UNIVERSITY OF CALIFORNIA, MERCED

Physics-Based Animation Models Using Fractional Calculus

A dissertation submitted in partial satisfaction of the requirements
for the degree Doctor of Philosophy

in

Electrical Engineering & Computer Science

by

Oktar Özgen

Committee in charge:

Professor Marcelo Kallmann, UC Merced, Committee Chair
Professor Shawn Newsam, UC Merced
Professor Carlos F.M. Coimbra, UC San Diego

2013

ABSTRACT OF THE DISSERTATION

Physics-Based Animation Models Using Fractional Calculus

by

Oktar Özgen
Doctor of Philosophy in Electrical Engineering & Computer Science
University of California, Merced, 2013
Professor Professor Marcelo Kallmann, UC Merced, Chair

Physics-based computer animation is an area of Computer Graphics with important applications ranging from visual effects in movies and computer games to simulation-based modeling and design. This dissertation introduces the use of fractional calculus for achieving new physics-based models in the areas of cloth and fluid simulation. Three main contributions are proposed. First, an underwater cloth deformation approach is introduced based on a half-derivative damping model. The cloth is represented by a system of particles connected by elastic elements with fractional damping terms that are able to well represent the history forces that happen underwater. As a result, underwater behavior is achieved without simulating the volumetric fluid around the cloth. Second, a new viscosity model employing fractional terms is proposed to better represent the behavior of colliding flows in simulations of Newtonian fluids based on Smoothed Particle Hydrodynamics (SPH). The fractional terms are integrated into the SPH equations in order to better capture the history-based exchanges that happen in regions of colliding flows. Finally, a model is introduced for simulating shear thickening fluids, which are a special type of non-Newtonian fluids that have not been simulated before in computer animation. The proposed model combines SPH forces with dynamic elastic forces that are modeled with history-based fractional stiffness terms. The model can successfully simulate both the solid and liquid-like behavior of a shear thickening fluid as well as the history-based phase transitions between the solid and liquid phases. The achieved model is able to produce results that are very similar to

real-life experiments. This dissertation illustrates the potential of fractional calculus for modeling physics-based behavior and several numerical and visual experiments are presented to demonstrate the advantages of the proposed models.

The dissertation of Oktar Özgen is approved, and it is acceptable in quality and form for publication on microfilm and electronically:

Professor Carlos F.M. Coimbra, UC San Diego

Professor Shawn Newsam, UC Merced

Professor Marcelo Kallmann, UC Merced, Committee Chair

University of California, Merced

2013

To my mother . . .

TABLE OF CONTENTS

LIST OF FIGURES

ACKNOWLEDGMENTS

The path to the completion of this dissertation was a rough one. The journey was a search. It was a search for solutions to the adaptation problems, cultural clashes, language barriers, search for social recognition, search for self awareness, search for a value set, search for a skill set, search for the meaning, search for the long lost past, search for a promising future, search for becoming a better man—and at times, it was a search for novel methods for the application of fractional calculus in computer graphics.

I would like to thank everyone who directly or indirectly helped me to get to the end of this journey. My family, my friends, my professors, my colleagues. I could not have done it without you.

I would like to start by thanking the most important person in my life. My mother, who raised me by herself in difficult conditions, always loved me unconditionally, not even complained once about me being so far away, gave me the freedom of establishing my own life and always supported me in every step I took. Second, my grandfather, who was the father figure I had growing up, who financially and morally supported me and always believed in me. I am sure he would have been very proud if he were still with us. Third, my sister, for being a loving and caring person and for taking care of my mom when I am not around. Lastly, my father, who passed his engineer genes to me and influenced me in choosing a scientific career.

Merced may not the best place to live, but without you, my friends, it would have been a lot worse. I would like to thank all my friends who, along the way, helped me, influenced me, changed me and laughed with me. I would like to thank Mentar Mahmudi, who has been the best friend one can ask for, and a brother to me. Without you, life would be extremely hard to deal with. I would like to also thank my close friends Janelle Szary, Görkem Erinç, Siyu Wu, Benjamin Balaguer and Nicola Basilico. It was a pleasure to know each and every one of you and share good memories with you. I would also like to thank my best friends in Turkey, Fatih Ömür, Mert Tuncer, Mert Tatver and Alp Aras, who always make me feel welcome home, and still connected, regardless of the years I have been away.

I would like to thank my colleagues with whom I have collaborated and exchanged ideas. I would like to thank the members of the UC Merced Computer Graphics Group; Mentar Mahmudi, David Huang, Carlo Camporesi and Robert Backman. I would like to thank Selçuk Sümengen and Lynnette Ramirez for collaborating with me in various papers.

I would like to thank the professors who thought and guided me during my PhD education. First, I would like to express my gratitude towards my committee members Professor Shawn Newsam and Professor Carlos F.M. Coimbra, for their valuable comments, questions and suggestions and dedication to their work as committee members. I would like to thank Professor Selim Balcısoy from Sabancı University who referred me to UC Merced and later collaborated with me in a paper. I would like to thank

Professor Eric Brown for collaborating with me. I would to especially thank Professor Carlos F.M. Coimbra for advising me on fluid mechanics, collaborating with me and providing me all the valuable methods from his own papers, which established the basis of my dissertation.

Last but not least, I would like to thank and express my gratitude to my adviser Professor Marcelo Kallmann. I would like to thank him for all the education and guidance, for valuable discussions on academia and life, for his contribution to my research, for his friendship and for being the greatest adviser one can ask for.

VITA

2001–2006	B.S. in Computer Science and Engineering, Sabanci University, Istanbul, Turkey.
2006–2007	Software Engineer, Garanti Technology, Istanbul, Turkey.
2007–2013	Teaching Assistant, Introduction to Java, Introduction to C++, Operating Systems, Computer Graphics, Database Systems, Electrical Engineering & Computer Science Department, UC Merced.
2007–2013	PhD Student - Research Assistant, Electrical Engineering & Computer Science Department, UC Merced.

PUBLICATIONS

O. Ozgen, M. Kallmann. Effects of Different Constraint Enforcement Methods for Verlet Integration in Cloth Simulation. In *Cognitive Animation Workshop*, 2008.

O. Ozgen, M. Kallmann, L.E.S Ramirez, C.F.M. Coimbra. Underwater Cloth Simulation with Fractional Derivatives. In *ACM Transactions on Graphics (TOG)*, v. 29-3, p. 1-9, presented at SIGGRAPH, 2010.

O. Ozgen, M. Kallmann. Directional Constraint Enforcement for Fast Cloth Simulation. In *Proceedings of the 4th international conference on Motion in Games(MIG)*, p. 424-435, 2011.

O. Ozgen, S. Sumengen, M. Kallmann, C.F.M. Coimbra, S. Balcisoy. Simulating Colliding Flows in SPH with Fractional Derivatives. In *Computer Animation and Virtual Worlds*, to appear.

O. Ozgen, E. Brown, M. Kallmann. Simulating Shear Thickening Fluids. Under review.

CHAPTER 1

Introduction

In the early 1990's, when Steven Spielberg's master piece movie Jurassic Park [Spi92] hit the movie theaters, it was a marking point in the history of computer graphics. Although computer graphics existed long before it was adopted by the movie industry, this was the first time computer graphics revealed its magic to large audiences. The degree of success achieved in the visual effects brought the dinosaurs to life in an amazingly realistic way. Since then, the popularity of computer graphics immensely increased. Today, it is widely used in movies, computer games as well as industrial applications. Computer graphics is a highly interdisciplinary science. It can essentially be located at the intersection of art and engineering. The artistic side of computer graphics dictates that the results should be visually realistic before everything else. A computer graphics scene is a success if it is believable in the eyes of an untrained observer, no matter what techniques are used. Unlike most sciences, numerically proving the efficiency of a method is not necessarily useful if it does not lead to visually convincing effects. Nevertheless, most visually striking effects seem to arise from a solid scientific background. A number of sciences contribute to the scientific foundations of computer graphics, such as physics, mechanics, dynamics, rheology, etc. The models based on these fields of science give rise to a subcategory named "physics-based animation".

Physics-based animation deals with the simulations of physical systems such as, soft body dynamics, fluid dynamics and rigid body dynamics. Physics-based models describe how various phenomena occur in reality. Hence, they make fantastic tools for animation. Once properly implemented, they can be employed for numerous interesting animations with little effort.

With a focus in describing the physical phenomena, many mathematical tools are used in sciences such as mechanical engineering, rheology and physics. Among these, fractional calculus is an important mathematical tool. Although it is widely used in areas such as fluid and solid mechanics, rheology, electromagnetism, electrochemistry, and biology, fractional calculus has not been explored before for applications in computer graphics. It is however a very powerful tool that can model memory-intense and delay systems as well as unsteady viscous and viscoelastic phenomena [OS74, MR93, Pod99, Hu05, Hil00, KST06]. These useful features have the potential to be employed in various physics-based simulations in computer graphics. In this dissertation, we introduce the first applications of fractional calculus in computer graphics. The focus of this dissertation is mainly on the applications of fractional calculus on soft

body dynamics, namely cloth simulation, Newtonian and non-Newtonian fluid simulations.

Cloth is an interesting material. It has a variety of complex behaviors. It stretches, compresses, bends and wrinkles. This wide range of interesting behavior made cloth a subject of interest in the history of art. The depiction of cloth was a major concern for artists since the Renaissance. In the Baroque era, correctly depicting the drapery of cloth in painting or sculpture was considered a major criteria for the success of an art piece. Accurate depiction of cloth was a major element that brings the realism of an art piece to the next level. The computer graphics community realized the need for animating the cloth right after the launch of early successful animation movies of 1990's such as Toy Story [Las95]. Toy Story produced by Pixar was a huge success. However, the characters with unrealistic looking clothes immediately captured the industry's attention. Since then, there has been ongoing research in the community on modeling more realistic looking cloth.

Cloth is modeled and animated with physics-based methods. Modeling cloth as a particle system is a very popular approach. The particles are generally connected to each other by springs that simulate the warp and weft based microstructure of actual clothes as depicted in Figure 2.1. The inner and outer forces on the particles are calculated and integrated in time with various integration methods. The challenge of cloth simulation is two-fold. First, the cloth should behave realistically. It shouldn't stretch, compress, shear or bend too much. It should react to the applied forces in a meaningful way. Second, the computation time of a cloth simulation should be minimized as much as possible. The simulation should run fast enough to render the cloth scenes in movies under some acceptable time limit. Moreover, the simulation should be real-time for computer games. We took on both challenges up to some extent. We focused on creating visual realism for the cloth in a scenario which was not previously addressed. We achieved very realistic results for a cloth moving underwater. In another paper, we also focused on improving the running times of cloth simulation scenes in computer games. Hence, we contributed in both aspects of the cloth simulation challenge.

Another interesting subject in physics-based animation is fluid simulation. Fluid is the general name of all substances that deform under applied stress. It includes liquids, gases and plasmas. It is possible to simulate water, smoke, fire and explosions using fluid simulation techniques. These effects add high levels of realism to movies and computer games. Therefore, fluid simulation emerges as a key topic in computer graphics. One of the studies presented in this dissertation is on water simulation. Water simulation is extremely rich in terms of visual results. It is possible to simulate rivers, oceans, waves, flows, splashes, floods and other forms of interesting phenomena to add high levels of realism. These effects are achieved using methods based on the well known Navier-Stokes equations. One very popular alternative to these equations is called the Smoothed Particle Hydrodynamics (SPH) method. SPH is a Lagrangian method that simulates the body of water as a collection of particles. The number of particles used in today's movies are on the scale of millions. Therefore, the computational cost of

these simulations is extremely high. The nature of the problem brings about the need of creating interesting effects with more efficient ways. This objective can be achieved in two ways. First, the running time of a simulation with a high number of particles can be optimized. Second, the accuracy of a simulation with relatively fewer particles can be improved. As a part of this dissertation, we focused on the latter and aimed to improve the accuracy of fluid simulation. In particular, we managed to increase the accuracy of a colliding flow animation that runs with relatively low number of particles, therefore eliminating the need for a high number of particles to achieve the same effect.

The introduction of physics-based models for materials which have no known prior simulation model is very important in the computer graphics community. After the tremendous developments in the major areas such as cloth and fluid simulations, the research community started to lean towards the challenge of simulating new materials. As a part of this trend, the simulation of non-Newtonian fluids has attracted much attention. Fluids such as honey, toothpaste and jelly which exhibit both liquid and solid-like behavior are targeted. These materials often require hybrid approaches that encompass both fluid simulation techniques and spring-mass techniques. As a continuation of this trend, we address the problem of simulating shear thickening fluids, a material which was never simulated before in the computer graphics community. Shear thickening fluids display very unconventional behavior that is not seen in any other Newtonian or non-Newtonian fluids. The fluid hardens when impacted by strong forces and softens in the absence of any forces. It changes its state from solid-like to liquid-like. The phase transitions are not immediate. They often require some time to occur because of the memory-intense nature of the material. This interesting type of non-Newtonian fluid has very interesting behaviors that can potentially lead to interesting visual effects and its memory-intense behavior can be simulated using fractional calculus. In Chapter 5, we present a method that can successfully simulate shear thickening fluids.

In summary, this dissertation introduces three novel methods based on fractional calculus for physics-based animations. First, an underwater cloth model with a fractional derivative based damping mechanism is presented in Chapter 3. Second, a fractional derivative based viscosity term is introduced to SPH equations to increase the accuracy of colliding flow animations. The method is explained in Chapter 4. Third, a novel fractional derivative based model is introduced in order to simulate shear thickening fluids. This contribution is detailed in Chapter 5. Aside from the main contributions, a method that increases the performance of cloth simulation in computer games is presented in Chapter 6.

CHAPTER 2

Literature Review

This chapter reviews the literature on fractional calculus, cloth simulation, Newtonian and non-Newtonian fluid simulations. In Section 2.1, we review the related literature on fractional calculus and discuss how fractional calculus relates to our studies. In Section 2.2, we present an overview of cloth simulation. In particular, we focus on the modeling of the cloth and the integration methods used. In Section 2.3, we first review the basics of fluid simulation. Then, we present the literature on the SPH method which forms the basis of our method presented in Chapter 5. In Section 2.4, we review physics papers related to shear thickening fluids. We aim to give the reader an introductory perspective on the behavior of this kind of fluid. A more specific related work review is later given in every chapter.

2.1 Fractional Calculus

The subject of Fractional Calculus (FC) [OS74] deals with the mathematical analysis of differentiation and integration to an arbitrary non-integer order. The traditional name of the discipline is a misnomer since any non-integer (rational or irrational) order of differentiation and integration is easily accounted for within the context of generalized differentiation.

In 1695, L'Hopital inquired of Leibniz what the meaning of half derivative could be. Leibniz replied: "This is an apparent paradox from which, one day, useful consequences will be drawn." The first usage of variable order notion of a derivative $D^n f(x)$ = $d^n f(x)/dx^n$ was by Leibniz. Following his footsteps, many mathematicians such as Euler, Laplace, Fourier, Abel, Liouville, Riemann, and Laurent contributed on the field of fractional calculus.

Historically, Abel was the first to apply FC to a real world problem [Abe23]. He used FC in the solution of the tautochronous problem. The problem is concerned with the determination of the shape of a frictionless plane curve for which the time taken by an object sliding without friction in uniform gravity to its lowest point is independent of its starting point. FC was first employed in the engineering community by Heaviside in 1892 [Hea92] [Hea93]. He proposed a fractional diffusion equation for the temperature distribution problem for electric transmission lines. Since these introductory works, FC definitions have been used in a variety of problems such as rheological models for the

deformation of viscoelastic materials, modeling the memory of systems, studying the effects of the history of systems in its current state, dynamical processes in fractals, porous materials, chemical reactions, diffusion, electrical circuits, chaotic processes, process control, fractional order physics, speech signals, cardiac tissue electrode interface, sound wave propagations, theory of viscoelasticity and fluid mechanics.

Beyond the fact that FC models provide superior modeling capability of memory-intense and delay systems, fractional modeling has been associated with the exact description of unsteady viscous and viscoelastic phenomena. Of particular importance to our work is the ability of non-integer derivatives to capture non-local behavior, and to interpolate between distinct dynamic regimes such as between the elastic and the viscous behavior [OS74, MR93, Pod99, Hu05, Hil00, KST06].

For instance, Coimbra et al. [CLL04] and L'Espérance et al. [LCT05] provide definitive experimental evidence of fractional history effects in the unsteady viscous motion of small particles in suspension [CR98, CK02b]. In Chapter 3, it is this same behavior [CLL04] that we propose to model by adding a fractional term to the equation of motion of particles in a cloth mesh moving underwater. The goal is to simulate the history-laden dynamics of the deformable cloth by introducing a non-local dynamic operator that accounts for an intermediate response between position and velocity of the medium. This viscoelastic term of order $1/2$ is able to simulate the fluid response to cloth accelerations without the complication of simulating the entire hydroelastic process.

The motivation for our approach to simulate cloth-fluid interactions follows from the study of individual particles moving in a viscous fluid [CR98]. Stokes [Sto45] pioneered the study of viscous particle motion by determining the force acting on a small fixed particle that is subjected to a uniform fluid velocity. Stokes related the undisturbed flow (or far-stream conditions) to the force acting on the sphere, thus creating a direct relationship between the resulting force (or torque) acting on a spherical particle and the kinematics of the free-stream flow.

Mathematically, Stokes determined the operator Λ_S that relates the force $\boldsymbol{F}$ acting on a particle to the background flow field $\boldsymbol{U}$, such that an expression of the form $\boldsymbol{F} = \Lambda_S(\boldsymbol{U})$ is determined. This is a major accomplishment because it allows for the calculation of the forces acting on the particle by considering only the undisturbed flow conditions as opposed to relating the force to the spatially dependent, non-uniform flow in the vicinity of the particle. The computation simplification of Stokes' analysis is such that, under special circumstances, one can predict accurately the rate of sedimentation of small particles without ever resolving or understanding the flow generated by the motion of the particles.

The resulting Stokes drag formula relates the force exerted on the sphere to the constant free-stream velocity $\boldsymbol{U}$, the dynamical viscosity of the fluid μ, and the radius a of the sphere in a linear way, given that the particle Reynolds number is maintained much smaller than unity.

Boussinesq [Bou85] and Basset [Bas88] independently extended Stokes' derivation to a case where the particle accelerates through the fluid due to a constant gravitational force but still neglecting the convective terms in the Navier-Stokes equation (valid only for small Reynolds numbers). The particle equation of motion with a constant forcing (the gravity term) is sometimes referred to as the BBO equation, due to the original contributions of Boussinesq, Basset and Oseen.

The BBO equation is an integro-differential equation that has a removable singularity in the integrand of the history term. The history or fractional term can be then derived directly from the Stokes operator Λ_S using Duhamel's Superposition Theorem. The history term is found to be simply $a\Lambda_S\nu^{-1/2}D^{1/2}\boldsymbol{V}$, where ν is the kinematic viscosity of the fluid and $D^{1/2}\boldsymbol{V}$ represents the half-derivative of the particle velocity [CR98], which can be computed with the following Riemann-Liouville differential operator:

$$D^{1/2}\boldsymbol{V} = \frac{1}{\Gamma(1/2)}\int_{-\infty}^{t}(t-\sigma)^{-1/2}\frac{d\boldsymbol{V}(\boldsymbol{\sigma})}{d\,t}d\sigma, \tag{2.1}$$

where Γ is the generalized factorial function and $\Gamma(1/2) = \sqrt{\pi}$.

The results above are important because they allow us to replace the complicated, evolving flow field around the particle with an integral force that can be calculated in an iterative way. The implications of this formulation in terms of computational time are clear. The fractional force plays the role of an intermediate dynamic variable between velocity and acceleration, and is exact for a single particle in low Reynolds flows.

In the context of cloth simulation, we are interested in the overall effect of a piece of cloth interacting with the flow. One key observation is that the immersed cloth will have its internal structures mixed with molecules of water and therefore both its constitutive material and its response with the surrounding volume of water will be better described as a flowable microstructure similar to the ones studied in Rheology [Lar98]. Our immersed cloth will therefore be better described as a viscoelastic microstructure exhibiting obvious fluid-like (viscous) and also solid-like (elastic) behaviors.

While an exact model for the immersed cloth microstructure cannot be devised, we recall that the intermediate behavior between viscosity and elasticity is, in its simplest form, a viscoelastic effect that can be approximated by a term on the half-derivative of the displacement [Coi03]. We therefore translate the particle history effect described by the half-derivative in Equation 2.1 into the immersed cloth by applying the half-derivative operator to the mesh displacements (instead of applying to the relative velocity).

The result is a viscoelastic term which is added to both the internal forces and the external drag of the cloth. The internal forces then become governed by viscoelastic spring elements and the external drag becomes the history (or Basset) drag. As a result, the achieved model simulates the different inertial responses between the fluid and the cloth in an efficient way. The model is presented in Chapter 3.

L'espérance at al. [LCT05] studied high frequency particle response in viscous fluid to validate the analytical solutions presented in [CR98]. In [CR98], it was shown that under certain conditions of particle size, fluid viscosity and oscillation frequency, the Basset force becomes dominant in the equation of motion. Therefore, in [LCT05], experiments are set up in a way to maximize the history force on the particle in suspension. The particle is tied to a thin tether to counter act the gravity. The cell filled with a low Reynolds number fluid is oscillated sinusoidally. A marker particle is also fixed on the cell's wall. A holography camera is used to track two particles. In the end, the particle response coefficient measured in the experiment and in the analytical solution are compared. As expected, the particle oscillated at the same frequency as the cell, but with different amplitude and phase. The experimental data and the theoretical results were in excellent agreement. This study demonstrated the effect of the Basset history force over a wide range of densities and frequencies.

The fundamental results from both [CR98] [LCT05] motivated us to use the fractional derivative terms for the simulation of a fluid phenomena in which history forces are dominant. This idea gave rise to the simulation of colliding flows. The particles in the simulation of colliding flows depict similar behavior to the oscillating particle in the experiment explained in [LCT05]. Thus, we aim to increase the physical accuracy of simulating flow collisions in low resolution simulations by utilizing a fractional derivative model. In Chapter 4, we propose a new SPH model with half-derivative viscosity terms to compensate the loss of information in low resolution simulations.

2.2 Cloth Simulation

The work on cloth simulation mostly focuses on the modeling of the cloth and on the integration methods used. Therefore, in the next sections, we review major works on cloth modeling and later, we present the main features of the integration methods used in cloth simulation.

2.2.1 Modeling the Cloth

In the literature of cloth simulation, the earliest efforts focused on simulating the cloth through the use of continuum mechanics. The cloth was considered as an elastic sheet and finite elements techniques were employed for simulations. Terzopoulos et al. [TPB87] [TF88] presented one of the earliest efforts to simulate elastically deformable models such as cloth, rubber and paper in computer graphics. Prior to this work, most of computer graphics modeling was kinematic. Kinematic models are considered passive. They neither interact with external forces nor change their shape over time. This work advocated the use of active models for the first time. The simulation of physical models that can react to applied forces and constraints was the main objective. The problem of cloth simulation is treated as a general deformable body problem, thus the

theory of elasticity is employed to form differential equations to model the behavior of non-rigid curves and surfaces as a function of time. Lagrange's equation of motion is used to model the dynamics of the cloth. The net external forces are given by the sum of inertial forces, damping forces and elastic forces. The elastic force term is discretized using finite element or finite difference approximation methods.

Carignan et al. [CYM92] was the first to recognize the need of simulating more complex cloth models. Prior to this work, only simple rectangular garments such as flags, table cloths, carpets were simulated. The authors considered modeling the cloth as a tailor would do in real life. Complex objects with many surface panels such as jackets and trousers were designed as an assembly of multiple individual two-dimensional garments. This kind of modeling posed the need of seaming different cloth pieces together and correctly calculating their interactions. Therefore, calculation of cloth-to-cloth and the cloth-to-body collisions had to be addressed. As the general framework, the elastic deformation model presented in [TPB87] was adapted. The only difference was the way the damping was handled. Instead of using a scalar damping term as in [TPB87], a dissipative damping term is used so that no matter where the energy comes from, it will be dissipated. Furthermore, a new collision response algorithm is proposed. In order to deal with the ill-effects of potential field method suggested by [TPB87], the law of conversation of momentum is proposed for the calculation of collision responses. This method eliminated the bouncy effects of collisions and resulted in perfectly inelastic bodies.

Baraff et al. [BW98] departed from the rectangular modeling and triangulated the cloth mesh. The continuum formulations are applied on per-triangle basis. The stretch and shear forces are formulated to resist in-plane stretching and compression and are calculated separately for each triangle. On the other hand, the bend forces are calculated based on the angle between triangles and they are responsible for out-of-plane deformations. One important feature of the method is that unlike previous metric-tensor-based formulations [TPB87] [CYM92], in this work, the deformation energies are formulated as quadratic functions of positions instead of quartic functions. Therefore, better performance is acquired.

Another popular method in cloth modeling is the mass-spring model. In this model, the cloth is treated as a particle system. The cloth is represented as a set of particles with given masses and particles are connected to each other through damped springs. Breen et al.[BHW92] were the first to depart from continuum mechanics and embrace a particle-based approach for cloth simulation. The microstructure of woven cloth is considered. Thus, the cloth was modeled as a set of intersecting warps and wefts with particles at their crossing points as shown in Figure 2.1. They employed potentially the most scientific approach in cloth modeling, by making use of the Kawabata machine. The Kawabata machine is a fabric testing device that measures the mechanical properties of fabrics. It is possible to measure the correct stretch, shear and bend energies of different types of garments such as cotton, wool and polyester. The measurements acquired from the Kawabata machine are used to describe the basic mechanical inter-

actions of the cloth such as stretching, trellising, bending and collision. As a result, the static draping of specific kinds of woven cloth was successfully computed.

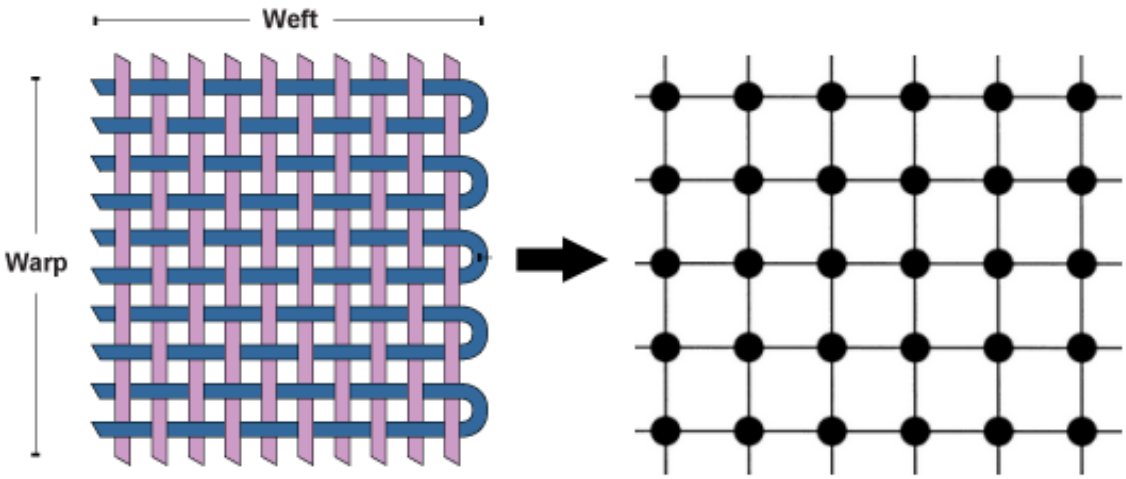

Figure 2.1: Cloth is modeled as a set of intersecting warps and wefts with particles at their crossing points.

Provot [Pro95] correctly pointed out an important drawback of particle-based cloth simulations. Cloth in real life is not a super-elastic material. It will not stretch forever if pulled with a great force. As a matter of fact, cloth is a very stiff material that requires the use of stiff springs when simulating. However, simulation of stiff springs has a technical speed barrier. One cannot use large timesteps in the simulation when the system is stiff. This would lead to numerical instability. Hence, springs should have moderate stiffness constants. This brings about the possibility of having highly elastic springs that can eventually lead to super-elastic clothes. In order to tackle the problem of super-elasticity, Provot [Pro95] proposed the addition of a post-processing step that iteratively restores the length of all over-stretched and over-compressed springs to a given acceptable value. This post-processing step that couples the time integration of the simulation is referred as constraint enforcement. It is especially used in cloth simulations in computer games because of its simplicity and high performance. Besides the main focus of this thesis, we also contributed to the improvement of constraint enforcement process in cloth simulations in [OK11], which will be further detailed in Chapter 6.

Eberhardt [EWS96] transitioned to the Lagrangian dynamics formulations. The cloth-specific inner forces are calculated based on the gradient of energy functions that originate from Kawabata measurements [BHW92]. The potential energy is calculated based on the external forces such as gravitation, air resistance, collision responses and air flows.

2.2.2 Integration Methods

In physical simulation, the objective is to accurately and efficiently compute the evolution of the physical system in time, given initial conditions. This class of problems is called initial value problems. In initial value problems, the behavior of the system is defined by an ordinary differential equation (ODE). Various numerical solutions can be used to solve the ODE's. Numerical methods used in cloth simulation are discussed below.

2.2.2.1 Euler's Method

Euler's method is the simplest numerical method widely used in the physically-based simulation community. Given the initial state of the particles, the method takes a forward step in time by a given step size h.

$$\boldsymbol{x}(t_0 + h) = \boldsymbol{x}(t_0) + hf(\boldsymbol{x}_0), \tag{2.2}$$

where $\boldsymbol{x}$ is the position and $f(\boldsymbol{x}) = \dot{\boldsymbol{x}} = \boldsymbol{v}$ is the derivative of the position. The new velocity at timestep $t + 1$ is calculated based on the current acceleration at timestep t. The new position at timestep $t + 1$ is calculated based on the current velocity at timestep t. Although it is a very simple and fast method, it has low accuracy. If the solution was always a line, then Euler's method would always be exactly correct all the time. However, the solution is a curve; which makes the method's accuracy highly dependent on the timestep size used . Large timestep size often leads to less accurate and possibly numerically unstable simulations. In case of stiff systems, in which the inner forces produced are relatively large, it is almost guaranteed that using large timesteps will make the simulation diverge, causing it to produce more and more errors. These kinds of simulations are called unstable simulations and visually they result in unrealistically moving particles. Therefore, for stiff systems, it becomes a necessity to use very small timesteps to make Euler's method stable. However, this leads to very poor performance since the simulation has to compute a large number of timesteps to move the simulation in time. One can easily see that the problem with Euler's method is that the slope calculated at the current position lacks accuracy and is the primary cause of error generation. The slope can be better approximated in other numerical methods such as Midpoint and Runge-Kutta methods.

2.2.2.2 Midpoint Method

The midpoint method attempts to correct the problematic slope in Euler's method by a two-step process. It first calculates the slope at the current position, moves forward in time for half of a timestep and re-calculates the slope for the second time. The average

of the first and second slopes are calculated and it is applied from the initial position for a full timestep and the next position is calculated.

$$\boldsymbol{x}(t_0 + h) = \boldsymbol{x}(t_0) + hf(\boldsymbol{x}_0 + \frac{h}{2}f(\boldsymbol{x}_0)), \tag{2.3}$$

When evaluating the performance of a numerical solution, one should focus on the performance of the most expensive step of the method. The most expensive step of the method is the calculation of the derivative. Since the derivative is calculated twice for the Midpoint method, the cost of calculating one timestep is more expensive than Euler's method. The Midpoint method is said to be second order method because the derivative is calculated twice. Although its performance is worse than Euler's method, it is more accurate and more stable because the calculation of the slope is improved. This method is also referred as Heun's method, Improved Euler method or Runge-Kutta (Order 2).

2.2.2.3 Runge-Kutta (Order 4)

The Runge-Kutta method goes even further and calculates an even better slope than the Midpoint method. It takes four virtual steps in time and calculates four slopes. A weighted average of these four slopes are later calculated and multiplied with the timestep and used to move a step forward in time. The equations of Runge-Kutta (Order 4) are given by:

$$\begin{aligned}
k_1 &= hf(\boldsymbol{x}_0, t), \\
k_2 &= hf(\boldsymbol{x}_0 + \frac{k_1}{2}, t_0 + \frac{h}{2}), \\
k_3 &= hf(\boldsymbol{x}_0 + \frac{k_2}{2}, t_0 + \frac{h}{2}), \\
k_4 &= hf(\boldsymbol{x}_0 + k_3, t_0 + h), \\
\boldsymbol{x}(t_0 + h) &= \boldsymbol{x}_0 + \frac{1}{6}k_1 + \frac{1}{3}k_2 + \frac{1}{3}k_3 + \frac{1}{6}k_4
\end{aligned} \tag{2.4}$$

This method is very accurate and stable. However, the cost of calculating one timestep is very expensive; therefore it is considered to be a very slow method. Eberhardt [EWS96] was the first the use this method in the scope of cloth simulation.

2.2.2.4 Backward Euler Method

In the forward Euler's method, the simulation takes a step into the future in a blind manner, without caring where the next position will be. On the other hand, the backward Euler's method forces the simulation to find a final state whose derivative points back to the initial state. In other words, the derivatives are calculated at the end of the time step, rather than the beginning. Therefore, the step in the backward Euler's method is based on the conditions at the end of the timestep as shown below:

$$\boldsymbol{x}(t_0 + h) = \boldsymbol{x}(t_0) + hf(\boldsymbol{x}(t_0 + h)), \tag{2.5}$$

This method requires a solution for a linear system, the performance of the solution of a single step is worst than the explicit methods. However, the method is extremely stable and it lets the simulation take very large timesteps, thus resulting in very good overall simulation performance. In cloth simulation literature, methods like Choleski factorization and Gauss-Seidel relaxation are used to solve the linear system. The most widely used method is introduced by Baraff et al. [BW98] in the milestone paper, where a modified version of conjugate gradient method is used.

2.3 Fluid simulation

2.3.1 Navier-Stokes Equations

In physics, the set of equations governing the motion of fluids are called Navier-Stokes equations. The first equation is acquired by applying Newton's second law to motion of fluids and it is called the momentum equation.

$$\frac{\partial \boldsymbol{u}}{\partial t} + \boldsymbol{u} \cdot \nabla \boldsymbol{u} + \frac{1}{\rho}\nabla p = \boldsymbol{g} + \nu \nabla \cdot \nabla \boldsymbol{u}, \tag{2.6}$$

where $\boldsymbol{u}$ is the velocity vector, ρ is the density, p is pressure, $\boldsymbol{g}$ is gravitational acceleration and ν is kinematic viscosity. The equation is a form of Newton's second law of motion, $f = ma$. It describes the acceleration of a fluid body due to the forces acting on it.

The forces acting on the fluid can be categorized in three. The first category of forces is called the body forces. The body forces are the forces that act on all the particles in the fluid. The gravity mg is the main body force used. In computer graphics, different kinds of artificial forces are occasionally used to create various visual fluid effects. These forces can also be categorized under the body forces.

The second fluid force is pressure. The pressure is symbolized by letter p and it is defined as the force per unit area over which that force is distributed. Particles move

from high pressure regions to low pressure regions. If the total forces acting on the particle is zero, the particle won't move. The imbalance of pressure around the particle is what makes it move. The force vector created by the pressure will point away from the high pressure towards the low pressure. Therefore, the pressure force will be the negative gradient of pressure ($-\nabla p$).

The third fluid force is the viscosity. The viscosity of a fluid is a measure of its resistance to the deformation. The viscosity term can be informally defined as the thickness of a fluid. For example, the alcohol has very low viscosity, whereas honey has high viscosity. The viscosity tries to make a particle move at the velocity of neighbor particles around it. It minimizes the difference of velocity between nearby particles. The differential operator that measures how far a quantity deviates from the average around it, is the Laplacian. Therefore, the viscosity force is defined as the Laplacian of velocity multiplied by the viscosity constant. ($\nu \nabla \cdot \nabla \boldsymbol{u}$)

The sum of all these forces should describe how the fluid accelerates. Using $F = ma$, the forces should sum to ma. The acceleration in the equation should be described as

$$\boldsymbol{a} = \frac{D\boldsymbol{u}}{Dt} \tag{2.7}$$

$D\boldsymbol{u}/Dt$ is the material derivative. In continuum mechanics, the material derivative describes the time rate of change of some physical quantity for a material element subjected to a space-and-time-dependent velocity field. It can be defined as:

$$\frac{d}{dt}q(t, \boldsymbol{x}) = \frac{\partial q}{\partial t} + \nabla q \cdot \frac{\boldsymbol{x}}{dt} = \frac{\partial q}{\partial t} + \nabla q \cdot \boldsymbol{u} = \frac{Dq}{Dt} \tag{2.8}$$

There are two terms constituting the material derivative. The first one is $\partial q/\partial t$. It measures how fast the quantity q changes at a fixed point in the fluid. The second one is $\nabla q\boldsymbol{u}$. It measures how much of the total change is due to the fluid flows passing through that region. Putting all three forces together, dividing all terms by volume V and defining the fluid density as $\rho = m/V$, we can get to the Equation 2.6.

The second Navier-Stokes equation is the incompressibility condition and it reads:

$$\nabla \cdot \boldsymbol{u} = 0 \tag{2.9}$$

It states that the divergence of the velocity field should be zero. Divergence is a vector operator that measures the magnitude of a vector field's source or sink at a given point, in terms of a signed scalar. More technically, the divergence represents the volume density of the outward flux of a vector field from an infinitesimal volume around a given point. Therefore, by using this equation, it is guaranteed that fluid entering and exiting a specific region of the fluid will be in balance. Thus, the fluid will be incompressible.

2.3.2 Lagrangian Vs. Eulerian Approaches

There are two major approaches to model fluid. The first approach is the Eulerian approach and it is commonly described by finite difference method [And95][Hir88][Wil99]. The methods based on the Eulerian perspective are called fixed grid methods. In these methods, the field quantities such as velocity, temperature, density are measured at fixed regions in space. The fluid flows through these fixed regions and might potentially change the field quantities, like changing the temperature or velocity of a given region. Eulerian methods are extensively used in computational fluid dynamics. Almost all hydrodynamics problems can be solved using these methods. Even simulations with large deformations such as explosions are numerically solved with fixed grid methods [Ben92] [CA87]. However, these methods have some known disadvantages.

First, it is very hard to simulate complex fluid geometries using this method. The simulations of breaking waves and water splashes are some examples of these complex geometries. It should often be coupled with a mesh generation process to map the complex geometry of the fluid to the problem domain. Second, the motion of the fluid cannot be tracked with a fixed mesh. It is only possible to have the time history of the field attributes. Third, the movement and the deformation of free surfaces are difficult to determine because of the fact that Eulerian methods track the mass and momentum on the cell boundaries. Finally, the size of the grid cells can cause computational complexity. Eulerian methods have to use a grid to completely cover the fluid body. Considering the fact that the grid should also cover regions where the fluid might potentially flow, the use of a coarser grid can be the only way to eliminate the computational complexity. As a result, increasing the resolution of the grid brings about a decrease in the accuracy of the simulation.

The second approach is the Lagrangian approach and it is generally represented by the finite element method [ZT00] [LQ03]. The Lagrangian approach is considered as material description due to the fact that the field quantities such as velocity, temperature, density are carried on the particles that constitute the fluid. The Lagrangian methods have numerous advantages. First, convective terms don't exist in the formulations; thus leading to a simpler and faster simulation. Second, since the field variables are measured on the particles, it is possible to track the history and the evolution of a given field variable on the desired part of the fluid. Third, simulating the complex geometries like water splashes and breaking waves do not pose any problems. These situations can be easily taken care of using an irregular mesh. Last but not least, since the grid is only defined on the problem domain, there is no need for extra grid regions outside the problem domain. Consequently, the Lagrangian methods are faster in terms of performance.

2.3.3 Smoothed Particle Hydrodynamics (SPH)

Smoothed Particle Hydrodynamics (SPH) is a mesh-free, particle-based Lagrangian method. SPH was first used to model astrophysical phenomena [GM77] [Luc77]. Since then, it has been used in fields such as astrophysics, ballistics, volcanology, oceanography, and fluid mechanics.

SPH has several advantages when compared against grid-based methods. First, SPH has an adaptive nature. The field attribute values are calculated on each timestep based on an arbitrary distribution of particles. Thanks to the smoothing kernel functions used in the formulations, even though the particles are unevenly distributed, field attributes can be calculated in a smooth manner throughout the fluid. This feature gives SPH the ability to successfully handling situations with large deformations. The second advantage is mesh-free structure. The adaptive nature allows SPH to have a mesh-free structure, in which there is no connectivity among particles. This structure comes in handy when dealing with problems where finite element methods have difficulties. Another important advantage of the method is that Lagrangian approach matches well with the particle-based structure. Particles carrying field attributes move due to external forces and inner interactions. By acting as both approximation points and material substructures, particles gain neat dual functionality.

SPH has a vast literature and there has been numerous review papers and books published on SPH [Ben90, Mon92, LL03]. In some resources the method is mentioned as Smoothed Particle Mechanics, as it is occasionally applied to solid mechanics problems. Only a selected set of papers will be mentioned here. SPH was first discovered by Gingold and Monaghan [GM77] and Lucy [Luc77] and it was used to model astrophysical phenomena. Following its initiation, SPH was used for the simulations of stellar collisions [Ben88] [Ben90] [Mon92] [FJ99], supernovas [HP99], formation of galaxies [ML91] and detonation of white dwarves [SBW99].

The first applications of SPH to the area of fluid dynamics include the simulations of elastic flow [Swe92], multi-phase flows [MK95], quasi-incompressible flows [Mon94] [MM97], flow through porous media [MYF99] [YFM99], heat conduction [Mon95] [CBC99] and shock simulations [MG83] [Mon89] [MM97].

In this section, we focused on a global review of SPH. We will narrow down our literature review of SPH to fluid simulation in computer graphics in Chapter 4. The first method to introduce SPH into Computer Graphics [MCG03] and its implementation details will be presented in Section 4.3 along with other SPH applications on the fluid simulation.

2.4 Shear Thickening Fluids

The simulation of shear thickening fluids involves concepts from different disciplines. We first review the previous work in physics that attempt to explain the behavior of shear thickening fluids. Then we shortly review a group of rheology and physics papers that use fractional calculus based approaches to describe the behavior of non-Newtonian fluids and viscoelasticity. The work in computer graphics that tackles the problem of simulating non-Newtonian or viscoelastic fluids is very limited yet most relevant to this dissertation. This group of work will be detailed in Chapter 5.

2.4.1 Physics and Rhelology

While several models exist for shear thickening, none have been able to quantitatively predict the dramatic impact resistance observed in dense suspensions such as cornstarch and water. One long-standard model for shear thickening attributes the increased dissipation rate of shear thickening to lubrication forces in the small gaps between densely packed particles [BB85] [WB09]. As the particles are sheared together the lubrication forces increase and the particles form 'hydroclusters' that separate slowly because of the large lubrication forces. Shear thickening is expected when the shear rate exceeds the relaxation rate of these hydroclusters. While this model has successfully described weaker shear thickening at intermediate packing densities, the calculated increase in effective viscosity has been no more than about a factor of 2. When this model is applied to a situation such as an object impacting a dense suspension, the object would be predicted to sink through the suspension with very little reduction in speed.

Another popular model connects shear thickening to dilatancy of a densely packed granular material. When a densely packed suspension is sheared, the particles form force chains between particles in contact and the packing dilates (expands), which causes it to push against the boundaries of the system. The boundaries can then apply a force that propagates back along the force chains. In this model, the energy dissipation in the dilated state comes from solid friction between particles, with forces determined by the stiffness of the boundary. This model has quantitatively explained orders-of-magnitude increase in viscosity observed in shear thickening suspensions in shear rheometry measurements, where the relevant boundary is usually the suspension-air interface and the stress is due to surface tension [BJ12]. However, the dramatic impact resistance observed in suspensions requires a larger effective viscosity that is still larger than found in these shear measurements by orders of magnitude.

A model that is specific to impact response was developed by Waitukaitis and Jaeger [WJ12], where they suggested that colliding particles collect to form a transient solid front that develops underneath the impact site and propagates downward. They found that for fast impacts in a deep pool of dense suspension, the deceleration of an impacting object could be accounted for by conservation of momentum as this solid region acts as

an effective mass accumulating on the end of the impacter. However, this deceleration is still not strong enough to account for the observations that a person can run on top of the surface of the fluid, for example. They also found that if the solid front reaches a hard wall at the bottom of the suspension before the impacter sinks in too far, the impacter may bounce off the surface of the fluid, demonstrating a storage of elastic energy either in the fluid or propagated through the fluid.

All of the aforementioned models predict forces that are at least an order-of-magnitude too small to explain the observation of a person running on top of a dense suspension, and a physical mechanism that can account for the most dramatic impact resistance is still unknown. No physics-based simulations have yet been able to show such dramatic impact resistance.

The importance of hysteresis (history effects) is recently emphasized for the correct description of the fluid. Deegan [Dee10] showed that indentations in the surface of a vibrated shear thickening fluid only become stable if there is hysteresis in the viscosity function. In a different study, Kann et al. [KSL11] suggested that oscillations in the velocity an object sinking in shear thickening fluid may also be due to hysteresis in the viscosity function.

2.4.2 Fractional Calculus and Viscoelasticity

Even though our model is the first to use fractional calculus for the simulation of non-Newtonian fluids in the realm of computer graphics, the approach comes from existing models developed in the physics and rheology fields. Fractional calculus is popular for describing the complex dynamics of relaxation, oscillation and viscoelastic behavior seen in non-Newtonian fluids. In general, fractional calculus is employed by replacing the time-derivatives of strain and stress in well-established ordinary models, by derivatives of fractional order. This results in a more appropriate model for fluids that exhibit both viscous and elastic behaviors. A detailed literature review on the attempts of using fractional derivative models to non-Newtonian fluids in order to obtain analytic solutions is presented in [QJ09]. The first work to adopt fractional derivatives to the problem of viscoelasticity was presented by [Ger38]. Following its introduction, fractional derivative based methods quickly gained momentum in the field and a number of studies has been conducted. Bagley and Torvik [BT83] demonstrated predictive constitutive relations based on fractional derivatives for the viscoelasticity of coiling polymers. Makris et al. [MDC93] was able to successfully fit a fractional derivative based Maxwell model to the experimental data, hence showing that a fractional Maxwell model can replace the Maxwell model for silicon fluids. Later, other applications of fractional Maxwell model are presented. In [HP08], a one-dimensional fractional derivative Maxwell model is used for modeling the linear viscoelastic response of some polymers in the glass transition. Palade et al. [PAH99], reduced the constitutive equation for the incompressible fluid to the linear fractional derivative Maxwell model in the context of small deformations hypothesis.

CHAPTER 3

Underwater Cloth Simulation with Fractional Derivatives

Figure 3.1: Our cloth model based on fractional derivatives is able to achieve realistic underwater behavior.

This chapter introduces the use of fractional differentiation for simulating cloth deformations underwater. The proposed approach is able to achieve realistic underwater deformations without simulating the Eulerian body of water in which the cloth is immersed. Instead, we propose a particle-based cloth model where half-derivative viscoelastic elements are included for describing both the internal and external dynamics of the cloth. These elements model the cloth responses to fluid stresses and are also able to emulate the memory-laden behavior of particles in a viscous fluid. As a result, we obtain *fractional cloths*, which are able to correctly depict the dynamics of the immersed cloth interacting with the fluid even though the fluid is not simulated. The proposed approach produces realistic underwater cloth deformations and has obvious advantages in simplicity and speed of computation in comparison to volumetric fluid simulation approaches.

3.1 Introduction

Cloth simulation has been a central topic in physically-based computer animation. Since the introduction of the first models of deformable surfaces [TPB87, TF88] sev-

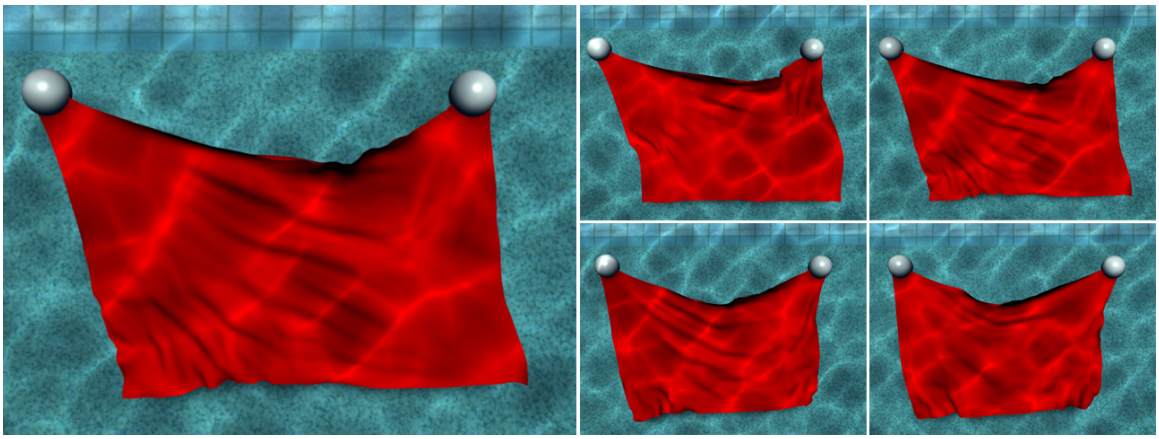

Figure 3.2: Examples of internal effects produced by the viscoelastic spring elements in our fractional cloth formulation.

eral research groups have achieved impressive cloth simulation results and their effects are now commonly seen in computer-generated movies and even in interactive applications such as in 3D computer games. Although several extensions have been proposed in the cloth simulation domain, no specific attention has been given for extending the regular spring mesh cloth model for producing deformations that simulate underwater behavior.

One logical approach for simulating cloth underwater is to embed the cloth model in a fluid simulation system. Such a coupled system can then be solved in different ways, for instance by coupling the Eulerian fluid to a Lagrangian (particle-based) cloth [RSG08], or also by a fully Lagrangian model [MST04]. Although required for accurately modeling the problem, such approaches are expensive to compute as they perform volumetric simulations of the Eulerian fluid surrounding the cloth surface. In particular for the cases when only the deformations of the immersed cloth are needed, a simplified simulation model would be much more practical.

This chapter builds on the key observation that the total force acting on an oscillating particle immersed in a Newtonian fluid is composed of force contributions which are proportional to: 1) the acceleration of the displaced fluid (the virtual mass force), 2) the velocity of the particle (the Stokes drag), and 3) the so-called history (or Basset) force, which accounts for history effects of the moving flow around the particle. These contributions can be derived exactly from the Navier-Stokes equations for the limit of small particles in viscous fluids at low Reynolds numbers. Most importantly, Coimbra and Rangel [CR98] have showed that the Basset force is mathematically equivalent to the half-derivative of the differential velocity between the particle and the far-stream flow.

Motivated by these observations we propose a cloth deformation model which employs the concept of non-integer derivatives for achieving underwater deformation effects. By doing so we obtain *fractional cloths*, which are able to simulate the dynamics of a

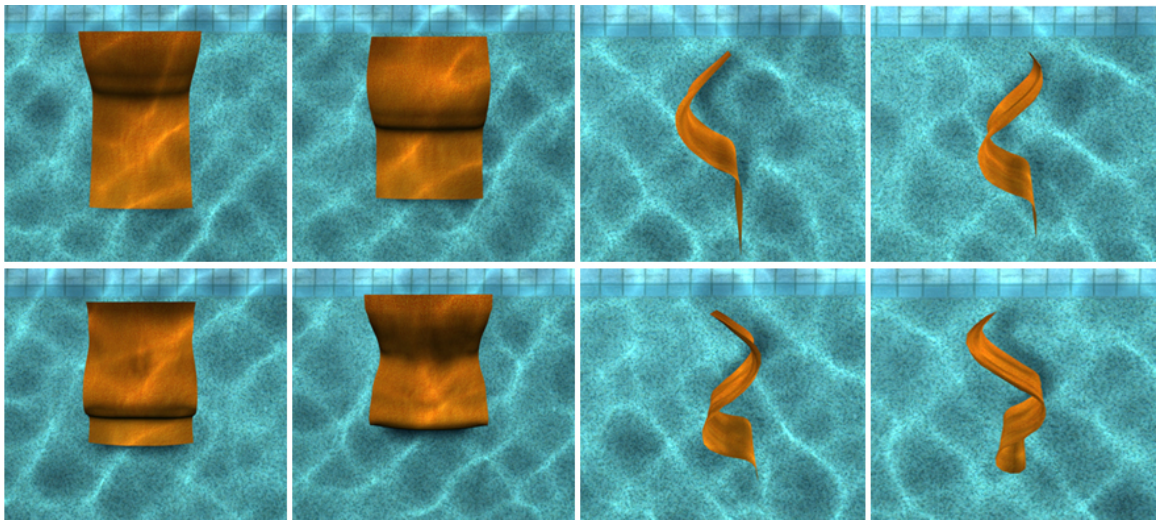

Figure 3.3: The cyclic motion applied to the top of this simulated piece of cloth generates realistic bump propagations.

network of immersed particles without performing expensive volumetric fluid simulations. The complex inertial-viscous balance between the cloth and the fluid is computed in our model by half-derivative terms, which dynamically modulate the cloth damping according to the previous states of the cloth, i. e., the cloth history.

An overview of fractional calculus for modeling the motion of particles is previously given in Section 2.1. After discussing related work in the following section we present our cloth model in Section 3.3. Finally, Section 3.4 presents and discusses the several obtained results and Section 3.5 concludes the chapter.

3.2 Related Work

Cloth simulation is a main topic in computer animation and a rich literature is available [CYM92, BHW92, Pro95, EWS96, VCT95, BW98, HB00, CK02a, TPB87, TF88]. We focus our review on how damping has been used in previous works.

Damping has always been an important aspect of cloth simulation systems. It has been mainly addressed as a viscous drag force with the goal of resisting motion, making particles gradually come to rest in the absence of other influences [BW97]. The viscous drag force is generally perceived as an external force proportional to the velocity: $\boldsymbol{F} = -k_d\dot{\boldsymbol{x}}$ [TPB87, TF88, Pro95], and is sometimes mentioned to be applied opposing the normal direction of the cloth surface [BW98, SSI09].

Damping has also been used as an internal force applied to particles from interaction with other particles. This type of damping is usually coupled with stretch, shear and bend forces, and is justified by the several benefits to the simulation: numerical stability,

oscillation prevention [BW98] and intrinsic damping property of fabrics [CK02a]. The damping force in this case is proportional to the relative velocity between a pair of connected particles, and is usually computed as $\boldsymbol{F}_{ij} = -k_d(\dot{\boldsymbol{x}}_i - \dot{\boldsymbol{x}}_j)$, where $\dot{\boldsymbol{x}}_i$ and $\dot{\boldsymbol{x}}_j$ are the velocities of the interacting particles i and j [DSB99, CK02a, HES03].

One particular way of including internal damping forces is proposed by Baraff and Witking [BW98] , where the damping forces only depend on the components of the velocities in the directions of each maintained condition (stretch, shear and bend). In contrast, other works are less detailed in the description of the used damping mechanism [EWS96, VCT95].

Models have also been proposed for achieving realistic cloth deformations under air flows. In particular, Ling et al. [LDG96] and Ling [Lin00] consider incompressible, irrotational and inviscid airflows, and employ two aerodynamic models to compute forces in response to the airflows. They used a quasi-steady force model to deduce the airflow effects from constant or slowly changing free stream velocities. An unsteady force model is used for the cases where the free stream velocity of the air flow changes rapidly and the flow relative to the cloth becomes unsteady.

Tu and Terzopoulos [TT94] describe one of the few works attempting to achieve underwater behaviors with a spring-mass system. They incorporate external forces which depend on the relative velocity between the surface of a deformable fish and the fluid. These forces are normal to the fish surface (similar to a damping force) and are used to achieve realistic locomotion underwater. The new half-derivative terms introduced by our work include additional fluid responses and could also be used to achieve locomotion patterns.

To sum up, we propose in this chapter the concept of employing damping forces which are modulated by half-derivative viscoelastic elements that simulate the Basset drag in Stokes flows. As explained in Section 2.1, our viscoelastic elements are made proportional to the half-derivative of the displacements in order to capture the macroscopic behavior of the whole cloth underwater. Our approach elegantly models the most important contributions of the history drag on the motion of the cloth and is able to produce realistic underwater deformations simulating the viscoelastic cloth-fluid interaction. We also address for the first time the topic of simulating underwater cloth behavior without recurring to a volumetric fluid simulation system.

3.3 Fractional Cloths

The mesh structure of our cloth model is based on the scheme presented by [CK02a]. The cloth is represented as a quadrilateral mesh of particles. Instead of regular damped springs, our particles are connected to each other by massless viscoelastic spring elements. There are three types of spring elements, which are responsible for the stretch, shear and bend forces as shown in Figure 3.5. The connectivity of the springs is

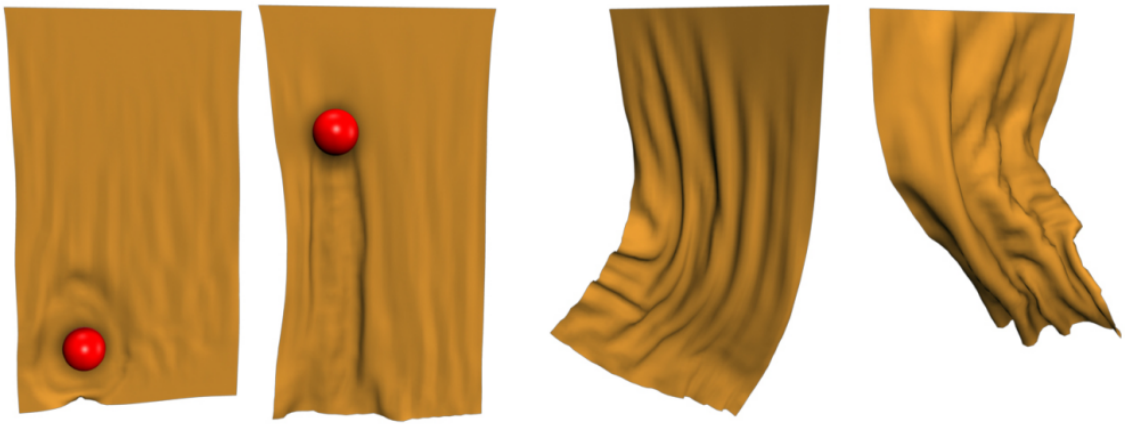

Figure 3.4: The left two figures show consistent propagation of waves caused by a sphere interacting with the cloth model. The right two figures show examples obtained with the cloth top edge being energetically moved sideways. The obtained long lasting wave propagations demonstrate consistent cloth-water inertial responses captured by the viscoelastic half-derivative terms of our fractional cloth model.

described in the following way: a particle indexed by $p(i, j)$ is connected by stretch springs to its neighbor particles indexed by $p(i \pm 1, j)$, $p(i, j \pm 1)$. The shear springs connect the particle to $p(i \pm 1, j \pm 1)$, and the bend springs connect the particle to $p(i \pm 2, j)$, $p(i, j \pm 2)$ and $p(i \pm 2, j \pm 2)$.

3.3.1 Forces

The dynamics of the system is governed by Newton's second law of motion $\boldsymbol{F} = m\boldsymbol{a}$, where m is the mass and $\boldsymbol{a}$ is the acceleration of a particle. The acceleration is computed at every time step based on the total force $\boldsymbol{F}$ applied to the particle, and which accounts for all external and internal forces.

The included external forces are: the gravitational force, the buoyancy of water, a regular viscous drag, and the history drag. The gravitational force and the buoyancy are computed together with:

$$\boldsymbol{F}_g = (\rho_c - \rho_w)\, V_c\, \boldsymbol{g}, \tag{3.1}$$

where ρ_c is the density of a cloth particle, ρ_w is the water density (approximately 1000 kg/m^3), V_c is an estimation of the volume of one cloth particle and $\boldsymbol{g}$ is the gravitational acceleration. The specification of different material densities will lead to different behaviors, ranging from a material which floats to one which quickly sinks. Correctly specifying the density of a wet material is not trivial; however, for the scope of computer animation applications the density can be seen as a parameter which can be fine-tuned in order to achieve different material properties.

External forces for modeling underwater currents can also be added to the global $\boldsymbol{F}_g$

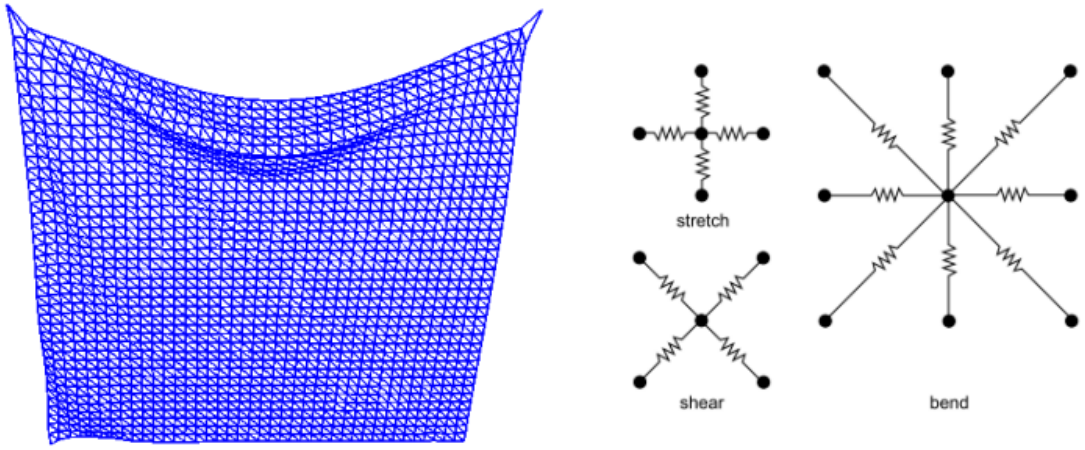

Figure 3.5: Cloth is represented as a quadrilateral mesh of particles. The particles are at the intersection points of horizontal and vertical lines. The particles are connected to each other by elastic springs. The connectivity of stretch, shear and bend springs are also shown.

force term. In our simulations, we tuned ρ_c and V_c such that the particles slightly sink in direction of the gravity force, similar to the behavior observed in real submerged cloths (see Figure 3.7). $\boldsymbol{F}_g$ is considered to be constant in the scope of this paper.

The regular viscous drag is responsible for dissipating kinetic energy and is modeled with:

$$\boldsymbol{F}_{vd} = -k_{vd}\dot{\boldsymbol{x}}, \tag{3.2}$$

where k_{vd} is the viscous drag coefficient and $\dot{\boldsymbol{x}} = \boldsymbol{v}$, is the velocity of the particle. Here we employ the viscous drag force opposing the velocity vector since this choice has showed to produce better results than other options, as for instance using a force opposing the surface normal. The history drag is then added to account for the inertial responses with the fluid and is computed as:

$$\boldsymbol{F}_{hd} = -k_{hd}\left(D^{1/2}\boldsymbol{x}\cdot\hat{\boldsymbol{v}}\right)\hat{\boldsymbol{v}}, \tag{3.3}$$

where k_{hd} is the history drag coefficient and $\hat{\boldsymbol{v}} = \boldsymbol{v}/|\boldsymbol{v}|$ is the unit velocity vector.

As discussed in Section 2.1, the history drag is a half-derivative of the displacements, however note that the drag force should oppose the direction of movement and therefore Equation 3.3 projects the computed half-derivative in the velocity direction. By including the viscous drag and the history drag with different coefficients a greater flexibility can be obtained in fine-tuning the overall drag effect.

The internal forces acting on a particle are the forces which originate from the stretch, bend and shear viscoelastic spring elements connected to each particle. The spring forces are responsible for all the internal deformation responses of the cloth and their

formulation also includes a half-derivative term. The spring force $\boldsymbol{F}_s$ exerted on particle i by the fractional viscoelastic spring element that links particle i to particle j, is formulated as:

$$\boldsymbol{F}_s = -k_s(|\boldsymbol{l}| - r)\frac{\boldsymbol{l}}{|\boldsymbol{l}|} - k_d\left[\frac{(D^{1/2}\boldsymbol{x}_i - D^{1/2}\boldsymbol{x}_j)\cdot\boldsymbol{l}}{|\boldsymbol{l}|}\right]\frac{\boldsymbol{l}}{|\boldsymbol{l}|}, \tag{3.4}$$

where $\boldsymbol{l} = \boldsymbol{x_i} - \boldsymbol{x_j}$ is the difference between the positions of the two particles, r is the rest length of the spring, k_d is the viscoelastic damping constant and k_s is the linear spring constant. Note that while a regular damped spring would have the damping proportional to the first time derivative of the position, our spring element has the damping proportional to the half-derivatives $D^{1/2}\boldsymbol{x}_i$ and $D^{1/2}\boldsymbol{x}_j$, achieving the memory-laden viscoelastic damping term.

3.3.2 Computing the Half Derivative Terms

Although the history integral of Equation 2.1 is singular, it can be numerically integrated with a consistent, first-order accurate, approximation suggested by Coimbra [Coi03], which is an adaptation of the algorithm for general convolution of integrals. The $1/2$ order derivative of $\boldsymbol{x}$ can thus be expressed as:

$$\begin{aligned} D^{1/2}\boldsymbol{x}_n = \frac{h}{6\sqrt{\pi}}\sum_{i=1}^{n-1}\Bigg[&\frac{\dot{\boldsymbol{x}}_{i-1}}{(nh-(i-1)h)^{1/2}} \\ &+ \frac{2\left(\dot{\boldsymbol{x}}_{i-1}+\dot{\boldsymbol{x}}_i\right)}{(nh-(i-1/2)h)^{1/2}} + \frac{\dot{\boldsymbol{x}}_i}{(nh-ih)^{1/2}}\Bigg] \\ &+ \frac{0.15\,h}{\sqrt{\pi}}\left[\frac{\dot{\boldsymbol{x}}_{n-1}}{h} + \frac{2\left(\dot{\boldsymbol{x}}_{n-1}+\dot{\boldsymbol{x}}_n\right)^{1/2}}{(0.55\,h)^{1/2}} + \frac{\dot{\boldsymbol{x}}_n}{(0.1\,h)^{1/2}}\right] \\ &+ \frac{0.05\,h}{\sqrt{\pi}}\left[\frac{8\sqrt{2}}{3}\frac{\dot{\boldsymbol{x}}_n}{(0.05\,h)^{1/2}} - \frac{4}{3}\frac{\dot{\boldsymbol{x}}_n}{(0.1\,h)^{1/2}}\right], \end{aligned} \tag{3.5}$$

where h is the timestep, i is the timestep index and n is the index of the most recent computed timestep.

As can be seen in this formulation, the system is storing and reusing the information of the past in order to calculate the current state. Note that the integral in the equation takes all the previous states of the simulation into consideration. At this point, an obvious problem quickly presents itself. Taking into account all the previous states for computing the current state clearly imposes a serious computational barrier. However, a careful analysis of Equation 3.5 will show that actually only the most recent terms are important to be considered.

Note that the right hand side of Equation 3.5 is a summation of weighted velocities. The weights in the summation cannot be computed in advance as they are different at

each timestep of the simulation. However by analyzing the evolution of these weights (see Figure 3.6), it is possible to observe that their values decay rapidly as we move away from the current state to the past. Therefore, the most recent states have much more influence on the final result of the equation than the states which are far away in the past and we can safely account only for a few of the most recent states when computing the summation of Equation 3.5.

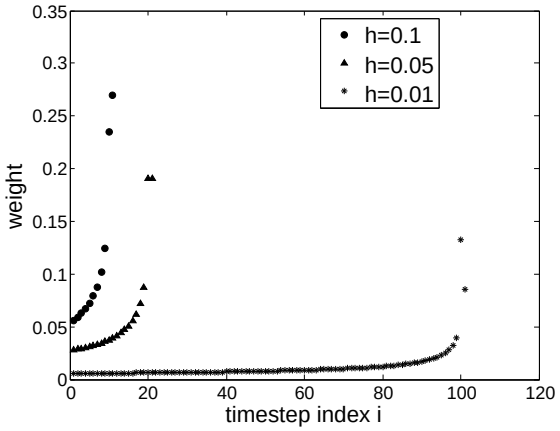

Figure 3.6: The plot shows the rapid decay of the weights as we move away from the most recent timestep index $n = 100$.

To illustrate the evolution of the weights we have computed their values for different simulation settings. The plot in Figure 3.6 shows the decay of the weights as a function of the timestep index i, and for three different values for the timestep h. It is possible to observe that the states closer to the most current timestep ($i = n, n-1, n-2$) clearly have much greater influence than the states further in the past.

We have also tested experimentally the effect on simulations with different numbers of previous states taken into account. Let u be the number of previous states considered in Equation 3.5, such that for a given u the summation index i will vary from $n+1-u$ to $n-1$. We indeed noticed that the change on the cloth behavior gets very small for $u > 3$. We therefore found $u = 3$ to be a good value for the performance/realism tradeoff. While greater values of u will increase the realism of the fluid response effect, the computational cost is also increased.

In Section 3.4 we present comparative results with real submerged cloths showing that the chosen $u = 3$ achieves sufficiently realistic results.

3.3.3 Numerical Integration

The addition of the fractional derivative terms into the equation of motion of the cloth requires the solution of a *Fractional Differential Equation* (FDE). Both the explicit and implicit numerical solution techniques available for Ordinary Differential Equations

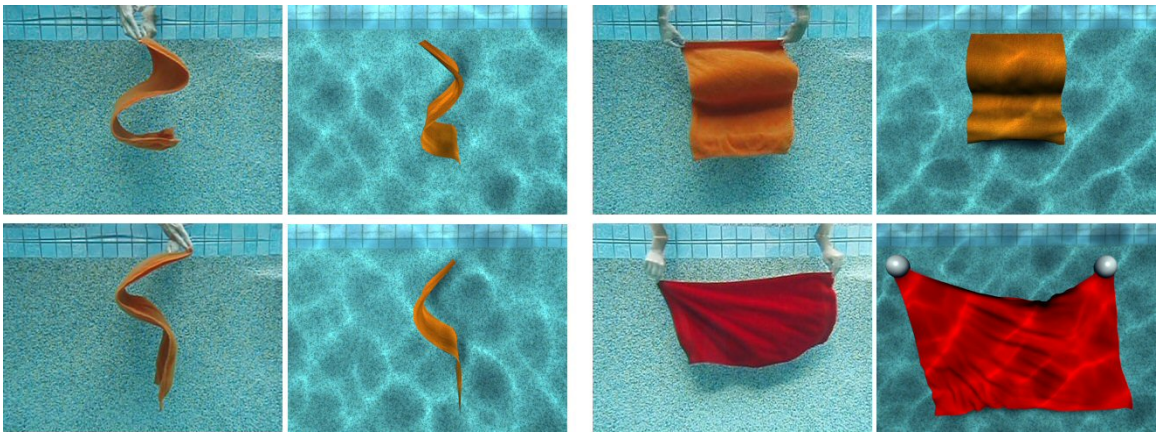

Figure 3.7: Comparisons between real cloths and simulated fractional cloths. In each example, a cyclic motion is applied to the top edge of the cloth. The simulations with the orange cloth model produce bumps which are well formed and which propagate in a consistent way, similarly to the ones observed in the real cloths. The red cloth simulation reproduces similar fine details in the interior of the cloth.

(ODEs) can be extended to FDE's.

Our present work does not focus on developing and comparing different integration methods, and in this thesis we have implemented an explicit Euler integration step in order to test and investigate the results achieved by our model. At each timestep the accelerations are computed from the overall force contributions to each particle, and then the system takes one step with:

$$\begin{aligned} \boldsymbol{x}(t+h) &= \boldsymbol{x}(t) + h\boldsymbol{v}(t), \\ \boldsymbol{v}(t+h) &= \boldsymbol{v}(t) + h\boldsymbol{a}(t). \end{aligned} \tag{3.6}$$

The explicit step has been sufficient for the examples simulated in this work and the inclusion of the half-derivative terms does not affect the stability of the integration. Coimbra [Coi03] suggests an improved explicit method for solving a 2^{nd} order differential equation with a general order damping term by using an adaptation of the Euler method. The approach approximates the first and second derivatives of $\boldsymbol{x}$ using a forward finite-differences method and uses an explicit marching method for the solution of the initial value problem.

Numerical solution techniques for FDEs consisting of an explicit predictor step followed by an implicit corrector step have also been developed [DFF05, Coi03]. For stiff equations these methods can be improved by repeatedly iterating the corrector step until a desired convergence is reached.

In addition to these methods, fully implicit integration schemes such as the approach suggested by Baraff and Witkin [BW98] can also be adapted to solve our FDEs. The

required partial derivatives of the half-derivative terms can be obtained by computing the partial derivatives of the Riemann-Liouville differential operator (shown in Equation 2.1).

3.4 Results and Discussion

Figures 3.1-3.4 present several results obtained with the proposed fractional cloth model. In order to demonstrate the capabilities of our model we have also compared our results against real pieces of cloth deforming underwater. The underwater behavior of real cloth is recorded through an underwater camera as shown in Figure 3.8

Figure 3.8: The underwater behavior of real cloth is recorded through an underwater camera.

Figure 3.7 shows several comparisons with real cloths immersed in a swimming pool. We tested the behavior of different kinds of fabrics underwater and rectangular shaped fabrics were the most suitable for performing the presented comparisons. Several of the fabrics just floated because they were not dense enough and cotton towels worked best for the experiment. Basic patterned movements such as swinging the cloth back and

forth, pulling it left and right, etc, were then applied to the immersed pieces of cloth for comparison with our simulated results. These simple motions were selected with the purpose to facilitate the evaluation of the comparisons.

It is possible to notice that when swinging the cloth back and forth, the cloth forms deep bumps which slowly and consistently propagate from the source of the applied motion (the top edge) towards the other end of the fabric (the bottom edge). The created bumps consistently last until the bottom edge of the cloth is reached and this behavior is a good justification of how fractional derivatives with their history-laden properties are suitable for simulating an underwater cloth. Fractional cloths are able to realistically simulate these observed effects as shown in Figure 3.9.

Figure 3.12 compares our fractional model with three regular cloth simulation models based on damped springs subjected to different drag forces. Note that all the cloths are under water buoyancy and the first three models do not include fractional terms. The first cloth model is affected by both viscous drag and normal drag forces. The second model is only affected by viscous drag forces. In the third model we only used normal drag forces. The employed normal drag force worked best in its linear form: $-k_d(\dot{\boldsymbol{x}}.\hat{\boldsymbol{n}})\hat{\boldsymbol{n}}$. In all cases we fine-tuned the drag coefficients in order to get as close as possible to the observed underwater behavior. These three regular models represent our best attempts to achieve underwater effects by tuning as much as possible the parameters of regular cloth deformation systems. As can be observed in Figure 3.12, although regular models are able to form some of the observed bumps, they are not always well shaped and several times they are overly long, covering the whole extent of the fabric and giving the cloth a bent look. The long-lasting shape and the consistent propagations cannot be correctly simulated with regular models.

In the right-most column of Figure 3.12 we show results obtained by our fractional cloth model. The parameter settings are the same as the ones used in the regular models, but now the internal damped springs are replaced by the viscoelastic spring elements and the history drag force $\boldsymbol{F}_{hd}$ is added. We can therefore evaluate the new effects produced by the half-derivative terms. The comparisons in Figure 3.12 show that fractional cloths were able to best reproduce the observed behavior of the real immersed cloths. The simulations formed well-shaped and deep bumps which were locally formed near the motion source (the top edge) and did not cover the whole extent of the cloth, similarly to the behavior observed in the real cloths.

While the performed comparisons (Figures 3.7 and 3.12) provide objective evaluations of our method in respect to global behaviors, the additional animations presented in Figures 3.1, 3.2, 3.4 and 3.11 show several realistic local deformation effects obtained.

We have also compared the behavior of our fractional cloth model against a regular model in a situation with more energetic motions. The simulated scene consists of a cloth model colliding and interacting with a moving solid sphere, as shown in Figure 3.10. The same simulation was performed with both our fractional model and our best regular cloth model, which is the one using only normal drag forces. The ob-

tained animations show that our fractional model can clearly generate waves on the cloth which propagate more consistently and which last longer, in comparison with the regular model. These animations are available in the videos accompanying this work.

Our simulations were conducted on a 2.4 GHz Intel Core 2 Duo desktop computer. Cloth meshes of both 2501 particles in a 61×41 mesh grid and 651 particles in a 31×21 mesh grid were used in the swinging simulations in order to observe the resulting effects of fractional terms in models with different resolutions. These simulations were performed with $u = 3$, i.e., using 3 states in the past for the computation of Equation 3.5. In order to handle collisions, these simulations also include a standard collision avoidance repulsive force opposing the direction of potential collisions. The implementation of the cloth with 31×21 mesh grid achieved the average computation time of about 9 frames per second for each cloth piece. The same simulations, when performed with the regular cloth models (without the fractional terms), could be computed in the order of 2 to 3 times faster, however without achieving the underwater behaviors. In contrast, reported times taken by complete fluid simulation systems commonly run in the order of a few minutes to several hours per frame of simulation in high performance machines [RSG08].

One of the caveats of our approach is that the large scale interactions of the flow with the cloth are not modeled. The same is true for the flow disturbances caused by the cloth from other regions in the flow. The model does however include the implicit effect of the presence of the cloth in the flow in the vicinity of each node of the calculation, and with this effect it is able to produce realistic results at a small fraction of the computational resources needed to model the two-way coupling between the flow and the cloth.

3.5 Conclusions

We have introduced a new methodology for modeling immersed deformable models which is based on the use of Fractional Calculus and we have demonstrated that our fractional cloth model is able to achieve realistic underwater behavior. We have also presented several results which clearly demonstrate the benefits of our method. The introduction of the history terms computed by the half-derivatives proves to be very effective in achieving the expected viscoelastic behaviors of the immersed cloth.

The fact that realistic behaviors can be achieved without modeling or simulating the body of water is remarkable and the proposed approach will prove itself useful to a number of applications employing deformable objects. For instance, the presented methodology can be integrated into the deformation equations of any spring-based structures such as the ones used in hair simulation or in several volumetric models common in surgical simulations. We believe that fractional derivatives coupled with physically based models have the potential to impact several applications in computer graphics.

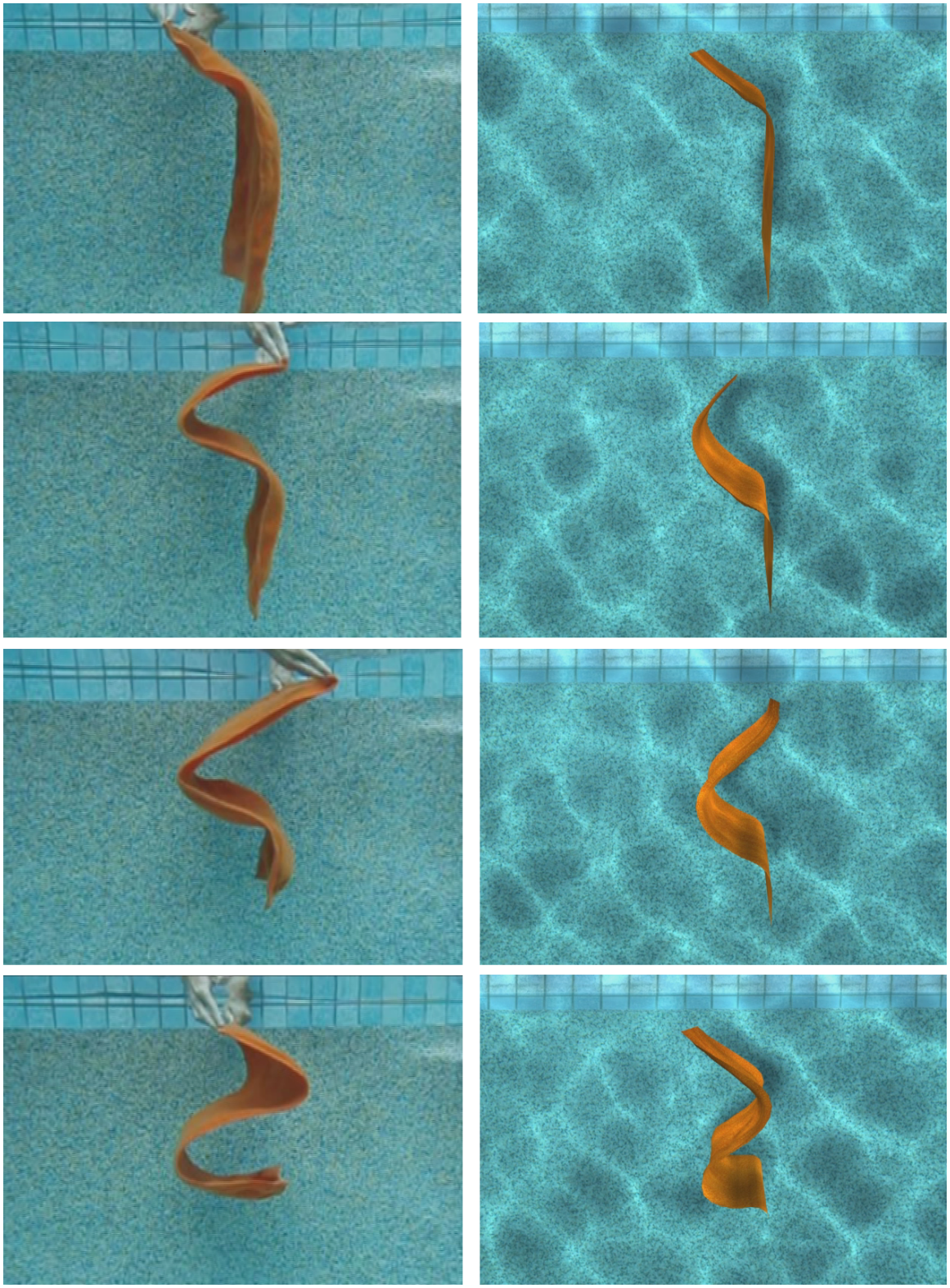

Figure 3.9: The figure shows the side-by-side comparison of our cloth model against the real cloth.

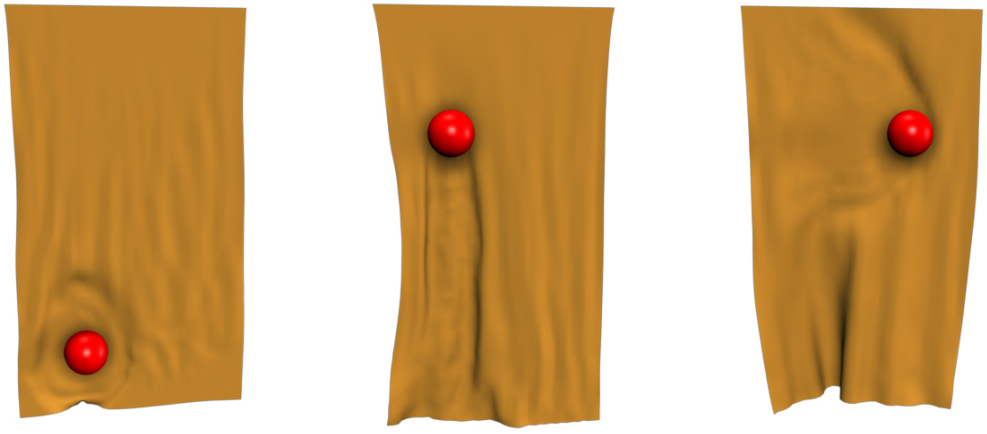

Figure 3.10: The figure shows the response of cloth when contacted with a solid sphere in various ways.

Figure 3.11: The figure shows snapshots from underwater dance animation.

Figure 3.12: Comparisons between regular cloth deformation models using various drag models (left three columns) and fractional cloths (right-most column).

CHAPTER 4

Simulating Colliding Flows in SPH with Fractional Derivatives

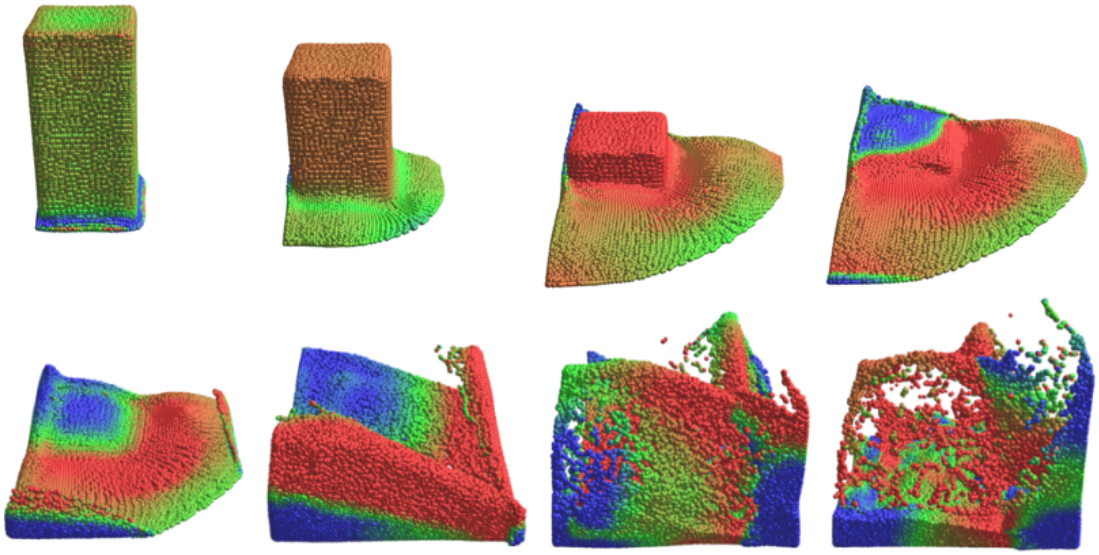

Figure 4.1: Example of a typical SPH simulation scenario. As demonstrated in several evaluations, our Fractional SPH model will improve the realism of the simulation in a chosen resolution.

This chapter describes a method based on fractional differentiation for improving the efficiency and realism of simulations based on Smoothed Particle Hydrodynamics (SPH). SPH represents a popular particle-based approach for fluid simulation and a high number of particles is typically needed for achieving high quality results. However, as the number of simulated particles increases, the speed of computation degrades accordingly. The proposed method employs fractional differentiation to improve the results obtained with SPH in a given resolution. The approach is based on the observation that effects requiring a high number of particles are most often produced from colliding flows, and therefore, when the modeling of this behavior is improved, higher-quality results can be achieved without changing the number of particles being simulated. Our method can be employed to reduce the resolution without significant loss of quality, or to improve the quality of the simulation in the current chosen resolution. The advantages of our method are demonstrated with several quantitative evaluations.

4.1 Introduction

The method of Smoothed Particle Hydrodynamics (SPH) has become a popular particle-based approach for fluid simulation because results incorporating complex interactions (e.g., splashes, coupling, etc.) can be obtained with relatively modest computational complexity [DC96, MCG03, ODA07, BT07, SP09]. Key to the quality of the results is the determination of an appropriate number of particles achieving sufficient volumetric density. While better results are, in principle, obtained with high concentrations of particles, the computational penalty is significant. Even if specific data structures are used to improve the computation performance, it is always a main concern in practical applications to reduce the computational time while still obtaining high quality simulation results. This topic is central to the effective implementation of SPH applications, due its suitability to high frame rate interactive applications such as for interactive virtual worlds.

This chapter presents a novel approach to address this problem with the introduction of *fractional derivatives* [OS74] [Pod99] to the SPH equations. The proposed method is able to achieve high quality simulation results with a relatively lower number of particles, achieving results similar to the ones obtained with higher resolution simulation (see Figure 4.1).

Our work is based on the observation that in regions of the fluid where the flows do not collide, a low resolution representation can still correctly represent the fluid because the flows evolve steadily in those regions. However, in regions where the flows collide, the responses to be simulated are higher in number and complexity and a low number of particles will not adequately capture all the occurring phenomena. In these situations, we show that the introduction of a history-laden viscous effect based on fractional forces will restore some of the lost behavior and improve the collision representation, leading to a lower concentration model that yields realistic behavior comparable to the ones obtained with higher concentrations of particles.

It is reasonable to expect that fractional terms can compensate for the loss of information in a particle-based flow and restore the expected behavior of the small volume of water represented by each particle. Fractional derivatives have been demonstrated to efficiently model memory-laden effects that are observed in immersed particles [CR98]. The force contributions acting on an oscillating particle immersed in a Newtonian fluid are proportional to: 1) the acceleration of the displaced fluid (the virtual mass force), 2) the velocity of the particle (the Stokes drag), and 3) the so-called history (or Basset) force, which accounts for history effects of the moving flow around the particle. In the limit of infinitesimal particle Reynolds numbers, the formulation including steady Stokes drag, virtual mass and the Basset fractional term is exact. These contributions can be derived directly from the Navier-Stokes equations for the limit of vanishingly small convective effects. Coimbra and Rangel [CR98] have shown that the Basset force is mathematically equivalent to the half-derivative of the differential velocity between the particle and the far-stream flow. These results indicate that the behavior of im-

mersed particles can be well represented with models based on fractional derivatives. The concept has been well demonstrated by Ozgen et al. [OKR10] on the problem of simulating cloth deformations with underwater behavior.

Our model compensates for the loss of information in a lower resolution simulation by adding a half-derivative viscosity in the regions of the flow detected to have significant variations in velocity, a situation that happens in colliding flows. When the simulation is implemented with low concentration of particles, local particle-particle interactions are not enough to correctly depict the flow interactions and memory-laden half-derivatives are used to compensate for the resolution decrease.

4.2 Background and Related Work

Smoothed Particle Hydrodynamics (SPH) has a long history in physics, developed in 1977 by Gingold and Monaghan [GM77] to model astrophysical phenomena, and extended to solve many problems in continuum mechanics. There are many uses of particle systems in Computer Graphics, however discrete formulation of continuous fields by particles was first introduced by Desburn et al. [DC96] for simulating highly deformable bodies. Muller et al. [MCG03] reached very promising results in particle-based fluid simulation for interactive applications using the SPH method. A very detailed study of SPH since its first emergence is presented by Monaghan [Mon05].

In recent years, new variations to the standard SPH models have also emerged. Solenthaler [SP09] proposed the PCISPH method for reducing the computation time of standard SPH and increasing the incompressibility of the fluid by employing a prediction-correction scheme based on particle pressures. Raveendran [RWT11] introduced a hybrid approach that uses a Poisson solver along with a local density correction step to increase the stability of SPH method in higher time steps. Solenthaler [SG11] proposed a two-scale simulation by merging the results from low and high resolution simulations running simultaneously. Adaptive time steps are employed by Ihmsen and Adams [IAG10, APK07] in SPH methods to increase the stability of the simulations. SPH applications based on parallel computing are also proposed by various groups [IAG10, IAB11, HKK07].

Several researchers have addressed the behavior of fluid flows in Computer Graphics. Elcott et al. [ETK07] modeled the fluid flows by satisfying the conservation of circulation along arbitrary loops. The nature of flows flowing through porous materials is described by Lenaerts et al. [LAD08] with the combination of SPH and the Law of Darcy. The behavior of turbulent fluids is defined using a large-scale numerical solver and physical energy models in [NSC08]. Pfaff et al. [PTS09] handles turbulence around objects immersed in flows by modeling turbulence formation with averaged flow fields. Another interesting work is the study of viscoelastic incompressible fluids in which additional elastic terms are integrated in the Navier-Stokes equations [GBO04a]. In Treuille [TLP06], a real-time method for acquiring detailed flows with a small number

of basis functions is presented. Our present work contributes to the field by introducing fractional derivatives as a new tool for improving fluid simulations.

In Section 2.1, we discuss the fundamental results on the motion of the particles in unsteady viscous fluids by [CR98] [LCT05]. Motivated by these results, in this chapter, we aim to increase the physical accuracy of simulating flow collisions in low resolution simulations by utilizing a fractional derivative model. We thus propose a new SPH model with half-derivative viscosity terms to compensate for the loss of information in low resolution simulations.

Our proposed model uses history-laden viscous terms that are made proportional to the half-derivative of the relative particle displacements in order to better capture the macroscopic behavior of colliding flows. We consider the history effects of interest by applying the half-derivative operator to the relative particle displacements. The achieved model is able to simulate the different inertial responses among the particles in an efficient way, while capturing the intermediate mechanical behavior between pure elasticity and pure viscosity.

The proposed model is presented in the next section. We then present two experiments that validate the proposed approach by showing that our model generates realistic results in a well-understood fluid simulation scenario and in comparison with an accurate Navier-Stokes solver (Section 4.4). The following section (Section 4.5) presents several evaluation results quantifying the improvements obtained in comparison with the standard SPH model. The final sections discuss the results obtained and then conclude the chapter.

4.3 Model

In this section we describe our proposed Fractional SPH model.

4.3.1 Standard SPH

The Smoothed Particle Hydrodynamics (SPH) model we employ is based on the scheme presented by Muller [MCG03]. SPH is a Lagrangian model where the fluid is represented by a set of particles that carry field attributes. An arbitrary attribute on a given particle's position is computed via smoothing kernels that only consider nearby particles within the core radius h. The smoothing of attributes is modeled with:

$$A_S(r) = \sum_j m_j \frac{A_j}{\rho_j} W(\mathbf{r} - \mathbf{r}_j, h), \tag{4.1}$$

where m_j is the mass, r_j is the position and ρ_j is the density of a particle j within the

core radius h of the smoothing kernel $W(r - r_j, h)$. A_j is the field attribute quantity at r_j.

At each timestep of the simulation, the density values of individual particles are evaluated first:

$$\rho_i = \sum_j m_j W(|\mathbf{r}_i - \mathbf{r}_j|, h), \tag{4.2}$$

then, the pressure is computed by the ideal gas state equation

$$p = k(\rho - \rho_0), \tag{4.3}$$

where k is a gas constant and ρ_0 is the rest density. Once the density and the pressure fields are computed, the pressure and viscosity forces acting on particle pairs are computed in a symmetric manner as proposed by Muller [MCG03]:

$$\mathbf{f}_i^{pressure} = -\sum_j m_j \frac{p_i + p_j}{2\rho_j} \nabla W(\mathbf{r}_i - \mathbf{r}_j, h), \tag{4.4}$$

$$\mathbf{f}_i^{viscosity} = \mu \sum_j m_j \frac{\dot{\mathbf{x}}_j - \dot{\mathbf{x}}_i}{\rho_j} \nabla^2 W(\mathbf{r}_i - \mathbf{r}_j, h), \tag{4.5}$$

where $\nabla W(\mathbf{r}_i - \mathbf{r}_j, h)$ is the gradient, $\nabla^2 W(\mathbf{r}_i - \mathbf{r}_j, h)$ is the Laplacian of the kernel, μ is the viscosity constant, and finally $\dot{\mathbf{x}}_i = \mathbf{v}_i$ and $\dot{\mathbf{x}}_j = \mathbf{v}_j$ are the velocity vectors of particles i and j respectively.

4.3.2 Fractional SPH

In order to represent the memory-laden characteristics of the fluid body, we introduce the fractional viscosity term of order $1/2$ to the motion of the particles. We achieve this by replacing the first time derivatives of the positions by the half-derivatives of the positions. As a result, the history-based viscosity is defined as:

$$\mathbf{f}_i^{viscosity} = \mu \sum_j m_j \frac{D^{1/2}\mathbf{x}_j - D^{1/2}\mathbf{x}_i}{\rho_j} \nabla^2 W(\mathbf{r}_i - \mathbf{r}_j, h) \tag{4.6}$$

where $D^{1/2}\mathbf{x}_i$ and $D^{1/2}\mathbf{x}_j$ are the half-derivatives of the positions of particles i and j respectively. Note that the viscosity force is now proportional to the difference of the half-derivatives, achieving the memory-laden viscosity needed to define the motion resulting from flow collisions.

The memory-laden viscosity is especially well suited for fluid phenomena occurring in intense flow collision regions. It is important to observe that in most situations a fluid simulation scenario will contain both regions with flow collisions and regions without any collisions. Our model will improve the quality of the simulation in the regions with flow collisions, which are often the regions producing the most interesting behaviors. Our model will not affect the results obtained in regions with steady flows.

4.3.3 Computing the Half Derivative Terms

In Coimbra [SCK05], a second-order accurate quadrature formula derived using the product trapezoidal method is suggested for derivative orders q in the $0 < q < 1$ range. This fractional-order differential operator reads:

$$D^q \mathbf{x}_n = \frac{h^{1-q}}{\Gamma(3-q)} \sum_{i=0}^{n} a_{i,n} D^1 \mathbf{x}_i, \tag{4.7}$$

$$a_{i,n} = \begin{cases} (n-1)^{2-q} - n^{1-q}(n+q-2) & \text{if } i = 0, \\ (n-i-1)^{2-q} - 2(n-i)^{2-q} + (n-i+1)^{2-q} & \text{if } 0 < i < n, \\ 1 & \text{if } i = n, \end{cases}$$

where q is the derivative order $(0 < q < 1)$, n is the index of the most recently computed timestep, $a_{i,n}$ is the weight of timestep index i at timestep n, and $D^1 \mathbf{x}_i = \mathbf{v}_i$ is the velocity of a particle at timestep i. In comparison to the method employed by Ozgen et al. [OKR10], this formulation is simpler and more accurate. In the presented simulations we have used this latter formulation with $q = 0.5$ to acquire the half derivatives.

The fact that computing the half derivative of the position of a particle makes use of all the past velocities of that particle seems to be a computational barrier at first. However, as stated in Ozgen [OKR10], an analysis on the evolution of the weights used for the fractional derivative computation shows that the most recent states have much more influence on the final result of the equation. The plot in Figure 4.2 shows the evolution of the weights as a function of the timestep index i for three different timestep values h. Thus, we only consider the last ten timesteps when computing the half-derivative terms.

4.4 Validation Experiments

In order to validate the approach taken by our model, we have performed two experiments designed to verify if the results produced by our simulations are in accordance with expected known results. The first experiment simulates a solid sphere in a viscous fluid, and the second experiment simulates the Shear Driven Cavity Test [LL03].

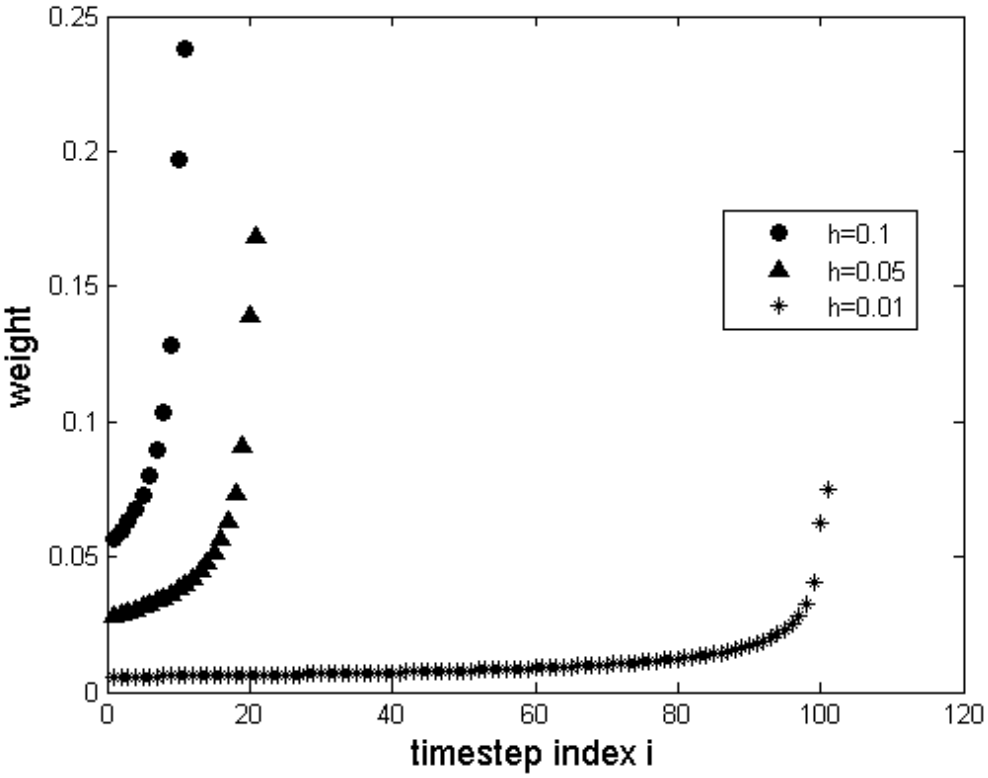

Figure 4.2: The plot shows how weights rapidly decay as we move away from the most recent timestep.

4.4.1 Solid Sphere in a Viscous Fluid

L'Esperance et al. [LCT05] describe the experiment where a single small rigid sphere is placed inside a container full of fluid, and the container is then continuously shaken by a sinusoidal force. The reason for shaking the container is to create a medium rich in oscillatory flows. The authors state that under certain conditions of particle size, fluid viscosity and oscillation frequency, the Basset history term in the equation of the particle motion becomes dominant. Because the Basset history term can be written in terms of the half-derivative of the particle's velocity [CR98], the authors show that the recorded path of the sphere in the experiment perfectly matches the solution to the sphere's equation including history effects through the use of fractional derivatives. In the described experiment the trajectory of the sphere will closely match a perfect sinusoidal curve. This provides a ground truth target to achieve with SPH simulations of the experiment, and provides us with an experiment to validate the results produced by our Fractional SPH model.

We have implemented an experimental simulation version of the scenario explained in L'Esperance et al. [LCT05]. We have prepared a 3D fluid simulation of 12K particles,

with the fluid located inside a closed rectangular container, and with a solid sphere immersed in the center of the container. The density of the solid sphere was set to be slightly greater than the density of the fluid by increasing the particle's mass by 50% and keeping the particle's volume the same. In the original experiment, the tracked particle was tethered by very thin copper wires to counteract the weight and buoyancy. We achieved the same effect by adjusting the gravity force so that the particle neither sank nor floated. We applied a sinusoidal shaking force to the fluid container, and the fluid was simulated with both standard SPH and our Fractional SPH.

We conducted three different experiments. In the first experiment, we simulated the water flow with the standard SPH model. In the second experiment, we simulated the water with 40% fewer particles but using our Fractional SPH with history forces included through fractional derivatives. In the last experiment, we simulated the water with the standard SPH in the same resolution of 40% less particles than the high resolution model. The recorded paths of the solid spheres in the three experiments are shown in Figure 4.3. It can be clearly seen that the inclusion of the history forces improved the results of the low resolution SPH by making the path of the solid sphere closer to the path of the solid sphere in the high resolution SPH. In addition, it can be observed that both the low resolution Fractional SPH and the high resolution SPH generated trajectories that are closer to a perfect sinusoidal curve.

We also obtain an indication that SPH simulations can reproduce fluid behaviors as they occur in real life and that by increasing the resolution we can improve the obtained results. This can be noticed by observing that the high resolution SPH experiment produces a curve (blue curve) that looks closer in shape to a well-defined sinusoidal shape than the low resolution SPH experiment (red curve). Therefore, it is reasonable to evaluate our proposed method by applying it at a given resolution and then comparing the results obtained against a higher resolution standard SPH simulation. We will adopt this comparison strategy in the evaluation experiments of Section 4.5. These preliminary experiments also indicate that the addition of fractional terms is beneficial and that higher resolutions in SPH will lead to improved behaviors. These results provide reasonable validation of our proposed approach.

4.4.2 Shear Driven Cavity Test

Our second validation experiment employed a standard test known as the Shear Driven Cavity Test (or Lid Driven Cavity Test) in Fluid Dynamics [LL03]. In this test, flow is generated by moving the top wall of a square box full of fluid while the other three walls are stationary. The top wall of the box moves in the horizontal direction with a constant speed, and the flow reaches a steady state after running the simulation for a while.

We implemented this test with our standard and Fractional SPH implementations, with various number of particles and Reynolds number values. All tests demonstrated that

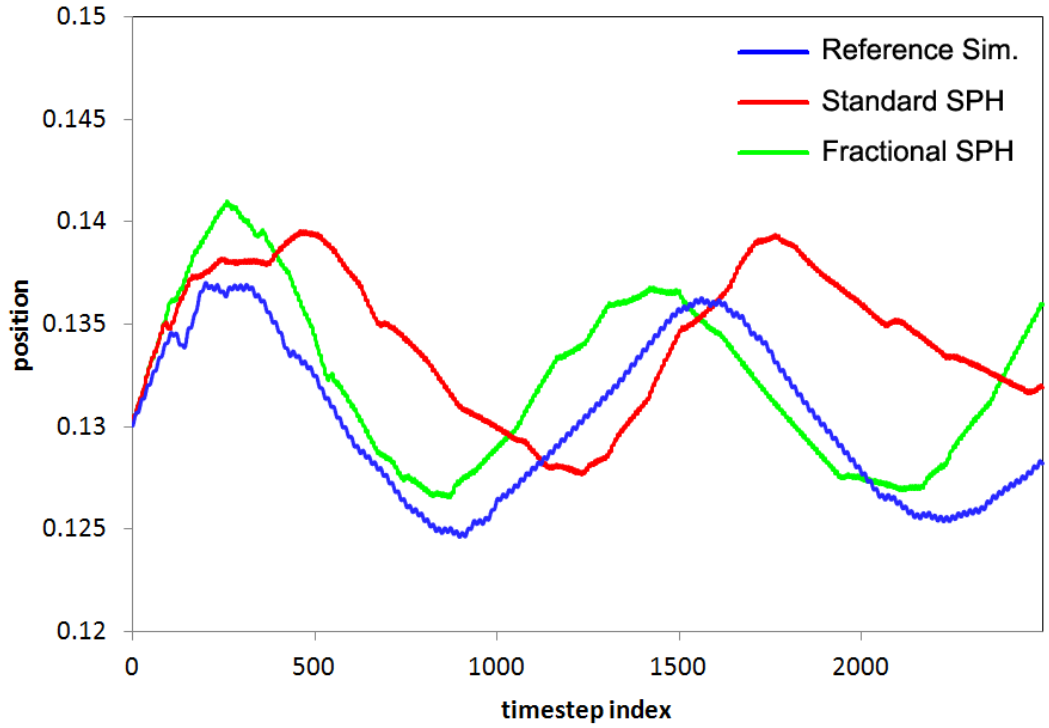

Figure 4.3: The figure shows the path followed by the solid sphere located in a fluid container subjected to a continuous sinusoidal shaking force. The blue, red and green curves show the path of the sphere in a fluid simulated with high resolution standard SPH, low resolution standard SPH and low resolution Fractional SPH, respectively.

our Fractional SPH model produced results closely matching the results computed by a high-precision fluid solver. We compared our results against the results generated by OpenFOAM, a grid-based solver widely employed by the CFD community [WG10]. One example of the obtained results are demonstrated in Figure 4.4. As it can be seen in the figure, standard SPH and Fractional SPH simulations with 40K particles follow the grid-based solution tightly, showing that the viscosity behavior of both fluids are valid and that the use of fractional derivatives in the viscosity formulation does not introduce any additional viscosity to the standard formulation. These results were computed with the simulations on the steady state phase. These results are important to demonstrate that the Fractional SPH model also performs well in steady flow cases. Our model will thus not degrade the results obtained in regions without colliding flows.

4.5 Evaluation Experiments

In order to quantify the improvements obtained with our Fractional SPH model we have designed a simulation scenario based on a rectangular grid that is rich in flow col-

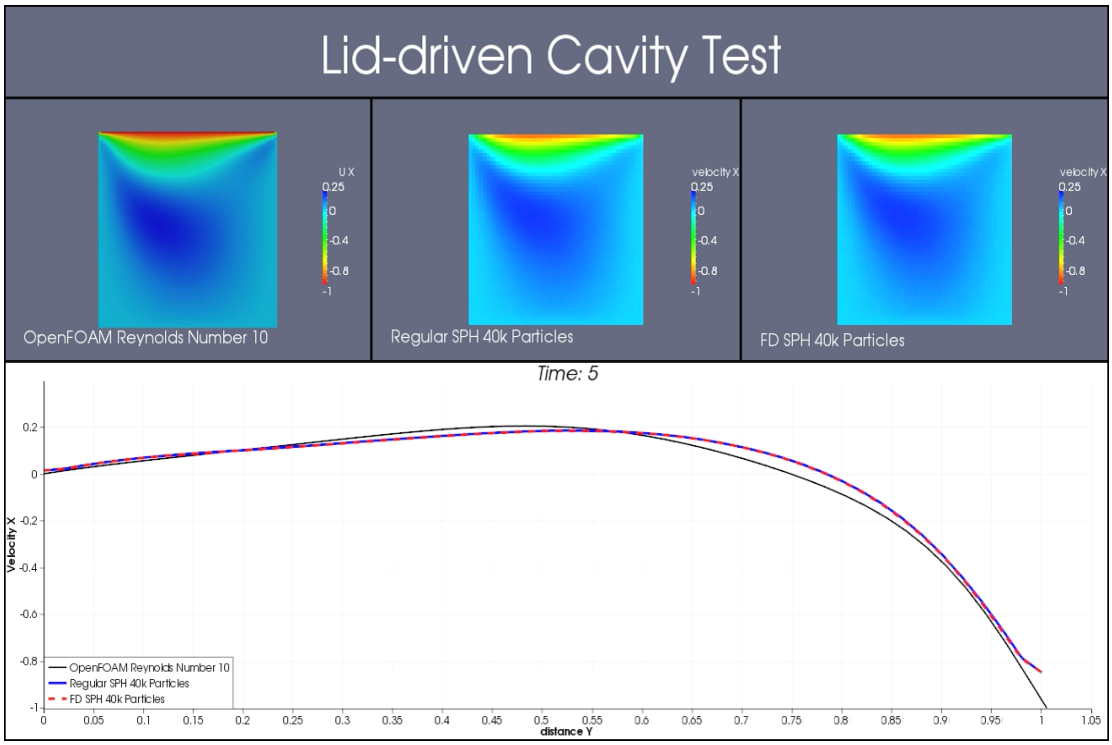

Figure 4.4: Lid-driven cavity test comparing OpenFOAM's grid-based Navier-Stokes solution (black curve), standard SPH (blue curve) and Fractional SPH (dashed red curve) with 40k particles. The velocities along the vertical line $x = 0.5$ passing by the center of the box are demonstrated at $t = 5s$ when the simulations are in steady state. The horizontal axis in the graph represents the vertical coordinates along the line $x = 0.5$. The similarity of the curves validate the viscosity behavior of the Fractional SPH simulation in a steady flow scenario.

lisions. We placed several force sources that continuously run and create flows moving towards opposite directions and colliding at several regions. Several evaluations were then performed by comparing low-resolution Fractional and standard SPH simulations against the corresponding higher resolution standard SPH simulation. As validated in the previous section, we consider the high-resolution standard SPH simulation to be the reference simulation for our comparisons. The low-resolution simulation that generated less error against the high-resolution simulation was considered to be physically more accurate. In these experiments we used explicit Euler numerical integration.

We discretized a 2D fluid grid by placing 48 equally distributed grid points. Our 2D grid scenario was borderless and cyclic, meaning that the particles penetrating a particular wall will come back to the scene from the opposite wall. Keeping all the initial conditions the same, we conducted three different simulations with this scenario. The first simulation was a high resolution standard SPH with 1500 particles. This one was considered to be the reference simulation. The second simulation was a low resolution standard SPH with 1250 particles and the third simulation was a low resolution Frac-

tional SPH with 1250 particles. As expected, our Fractional SPH model has showed to generate much less error rates than standard SPH at the regions with intense flow collisions. We measured the average velocity vectors of 48 different circular regions centered at the grid points. The average velocity of a circular region for a given timestep is calculated by averaging the angular difference of the velocities of all particles within that region at that timestep. The placement of source forces, measurement regions, and the shape of the grid is presented in Figure 4.5.

The angular difference of the average velocity vectors of each region was first computed for every timestep, then, for all the regions, we took the average of the angular errors by dividing the error sum to the total number of timesteps. The average error that each region produces with respect to the high resolution simulation was obtained with:

$$\epsilon_j = \sum_{i=1}^{n} \frac{1}{n} \arccos\left(\frac{\mathbf{v}_{ji_{SE}} \cdot \mathbf{v}_{ji_{SHR}}}{|\mathbf{v}_{ji_{SE}}||\mathbf{v}_{ji_{SHR}}|}\right), \tag{4.8}$$

where ϵ_j is the average error generated at region j, i is the timestep index, n is the total number of timesteps, and $\mathbf{v}_{ji_{SE}}$ and $\mathbf{v}_{ji_{SHR}}$ are the average velocities of region j at timestep i, for the evaluated simulation SE and the high resolution simulation SHR, respectively.

With this scheme, we are able to compare the number of grid regions that performed better with each SPH formulation. The method that resulted with more successful grid regions is considered to perform better overall. The next section presents and discusses in detail the results obtained. Additional results with 3D simulations are also presented.

4.6 Results and Discussion

We first analyze the results presented in Figure 4.5. The arrows show the placement and the direction of force sources and the circles indicate the 48 comparison regions. The color range is between red and green. The color red shows that standard SPH is producing less error and the green shows that Fractional SPH is producing less error. The yellow regions are the ones where both simulations create almost the same error rates. As demonstrated in this figure, the regions where flows collide tend to be green most of the time. This indicates that the fluid behavior that emerges from intense flow collisions is better modeled by Fractional SPH. The yellow regions are where few flow collisions occur and flows mostly tend to move to a fixed direction. As expected, no significant differences exist between the two simulations on these regions. The few orange regions where Fractional SPH is performing slightly worse than the standard SPH are always located near the force sources. These regions produce slightly worse error rates because they have always relatively less particles in them, due the fact that the continuous flows are generated from these regions. The lack of enough particles in these regions leads to a less accurate error measurement.

Figure 4.6 presents the comparison of angular error rates (in radians) of regional velocities, generated by both standard SPH and Fractional SPH, at a region rich in flow collisions, over the course of 500 timesteps. As can be observed, the error rate tends to be small at the beginning because of the same initial positions of the particles. However, the Fractional SPH produces less error than standard SPH as the simulation progresses.

Figure 4.7 shows the same results on a different region of the grid. This region is on the middle right of the grid and it is poor in flow collisions. As can be seen on the graph, the difference between the error rates generated between standard and Fractional SPH tend to decrease on this kind of regions. This result is expected in the sense that the difference of velocities of two particles moving towards the same direction with similar speed causes the viscosity to vanish.

Aside from comparing the direction of regional velocities, we also compared the evolution of velocity magnitudes on each region over the course of 500 timesteps. As it was the case for velocity directions, velocity magnitudes tend to be more similar to the high resolution simulation in Fractional SPH than standard SPH. Figure 4.8 shows the velocity magnitudes of a region rich in flow collisions, which is placed in the middle of the grid, for 3 simulations. The blue line is the high resolution reference simulation, the red line is the standard SPH and the green line is the Fractional SPH. As can be observed, Fractional SPH is better at approximating the high resolution simulation. Figure 4.9 shows that the difference between the effects of standard and Fractional SPH is minimized on regions poor in flow collisions.

We have also experimented with 3D SPH simulations following the same flow scenario but with the border of the fluid grid acting like walls. We used 12K particles for the high resolution reference simulation. We increased the gap between high and low resolution fluids by only using 3K particles for low resolution simulations. The same comparison scheme is employed to compare the simulations and calculate the error rates. The time of evaluation was extended to 1000 timesteps in order to have a wider comparison window.

Figure 4.10 and Figure 4.11 show the error produced by regional velocity directions and magnitudes, respectively. The comparisons were conducted at a region rich in flow collisions, over the course of 1000 timesteps. As can be observed in the figures, our method produces less error than the standard SPH in 90% of the timesteps.

In terms of performance, our method runs real-time for simulations with up to 5k particles and runs with 4 FPS for simulations with 20k particles on an AMD Athlon II X4 3.2 GHz computer. The use of half derivatives in the SPH implementation does not affect the complexity or the running time of the algorithm. In Equation 4.7, the weights are always calculated based on the terms q and $n - i$. The value of q must always be equal to 0.5 to acquire the half derivatives. Given that we only use the ten most recent terms of the history terms, $n - i$ terms always stay the same for all the ten weights, except for the first ten timesteps. Because Equation 4.7 makes use of the past particle velocities, we require some extra memory space to store the previous velocities.

Therefore, the weights can be precomputed and used in combination with pre-recorded velocities.

Fractional SPH also proved itself useful by allowing larger timesteps in the integration. It was observed that our Fractional SPH simulations were more stable than standard SPH when using large timesteps especially for viscous fluids. In Figure 4.12, standard SPH and Fractional SPH simulations are compared for different sizes of timesteps. Fractional SPH allowed 2 times larger timesteps, while standard SPH becomes unstable after a small increase. We also noticed that our method performed better in the early stages of the Shear Driven Cavity test, when the flows were not stabilized yet. This was observed in the 3D version of Shear Driven Cavity test and some results are presented in Figure 4.13. Additional examples demonstrating 3D simulations with our Fractional model are presented in Figure 4.14 and Figure 4.15.

In summary, we demonstrate that Fractional SPH always produces better results than the standard SPH in regions where flow collisions are detected. Both in 2D and 3D simulations, our Fractional SPH model produced less error on 73% of our grid regions, in comparison with the standard SPH. Note that this percentage depends directly on the density of colliding flows on the scene. A scenario with more colliding flows will definitely increase the success rate of Fractional SPH. The videos accompanying this submission provide additional examples and results.

4.6.1 Limitations

Although our model is able to accurately model colliding flows, it does not change the accuracy of the simulation on regions where the flows are stable and one-directional. When particles move towards similar directions with similar velocity magnitudes, the differential velocity gets close to zero and this lowers down the effect of viscosity terms. Since our contribution is on the viscosity term, the history effects have only limited effect on the overall behavior of particles. In these cases, there is no clear need for fractional viscosity terms. Therefore, one can take an hybrid approach to employ history terms only in regions where colliding flows are detected to occur. These terms can be omitted in regions of stable flows to further increase the performance of the simulation.

4.7 Conclusions

We have introduced a new methodology for fluid simulation based on the use of Fractional Calculus with Smoothed Particle Hydrodynamics. We have demonstrated in several experiments that our method can better simulate observed fluid behavior emerging from flow collisions. The fact that the memory-laden viscosity terms modeled by fractional derivatives are able to increase the accuracy of low resolution SPH simulations is promising as a technique to improve the quality and computational efficiency of SPH.

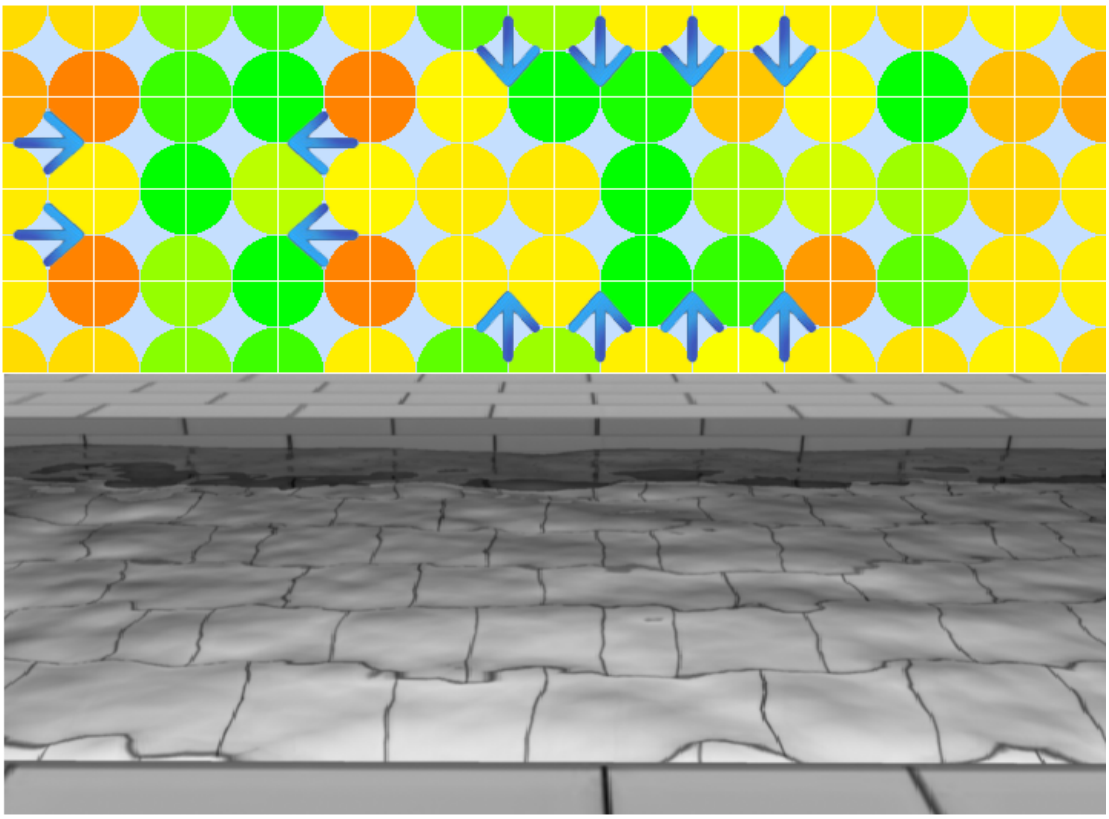

Figure 4.5: Simulation scheme (top image) and its corresponding fluid simulation scenario (bottom image). The simulation scheme shows both the flow generation configuration and the results of our experiments. The arrows show the force sources creating the flows. The circles show the regions where we computed the average velocity directions in order to quantitatively compare the simulations. The colors are in the red-green range and represent the difference of angular velocity error rates produced by the low resolution standard SPH and low resolution Fractional SPH when compared against the higher resolution standard SPH. Our Fractional SPH method performs better than the standard SPH in the green regions, the performance is similar in yellow regions, and our method performs slightly worse in the orange regions.

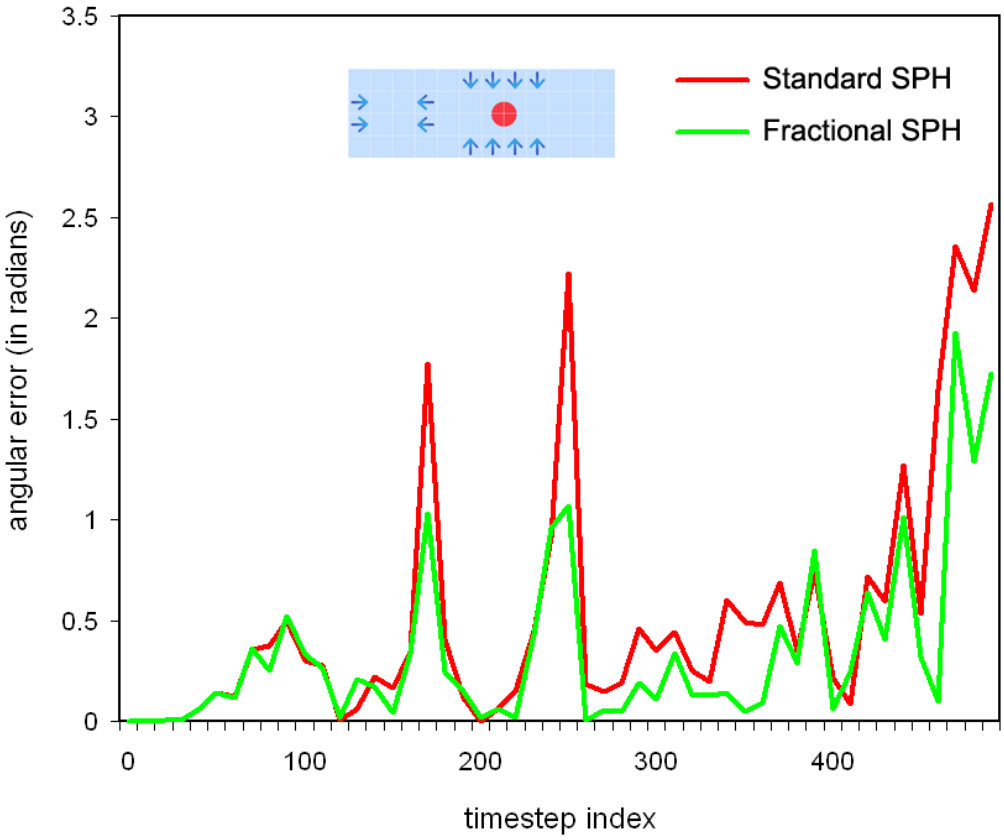

Figure 4.6: The plot shows the error rates in velocity directions of the region marked with the red disc, over 500 timesteps of simulation. The red and green lines represent the simulations of low resolution standard SPH and low resolution Fractional SPH compared against higher resolution standard SPH. The region marked with the red disc is rich in flow collisions.

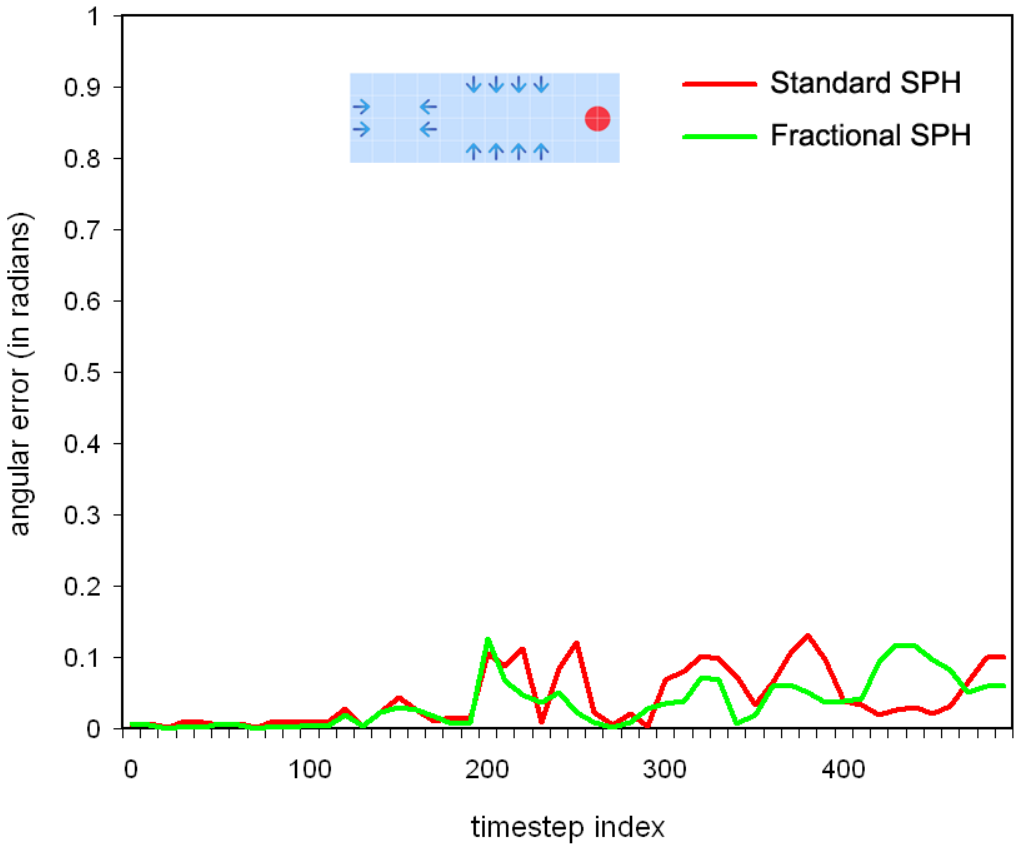

Figure 4.7: The plot shows the error rates in velocity directions of the region marked with the red disc, over a simulation of 500 timesteps, for 2D simulations. The red and green lines represent the simulations of low resolution standard SPH and low resolution Fractional SPH compared against higher resolution standard SPH, respectively. Here the region marked with red disc is poor in flow collisions.

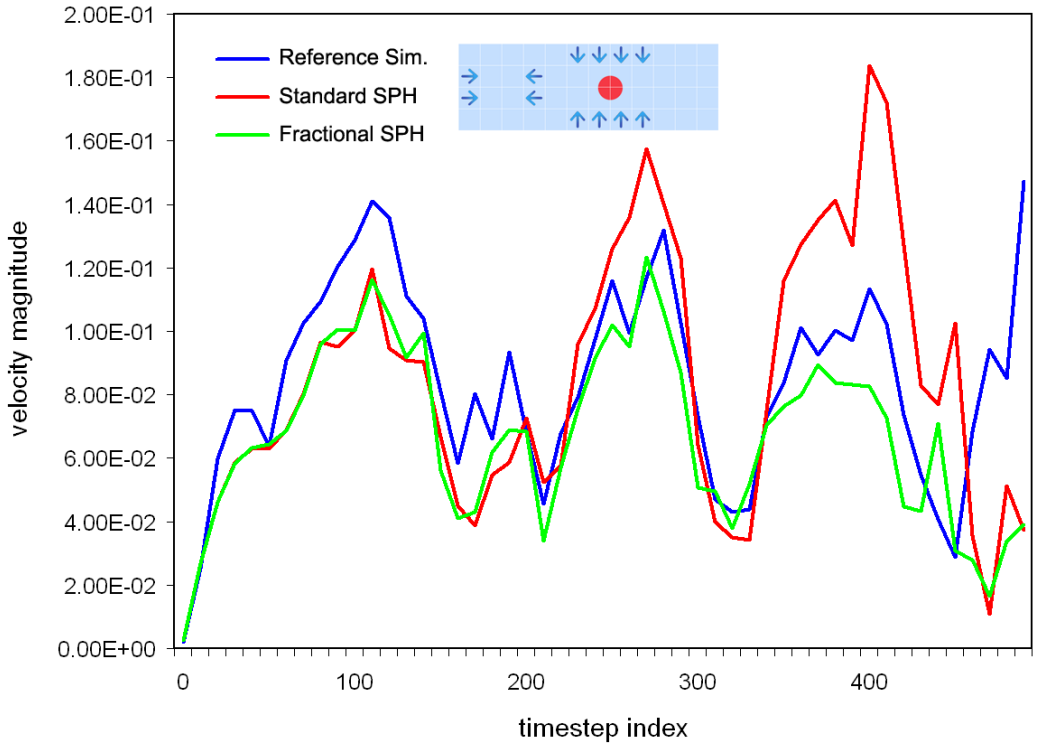

Figure 4.8: The plot shows the velocity magnitudes of the region marked with the red disc, over a simulation of 500 timesteps, for 2D simulations. The blue, red and green lines represent the simulations of high resolution standard SPH, low resolution standard SPH and low resolution Fractional SPH, respectively. The region marked with red disc is rich in flow collisions.

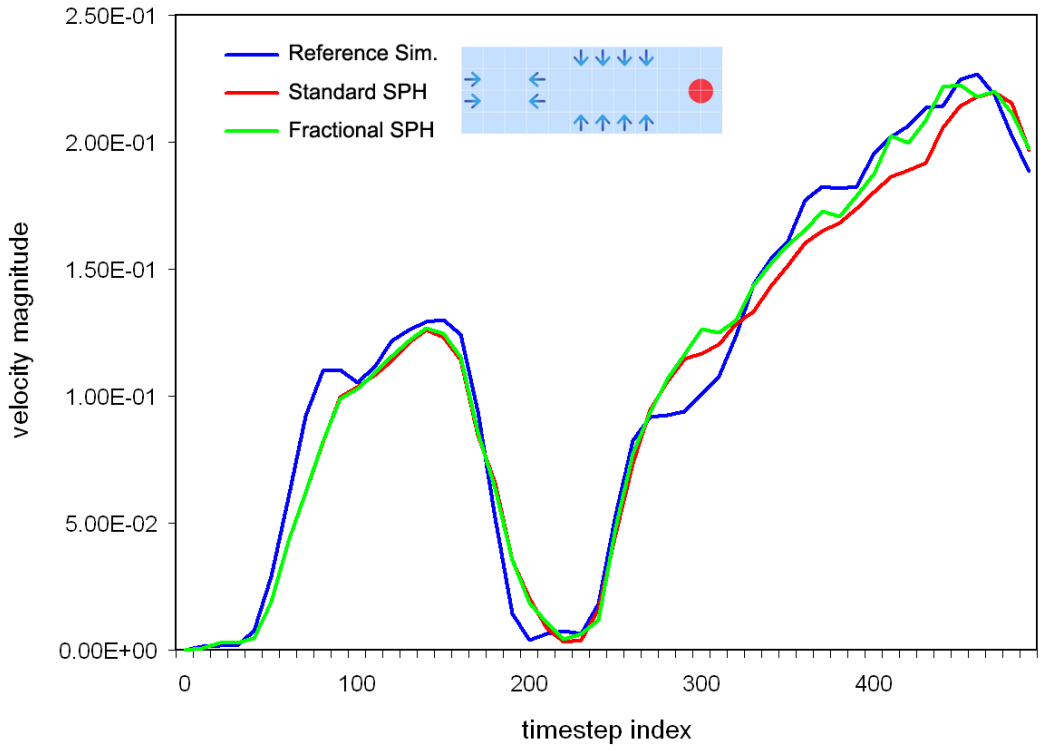

Figure 4.9: The plot shows the velocity magnitudes of the region marked with the red disc, over a simulation of 500 timesteps, for 2D simulations. The blue, red and green lines represent the simulations of high resolution standard SPH, low resolution standard SPH and low resolution Fractional SPH, respectively. The region marked with red disc is poor in flow collisions.

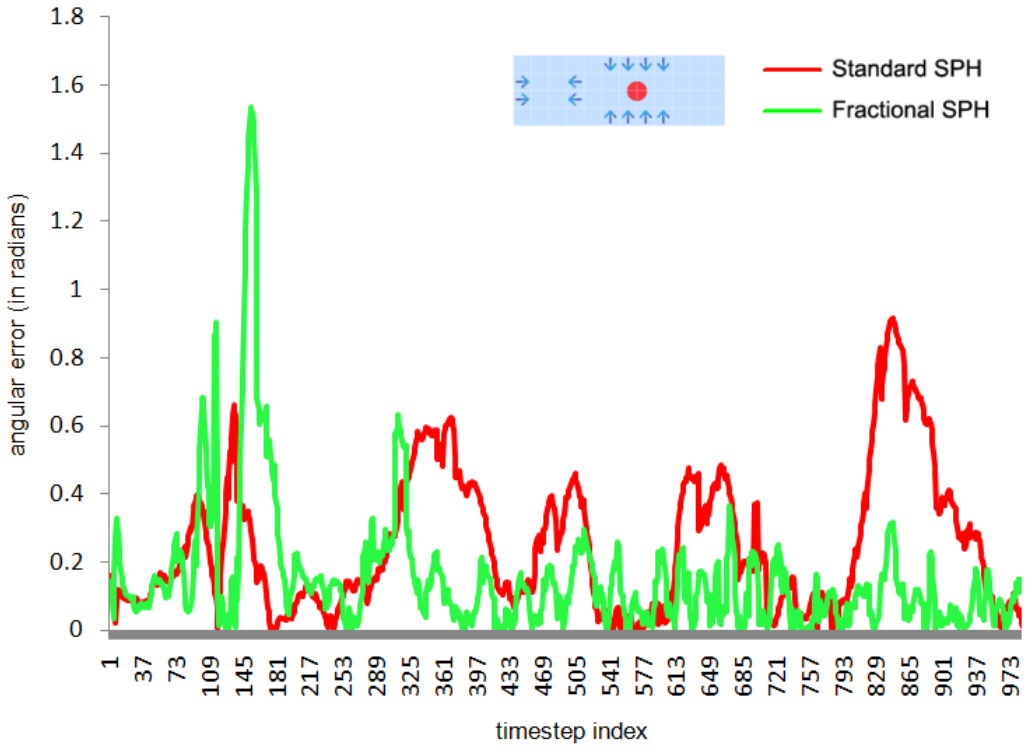

Figure 4.10: The plot shows the error rates in velocity directions of the region marked with the red disc, over a simulation of 1000 timesteps, in 3D simulations. The red and green lines represent the simulations of low resolution standard SPH and low resolution Fractional SPH compared against higher resolution standard SPH, respectively. The region marked with the red disc is rich in flow collisions.

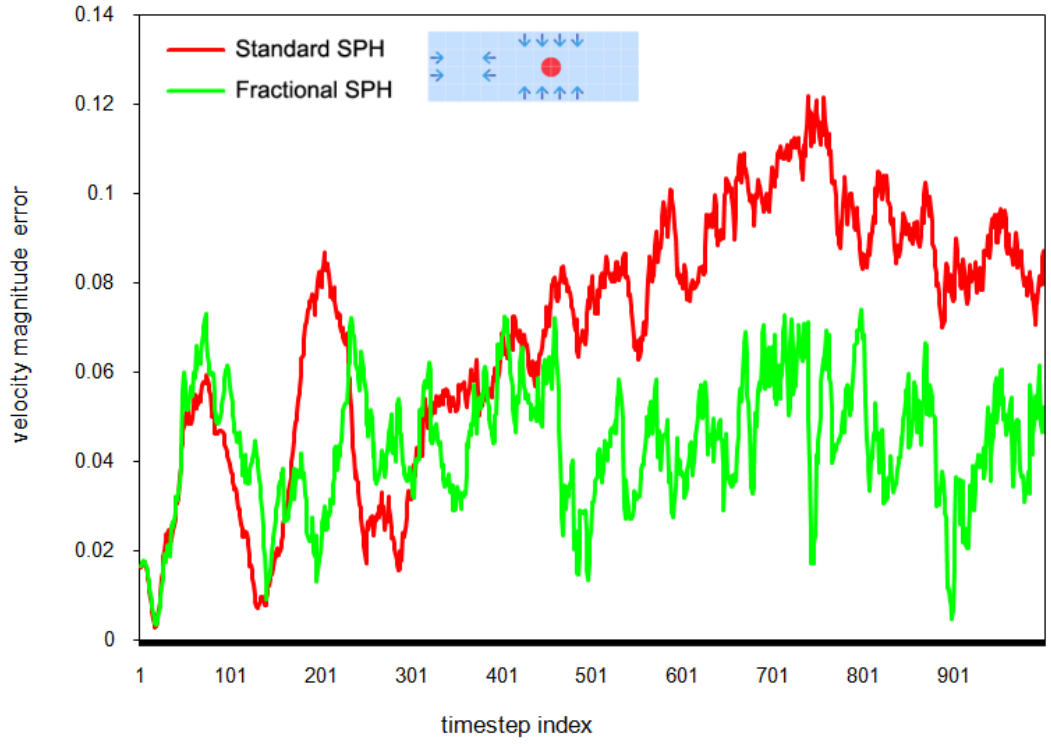

Figure 4.11: The plot shows the velocity magnitudes of the region marked with the red disc, over a simulation of 1000 timesteps, in 3D simulations. The red and green lines represent the simulations of low resolution standard SPH and low resolution Fractional SPH compared against higher resolution standard SPH, respectively. The region marked with red disc is rich in flow collisions.

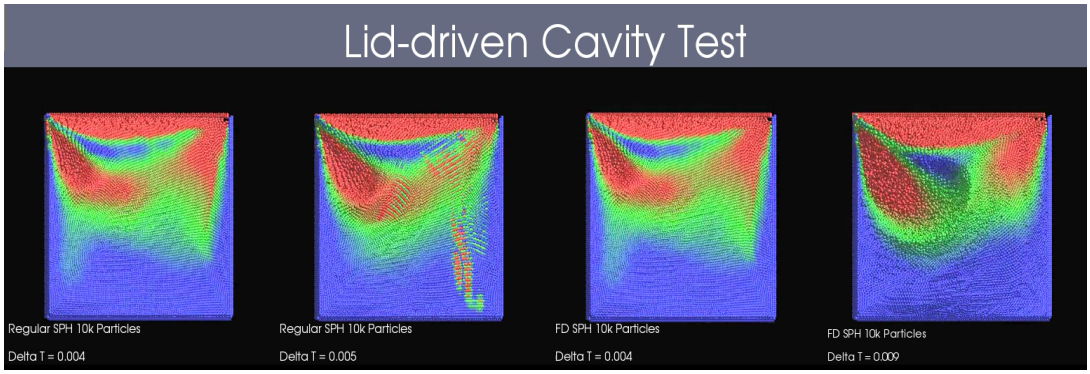

Figure 4.12: Lid-driven cavity test for comparing the stability of standard SPH and Fractional SPH simulations.

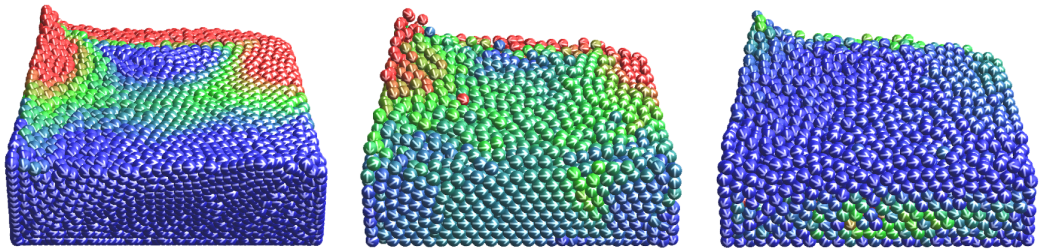

Figure 4.13: Average velocities and velocity directions obtained in the Shear Driven Cavity Test with different SPH simulations: 21K standard SPH (left), 6K Fractional SPH (center) and 6K standard SPH (right). Colors red, green and blue represent high, medium and low velocities respectively. The color distribution and regional velocity directions obtained with the Fractional SPH simulation are very similar to the ones in the high resolution reference simulation. This was not the case for the standard SPH in the same low resolution.

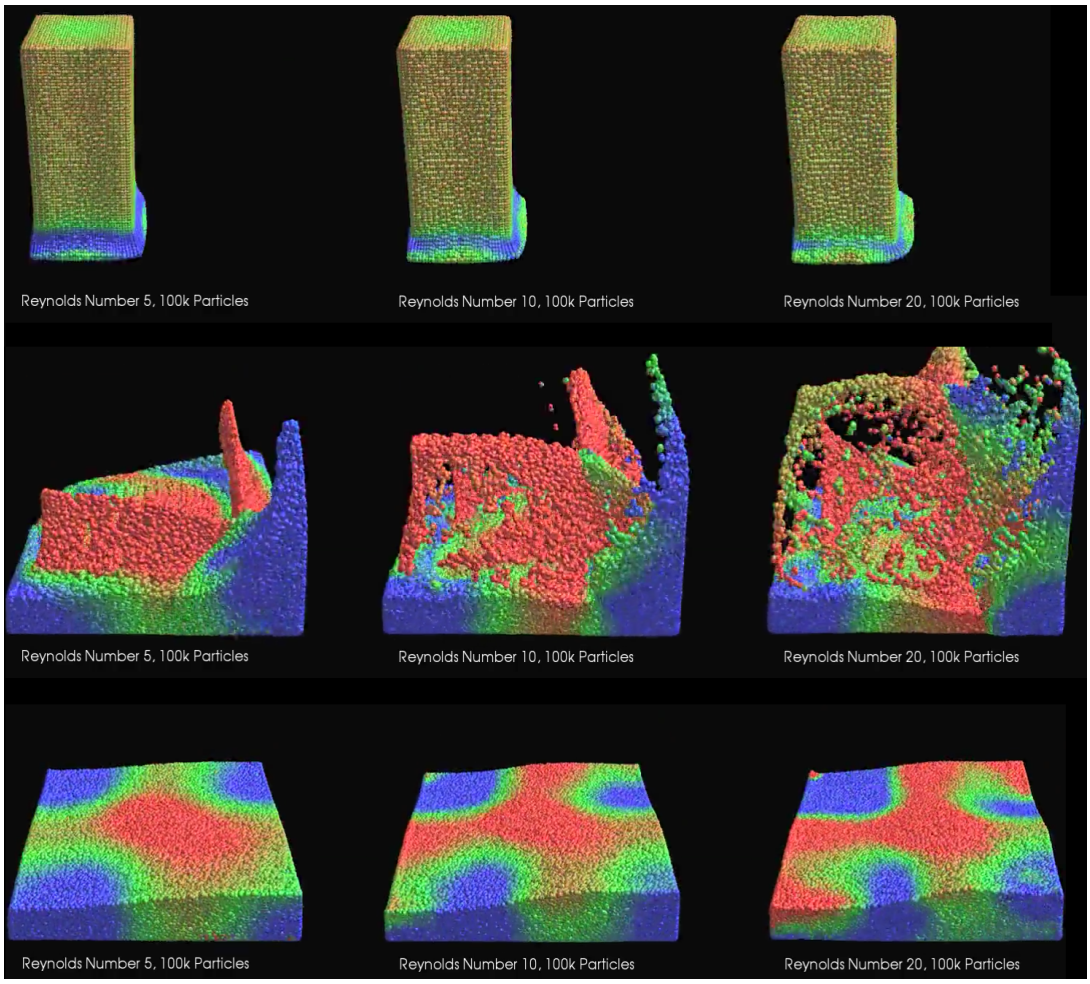

Figure 4.14: Dam break test using Fractional SPH, which shows the viscosity behavior of fluids with various Reynolds numbers.

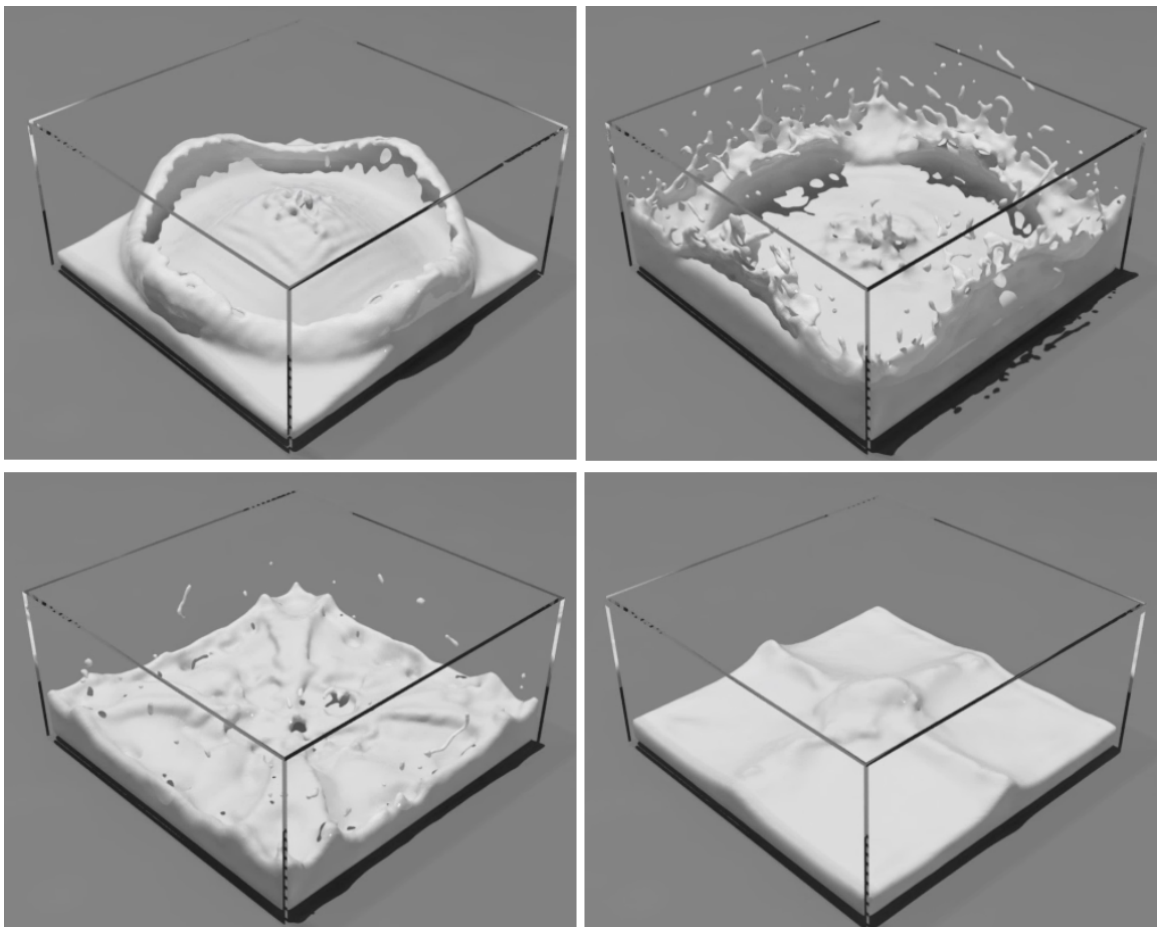

Figure 4.15: The rendered scene from another dam break experiment using Fractional SPH.

CHAPTER 5

Simulating Shear Thickening Fluids

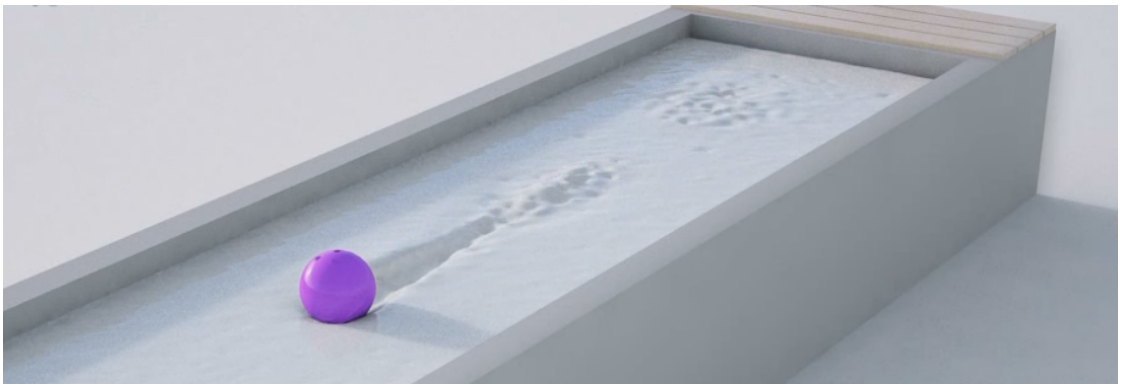

Figure 5.1: Our shear thickening fluid model based on fractional derivatives is able to achieve realistic behavior.

This chapter presents a computational model for simulating non-Newtonian shear thickening fluids. Shear thickening fluids have a unique and unconventional behavior that has not been simulated before and yet they often appear in real world scenarios such as when sinking in quicksand or when experimenting with popular corn starch and water mixtures. The fluid exhibits unique phase changes between solid and liquid states, great impact resistance in its solid state and strong hysteresis effects. Our proposed approach builds on existing non-Newtonian fluid models in computer graphics and introduces an efficient history-based stiffness term that is essential to enable shear thickening behavior. The history-based stiffness is formulated through the use of fractional derivatives, leveraging the fractional calculus ability to depict both the viscoelastic behavior and the history effects of memory-intensive systems. Simulations produced by our method are compared against real experiments and the results demonstrate that the proposed model successfully captures a wide range of behaviors observed in shear thickening fluids.

5.1 Introduction

In the field of rheology, the mechanics of fluids is typically characterized by a viscosity function, where the viscosity is defined as the ratio of shear stress to shear rate in a steady state flow. For a Newtonian fluid like water, this viscosity is a single-valued

constant, but in non-Newtonian fluids this viscosity may be a function of shear rate and other parameters. Dense suspensions of hard particles are known to be shear thickening, meaning that the viscosity function increases with shear rate over some range.

Shear thickening fluids (STFs) form a sub-category of non-Newtonian fluids exhibiting unconventional behavior that is not seen in other Newtonian or non-Newtonian fluids. These fluids harden when agitated by strong forces and soften in the absence of any forces. Moreover, they are memory-intensive fluids in the sense that they keep their solid-like state for a given amount of time even after the forces are applied, i.e., their memory only fades away after some time.

The most known example of shear thickening fluids is probably the corn starch and water mixture. The odd behavior of this fluid has recently attracted much public attention and videos showing experiments involving mixtures of corn starch and water are widely spread over the Internet. In these experiments, it is possible to witness the dramatic impact resistance of the fluid to the point that a person can run on top of the surface of the fluid [Tim09]. Another STF that has received some attention in several movies is quicksand. Applications of STFs have also been developed, for example, the United States Army Research Laboratory has invented a "Liquid Body Armor", an armor suit that has layers of shear thickening fluid and Kevlar mixture. The fabric made of STF is highly resistant to penetration when impacted by a spike, knife or bullet, without compromising its weight, comfort or flexibility [Han06] [Wil07].

Despite the fact that there is a number of mathematical models aiming to explain the behavior of shear thickening fluids, no animations of such fluids have been achieved before. We introduce in this chapter a first model that is able to produce visually successful simulations of such materials.

5.2 Related Work

To the best of our knowledge, there is no prior study on the simulation of shear thickening fluids in the Computer Graphics community. A number of works on the simulation of other types of non-Newtonian fluids have however been developed, such as for melting and the flow of deformable objects along with the animation of viscoelastic fluids. We review these works on the following paragraphs.

The most relevant papers to our work are the ones addressing melting and flow of materials. Terzopoulos and Fleischer [TPF89] developed a method that can simulate non-rigid objects that are capable of heat conduction, thermoelasticity and melting. The particles that constitute the material are connected by elastic springs and, in order to model the melting effect, the stiffness of the springs decreases as the temperature rises, and the particles break free when the springs disappear. Our proposed model is also based on the approach of using adaptive springs, which will in our case be augmented with fractional terms in order to achieve shear thickening fluids.

Desbrun and Gascuel [DG95] also focused on the flow of soft substances. They employed a hybrid model by combining an implicit surface model with a particle system. Successful results were presented on the animation of soft materials that can undergo fusion. Desbrun and Cani [DC96] demonstrated the first application of SPH in Computer Graphics for the animation of inelastic deformable bodies with a wide range of stiffness and viscosity. The first results included a 2D animation of a deformable solid and the flow of a viscoelastic material.

Carlson and Turk [CMI02] took on the challenge of animating materials that exhibit both fluid and solid-like behavior. The focus was on materials that can melt, flow and solidify. The fluid is modeled with variant viscosity. A modification of Marker-and-Cell method is used to deal with the computational barrier of simulating highly viscous fluids. Therefore, the solidity of the materials was in fact achieved by simulating a highly viscous fluid. No elastic elements were added to the fluid equations.

Goktekin et al. [GBO04b] presented a method that animates viscoelastic fluids such as mucus, liquid soap, toothpaste or clay. The method introduced elements of elasticity into an Eulerian implementation of Navier-Stokes equations. The total strain is defined as the sum of elastic and plastic strain. Additionally, Von Mises' yield condition is used to determine when the plastic flow should occur. When the magnitude of the strain deviation crosses some threshold, the plastic flow occurs. This condition guarantees a smooth transition from elasticity to viscosity. The authors presented a variety of successful results on the behavior of viscoelastic fluids.

Clavet et al. [CBP05] developed a powerful and simple method that can efficiently and realistically simulate viscoelastic fluids. The method combines a simplified SPH with a mass-spring system. The SPH is responsible for the fluid behavior while the springs are responsible for the elasticity and plasticity effects that make the material more solid-like. The method computes an additional pressure term named near-pressure, along with the conventional pressure term calculated through SPH. This term is used to solve the clustering problem which is a common problem in SPH implementations. Furthermore, the nearby particles are also connected to each other by springs. These springs have varying rest lengths and are removed when the distance between a pair of particles is greater than a threshold. The use of adaptive springs along with the double density relaxation method results in a very strong animation tool for viscoelastic models. This method constitutes the starting point for our shear thickening fluid animation model.

5.3 The Model

The starting point of our shear thickening simulation model is based on the model presented in [CBP05]. This is a Lagrangian model that combines SPH forces with a dynamic mass-spring system. The fluid is modeled as a particle system. Particles have attributes such as position, velocity, force, density and pressure. In each simulation step, first the forces are calculated; then, a new position and velocity is calculated for

each particle using a prediction-relaxation approach.

The forces are calculated among pairs of particles. Since considering all pair-wise connections would result in $O(n^2)$ running time for the algorithm, only the interactions of close-by particles are taken into account. Therefore, the fluid container is discretized into cubic regions of dimension h. Forces are only calculated among pairs of particles that are in the same or adjacent grid regions.

The simulated material shows both liquid-like and solid-like behavior depending on the forces applied. The density and viscosity forces are responsible for making the material act like a fluid. The material should flow from high density regions to low density regions and should be incompressible. All these features are acquired though the density and viscosity calculations that arise from classical SPH formulations. One clear difference that this SPH implementation has from the previous SPH implementation presented in Chapter 4, is that this SPH implementation should generate higher viscosity forces to create a non-Newtonian fluid. The fluid is supposed to act similar to materials like honey, shampoo or starch. This directly implies that the kinematic viscosity constant should be greater than the previous SPH model presented. However, even increasing the viscosity constant will not be enough to make the fluid behave like a non-Newtonian fluid. Therefore, to account for the elasticity and plasticity features of non-Newtonian fluids, elastic springs are added into the formulations. While these springs generate elasticity forces like a regular Hooke's spring, they might be removed, added or their rest length might be updated depending on given conditions during the simulation. This adaptive spring model will in fact create the plasticity of the material.

The existence of viscosity, elasticity and plasticity terms will lead to the simulation of so-called viscoelastic material or a non-Newtonian fluid. However, these components won't suffice for creating a shear thickening fluid, which is a subcategory of non-Newtonian fluids. In order to upgrade the non-Newtonian fluid to the shear thickening fluid, history-based spring elements will be added to the system to acquire a system that solidifies when subjected to impact forces. The system should keep its solid state for a given amount of time even when there are no more forces acting on it. At the end of this period of time, the material should resume its viscous liquid state. As a conclusion, the model should be able to transition between solid and liquid states.

The density forces are calculated in accordance with classical SPH formulations. The density of a particle is computed based on the distance to other particles in its vicinity. The pair-wise distances are smoothed through a smoothing kernel function. The function generates weights for all the surrounding particles and the distances are multiplied by these weights to compute final densities. The density for a particle i is given in terms of the particle's distance to its neighbor particles j:

$$\rho_i = \sum_j (1 - r_{ij}/h)^2, \tag{5.1}$$

where r_{ij} is the distance between two particles i, j and h is the SPH kernel radius. The

densities are then used to calculate the pressures p at particle locations:

$$p_i = k(\rho_i - \rho_0), \tag{5.2}$$

where ρ_i is the density of the particle i, ρ_0 is the rest density of the fluid and k is the gas constant. The pressure is computed based on the difference of the current density and the rest density of the fluid. The pressure will then be used to calculate the pressure force acting on each particle. The displacement created by the pressure force is given as:

$$\boldsymbol{D}_{ij} = \Delta t^2 p_i (1 - r_{ij}/h)\hat{\boldsymbol{r}}_{ij}, \tag{5.3}$$

where Δt is the timestep, p_i is particle's pressure, r_{ij} is the distance between two particles, h is the radius of the kernel and $\hat{\boldsymbol{r}}_{ij}$ is the normalized difference vector between two particles. In addition to the classical density and pressure calculation, [CBP05] adds a secondary density term for a particle solely for the close vicinity of that particle. Unlike the first kernel function which is quadratic, the near-pressure kernel function is cubic. The density term is called the near-density and it is computed as:

$$\rho_i^{near} = \sum_j (1 - r_{ij}/h)^3. \tag{5.4}$$

The pressure that is computed based on the near-density is called the near-pressure and reads:

$$p_i^{near} = k^{near} \rho_i^{near}, \tag{5.5}$$

where k^{near} is a constant and ρ_i^{near} is the near-density. The near-pressure is calculated independent of the rest density. The displacement caused by the near-pressure is defined as:

$$\boldsymbol{D}_{ij} = \Delta t^2 p_i^{near} (1 - r_{ij}/h)^2 \hat{\boldsymbol{r}}_{ij}. \tag{5.6}$$

The near pressure term solves an important problem. In SPH simulations, sometimes the particles can come to a rest by pulling a great number of other particles. This problem is known as the particle clustering problem. It is shown that the near-pressure term solves the particle clustering problem to a great extent. Additionally, near-pressure adds more non-Newtonian fluid features to the material by generating a suited surface tension.

The viscosity force between two particles is calculated based on the difference between their velocities, in accordance with the Navier-Stokes equations and the SPH method

presented in Chapter 4. The difference in velocity has both linear and quadratic effects. The algorithm for the viscosity calculations are given in Algorithm 1.

Algorithm 1 ApplyViscosity

1. $q \leftarrow r_{ij}/h$
2. //If particle j is in the kernel radius
3. **if** q $<$1 **then**
4. $\quad u \leftarrow (\boldsymbol{v}_i - \boldsymbol{v}_j) \cdot \hat{\boldsymbol{r}}_{ij}$
5. $\quad$ **if** u $>$0 **then**
6. $\quad\quad$ //Linear and quadratic impulses
7. $\quad\quad \boldsymbol{I} \leftarrow \Delta t(1-q)(\sigma u + \beta u^2)\hat{\boldsymbol{r}}_{ij}$
8. $\quad\quad \boldsymbol{v}_i \leftarrow \boldsymbol{v}_i - \boldsymbol{I}/2$
9. $\quad\quad \boldsymbol{v}_j \leftarrow \boldsymbol{v}_j + \boldsymbol{I}/2$
10. $\quad$ **end if**
11. **end if**

The elasticity component originates from the spring-mass system. Particle pairs that are close up to a given distance are connected to each other by elastic springs and the spring force is given by:

$$F = -k_{min}\, \boldsymbol{x}, \tag{5.7}$$

where k_{min} is the spring constant. The springs only exist between pairs of particles that are closer to each other than a given distance. If the particles move further away from each other, then the spring is removed. The rest lengths of the springs are also updated in every simulation step. The addition and removal of springs along with the rest length updates create the effect of plasticity. The material can be stretched without causing great forces that want to recover its shape. If the material is elongated more than a given amount, it never tries to go back to its previous shape because the springs are now removed. The plasticity model obeys the von Mise's condition which states that the plastic flow should occur only if the deformation is large enough [Fun65]. In the context of the spring system, this means that the rest length of a spring should only change if its elongation is greater than some threshold. Therefore, the rest length update is calculated as:

$$\Delta L = \Delta t\, \alpha\, sign(r - L)\, max(0, |r - L| - \gamma L), \tag{5.8}$$

where α is the plasticity constant, L is the spring rest length and γ is the yield ratio.

Shear thickening fluids have two very important additional characteristics when compared to viscoelastic fluids. First, they change their state from liquid-like to solid-like when subjected to great forces. As long as forces are present, the solid-like state is preserved. In the absence of forces, the material will soften and return to its liquid-like

structure. Second, shear thickening fluids are memory-intensive materials. The present state of memory materials is defined in terms of past states, and the materials remember the past forces that affected them up to some extent. When a shear thickening fluid is in its solid-like state and all forces are ceased, it will take some time to go back to its liquid-like state. Likewise, when the fluid is in its liquid state and is agitated by mild forces, it will slowly move from liquid to solid state over time.

The most critical factor that will transform a viscoelastic fluid into a shear thickening fluid is the inclusion of history-based spring elements. These elements have a history-based stiffness constant that injects the information of the past into the system. The history-based stiffness terms are achieved by multiplying the stiffness constant of the dynamic springs by the magnitude of the fractional derivative of the position of the connected particle. The history-based spring element is given by:

$$F = -k_{hist} \left\| D^q \boldsymbol{x} \right\| \boldsymbol{x}, \tag{5.9}$$

where $k_{hist} \left\| D^q \boldsymbol{x} \right\|$ is the history-based stiffness, $D^q \boldsymbol{x}$ is the q^{th} order derivative of the position and q is a non-integer between 0 and 1. The approach used to achieve the history effects relies on the properties of fractional calculus. As mentioned in Section 2.1, fractional calculus is widely used for the description of viscoelastic phenomena and it provides a neat tool to simulate history effects. A second order numerical solution of the fractional derivative of order q for values $0 < q < 1$ is given in [SCK05] and it reads:

$$D^q \mathbf{x}_n = \frac{\Delta t^{1-q}}{\Gamma(3-q)} \sum_{p=0}^{n} a_{p,n} D^1 \mathbf{x}_p, \tag{5.10}$$

$$a_{p,n} = \begin{cases} (n-1)^{2-q} - n^{1-q}(n+q-2) & \text{if } p = 0, \\ (n-p-1)^{2-q} - 2(n-p)^{2-q} + (n-p+1)^{2-q} & \text{if } 0 < p < n, \\ 1 & \text{if } p = n, \end{cases}$$

where q is the derivative order such that $0 < q < 1$, n is the index of the most recently computed timestep, $a_{p,n}$ is the weight of the past timestep p computed at the current timestep n, Γ is the Gamma function, and $D^1 \mathbf{x}_p = \mathbf{v}_p$ is the velocity of the particle at past timestep p.

The fractional derivative of the position at the current time step is thus computed based on the velocities of the past, and the past velocities are prioritized by given weights. The near past has more influence on the current state than the far past. The evolution of the weights generated by the fractional derivative operator is shown in Figure 5.2 for q values of 0.2, 0.5 and 0.8.

We compute the total spring force as the sum of forces generated by the history-based spring element and the regular elastic spring:

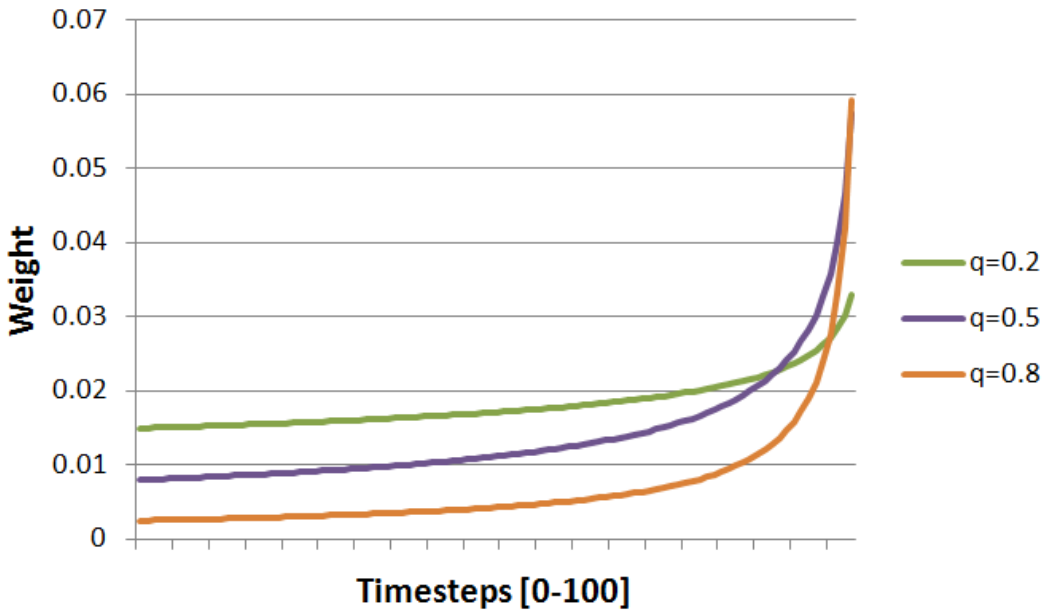

Figure 5.2: Weights produced by fractional derivatives of order 0.2, 0.5 and 0.8.

$$F = -k_{min}\,\boldsymbol{x} - k_{hist}\,\|D^q\boldsymbol{x}\|\,\boldsymbol{x}. \tag{5.11}$$

Substituting Equation 5.10 into Equation 5.11 and rearranging, we obtain:

$$F = -1\,(k_{hist}\,\frac{\Delta t^{1-q}}{\Gamma(3-q)}\,\sum_{p=0}^{n} a_{p,n}|\boldsymbol{v}_p| + k_{min})\,\boldsymbol{x}, \tag{5.12}$$

where k_{hist} is the control parameter for the history-based spring, $a_{p,n}$ is the weight of the past timestep p computed at the current timestep n, $\boldsymbol{v}_p$ is the velocity of the particle at the past timestep p, k_{min} is the stiffness constant of the elastic spring, and $\boldsymbol{x}$ is the vector of elongation or compression. Note that this formulation successfully models both of the desired features of shear thickening fluids. First, the spring immediately becomes very stiff when the velocity $\boldsymbol{v}_p$ dramatically increases in the near past, what results in the mass-spring system exhibiting solid-like behavior in regions where great impact forces are applied. Second, the spring will gain memory thanks to the weights $a_{p,n}$ that apply to the velocities of the past. This results in the overall stiffness of the springs gradually rising under continuous application of forces, and gradually decreasing under the absence of forces. The overall behavior achieved by the spring-mass system will lead to the shear thickening fluid slowly changing its phase both from liquid to solid and from solid to liquid.

Although k_{hist} and k_{min} are both denoted with the letter k in accordance with the Hooke's law, they serve different purposes. Constant k_{min} is the stiffness of the elastic

spring and it specifies the minimum stiffness that the spring can have. The history-based stiffness originating from the fractional derivative is an additional term on top of the default stiffness. In the absence of great velocities, notice that the first term in the parenthesis of Equation 5.12 vanishes and the system behaves exactly like a regular viscoelastic fluid as described in [CBP05]. However, when the particle gains instantaneous or cumulative high velocity over time, the first term adds extra stiffness to the spring. Constant k_{hist} is the parameter that controls the amount of extra stiffness that will be created due to the velocity increase. Constant k_{hist} can be used to give to the animator the ability to implicitly control the maximum stiffness that the system can gain. In cases where the integration method cannot keep the system numerically stable because of excess of stiffness, k_{hist} can be fine-tuned to keep the simulation stable. In our experiments, we have set k_{hist} so that the added stiffness can increase up to approximately ten times k_{min}.

One important numerical drawback of the fractional derivative formulation is that it is a global derivative over the whole history of the particle. Clearly, this would pose performance problems as the simulation moves forward since the cost of calculating each simulation step would be always greater than the cost of the previous step. Therefore we limit the amount of history to be considered by changing the start index of the sum in Equation 5.12 from $p = 0$ to $p = n - d$, where d is the number of time steps in the past that we would like to take into account. Parameter d is also a useful parameter to model the characteristics of the material. If the material being modeled has long-term memory, it makes sense to use a large number of past states. On the other hand, if the material has short-term memory, a small value for d should be chosen.

We have investigated the behavior of a single particle under the influence of our history-based spring element in order to analyze the evolution of the history-based stiffness with respect to the particle's velocity magnitude. Figures 5.5 and 5.6 present several history-based stiffness values generated by various memory settings (various d values) and for history-based spring terms with various q values. Figure 5.5 demonstrates how the history-based stiffness changes when the particle gains a temporarily high velocity due to an impact. Figure 5.6 shows the evolution of the history-based stiffness when the particle moves with a steady velocity for a longer period time. It is possible to observe that the history-based stiffness increases more and lasts for a longer period when more states in the past are considered. Furthermore, it can be seen in the graph that limiting the number of past timesteps naturally creates a maximum limit for history-based stiffness, leading to obvious advantages in terms of controlling the numerical stability. In real experiments with shear thickening fluids composed of corn starch and water we have observed that the material has the approximate memory of one second. Therefore we have chosen $d = 1/\Delta t$ in our simulations.

Another important modeling decision is the choice of the fractional derivative order. Equation 5.10 is defined for values of q such that $0 < q < 1$. The evolution of the weights generated by the fractional derivative for different values of q is presented in Figure 5.2. It can be observed that as q gets closer to 1, the history effects tend to

disappear. As the value of q gets close to 0, the effect of the past time steps tend to become similar. Both extreme cases do not seem to be useful for our problem. From previous work on fractional derivatives on memory systems we know that the half-derivative ($q = 0.5$) is the best choice for describing a number of phenomena in rheology [CR98, Coi03, CLL04, LCT05, SCK05]. We have therefore used $q = 0.5$ as the fractional derivative order in our experiments.

Different values of q can be always specified by animators in order to experiment with different history effects. For the sake of examining the different behaviors that can be achieved, an analysis of the history effects generated with $q = 0.2$, $q = 0.5$, and $q = 0.8$ is also presented in Figures 5.5 and 5.6.

Considering all discussed parameters and adapting Equation 5.12 to the case of two particles connected by a spring, the total force between two given connected particles is:

$$\begin{aligned} F = -1\,(k_{hist}\,\frac{\Delta t^{0.5}}{\Gamma(2.5)}\,\sum_{p=n-d}^{n} a_{p,n}\,|\boldsymbol{v}_{pi} - \boldsymbol{v}_{pj}| + k_{min}) \\ \frac{\boldsymbol{x}_i - \boldsymbol{x}_j}{|\boldsymbol{x}_i - \boldsymbol{x}_j|}\,(|\boldsymbol{x}_i - \boldsymbol{x}_j| - L), \end{aligned} \tag{5.13}$$

where $\boldsymbol{v}_{pi}$ and $\boldsymbol{v}_{pi}$ are the velocities of the particles i and j at the past timestep p and L is the rest length of the spring. As part of the employed prediction-relaxation integration scheme [CBP05], the displacement generated by the spring force is computed by multiplying the spring force by Δt^2, and then the displacement is applied to the positions of the particles.

Experiments with real shear thickening fluids show that shear thickening can support objects with extreme masses. A bowling ball can jump and roll, and a person can literally walk on a shear thickening fluid while the fluid is in its solid-like state. Experiments in physics and rheology stress the importance of solid container walls in this particular behavior [BJ12]. The force applied to the surface is eventually dissipated to the boundaries of the container and solid walls help the fluid to support even greater masses.

Our network of history-based springs can simulate this behavior up to some extent. To improve the overall results an additional technique is employed for making sure that the particles at the boundary of the fluid are highly sticky to the boundary walls. For a particle at the boundary, its velocity is separated into its normal and tangential components. We completely cancel the normal velocity component that is perpendicular to the wall. We also lower the tangential velocity component in a proportional way to the average of the history-based stiffness of the springs that the particle is connected to. In this way when the fluid exhibits solid-like behavior, the particles at the boundaries

are almost fixed. As the stiffness of the whole system decreases, the particles at the boundaries become less sticky.

The algorithm for the calculation of a simulation step is shown in Algorithm 2. In this algorithm, time integration is done through the prediction-relaxation approach. In the prediction-relaxation approach, first the forces originating from gravity and viscosity are calculated and the velocity is updated based on these forces. The particle is then virtually moved to its predicted position based on the current velocity. At this predicted position, the density forces of SPH and the spring forces that account for the elasticity are calculated and the displacement due to these forces is immediately applied to the particle. Lastly, the collisions between the particles and the walls of the container are resolved and the positions of particles are corrected. The velocity of the particle is implicitly calculated at the end of the timestep.

Algorithm 2 Simulation Step

1. //Apply gravity
2. **for all** particle i **do**
3. $\quad \boldsymbol{v}_i \leftarrow \boldsymbol{v}_i + \Delta t \boldsymbol{g}$
4. **end for**
5. //Update velocities with viscosity forces
6. ApplyViscosity();
7. **for all** particle i **do**
8. $\quad$ //Save previous position
9. $\quad \boldsymbol{x}_i^{prev} \leftarrow \boldsymbol{x}_i$
10. $\quad$ //Advance to predicted position
11. $\quad \boldsymbol{x}_i \leftarrow \boldsymbol{x}_i + \Delta t \boldsymbol{v}_i$
12. **end for**
13. //Add, remove springs or change their rest lengths if necessary
14. AdjustSprings();
15. //Apply dispacements due to history-based spring elements
16. ApplySpringDisplacements();
17. //Apply dispacements due to pressure forces
18. DoubleDensityRelaxation();
19. //Use previous position to compute next velocity
20. **for all** particle i **do**
21. $\quad \boldsymbol{v}_i \leftarrow (\boldsymbol{x}_i - \boldsymbol{x}_i^{prev})/\Delta t$
22. **end for**

5.4 Results and Discussion

There is an abundance of real experiments involving shear thickening fluids, namely with corn starch and water mixtures, that show us how the material behaves in different

Figure 5.3: The snapshots from the vibration experiment show finger-like formations with various shapes.

conditions. We evaluate our model by visually comparing our simulations with these real experiments.

In our first experiment we have simulated a scene where a bowling ball is rolled on the surface of the shear thickening fluid. The real version of this experiment is important in many aspects. First, it shows that the fluid can support a high mass object such as a bowling ball and that the solid-like phase of the material can make the bowling ball slightly bounce on the surface. Second, the bowling ball can even roll on the fluid surface as if rolling on a solid surface. Third, even when the motion of the bowling ball ends, it still stays on the surface of the fluid for a given time, before it slowly starts to sink. This scene clearly shows the history effects in action in the mechanics of the fluid, and our model was able to successfully replicate the observed behavior in simulation. The fluid immediately solidified under the high impact of the ball and acted somewhat like a spring mattress. It kept its solid-state for a while, letting the bowling ball to roll on the surface, and when the ball stopped the fluid gradually softened and entered into its liquid state allowing the ball to sink. These phases are illustrated by the snapshots presented in Figure 5.7.

In our second experiment we have simulated the shear thickening fluid subjected to vibration forces. In the real experiments the fluid is placed on a speaker creating vibrations that are transmitted to the fluid. After some time the fluid slowly starts changing its phase from liquid-like to solid-like. First the fluid forms holes and starts clustering at some random regions. Then the clustered regions slowly grow and form finger-like formations. The finger-like formations gradually get higher until they either merge with other parts of the fluid or break off from the main fluid body. We have performed a similar experiment in simulation with a fluid with 10K particles. We have placed equally distributed vibration sources at the bottom of the container and we have added random changes in the vibration frequency throughout the simulation. We have simulated both a fluid with history-based springs and a fluid with no history terms. The results show that our fluid model with history effects is able to successfully simulate the finger-like formations as shown in Figure 5.3. The vibration forces cause an accumulated history-based stiffness on the springs and the particles naturally start to climb on each other. On the other hand, the fluid without history terms could not achieve any of the finger-like formations. The results of this experiment are compared against the real experiment in

Figure 5.4: Comparison between the real [You08] and the animated shear thickening fluid in the context of the vibration experiment.

Figure 5.4.

Our vibration experiment shows that finger-like formations require history terms, in agreement with the model of [Dee10]. The obtained results are the first to confirm that prediction in a dynamic simulation. The fact that several of the phenomena associated with shear thickening have been simulated for the first time with a model with strong history effects suggests that hysteresis may be more generally important to shear thickening than previously indicated. These effects have been ignored in the standard models for shear thickening such as hydroclustering [BB85] or dilatancy [BJ12] and this insight may lead to improvements of those models so that they can finally predict some of the interesting phenomena associated with shear thickening.

In terms of performance, the running time obtained with our system is approximately 1.5 seconds per frame for a simulation of 20K particles in an Intel Core(TM) i7-2600K 3.4 GHz computer. The added computation time by the inclusion of history-based spring forces is minimal and these terms do not affect the complexity of the algorithm. The weights of the fractional derivative term can be precomputed and used in combination with the stored past velocities. Equation 5.13 requires the algorithm to store the past velocities of the particles only up to a given number of timesteps, leading to some extra memory space required.

5.4.1 Limitations

As for limitations, although our model is able to imitate shear thickening fluids to a great extent, we have observed that in the vibration experiment the finger-like formations did not grow as tall as in the real experiment. This prevented the replication of finger-like formations falling and merging with each other. Additional experimentation with parameters and particle resolutions is required in order to determine the limits of the model more accurately.

5.5 Conclusions

We have introduced a new model for simulating shear thickening fluids. Our model has successfully reproduced in simulation a wide range of behaviors that can be observed in real experiments, and has demonstrated the importance of fractional calculus in capturing the memory effects exhibited by shear-thickening materials. Our results show that complex relationships between hysteresis, viscoelasticity and flows can be successfully reproduced, and the proposed techniques show great potential to be extended for addressing other types of mixtures.

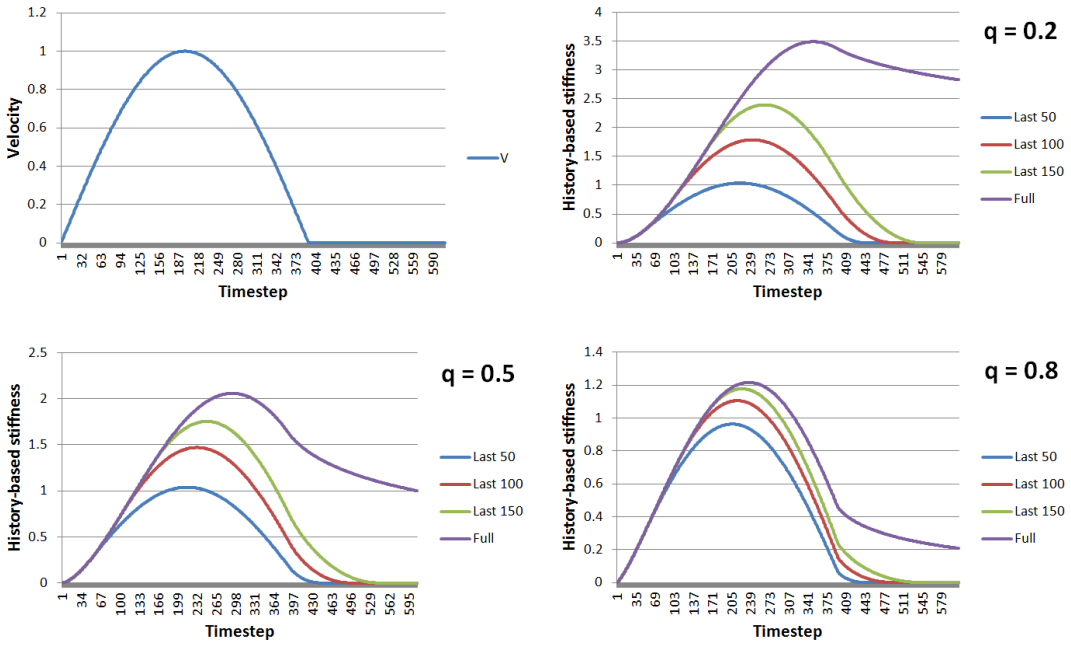

Figure 5.5: The upper left graph shows the velocity of a single particle subjected to a temporary velocity increase due to an impact. The next three graphs (in left-right, top-down order) show the evolution of the history-based stiffness values of the spring connected to the same particle, when its velocity changes according to the upper left graph. Curves for various d values are given in each graph. The second, third and forth graphs are based on the order of the fractional derivative $q = 0.2$, $q = 0.5$ and $q = 0.8$ respectively.

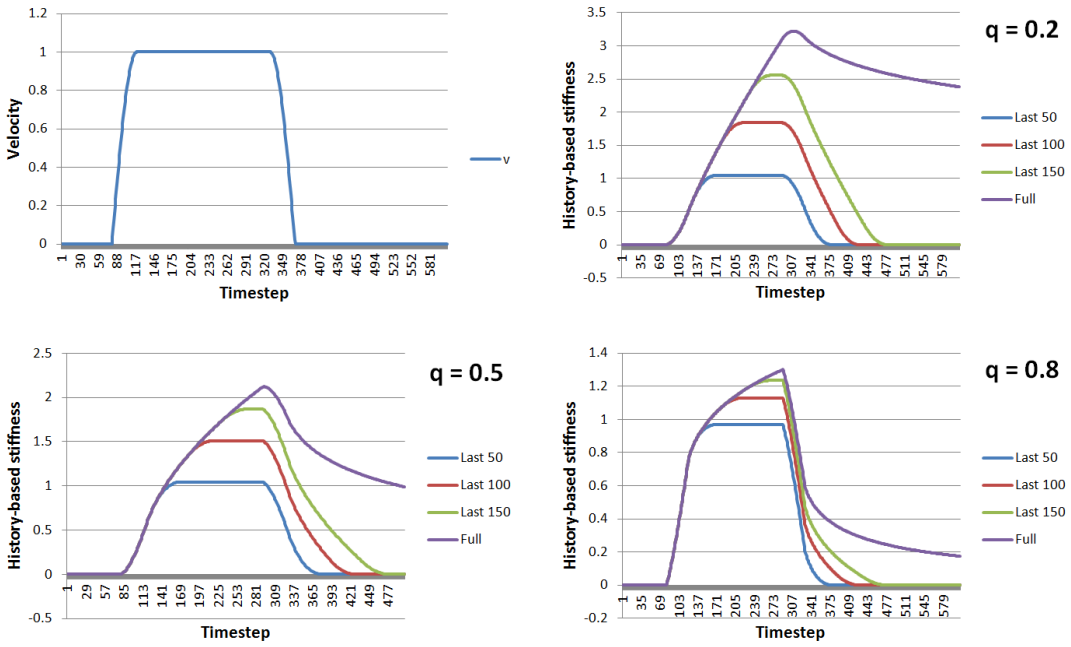

Figure 5.6: The shown graphs present the same plots as in Figure 5.5, except for the upper left graph which shows the velocity of a single particle moving with constant velocity for a given period of time.

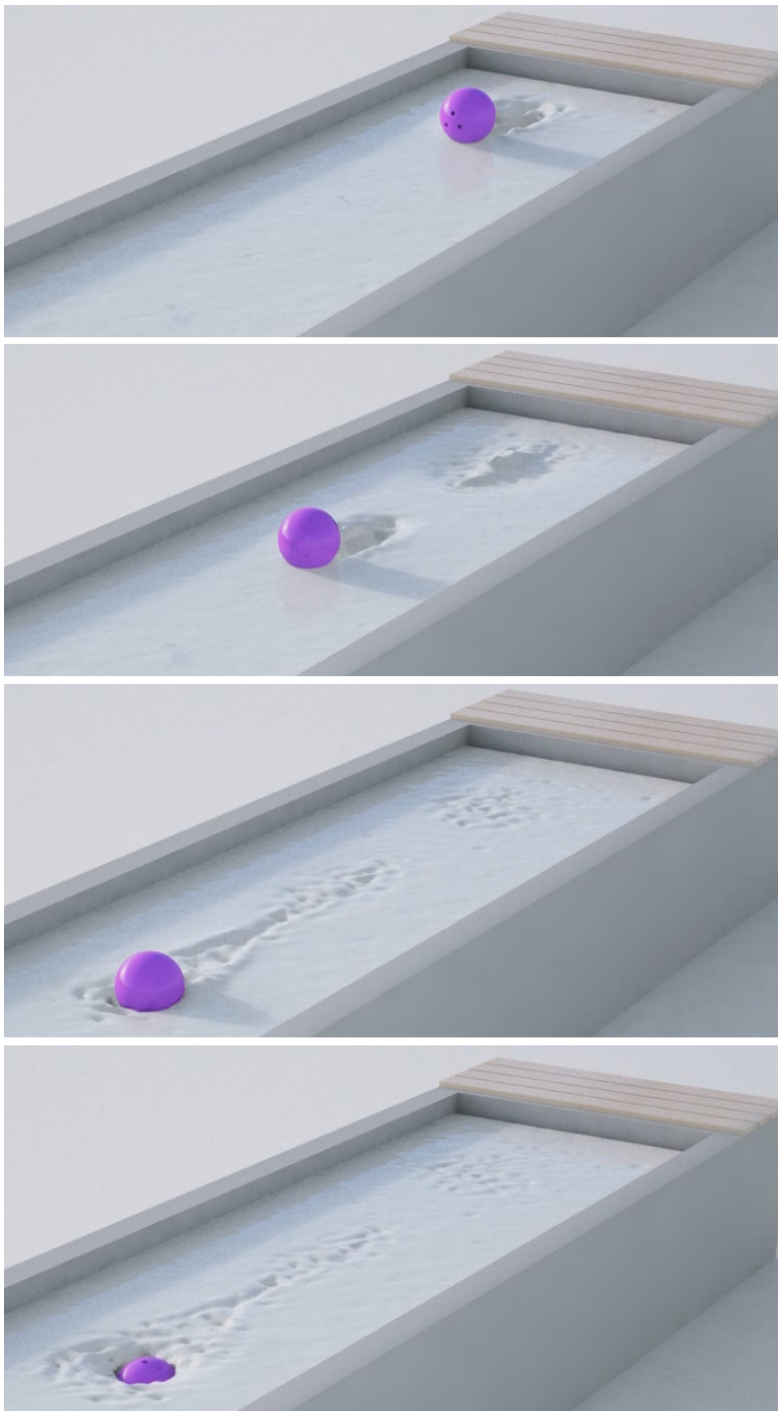

Figure 5.7: The first image (in top-down order) shows the first moment of impact where the shear thickening fluid successfully carries the mass of the bowling ball and makes it bounce. The second image shows the second moment of impact after the bounce. The bowling ball still won't sink but slows down. The third image shows the moment where the bowling ball comes to a stop. Since the fluid is still in its solid-like state, the bowling ball will not immediately sink. The fourth image shows the end of the simulation where the history-based stiffness starts to vanish and the fluid transitions to fluid-like state, finally resulting in the sinking of the bowling ball.

CHAPTER 6

Directional Constraint Enforcement for Fast Cloth Simulation

We introduce a new method that greatly improves the iterative edge length constraint enforcement frequently used in real-time cloth simulation systems for preventing overstretching. Our method is based in the directional enforcement of constraints and on the simultaneous progressive scanning of the cloth edges, starting from fixed vertices and propagating on the direction of gravity. The proposed method successfully detects the meaningful springs to be corrected and ignores the ones that do not have any significance on the overall visual result. The proposed approach is simple and robust and is able to achieve realistic cloth simulations without overstretching, without causing any visual artifacts, and by dramatically decreasing the computational cost of the constraint enforcement process.

6.1 Introduction

Cloth simulation is widely used in computer graphics and many systems are able to simulate in real time moderately complex cloth models. Among the several possible approaches for achieving deformable surfaces, particle-based systems remain the most popular model for achieving interactive, real-time results. One common undesired behavior in particle-based cloth models is the effect of overstretching, and specific geometric edge length enforcement procedures are commonly employed.

Iterative enforcement of constraints coupled with explicit integration methods is well suited for cloth simulation systems with modest particle numbers. This combination successfully avoids the requirement of solving systems of equations and therefore leads to computationally fast and visually satisfying simulations, suitable for computer games. Although research has been developed on the problem of constraint enforcement in general, the useful iterative geometric edge length enforcement method has not been improved since its introduction by Provot [Pro95]. We present in this chapter a new approach that greatly improves this method. Our method is simple to be implemented, robust, and is able to achieve realistic stiff cloth behavior without any visual artifacts and improves the computation time of the regular iterative constraint enforcement by up to 80%.

The proposed method is based on computing a meaningful one-pass ordering of edge

corrections by considering the direction in which stretching occurs most. First, we experimentally determine the correction priorities over the different types of considered springs in order to minimize the stretch effect of the gravitational pull. Then, we employ corrections starting from all the fixed vertices in a given simulation and synchronously expanding towards the bottom of the cloth. We also limit the correction number of each vertex. By prioritizing the correction order of different types of springs and limiting the correction number of all vertices to a small number, our method successfully detects the important springs to be corrected and ignores the ones that do not have any significance on the overall visual result. Our final method is able to dramatically decrease the cost of the iterative constraint enforcement process and at the same time successfully eliminates undesired overstretching effects.

6.2 Related Work

Cloth simulation is a central topic in computer graphics and there is a rich literature available [CYM92, BHW92, Pro95, EWS96, VCT95, BW98, HB00, CK02a, TPB87, TF88]. We focus our review on how constraint enforcement has been used in previous works.

A common approach for the iterative constraint enforcement procedure is proposed by Provot [Pro95]. In this method, the cloth is scanned in an iterative way and all springs that are disturbed more than a given threshold in respect to their rest lengths are identified. These springs are then restored to their rest length by moving the two particles connected to a spring along the spring axis, bringing them closer if the spring is over stretched; or further away from each other if the spring is over compressed (see Fig. 6.1).

The result of the described procedure is that a stiffer cloth behavior is produced. Note however that not all springs can be perfectly corrected since one spring correction can alter a previously corrected spring. There are a number of additional limitations inherent to this kind of brute-force correction approach. One corrective iteration over the cloth does not always guarantee a visually satisfying result, and in addition, the cloth might exhibit an oscillatory behavior in several areas. To overcome these limitations, a common approach is to apply the corrective procedure several times until a visually satisfying result can be obtained. Obviously, increasing the number of constraint enforcement iterations will decrease the performance of the simulation and there is no reliable way to determine how many iterations would be necessary to achieve the desired results. In [Pro95] it is mentioned that the order of correction of springs depends entirely on their used data structure and in cases where constraints are globally extending to the whole cloth object, the order of springs would probably have more importance and should be studied. This observation is the primary motivation for our proposed directional constraint enforcement method.

Aside from iterative approaches, it is also possible to address the problem by solving a

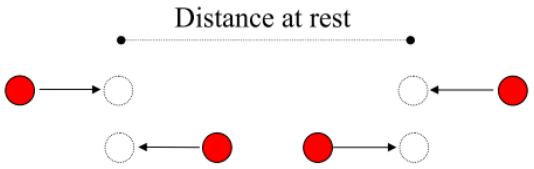

Figure 6.1: Direction of correction for two particles connected by a spring.

system of non-linear constraint equations. Terzopoulos et al [TPB87] used the Gauss-Seidel method to approximate the solution with a linear system of equations [TPB87]. Another popular method is called the Reduced Coordinate Formulation. In this formulation, an unconstrained system with a given number of degrees of freedom is subject to a set of constraints that remove some of the degrees of freedom and parameterizes the rest; thus leading to generalized coordinates [IC87]. Lagrange multiplier formulations are also widely used and have the advantage of defining the velocity and other types of constraints which cannot be formulated using the Reduced Coordinates method [Bar96, BW97].

Among these several approaches, our proposed method is most suitable for applications where speed of computation and simplicity of implementation are most important, such as in computer games and real-time simulations.

6.3 Cloth Model

The mesh structure of our cloth model is based on the scheme presented by [CK02a]. The cloth is represented as a quadrilateral mesh of particles. Particles are connected to each other by massless springs. There are three types of spring elements, which are responsible for the stretch, shear and bend forces. The connectivity of the springs is described in the following way: a particle indexed by $p(i, j)$ is connected by stretch springs to its neighbor particles indexed by $p(i \pm 1, j)$, $p(i, j \pm 1)$. The shear springs connect the particle to $p(i \pm 1, j \pm 1)$, and the bend springs connect the particle to $p(i \pm 2, j)$, $p(i, j \pm 2)$ and $p(i \pm 2, j \pm 2)$.

6.3.1 Forces

The dynamics of the system is governed by Newton's second law of motion $\vec{F} = m\vec{a}$, where m is the mass and $\vec{a}$ is the acceleration of a particle. The acceleration is computed at every time step based on the total force $\vec{F}$ applied to the particle, and which accounts for all external and internal forces.

The gravitational force is given by $\vec{F}_g = m\,\vec{g}$ and it is the only external force in our system. The internal forces acting on a particle are the forces which originate from the

stretch, bend and shear springs connected to each particle. The spring force $\vec{F_s}$ between two particles is given by Hooke's law:

$$\vec{F_s} = -k_s(|\vec{l}| - r)\frac{\vec{l}}{|\vec{l}|} \tag{6.1}$$

where $\vec{l} = \vec{x_i} - \vec{x_j}$ is the difference between the positions of the two particles, r is the rest length of the spring and k_s is the linear spring constant.

6.3.2 Verlet Integration

Although it is possible to couple the constraint enforcement process with various integration methods, we use the Verlet integration method in this work. The Verlet integration is a numerical method that originated from the field of molecular dynamics. Thanks to its simplicity, performance and stability, it has become popular for real-time cloth simulation, in particular in computer games.

The Verlet method stores the current and previous positions of each particle as the state of the system. The velocity is thus implicitly represented by positions. The position update step for each particle is computed with:

$$\vec{x}_{t+1} = 2\vec{x}_t - \vec{x}_{t-1} + \vec{a}h^2, \tag{6.2}$$

where $\vec{x}_{t+1}$, $\vec{x}_t$ and $\vec{x}_{t-1}$ are the positions of a particle at the next, current and previous timesteps, $\vec{a}$ is the current acceleration influencing the particle and h is the integration timestep. The acceleration influencing the particle is calculated based on the total force acting on it. The damping of the system might be fine-tuned by changing the constant multiplier (of value 2) in the equation. Decreasing the value to less than 2 will increase the damping; whereas increasing to more than 2 will decrease the damping.

Because the velocity is implicitly calculated, the method tends to stay stable even when relatively large timesteps are used. However, although the simulation remains fast and stable, the use of large timesteps will lead to overshooting of positions and result in a super-elastic cloth. Therefore, simulations relying on Verlet integration are often coupled with an iterative constraint enforcement process that attempts to restore the original distances between each pair of particles connected by a spring in the cloth model.

6.4 Directional Constraint Enforcement

Our directional constraint enforcement method achieves improved results with a meaningful traversal order of spring corrections. Our method minimizes the number of total corrections needed by preventing redundant repetition of corrections and avoiding the oscillation behavior that often results from a simple iterative constraint enforcement traversal.

Our method is based on the observation that the most stretched springs tend to be the ones in the regions most influenced by the contact forces in a given simulation scenario. Therefore, the most stretched regions on the surface of the cloth are expected to be the regions close to fixed vertices of the cloth, which are vertices that tend not to move relative to their respective colliding object. By defining a traversal order that first corrects the springs around the fixed vertices of the cloth, and expanding the corrections on the direction of the gravity, it is possible to ensure that corrections will be effective and without redundancy.

Our algorithm assumes that the cloth stretches under gravity and has a set of fixed vertices. These fixed vertices correspond to vertices that attach the cloth to a character's body, a flag to a flag pole or a table cloth to the table, etc. Since there may possibly be more than one fixed vertex to start the correction expansion, it is important to synchronize the corrections originating from different fixed vertices. Our experiments showed that failing to synchronize the expansions are causing serious visual artifacts; and the simultaneous expansion remains an important aspect of the algorithm.

It is also crucial to analyze the overall visual look of the cloth according to the order chosen on the different types of springs. If we categorize the different types of springs along with their elongation directions, we can come up with three major categories: 1) stretch springs elongating towards left or right (horizontal stretch springs), 2) stretch springs elongating towards the bottom of the cloth (vertical stretch springs) and 3) shear springs elongating towards the bottom of the cloth. Note that we do not take into account springs elongating upwards, to make sure that the correction will be expanded towards the gravity direction.

It is not straightforward to determine which type of spring should be given correction priority over other types. We have experimented with different orderings for the three different types of springs, leading to six different combinations. Two of these combinations produced similar correction orders as others; so, we ended up having four different correction orders possible. The correction maps of these four different orderings and their associated deformation results are shown in Fig. 6.2. The experiment of switching among the different orderings to observe the resulting cloth postures can be found on the accompanying video of this work. This experiment showed that the best-looking deformations are obtained with the following order of correction: 1) Horizontal stretch springs, 2) Vertical stretch springs, 3) Shear springs.

With the ordering for processing the different types of springs determined, an edge

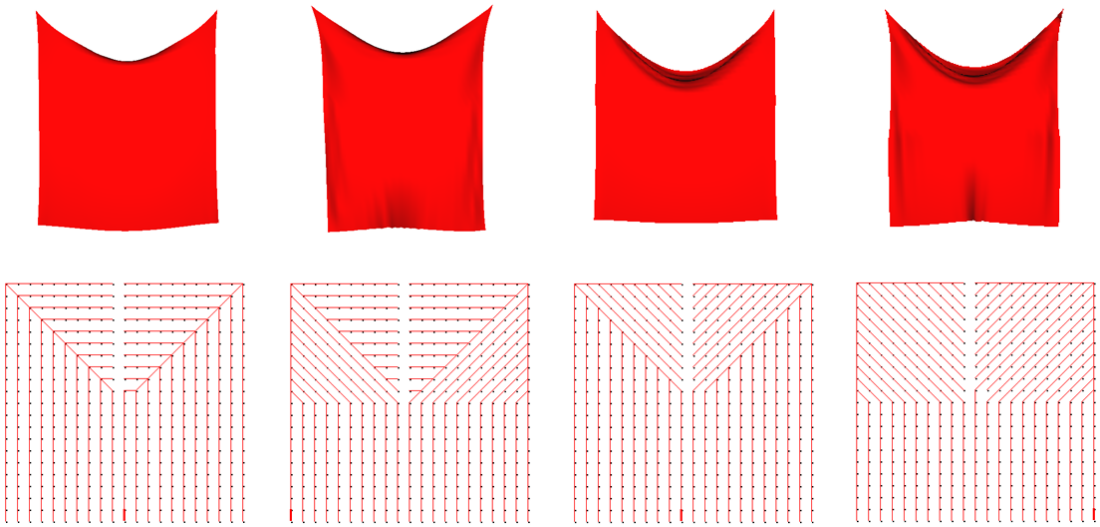

Figure 6.2: Correction patterns and their respective deformation results obtained from choosing different correction orders among the different types of springs.

traversal process respecing this ordering can then be devised. Algorithm 3 summarizes our final edge correction ordering generation process. Details of the algorithm are given in the next paragraphs.

Algorithm 1 receives as input the set of fixed vertices V_f. At the beginning of the algorithm, we use an empty queue Q to maintain the synchronous order of correction expansions. When the algorithm terminates, list L will be filled with the right order of edges to be corrected during run time.

The overall algorithm proceeds as follows: first, all the fixed vertices are pushed into the expansion queue Q. Then, until there is no vertex left in the queue, the vertices are popped and expanded one by one. Procedure *GetHorizontalStretchSprings* (v_s) returns the horizontal stretch springs connected to the vertex v_s. Similarly, *GetVerticalStretchSprings* (v_s) returns the vertical stretch springs connected to the vertex v_s, and *GetShearSprings* (v_s) returns the shear springs connected to the vertex v_s. Finally, procedure *GetGoalVertex* (v_s, s) returns the vertex that is connected to the vertex v_s by the spring s.

For each vertex expanded we first find all the horizontal stretch springs connected to it. Then, we find the goal vertices that the vertex is connected to through these springs. Finally, we try to expand the correction map towards those vertices. When the *Expand* procedure is called, the expansion to the goal vertex is not guaranteed. The algorithm first checks how many times the goal vertex has been visited. If it already reached the visit limit number then we do not expand. If the visit limit is not reached yet, the algorithm expands to the goal vertex and adds the edge from the current vertex to the goal vertex to the ordered correction list L. The visit count of the goal vertex will be incremented and the goal vertex will be added to the expansion queue Q. These

Algorithm 3 Ordered Edge Correction

Compute_Correction_List (V_f)

1. $L \leftarrow null$
2. **for all** v such that $v \in V_f$ **do**
3. $Q.push(v)$
4. **end for**
5. **while** Q is not empty **do**
6. $v_S \leftarrow Q.pop()$
7. $S_h \leftarrow$ *GetHorizontalStretchSprings* (v_s)
8. **for all** s such that $s \in S_h$ **do**
9. $v_g \leftarrow$ *GetGoalVertex* (v_s, s)
10. *Expand* (L, v_s, v_g)
11. **end for**
12. $S_v \leftarrow$ *GetVerticalStretchSprings* (v_s)
13. **for all** s such that $s \in V_s$ **do**
14. $v_g \leftarrow$ *GetGoalVertex* (v_s, s)
15. *Expand* (L, v_s, v_g)
16. **end for**
17. $S_s \leftarrow$ *GetShearSprings* (v_s)
18. **for all** s such that $s \in S_s$ **do**
19. $v_g \leftarrow$ *GetGoalVertex* (v_s, s)
20. *Expand* (L, v_s, v_g)
21. **end for**
22. **end while**
23. **return** L

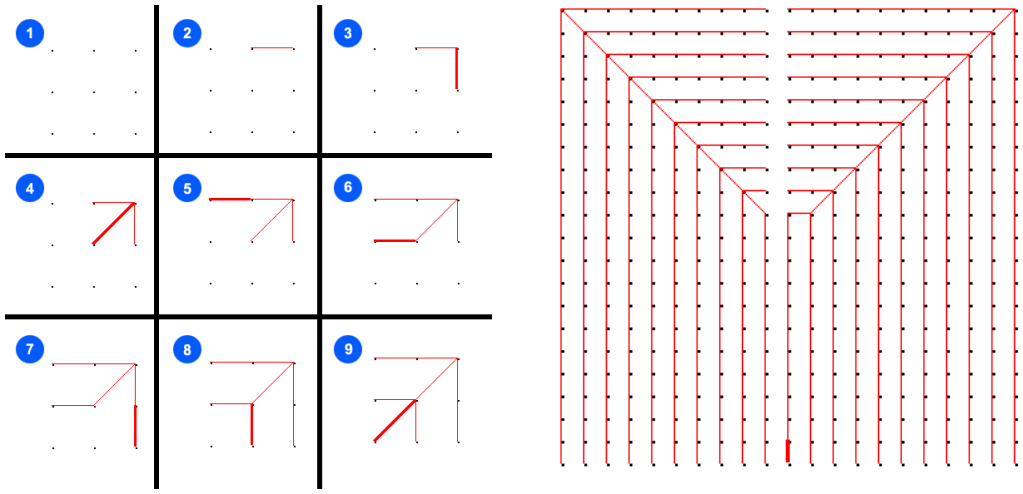

Figure 6.3: Left: step-by-step order of correction starting from an upper right fixed vertex of the cloth. Right: correction map created for a cloth fixed from the two top corners.

procedures will be repeated for vertical stretch springs and shear springs as well. The algorithm will stop when there is no vertex left in the queue. At the end of the algorithm, we will have the ordered correction list L completed.

List L is precomputed in advance for each set of fixed vertices to be considered during the real-time simulation. Then, after each integration step in the simulation, the iterative constraint enforcement step will traverse the edges in L, correcting only the position of the second vertices in each edge, satisfying the desired length of each edge. Example results obtained with the algorithm are shown in Figure 6.3.

6.4.1 Using multiple correction maps

Our method relies on precomputed fixed orderings (encoded in list L) in order to maximize computation performance in real time. Since the set of fixed vertices may change during a simulation, the validity of the precomputed list L may be reduced depending on how irregular and unpredictable the deformation events are. We tested our method on various scenes such as a planar cloth interacting with a solid ball and a complex cloth model interacting with a women model performing a "catwalk". At both simple and complex scenarios, the obtained varied cloth-object collisions and cloth-cloth collisions did not seem to cause any significant degradation in the obtained quality of results.

It is also possible to precompute several correction maps to further increase the performance and the accuracy of the animation. Precomputed maps corresponding to different collision states that repeatedly occur in the animation can be used by switching between them according to events in the simulation. For example, in the animation of the walk-

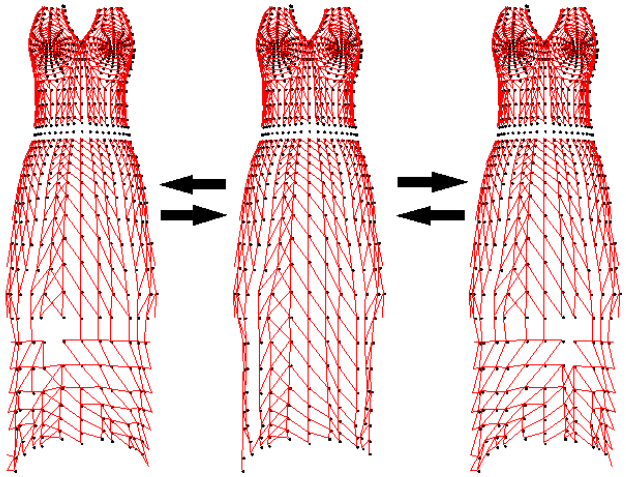

Figure 6.4: Three alternating correction maps used for the long skirt simulation during walking: map with right knee collision (left), map with no knee collision (center), and map with left knee collision (right).

ing character wearing a long skirt, the character's knees are interchangeably colliding with the skirt generating a visible event with impact on the edge correction list. Everytime a knee collides with the skirt, the colliding vertices are going to be corrected by the collision detection module; we can as well treat them as fixed vertices in additional correction lists. We have obtained improved results with three correction maps for the cases where 1) the knees do not touch the skirt, 2) only the right knee touches the skirt and 3) only the left knee touches the skirt. As shown in Figure 6.4, everytime we detect a knee collision (or obsence of collision) we switch to its corresponding correction map. If the deformation is specific for one given walking animation, such events can be encoded in the time-parameterization of the animation, eliminating decisions based on collision detection events. The use of multiple correction maps is therefore suitable for simulations dependable on cyclic animations such as walking.

6.5 Results

In order to test the applicability of our method to different cloth simulation scenarios we have tested our directional correction algorithm with cloths with different numbers of fixed vertices and at different places. Figure 6.5 shows the different correction maps produced for different initial sets of fixed vertices. Although the order of the corrections change, the number of springs to be corrected stays practically the same. The only exception is the scenario with the falling cloth on the table. In that scenario, a large circular set of fixed vertices is naturally marked as fixed, and the number of springs to be corrected is much less.

We have also tested our method on a more complex long skirt cloth model interacting

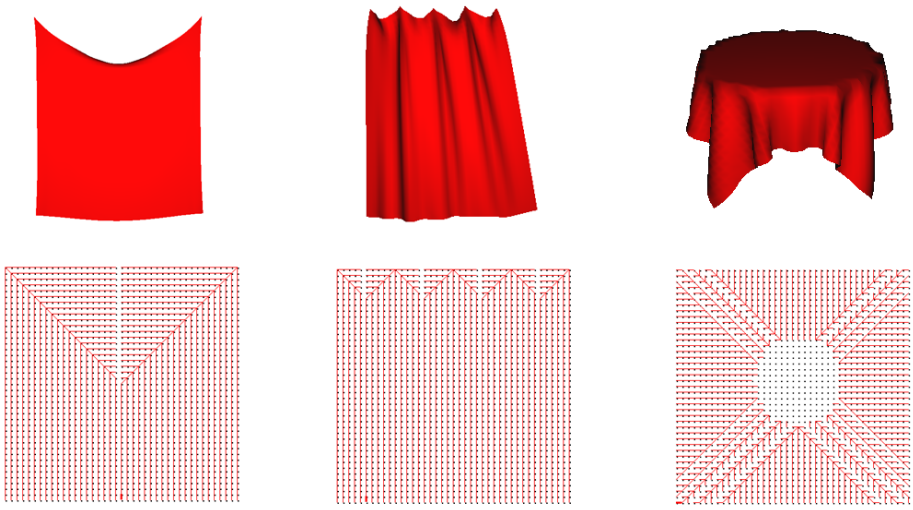

Figure 6.5: The figure shows the correction maps generated by our algorithm and their associated cloth simulation scenarios.

with a walking character. Our method proved to be efficient and did not cause any undesired visual artifacts. Additional examples of edge correction orders obtained for the skirt cloth model are shown in Figure 6.6 and rendered results are shown in Figure 6.7.

Our correction methodology is dependent on the set of fixed vertices on the surface of the cloth. In most of the cloth simulation scenarios, such as hanging clothes, capes, flags, table clothes, or dressed characters, the set of fixed vertices stays the same during the whole animation. Therefore, in these cases the correction map would only be precomputed once and in a very small amount of time. The precomputation time of the correction map is about 0.52 seconds for a cloth of 1681 particles on a 2.4 GHz Intel Core 2 Duo computer.

In terms of visual quality, our model achieves visually satisfying results from a single correction pass over the list of edges. This is a major improvement in respect to the original (non-ordered) constraint enforcement procedure that relies on several passes to achieve good results. One aditional important advantage of our method is that the directional correction completely eliminates the shaky cloth behavior that is frequently seen in multiple pass procedures.

6.5.1 Limitations

As previously stated, our correction methodology is dependent on how predictable the set of fixed vertices is. In cloth simulation scenarios where the set of fixed points change very frequently, our method would need to update the correction map after each

significant change. In such situations an automatic procedure for detecting the validity of correction list L and triggering an ordering update everytime the set of fixed vertices significantly changes from the original set can be integrated in the simulation system. However, this would still maintain the negative effect of penalizing the overall performance of the simulation. On the other hand, if the change on the set of fixed vertices is not frequent, this will not represent a major slow down in the overall computation time.

We are also assuming that the cloth stretches under one single major constant external force (gravity). Although this assumption addresses the vast majority of cloth simulation scenarios, more complex situations where the stretch direction would be affected by multiple varying external forces have not been tested.

6.6 Conclusions

We have introduced a new methodology for determining an efficient and effective traversal order for the iterative edge length enforcement in cloth simulations. Our method is robust, simple to be implemented and is able to achieve realistic stiff cloth behavior without causing overstretching or visual artifacts (like shaky effects). Furthermore, our method improves the computation time by 80% with respect to the regular iterative constraint enforcement procedure. We believe that our directional constraint enforcement approach will prove itself useful to a number of scenarios employing particle-based deformable models, and in particular in real-time applications such as computer games.

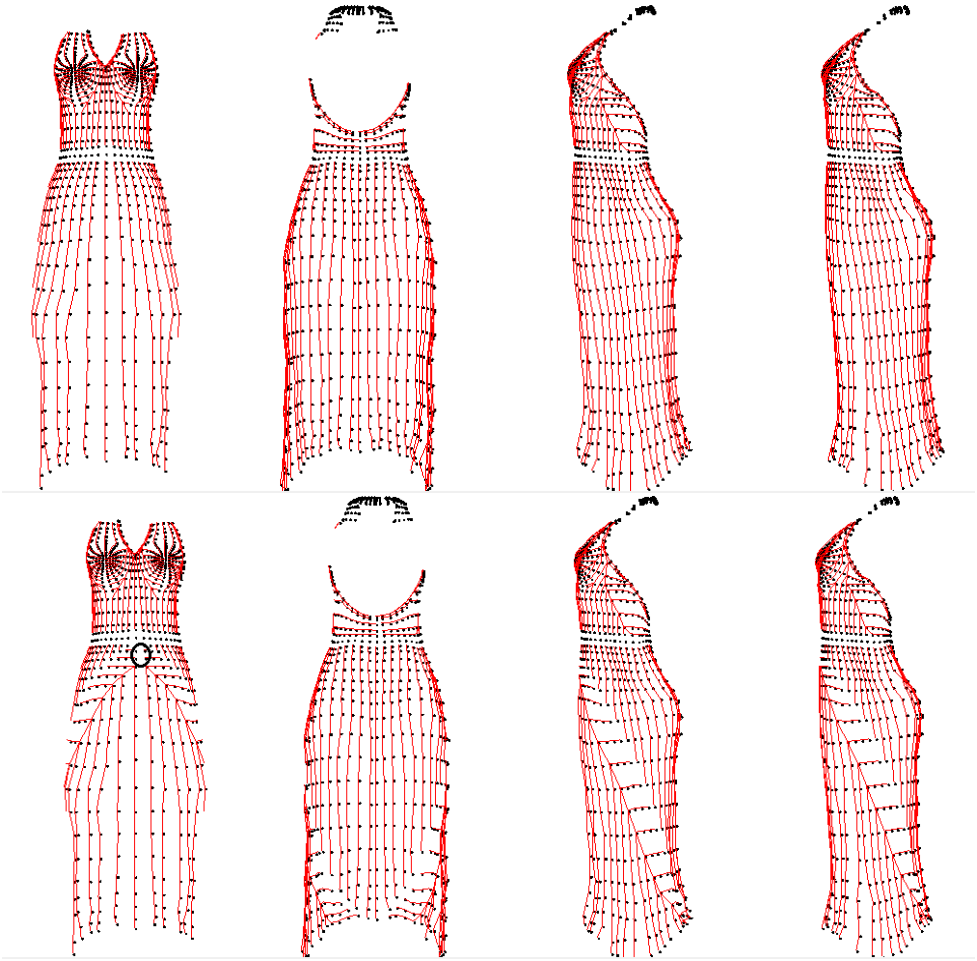

Figure 6.6: Top row: Order of correction obtained with fixed vertices selected at the shoulder and belt regions. Bottom row: A small number of vertices (shown by the circle on the bottom-left image) is added to the set of fixed vertices. Notice how the new correction order obtained is different and how the correction map visually corresponds to the expected buckling of the cloth.

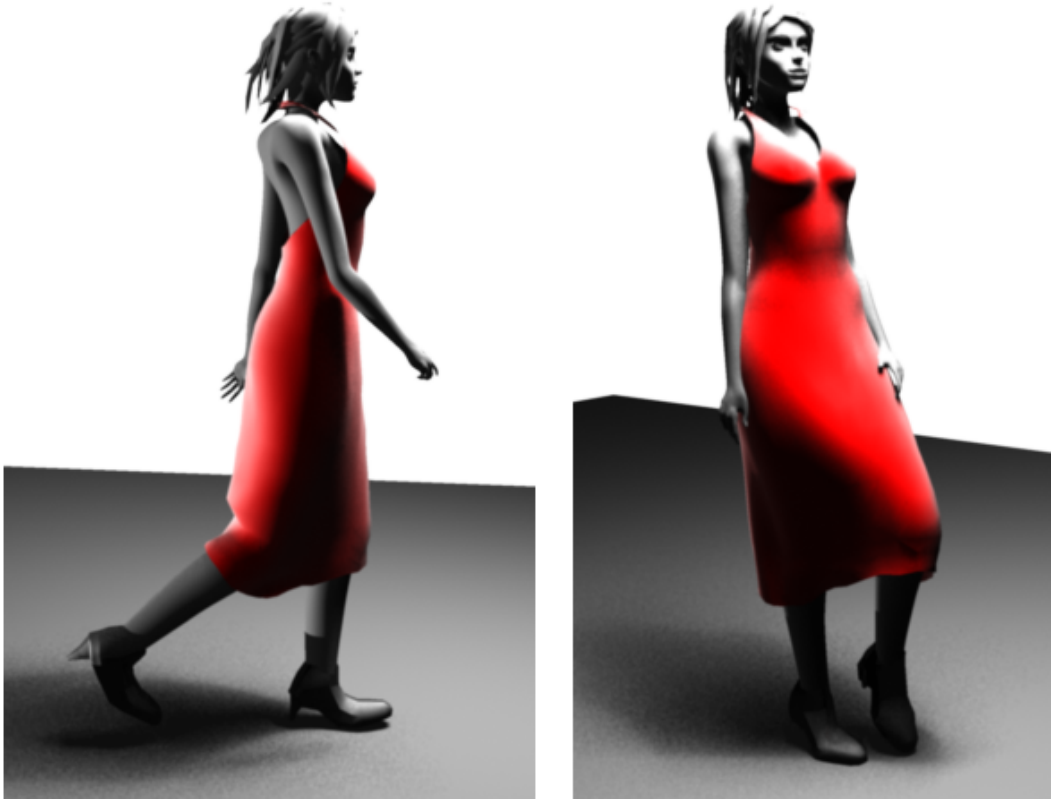

Figure 6.7: Example of a long skirt that would significantly overstretch if no constraint enforcement is made. Our method eliminates the overstreching in an efficient manner.

CHAPTER 7

Conclusions

In this dissertation, we introduced three physics-based animation models based on fractional calculus. These methods are applied to the problems in popular areas of computer graphics such as cloth simulation and Newtonian and non-Newtonian fluid simulation. In this chapter, we will discuss our main contributions and conclude the dissertation.

7.1 Summary of Contributions

Our primary objective was to explore the contributions that fractional calculus can offer the computer graphics community. To that end, this dissertation presents three contributions: (1) underwater cloth model with history-based damping, (2) SPH model with history-based viscosity term that can better represent the dynamics of colliding flows, (3) shear thickening fluid model with history-based stiffness.

7.1.1 Underwater Cloth

Our first contribution emerged from fluid mechanics findings showing that the history forces on a solid particle moving underwater can be calculated based on the half-derivative of the particle's velocity. This implies that underwater behavior can be acquired up to some extent without simulating the body of fluid around the particle. We extended this approach to a particle system that constitutes the cloth model. We replaced the integer order derivatives with half-derivatives in the damping equations of the cloth, therefore achieving realistic underwater cloth motion.

Creating similar effects using conventional methods requires a simulation of the fluid body around the cloth. This kind of simulation is known to take hours of computation time even when a modest number of particles is used. With respect to this aspect, the creation of underwater behavior without simulating the body of water is remarkable and it saves considerable time for animators for the creation of underwater effects.

Furthermore, the fact that fractional terms well model history effects can be clearly seen in our animations. In real life experiments, force impulses applied to cloth underwater show distinctive lasting effects. The evaluations in Figure 3.10 especially show very clearly how well this behavior can be achieved in animation. A solid sphere colliding

with the cloth creates long lasting propagations. These effects are a direct result of history effects.

Diverging from the main focus of the dissertation, we also contributed in the simulation of cloth in computer games. We increased the efficiency of the constraint enforcement procedure by decreasing the running time by 80%. The model is simple, efficient and has the potential to be very useful in real-time cloth simulation scenarios.

7.1.2 Colliding Flows

After the first application of fractional calculus on cloth simulation, we aimed to extend our research to fluid simulation. The main motivation for this research originated from fluid mechanics studies that present definitive experimental evidence of fractional history effects in the unsteady viscous motion of oscillating particles in Newtonian fluid. We applied this idea to a simulation scenario rich in unsteady particle motion. We used half-derivative based viscosity terms in the equations of a very popular particle-based fluid simulation method called SPH. The model successfully improved the accuracy of the simulation in regions where flows collide and create unsteady behavior.

The fact that our model modifies the viscosity terms in the SPH method raises the question of whether the resulting effects are due to a change in viscosity of the fluid. We specifically addressed this concern by validating our method against fluid simulations from benchmark software and showed that our simulations in fact do not add any artificial viscosity and hence produce valid results. Furthermore, we numerically compared our results against the classical SPH method and concluded that our fractional SPH produces more accurate results for colliding flow regions.

In the underwater cloth research, it was not possible to numerically compare our method with any other work in the literature as there is no prior effort to simulate similar scenarios in computer graphics. However, in the research on the colliding flow animation, our results are mostly numerically evaluated. This is important for the completeness of this dissertation in terms of providing both visual and numerical evidence for fractional calculus based methods.

7.1.3 Shear Thickening Fluids

The third major contribution is the application of fractional calculus for the simulation of shear thickening fluids. This type of fluid has very unconventional behaviors of phase changing and solidification upon impact. These features makes shear thickening fluids an interesting simulation topic for computer graphics. Motivated by the physics and rheology papers that model non-Newtonian fluid behavior through fractional derivative based elasticity terms, we aimed to adapt the idea to computer graphics. We presented a combination of SPH and mass-spring system to simulate shear thickening fluids. Our

contribution is mainly on the employment of fractional derivatives in order to create history-based stiffness for springs connecting the particles.

Our model simulates shear thickening fluid in a great extent and is able to produce most of the interesting behaviors that this fluid exhibits. We were able to demonstrate simulations very similar to real life experiments, strongly validating our results. Shear thickening fluids were never simulated before in computer graphics and our work is the first to simulate this material.

7.2 Future Directions

For the future work, we believe the use of fractional calculus based models will be proven useful in a variety of problems. The animations of deformable objects moving underwater can definitely benefit from the ability of fractional calculus based methods in simulating the history forces. More realistic animation of hair moving underwater can be achieved. Moreover, the movements of the body parts of sea animals such as the bell and tentacles of a jellyfish can be described more accurately.

The study of the effects of different orders of fractional derivatives can be an interesting research direction. This would require numerical data from physics and rheology experiments of materials that have some kind of memory effects. This data can later be compared against models employing various orders of fractional derivatives and any possible correlations can be investigated. Materials exhibiting any kind of memory effects, hysteresis or viscoelastic behavior would be a good candidate to model with fractional calculus.

Another important future direction would be the application of fractional calculus based models for organ or tissue simulations. Organs have viscoelastic behavior that can be modeled with fractional calculus. This kind of research would be particularly beneficial for surgical simulations that require a high precision level of accuracy.

REFERENCES

[Abe23] N.H. Abel. "Solution de quelques problèmes à l'aide d'intégrales définie." *Werke*, **1**:10, 1823.

[And95] J.D. Anderson. *Computational fluid dynamics: the basics with applications*. McGraw-Hill, 1995.

[APK07] B. Adams, M. Pauly, R. Keiser, and L.J. Guibas. "Adaptively sampled particle fluids." p. 48, 2007.

[Bar96] D. Baraff. "Linear-time dynamics using Lagrange multipliers." In *Computer Graphcis Proceedings, Annual Conference Series*, pp. 137–146, New York, NY, USA, 1996.

[Bas88] A.B. Basset. "On the motion of a sphere in a viscous liquid." *Transactions of the Royal Philosophy Society of London*, (179):43–63, 1888.

[BB85] J.F. Brady and G. Bossis. "The rheology of concentrated suspensions of spheres in simple shear flow by numerical simulation." *J. Fluid Mech.*, **155**:105, 1985.

[Ben88] W. Benz. "Applications of smoothed particle hydrodynamics (SPH) to astrophysical problems." *Computer Physics Communications*, **48**:97–105, 1988.

[Ben90] W. Benz. *Smoothed Particle Hydrodynamics: A Review*. Kluwer Academic, Boston, 1990.

[Ben92] D.J. Benson. "Computational methods in Lagrangian and Eulerian hydrocodes." *Comput. Methods Appl. Mech. Eng.*, **99**(2-3):235–394, September 1992.

[BHW92] D.E. Breen, D.H. House, and M.J. Wozny. "Predicting the drape of woven cloth using interacting particles." In *Proc. of SIGGRAPH'94*, pp. 365–372, New York, NY, USA, 1992. ACM.

[BJ12] E. Brown and H.M. Jaeger. "The role of dilation and confining stress in shear thickening of dense suspensions." *J. Rheology*, **56**(4):875–923, 2012.

[Bou85] J. Boussinesq. "Sur la résistance qu'oppose un liquide indéfini en repos, sans pesanteur, au mouvement varié d'une sphère solide qu'il mouille sur toute sa surface, quand les vitesses restent bien continues et assez faibles pour que leurs carrés et produits soient négligeables." *C. R. Acad. Sci.*, pp. 935–937, 1885.

[BT83] R. L. Bagley and P. J. Torvik. "A theoretical basis for the applications of fractional calculus to viscoelasticity." *Journal of Rheology*, **27**:201–210, 1983.

[BT07] M. Becker and M. Teschner. "Weakly compressible SPH for free surface flows." In *SCA '07: Proceedings of the 2007 ACM SIGGRAPH/Eurographics symposium on Computer animation*, pp. 209–217, Aire-la-Ville, Switzerland, Switzerland, 2007. Eurographics Association.

[BW97] D. Baraff and A. Witkin. "Physically Based Modeling: Principles and Practice." In *ACM SIGGRAPH '97 Course Notes*, 1997.

[BW98] D. Baraff and A. Witkin. "Large steps in cloth simulation." In *Proc. of SIGGRAPH'98*, pp. 43–54, New York, NY, USA, 1998.

[CA87] E. Charles and Jr. Anderson. "An overview of the theory of hydrocodes." *International Journal of Impact Engineering*, **5**:33 – 59, 1987.

[CBC99] J.K. Chen, J.E. Beraun, and T.C. Carney. "A corrective smoothed particle method for boundary value problems in heat conduction." *Computer Methods in Applied Mechanics and Engineering*, **46**:231–252, 1999.

[CBP05] S. Clavet, P. Beaudoin, and P. Poulin. "Particle-based viscoelastic fluid simulation." In *Proceedings of the 2005 ACM SIGGRAPH/Eurographics symposium on Computer animation*, SCA '05, pp. 219–228, New York, NY, USA, 2005. ACM.

[CK02a] K. Choi and H. Ko. "Stable but responsive cloth." In *Proc. of SIGGRAPH'02*, pp. 604–611, New York, NY, USA, 2002. ACM.

[CK02b] C.F.M. Coimbra and M.H. Kobayashi. "On the viscous motion of a small particle in a rotating cylinder." *Journal of Fluid Mechanics*, (469):257–286, 2002.

[CLL04] C.F.M. Coimbra, D. L'Espérance, A. Lambert, J.D. Trolinger, and R.H. Rangel. "An experimental study on the history effects in high-frequency Stokes flows." *Journal of Fluid Mechanics*, (504):353–363, 2004.

[CMI02] M. Carlson, P. Mucha, B. Van Horn III, and G. Turk. "Melting and Flowing." *Proc. of SIGGRAPH'02*, pp. 167–174, 2002.

[Coi03] C.F.M. Coimbra. "Mechanics with Variable Order Operators." *Annalen der Physik*, (12):692–703, 2003.

[CR98] C.F.M. Coimbra and R.H. Rangel. "General solution of the particle equation of motion in unsteady Stokes flows." *Journal of Fluid Mechanics*, (370):53–72, 1998.

[CYM92] M. Carignan, Y. Yang, N. Magenenat-Thalmann, and D. Thalmann. "Dressing animated synthetic actors with complex deformable clothes." In *Proc. of SIGGRAPH'92*, pp. 99–104, New York, NY, USA, 1992. ACM.

[DC96] M. Desbrun and M.P. Cani. "Smoothed particles: A new paradigm for animating highly deformable bodies." *Computer Animation and Simulation*,

pp. 61–76, 1996.

[Dee10] R.D. Deegan. “Stress hysteresis as the cause of persistent holes in particulate suspensions.” *Phys. Rev. E*, **81**, 2010.

[DFF05] K. Diethelm, N.J. Ford, A.D. Freed, and Yu. Luchko. “Algorithms for the fractional calculus: A selection of numerical methods.” *Computer Methods in Applied Mechanics and Engineering*, **194**:743–773, 2005.

[DG95] M. Desbrun and M.P. Gascuel. “Animating soft substances with implicit surfaces.” *Proc. of SIGGRAPH’95*, pp. 287–290, 1995.

[DSB99] M. Desbrun, P. Schröder, and A. Barr. “Interactive animation of structured deformable objects.” In *Proc. of Graphics interface*, pp. 1–8, San Francisco, CA, USA, 1999.

[ETK07] S. Elcott, Y. Tong, E. Kanso, P. Schröder, and M. Desbrun. “Stable, circulation-preserving, simplicial fluids.” *ACM Trans. Graph.*, **26**(1):4, 2007.

[EWS96] B. Eberhardt, A. Weber, and W. Strasser. “A fast, flexible, particle-system model for cloth draping.” In *IEEE Computer Graphics and Applications’96*, pp. 52–59. IEEE, 1996.

[FJ99] A.R. Frederic and C.L. James. “Smoothed particle hydrodynamics calculations of stellar interactions.” *Annual Review of Astronomical and Astrophysics*, **109**:213–230, 1999.

[Fun65] Y.C. Fung. *Foundations of Solid Mechanics*. Prentice-Hall, 1965.

[GBO04a] T.G. Goktekin, A.W. Bargteil, and J.F. O’Brien. “A method for animating viscoelastic fluids.” In *SIGGRAPH ’04: ACM SIGGRAPH 2004 Papers*, pp. 463–468, New York, NY, USA, 2004. ACM.

[GBO04b] T.G. Goktekin, A.W. Bargteil, and J.F. O’Brien. “A method for animating viscoelastic fluids.” In *ACM SIGGRAPH 2004 Papers*, SIGGRAPH ’04, pp. 463–468, New York, NY, USA, 2004. ACM.

[Ger38] A. Germant. “On fractional differentials.” *Philosophical Magazine*, **25**:540–549, 1938.

[GM77] R.A. Gingold and J.J. Monaghan. “Smoothed Particle Hydrodynamics: Theory and Application to Non-spherical stars.” *Monthly Notices of the Royal Astronomical Society*, **181**:375–389, 1977.

[Han06] M. Hanlon. “New Shear Thickening Fluid (STF) enables flexible, comfortable armor.”, 2006.

[HB00] D.H. House and D.E. Breen, editors. *Cloth Modeling and Animation*. A. K. Peters, Ltd., Natick, MA, USA, 2000.

[Hea92] O. Heaviside. “Electrical papers.” *Macmillan, London*, 1892.

[Hea93] O. Heaviside. “On operatorsin physical mathematics.” *Proc. Roy. Soc.*, **52**:505, 1893.

[HES03] M. Hauth, O. Etzmuß, and W. Straßer. “Analysis of numerical methods for the simulation of deformable models.” *The Visual Computer*, **19**(7-8):581–600, 2003.

[Hil00] R. Hilfer. *Applications of Fractional Calculus in Physics*. World Scientific, River Edge, NJ, 2000.

[Hir88] C. Hirsch. *Numerical Computation of Internal and External Flows*, volume 1. Wiley-Interscience, 1988.

[HKK07] T. Harada, S. Koshizuka, and Y. Kawaguchi. “Smoothed Particle Hydrodynamics on GPUs.” *Structure*, pp. 1–8, 2007.

[HP99] J. Hultman and A. Pharayn. “Hierarchical, dissipative formation of elliptical galaxies: is thermal instability the key mechanism? hydrodynamic simulations including supernova feedback multi-phase gas and metal enrichment in cdm: structure and dynamics of elliptical galaxies.” *Astronomy and Astrophysics*, **347**:769–798, 1999.

[HP08] A. Heibiga and L.I. Palade. “On the rest state stability of an objective fractional derivative viscoelastic fluid model.” *J. Math. Phys.*, **49**:1–22, 2008.

[Hu05] Y. Hu. “Integral Transformations and Anticipative Calculus for Fractional Brownian Motions.” In *Memoirs of the American Mathematical Society*, 2005.

[IAB11] M. Ihmsen, N. Akinci, M. Becker, and M. Teschner. “A Parallel SPH Implementation on Multi-Core CPUs.” *Computer Graphics Forum*, **0**(0):1–12, 2011.

[IAG10] M. Ihmsen, N. Akinci, M. Gissler, and M. Teschner. “Boundary Handling and Adaptive Time-stepping for PCISPH.” In *VRIPHYS*, pp. 79–88, 2010.

[IC87] P.M. Isaacs and M.F. Cohen. “Controlling dynamic simulation with kinematic constraints, behavior functions and inverse dynamics.” In *Computer Graphcis Proceedings, Annual Conference Series*, pp. 131–140, New York, NY, USA, 1987.

[KSL11] S. von Kann, J.H. Snoeijer, D. Lohse, and D. van der Meer. “Nonmonotonic settling of a sphere in a cornstarch suspension.” *Phys. Rev. E*, **84**, 2011.

[KST06] A.A. Kilbas, H.M. Srivastava, and J.J. Trujillo. *Theory and Applications of Fractional Differential Equations*. Amsterdam, The Netherlands, 2006.

[LAD08] T. Lenaerts, B. Adams, and P. Dutré. "Porous flow in particle-based fluid simulations." In *SIGGRAPH '08: ACM SIGGRAPH 2008 papers*, pp. 1–8, New York, NY, USA, 2008. ACM.

[Lar98] R.G. Larson. *The Structure and Rheology of Complex Fluids*. Oxford University Press, USA, October 1998.

[Las95] J. Lasseter. "Toy Story.", 1995.

[LCT05] D. L'Espérance, C.F.M. Coimbra, J.D. Trolinger, and R.H. Rangel. "Experimental verification of fractional history effects on the viscous dynamics of small spherical particles." *Experiments in Fluids*, (38):112–116, 2005.

[LDG96] L. Ling, M. Damodaran, and R.K.L. Gay. "Aerodynamic force models for animating cloth motion in air flow." *The Visual Computer*, **12**(2):84–104, 1996.

[Lin00] Li Ling. "Aerodynamic Effects." In D.H. House and D.E. Breen, editors, *Cloth Modeling and Animation*, pp. 175,195. A. K. Peters, Ltd., Natick, MA, USA, 2000.

[LL03] G.R. Liu and M.B. Liu. *Smoothed Particle Hydrodynamics: A Meshfree Particle Method*. World Scientific Publishing Company, 2003.

[LQ03] G.R. Liu and S.S. Quek. *The Finite Element Method, a practical course*. Butterworth Herinemann, 2003.

[Luc77] L.B. Lucy. "A numerical approach to the testing of the fission hypothesis." *Astronomical Journal*, **82**:1013–1024, 1977.

[MCG03] M. Müller, D. Charypar, and M. Gross. "Particle-based fluid simulation for interactive applications." In *SCA '03: Proceedings of the 2003 ACM SIGGRAPH/Eurographics symposium on Computer animation*, pp. 154–159, Aire-la-Ville, Switzerland, Switzerland, 2003. Eurographics Association.

[MDC93] N. Makris, G.F. Dargush, and M.C. Constantinou. "Dynamic analysis of generalized viscoelastic fluids." *Journal of Engineering Mechanics*, **119**(8):1663–1679, 1993.

[MG83] J.J. Monaghan and R.A. Gingold. "Shock simulation by the particle method SPH." *Journal of Computational Physics*, **52**:374–389, 1983.

[MK95] J.J. Monaghan and A. Kocharyan. "SPH simulation of multi-phase flow." *Astrophysical Journal*, **87**:225–235, 1995.

[ML91] J.J. Monaghan and J.C. Lattanzio. "A simulation of the collapse and fragmentation of cooling molecular clouds." *Astrophysical Journal*, **375**:177–189, 1991.

[MM97] J.P. Morris and J.J. Monaghan. "A switch to reduce SPH viscosity." *Journal of Computational Physics*, **136**:41–50, 1997.

[Mon89] J.J. Monaghan. "On the problem of penetration in particle methods." *Journal of Computational Physics*, **82**:1–15, 1989.

[Mon92] J.J. Monaghan. "Smoothed particle hydrodynamics." *Annual Review of Astronomical and Astrophysics*, **30**:543–574, 1992.

[Mon94] J.J. Monaghan. "Simulating free surface flow with SPH." *Journal of Computational Physics*, **110**:399, 1994.

[Mon95] J.J. Monaghan. "Heat conduction with discontinuous conductivity." *Applied Mathematics Reports and Preprints, Monash University*, **95**:18, 1995.

[Mon05] J J Monaghan. "Smoothed particle hydrodynamics." *Reports on Progress in Physics*, **68**(8):1703, 2005.

[MR93] K.S. Miller and B. Ross. *An Introduction to the Fractional Calculus and Fractional Differential Equations*. John Wiley and Sons, New York, NY, 1993.

[MST04] M. Müller, S. Schirm, M. Teschner, B. Heidelberger, and M. Gross. "Interaction of fluids with deformable solids: Research Articles." *Computer Animation and Virtual Worlds*, **15**(3-4):159–171, 2004.

[MYF99] J.P. Morris, Y.Zhu, and P.J. Fox. "Parallel simulation of pore-scale flow though porous media." *Computers and Geotechnics*, **25**:227–246, 1999.

[NSC08] R. Narain, J. Sewall, M. Carlson, and M.C. Lin. "Fast animation of turbulence using energy transport and procedural synthesis." In *SIGGRAPH Asia '08: ACM SIGGRAPH Asia 2008 papers*, pp. 1–8, New York, NY, USA, 2008. ACM.

[ODA07] G. Oger, M. Doring, B. Alessandrini, and P. Ferrant. "An improved SPH method: Towards higher order convergence." *J. Comput. Phys.*, **225**(2):1472–1492, 2007.

[OK11] O. Ozgen and M. Kallmann. "Directional constraint enforcement for fast cloth simulation." In *Proceedings of the 4th international conference on Motion in Games*, MIG'11, pp. 424–435, Berlin, Heidelberg, 2011. Springer-Verlag.

[OKR10] O. Ozgen, M. Kallmann, L.E.S. Ramirez, and C.F.M. Coimbra. "Underwater cloth simulation with fractional derivatives." *ACM Transactions on Graphics (TOG)*, **29**(3):1–9, 2010.

[OS74] K.B. Oldham and J. Spanier. *The Fractional Calculus*. Academic Press, New York, NY, 1974.

[PAH99] L.I. Palade, R.R. Attane, P. Huilgol, and B. Mena. "Anomalous stability behavior of a properly invariant constitutive equation which generalise fractional derivative models." *Int. J. Eng. Sci.*, **37**:315–329, 1999.

[Pod99] I. Podlubny. *Fractional Differential Equations*. Academic Press, San Diego, CA, 1999.

[Pro95] X. Provot. "Deformation constraints in a mass-spring model to describe rigid cloth behavior." In *Proc. of Graphics Interface '95*, pp. 147–155, New York, NY, USA, 1995.

[PTS09] T. Pfaff, N. Thuerey, A. Selle, and M. Gross. "Synthetic turbulence using artificial boundary layers." In *SIGGRAPH Asia '09: ACM SIGGRAPH Asia 2009 papers*, pp. 1–10, New York, NY, USA, 2009. ACM.

[QJ09] H. Qi and H. Jin. "Unsteady helical flows of a generalized Oldroyd-B fluid with fractional derivative." *Nonlinear Analysis: Real World Applications*, **10**(5):2700–2708, 2009.

[RSG08] A. Robinson-Mosher, T. Shinar, J. Gretarsson, J. Su, and R. Fedkiw. "Two-way coupling of fluids to rigid and deformable solids and shells." *ACM Transactions on Graphics (Proc. of SIGGRAPH'08)*, **27**(3):1–9, 2008.

[RWT11] K. Raveendran, C. Wojtan, and G. Turk. "Hybrid smoothed particle hydrodynamics." In *Proceedings of the 2011 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, SCA '11, pp. 33–42, New York, NY, USA, 2011. ACM.

[SBW99] D.G. Senz, E. Bravo, and S.E. Woosley. "Single and multiple detonations in white dwarfs." *Astronomy and Astrophysics*, **349**:177–188, 1999.

[SCK05] C.M. Soon, C.F.M. Coimbra, and M.H. Kobayashi. "The variable viscoelasticity oscillator." *Annalen der Physik*, (14):378–389, 2005.

[SG11] B. Solenthaler and M. Gross. "Two-scale particle simulation." *ACM Trans. Graph.*, **30**(4):81:1–81:8, July 2011.

[SP09] B. Solenthaler and R. Pajarola. "Predictive-corrective incompressible SPH." In *SIGGRAPH '09: ACM SIGGRAPH 2009 papers*, pp. 1–6, New York, NY, USA, 2009. ACM.

[Spi92] S. Spielberg. "Jurassic Park.", 1992.

[SSI09] A. Selle, J. Su, G. Irving, and R. Fedkiw. "Robust High-Resolution Cloth Using Parallelism, History-Based Collisions, and Accurate Friction." *Transactions on Visualization and Computer Graphics*, **15**(2), 2009.

[Sto45] G.G. Stokes. "On the theories of internal friction of fluids in motion." *Transactions of the Cambridge Philosophy Society*, (8):287–305, 1845.

[Swe92] J.W. Swegle. “Report at Sandia National laboratories.” 1992.

[TF88] D. Terzopoulos and K. Fleischer. “Deformable models.” *The Visual Computer*, **4**(6):306–331, 1988.

[Tim09] Discovery Channel Time Warp. “`www.youtube.com/watch?v=S5SGiwS5L6I`.”, 2009.

[TLP06] A. Treuille, A. Lewis, and Z. Popović. “Model reduction for real-time fluids.” In *ACM SIGGRAPH 2006 Papers*, pp. 826–834, New York, NY, USA, 2006. ACM.

[TPB87] D. Terzopoulos, J. Platt, A. Barr, and K. Fleischer. “Elastically deformable models.” In *Proc. of SIGGRAPH ’87*, pp. 205–214, New York, NY, USA, 1987. ACM.

[TPF89] D. Terzopoulos, J. Platt, and K. Fleischer. “Heating and melting deformable models.” *Graphics Interface*, pp. 219–226, 1989.

[TT94] X. Tu and D. Terzopoulos. “Artificial fishes: physics, locomotion, perception, behavior.” In *Proc. of SIGGRAPH ’94*, pp. 43–50, New York, NY, USA, 1994. ACM Press.

[VCT95] P. Volino, M. Courchesne, and N. Magnenat Thalmann. “Versatile and efficient techniques for simulating cloth and other deformable objects.” In *Proc. of SIGGRAPH’95*, pp. 137–144, New York, NY, USA, 1995. ACM.

[WB09] N.J. Wagner and J.F. Brady. “Shear thickening in colloidal dispersions.” *Phys. Today*, **27**, 2009.

[WG10] H. Weller and C. Greenshields. “OpenFOAM.”, April 2010.

[Wil99] M.L. Wilkins. *Computer simulation of dynamic phenomena*. Springer-Verlag Berlin Heidelberg, 1999.

[Wil07] T.V. Wilson. “How Liquid Body Armor Works.”, 2007.

[WJ12] S.R. Waitukaitis and H.M. Jaeger. “Impact-activated solidification of dense suspensions via dynamic jamming fronts.” *Phys. Today*, **487**:205–209, 2012.

[YFM99] Y.Zhu, P.J. Fox, and J.P. Morris. “A pore-scale numerical model for flow through porous media.” *International Journal for Numerical and Analytical methods in Geomechanics*, **23**:881–904, 1999.

[You08] Youtube.com. “Non-Newtonian Fluid on a Speaker Cone. `www.youtube.com/watch?v=3zoTKXXNQIU`.”, 2008.

[ZT00] O.C. Zienkiewicz and R. L. Taylor. *The Finite Element Method*. Butterworth Herinemann, 5 edition, 2000.