint $face_i \cdot \vec{x}_1$ or $face_i \cdot \vec{x}_2$

Update trimmed face endpoint plane intersection

$$face_i \cdot sharedplane_{trimmed} = plane_j$$

end for

end for

Order faces by angle or pseudo-angle

Generate a list of open sections between face endpoints without shared planes

Sum volumes and moments of each face and open-section

$$\text{return: } \vec{x}_c = \frac{\sum \vec{x}_{c,i} V_i}{V}, V = \sum V_i$$

Chapter 8: Conclusions

Smoothed Particle Hydrodynamics is a powerful computational fluid dynamics method. In this thesis, a high-pressure multi-fluid deflagration test case was analyzed both by experiment and using a simplified 2D simulation. Chapter 2 presented and discussed the experimental results of the multi-fluid test case. Chapter 3 detailed the results and analysis gained from the simplified 2D simulation. Deficiencies were observed in the SPH method related to thin boundaries and gas rarefaction. These deficiencies were identified as opportunities to add to the capability of SPH and were used to guide the development of new computational geometry algorithms to improve the accuracy and efficiency of SPH simulations. Chapter 4 presented a new boundary condition for flexible and thin walls. Chapter 5 detailed a new algorithm for spatial filtering to reduce particle disorder. This method has the potential both to address disorder and clumping as well as potentially to condition the particle spacing of domain discretizations after resolution adaptation. Chapter 6 presented an iterative inter-domain property transfer method intended to facilitate copying domain properties to alternate fluid discretizations while strictly conserving the net quantity of conserved fluid properties. Chapter 7 detailed a finite-radius Voronoi tessellation method meant to be used in conjunction with the iterative property transfer method presented in chapter 6. The new Voronoi method allows for the parallel generation of Voronoi cells which are clipped to a finite radius. This modification allows for compact particle support to be used when generating Voronoi cells.

An immediate and expected deficiency lies with the highly simplified pressure ramp to mimic gas combustion. By analyzing figures 3.2 and 3.3, it is clear that using a simple pressure ramp cannot accurately model the highly non-linear behavior of the pressure in the true system. This suggests that either a full combustion model must be fully simulated or a much more capable low-order combustion model must be implemented. Future work

is necessary to address this issue.

With further development of these computational geometry methods, it is hoped that meaningful contributions to the capability of SPH can be made.

Bibliography

Bibliography

- [1] 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, no. 3, pp. 375–389, 1977.
- [2] L. B. Lucy, “A numerical approach to the testing of the fission hypothesis,” *The astronomical journal*, vol. 82, pp. 1013–1024, 1977.
- [3] J. J. Monaghan, “Simulating free surface flows with sph,” *Journal of computational physics*, vol. 110, no. 2, pp. 399–406, 1994.
- [4] L. D. Libersky and A. G. Petschek, “Smooth particle hydrodynamics with strength of materials,” in *Advances in the free-Lagrange method including contributions on adaptive gridding and the smooth particle hydrodynamics method*. Springer, 1991, pp. 248–257.
- [5] V. Springel, “Smoothed particle hydrodynamics in astrophysics,” *arXiv preprint arXiv:1109.2219*, 2011.
- [6] 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, no. 1-4, pp. 347–373, 1996.
- [7] A. Kubit, T. Trzepieciński, R. Kiciński, and K. Jurczak, “Three-dimensional smooth particle hydrodynamics modeling and experimental analysis of the ballistic performance of steel-based fml targets,” *Materials*, vol. 15, no. 10, p. 3711, 2022.
- [8] M. Liu, G. Liu, K. Lam, and Z. Zong, “Meshfree particle simulation of the detonation process for high explosives in shaped charge unlined cavity configurations,” *Shock Waves*, vol. 12, no. 6, pp. 509–520, 2003.
- [9] H. Yan, Z. Wang, J. He, X. Chen, C. Wang, and Q. Peng, “Real-time fluid simulation with adaptive sph,” *Computer Animation and Virtual Worlds*, vol. 20, no. 2-3, pp. 417–426, 2009.
- [10] S. Manenti, S. Sibilla, M. Gallati, G. Agate, and R. Guandalini, “Sph simulation of sediment flushing induced by a rapid water flow,” *Journal of Hydraulic Engineering*, vol. 138, no. 3, pp. 272–284, 2012.
- [11] D. Violeau and B. D. Rogers, “Smoothed particle hydrodynamics (sph) for free-surface flows: past, present and future,” *Journal of Hydraulic Research*, vol. 54, no. 1, pp. 1–26, 2016.

- [12] J. J. Monaghan and J. C. Lattanzio, “A refined particle method for astrophysical problems,” *Astronomy and astrophysics*, vol. 149, pp. 135–143, 1985.
- [13] P. Tait, “Physics and chemistry of the voyage of hms challenger,” *Vol. II, Part IV, SP LXI*, 1888.
- [14] F. D. Murnaghan, “The compressibility of media under extreme pressures,” *Proceedings of the National Academy of Sciences*, vol. 30, no. 9, pp. 244–247, 1944.
- [15] R. H. Cole and R. Weller, “Underwater explosions,” *Physics Today*, vol. 1, no. 6, p. 35, 1948.
- [16] J. R. Macdonald, “Some simple isothermal equations of state,” *Reviews of Modern Physics*, vol. 38, no. 4, p. 669, 1966.
- [17] G.-R. Liu and M. B. Liu, *Smoothed particle hydrodynamics: a meshfree particle method*. World scientific, 2003.
- [18] P. W. Cleary, “Modelling confined multi-material heat and mass flows using sph,” *Applied Mathematical Modelling*, vol. 22, no. 12, pp. 981–993, 1998.
- [19] H. Wendland, “Piecewise polynomial, positive definite and compactly supported radial functions of minimal degree,” *Advances in computational Mathematics*, vol. 4, no. 1, pp. 389–396, 1995.
- [20] J. Bonet and T.-S. Lok, “Variational and momentum preservation aspects of smooth particle hydrodynamic formulations,” *Computer Methods in applied mechanics and engineering*, vol. 180, no. 1-2, pp. 97–115, 1999.
- [21] J. J. Monaghan and R. A. Gingold, “Shock simulation by the particle method sph,” *Journal of computational physics*, vol. 52, no. 2, pp. 374–389, 1983.
- [22] 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, no. 5, pp. 537–561, 2013.
- [23] J. P. Morris and J. J. Monaghan, “A switch to reduce sph viscosity,” *Journal of Computational Physics*, vol. 136, no. 1, pp. 41–50, 1997.
- [24] A. Colagrossi and M. Landrini, “Numerical simulation of interfacial flows by smoothed particle hydrodynamics,” *Journal of computational physics*, vol. 191, no. 2, pp. 448–475, 2003.
- [25] M. Meister and W. Rauch, “Simulating rising bubble problems with smoothed particle hydrodynamics,” in *IMA Conference on Mathematical Modelling of Fluid Systems*, vol. 10, 2014, p. 12.
- [26] Q. Yang, “Sph simulation of fluid-structure interaction problems with application to hovercraft,” Ph.D. dissertation, Virginia Tech, 2011.
- [27] A. Keil, “The response of ships to underwater explosions,” DAVID TAYLOR MODEL BASIN WASHINGTON DC, Tech. Rep., 1961.

- [28] M. Holt, "Underwater explosions," *Annual review of fluid mechanics*, vol. 9, no. 1, pp. 187–214, 1977.
- [29] T. M. Keevin and G. L. Hempen, "The environmental effects of underwater explosions with methods to mitigate impacts," Tech. Rep., 1997. [Online]. Available: <http://www.dtic.mil/docs/citations/ADA575523>
- [30] M. H. Nielsen, L. Bach, and S. M. Bollwerk, "Spreading of sediment due to underwater blasting and dredging: Field observations from quay construction in sisimiut, greenland," *Ocean & Coastal Management*, vol. 116, pp. 512–522, 2015.
- [31] D. R. Ketten, "Estimates of blast injury and acoustic trauma zones for marine mammals from underwater explosions," *Sensory systems of aquatic mammals*, pp. 391–407, 1995.
- [32] F. V. de Camargo, "Survey on experimental and numerical approaches to model underwater explosions," *Journal of marine science and engineering*, vol. 7, no. 1, 2019. [Online]. Available: <https://doaj.org/article/51c0cea1d2f6494185723569029fea2a>
- [33] A. S. Committee *et al.*, *AIAA guide for the verification and validation of computational fluid dynamics simulations*. American Institute of aeronautics and astronautics., 1998.
- [34] F. Stern, R. V. Wilson, H. W. Coleman, and E. G. Paterson, "Comprehensive approach to verification and validation of cfd simulations—part 1: Methodology and procedures," *Journal of fluids engineering*, vol. 123, no. 4, pp. 793–802, 2001.
- [35] P. J. Roache, "Need for control of numerical accuracy," *Journal of Spacecraft and Rockets*, vol. 27, no. 2, pp. 98–102, 1990.
- [36] L. E. Schwer, "Guide for verification and validation in computational solid mechanics," 2009.
- [37] J. P. Kleijnen, "Verification and validation of simulation models," *European journal of operational research*, vol. 82, no. 1, pp. 145–162, 1995.
- [38] W. L. Oberkampf and T. G. Trucano, "Verification and validation in computational fluid dynamics," *Progress in aerospace sciences*, vol. 38, no. 3, pp. 209–272, 2002.
- [39] O. Balci, "Credibility assessment of simulation results," in *Proceedings of the 18th conference on Winter simulation*, 1986, pp. 38–44.
- [40] R. G. Sargent, "Verification, validation and accreditation of simulation models," in *2000 Winter Simulation Conference Proceedings (Cat. No. 00CH37165)*, vol. 1. IEEE, 2000, pp. 50–59.
- [41] H. Wang, X. Zhu, Y. S. Cheng, and J. Liu, "Experimental and numerical investigation of ship structure subjected to close-in underwater shock wave and following gas bubble pulse," *Marine Structures*, vol. 39, pp. 90–117, 2014.
- [42] Y. Chen, Z. Tong, H. Hua, Y. Wang, and H. Gou, "Experimental investigation on the dynamic response of scaled ship model with rubber sandwich coatings subjected to underwater explosion," *International Journal of Impact Engineering*, vol. 36, no. 2, pp. 318–328, 2009.

- [43] Z. Zhang, Y. Wang, H. Zhao, H. Qian, and J. Mou, "An experimental study on the dynamic response of a hull girder subjected to near field underwater explosion," *Marine Structures*, vol. 44, pp. 43–60, 2015.
- [44] P. Cui, A.-m. Zhang, S.-p. Wang, and Q.-x. Wang, "Experimental investigation of bubble dynamics near the bilge with a circular opening," *Applied Ocean Research*, vol. 41, pp. 65–75, 2013.
- [45] X. Cui, X. Yao, and Y. Chen, "A lab-scale experiment approach to the measurement of wall pressure from near-field under water explosions by a hopkinson bar," *Shock and Vibration*, vol. 2018, pp. 1–15, 2018. [Online]. Available: <http://search.proquest.com/docview/2111086828/>
- [46] A. De, A. Niemiec, and T. F. Zimmie, "Physical and numerical modeling to study effects of an underwater explosion on a buried tunnel," *Journal of Geotechnical and Geoenvironmental Engineering*, vol. 143, no. 5, 2017.
- [47] G. Yang, G. Wang, W. Lu, L. Wu, P. Yan, and M. Chen, "Experimental and numerical study of damage characteristics of rc slabs subjected to air and underwater contact explosions," *Marine structures*, vol. 66, pp. 242–257, 2019.
- [48] V. Jones, K. V. Pol, J. Gilbert, and L. McCue-Weil, "Underwater gas expansion and deflagration," Gallery of Fluid Motion arXiv:1310.3523, 2013.
- [49] V. Jones, J. Gilbert, K. V. Pol, and L. McCueWeil, "An underwater explosion experimental dataset towards validation of fsi codes," in *Multi-Agency Craft Conference*. Virginia Beach United States: American Society of Naval Engineers, May 2014.
- [50] V. Jones, L. McCue, J. Gilbert, K. V. Pol, and S. Letourneau, "Underwater deflagration acting on a flat plate," Gallery of Fluid Motion, 2020.
- [51] M. S. Ebaid and K. J. Al-Khishali, "Measurements of the laminar burning velocity for propane: Air mixtures," *Advances in Mechanical Engineering*, vol. 8, no. 6, pp. 1–17, 2016.
- [52] LaVision, "Davis 8.2.3," 2015, <https://www.lavision.de/>.
- [53] V. Bychkov, A. Petchenko, V. Akkerman, and L.-E. Eriksson, "Theory and modeling of accelerating flames in tubes," *Physical Review E*, vol. 72, no. 4, p. 046307, 2005.
- [54] V. Bychkov, V. Akkerman, G. Fru, A. Petchenko, and L.-E. Eriksson, "Flame acceleration in the early stages of burning in tubes," *Combustion and Flame*, vol. 150, no. 4, pp. 263–276, 2007.
- [55] A. Mazas, D. Lacoste, and T. Schuller, "Experimental and numerical investigation on the laminar flame speed of ch4/o2 mixtures diluted with co2 and h2o," in *Turbo Expo: Power for Land, Sea, and Air*, vol. 43970, 2010, pp. 411–421.
- [56] K. de Graff, "The bubble dynamics and pressure field generated by a seismic airgun," Ph.D. dissertation, University of Tasmania, 2013.

- [57] V. Jones, Q. Yang, and L. M. Weil, “Sph boundary deficiency correction for improved boundary conditions at deformable surfaces,” *Ship Science and Technology*, vol. 4, no. 7, pp. 21–30, 2010.
- [58] J. J. Monaghan, A. Kos, and N. Issa, “Fluid motion generated by impact,” *Journal of waterway, port, coastal, and ocean engineering*, vol. 129, no. 6, pp. 250–259, 2003.
- [59] J. J. Monaghan and J. B. Kajtár, “Sph particle boundary forces for arbitrary boundaries,” *Computer physics communications*, vol. 180, no. 10, pp. 1811–1820, 2009.
- [60] L. D. Libersky, A. G. Petschek, T. C. Carney, J. R. Hipp, and F. A. Allahdadi, “High strain lagrangian hydrodynamics: a three-dimensional sph code for dynamic material response,” *Journal of computational physics*, vol. 109, no. 1, pp. 67–75, 1993.
- [61] P. Randles and L. D. Libersky, “Smoothed particle hydrodynamics: some recent improvements and applications,” *Computer methods in applied mechanics and engineering*, vol. 139, no. 1-4, pp. 375–408, 1996.
- [62] G. Liu and Y. Gu, “A local radial point interpolation method (lrpim) for free vibration analyses of 2-d solids,” *Journal of Sound and vibration*, vol. 246, no. 1, pp. 29–46, 2001.
- [63] M. Liu, G. Liu, and K. Lam, “Investigations into water mitigation using a meshless particle method,” *Shock waves*, vol. 12, no. 3, pp. 181–195, 2002.
- [64] J. Feldman and J. Bonet, “Dynamic refinement and boundary contact forces in sph with applications in fluid flow problems,” *International journal for numerical methods in engineering*, vol. 72, no. 3, pp. 295–324, 2007.
- [65] M. Ferrand, D. 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, no. 4, pp. 446–472, 2013.
- [66] A. Mayrhofer, M. Ferrand, C. Kassiotis, D. Violeau, and F.-X. Morel, “Unified semi-analytical wall boundary conditions in sph: analytical extension to 3-d,” *Numerical Algorithms*, vol. 68, no. 1, pp. 15–34, 2015.
- [67] J. Monaghan, “On the problem of penetration in particle methods,” *Journal of Computational physics*, vol. 82, no. 1, pp. 1–15, 1989.
- [68] M. Antuono, B. Bouscasse, A. Colagrossi, and S. Marrone, “A measure of spatial disorder in particle methods,” *Computer Physics Communications*, vol. 185, no. 10, pp. 2609–2621, 2014.
- [69] N. J. Quinlan, M. Basa, and M. Lastiwka, “Truncation error in mesh-free particle methods,” *International Journal for Numerical Methods in Engineering*, vol. 66, no. 13, pp. 2064–2085, 2006.
- [70] J. J. Monaghan, “Sph without a tensile instability,” *Journal of computational physics*, vol. 159, no. 2, pp. 290–311, 2000.

- [71] S. Marrone, A. Colagrossi, D. Le Touzé, and G. Graziani, “Fast free-surface detection and level-set function definition in sph solvers,” *Journal of Computational Physics*, vol. 229, no. 10, pp. 3652–3663, 2010. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0021999110000343>
- [72] X. Zheng, W. Duan, and Q. Ma, “A new scheme for identifying free surface particles in improved sph,” *Science China Physics, Mechanics and Astronomy*, vol. 55, no. 8, pp. 1454–1463, 2012.
- [73] A. Chaniotis, D. Poulidakos, and P. Koumoutsakos, “Remeshed smoothed particle hydrodynamics for the simulation of viscous and heat conducting flows,” *Journal of Computational Physics*, vol. 182, no. 1, pp. 67–90, 2002.
- [74] P. Koumoutsakos, “Multiscale flow simulations using particles,” *Annu. Rev. Fluid Mech.*, vol. 37, pp. 457–487, 2005.
- [75] R. Vacondio, B. Rogers, P. K. Stansby, P. Mignosa, and J. Feldman, “Variable resolution for sph: a dynamic particle coalescing and splitting scheme,” *Computer Methods in Applied Mechanics and Engineering*, vol. 256, pp. 132–148, 2013.
- [76] Q. Xiong, B. Li, and J. Xu, “Gpu-accelerated adaptive particle splitting and merging in sph,” *Computer Physics Communications*, vol. 184, no. 7, pp. 1701–1707, 2013.
- [77] R. Winchenbach, H. Hochstetter, and A. Kolb, “Infinite continuous adaptivity for incompressible sph,” *ACM Transactions on Graphics (TOG)*, vol. 36, no. 4, pp. 1–10, 2017.
- [78] T. M. Liebling and L. Pournin, “Voronoi diagrams and delaunay triangulations: Ubiquitous siamese twins,” *Documenta Mathematica, ISMP*, pp. 419–431, 2012.
- [79] G. Voronoi, “Nouvelles applications des paramètres continus à la théorie des formes quadratiques. deuxième mémoire. recherches sur les paralléloèdres primitifs.” *Journal für die reine und angewandte Mathematik (Crelles Journal)*, vol. 1908, no. 134, pp. 198–287, 1908.

Curriculum Vitae

Van received his Bachelor of Arts from Virginia Tech in 2008. He continued his graduate education there before transferring to George Mason University in 2019.