EXTENDING WAVE PARTICLES TO SIMULATE CORRECT OBSTACLE COLLISIONS AND WAVE DIFFRACTION

An Undergraduate Research Scholars Thesis

by

BRENNEN RAY TAYLOR

Submitted to the Undergraduate Research Scholars program at
Texas A&M University
in partial fulfillment of the requirements for the designation as an

UNDERGRADUATE RESEARCH SCHOLAR

Approved by Research Advisor: Dr. John Keyser

May 2017

Major: Computer Science

TABLE OF CONTENTS

Page

ABSTRACT

Extending Wave Particles to Simulate Correct Obstacles Collisions and Wave Diffraction

Brennen Ray Taylor
Department of Computer Science and Engineering
Texas A&M University

Research Advisor: Dr. John Keyser
Department of Computer Science and Engineering
Texas A&M University

One common research area in computer graphics involves the real-time simulation of water surfaces. Uses of water simulation in movies or video games commonly include simulating large bodies of water, such as oceans or lakes. The size of these environments tends to make realistic animation by hand a nearly impossible task, driving the need for both realistic and computationally light simulation methods. Approaches taken to solve this problem include parametric representations of water surfaces, Eulerian grid-based approaches, and Lagrangian particle-based approaches. More recently, a method based on wave particles was proposed, offering a real-time, computationally light, and scalable technique to model wave simulation using modern graphics hardware. However, this method does not support correct behavior of wavefronts when colliding with the edge of obstacles, leading to too little or too much water being reflected off obstacles. In addition, this method also lacks the ability to model diffraction, the bending of surface waves when they approach an obstacle or slit. In this work, we propose extensions of the wave particle simulation to include splitting of wave particles to support more correct collision with obstacle edges, as well as surface wave diffraction. This extension aims to add a level of realism in environments where waves will interact with obstacles. We discuss the

theory and implantation behind the extension of wave particle behaviors to simulate splitting at the edge of obstacles, as well as diffraction. In addition, studies show that the method maintains scalability of wave particles, allowing for large numbers of objects and wave particles to be simulated in real-time.

ACKNOWLEDGEMENTS

I would like to thank my faculty advisor, Dr. John Keyser, for all his guidance throughout this research process.

I would also like to thank Dr. Nancy Amato for the opportunity to first venture into undergraduate research.

I would like to thank my parents, Wendi and Ray Taylor, for everything they have done for me, and for always giving me their support.

Finally, I would like to thank my girlfriend, Sara Maynard, for all the support through this process, and for the encouragement to always keep going.

DEDICATION

I would like to dedicate this thesis to my mother, Wendi Rae Taylor, who taught me the lessons of true courage, strength and love during her fight with Stage-4 breast cancer. May she always be remembered.

SECTION I

INTRODUCTION

Simulating the surface of water waves is an important effect for many discipline areas. The two most notable in computer graphics are the movie industry and the video-game industry. Despite the usefulness of such an effect, many simulation techniques in use forgo physical accuracy for shorter computation time.

While some water simulation techniques can be very realistic and accurate, these simulations usually require a large amount of computational effort, making them candidates only for offline simulation. Thus, these are applicable to only movies, or possibly video game cinematics. On the other hand, real time graphics simulations, such as video games, require modeling to be performed within a small time constraint, on the scale of 16.7ms to 33.3ms per simulation step. Due to this time constraint, many of these real-time methods sacrifice physical accuracy of the simulation for lower computation times. While this exchange allows for simulation of water in real time, the water, if observed, will behave in a noticeably unrealistic manner. As real time simulations move closer to the goal of simulating physically accurate and photorealistic worlds, the simulations must expand to support modeling more detailed water behavior.

In this work, we present our expansion of one such real time simulation technique, wave particles (Yuksel et al., 2007), to support both correct obstacle boundary collisions and diffraction of waves around obstacles. As many use case environments involve water interacting with obstacles, supporting these behaviors is important for the simulation to look realistic in common environments. Our expansion successfully extends the previous method, improving the accuracy of the previous method in more common, obstacle rich environments, while still satisfying the real-time constraint.

In the next section, we give an overview of previous work, as well as discuss in more detail the work our method expands upon. Section 3 describes our collision and diffraction simulation techniques, as well as any key implementation details. In Section 4, we discuss our test cases and results. Finally, Section 5 discusses conclusions we drew from our experiments, as well future directions our work could be taken.

SECTION II

RELATED WORK

There have many different approaches to the problem of simulating water surfaces in real-time. Some early approaches concentrated on modeling water as a surface using Fourier Synthesis (Mastin et al., 1987) or parametric representations (Fournier & Reeves, 1986; Peachey, 1986; Schachter, 1980; Ts'o & Barsky, 1987). Other approaches focused on shallow water equations using height fields, including a 2D Navier-Stokes solution to generate a pressure defined height field (Chen & Lobo, 1995). Yet another approach focuses on Eulerian grid-based solutions to Navier-Stokes equations (N. Foster & Metaxas, 1997; Nick Foster & Fedkiw, 2001; Stam, 1999). One approach (Irving et al., 2006) focuses on modeling realistic 3d behavior of the water surface, such as splashes and breaking of waves. Another method (Chentanez & Müller, 2011a) expands on this, maintaining the detail of the water surface while simulating in a 33.3ms time constraint.

Other creative approaches have been introduced. One such recent work uses a model reduction approach, paired with offline training computation, to generate a solution in scenarios in which it's been trained (Treuille et al., 2006). Another recent approach uses wave particles to track wave motion and generate a height map based on their position (Yuksel et al., 2007). Another recent approach presents a hybrid solution, combining a grid based approach for the main water surface simulation with a particle based approach for modeling spray, splash and foam (Chentanez & Muller, 2010). In this work, we expand upon the wave particle wave particle method (Yuksel et al., 2007), and thus take time to describe this method in more detail.

Wave particles

Wave particles introduced a unique solution for simulating wave surfaces (Yuksel et al., 2007). One of the method's greatest strengths over a full fluid simulation was its speed and stability. In addition, it also provides a generalized form which is sufficient to represent both deep ocean and constrained environments, while maintaining a simple computation framework. As our method extends upon wave particles (Yuksel et al., 2007), we now describe it in more detail.

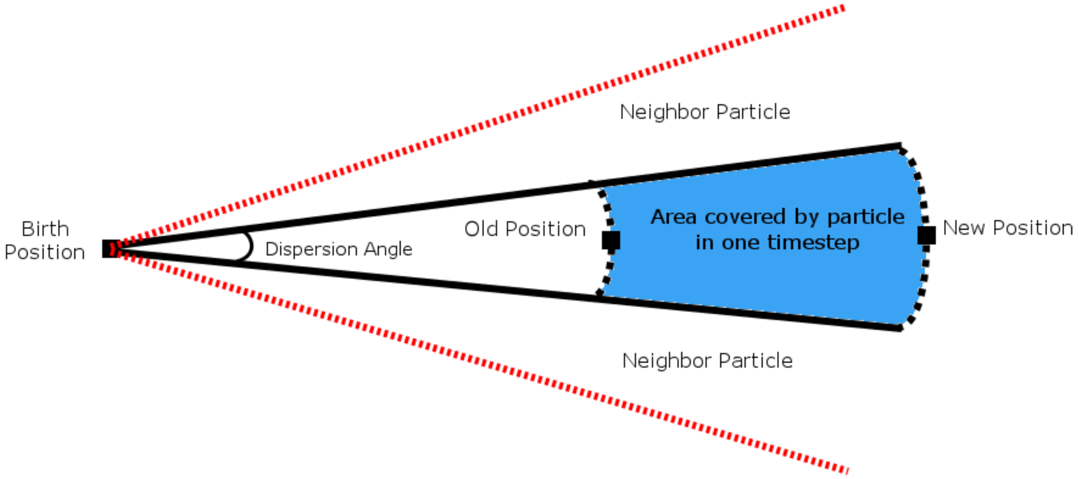

Figure 1. The Components that Make up a Single Wave Particle in a Timestep. The particle moves from the old position, to the new, traveling from the birth position. It is responsible for the region of space in its dispersion angle, marked blue in this diagram.

Water waves take the form of wavefronts moving over the surface of a body of water. However, the wave particles method simplifies the simulation of these wavefronts by representing each as a distribution of wave particles [Figure 1]. These wave particles are then used to generate a heightfield of the water surface containing wavefronts in their correct form.

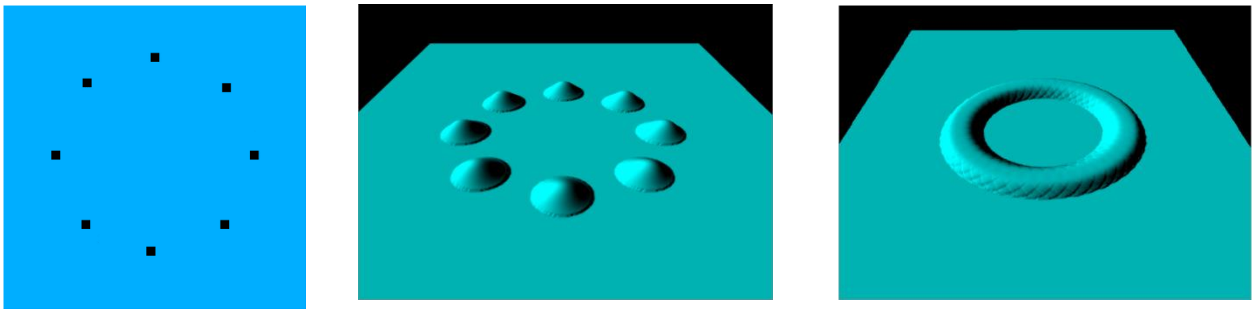

Figure 2. Wave Particle Model. This shows how wave particles represent wavefronts. (Left) Wave particles in the 2D plane. (Center) Particles rendered to a heightmap. (Right) The entire wavefront formed by a full set of particles.

Wave particles exist in a two-dimensional space, propagating only in an x, z plane over time [Figure 2]. Each particle is responsible for an angle of dispersion of a newly generated wavefront. The particle is responsible for representing the section of the wavefront contained within its angle of dispersion. Each particle represents a cosine local deviation function centered around the particle's current position. Each particle's distribution is then rendered to a texture with additive blending so all wave particles' local deviation functions are blended together. This texture acts as a heightmap for the water surface, with each wavefront correctly represented as the sum of all its wave particles' local deviation functions. Finally, a grid is rendered across a distribution of x, z -values, using the corresponding value of the heightmap as each grid point's y-value.

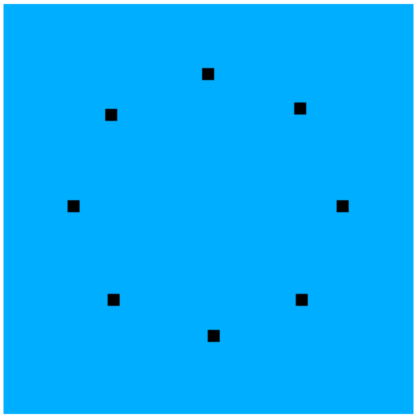

Figure 3. A Group of Wave Particles Before Subdivision. The particles are too spread out, each responsible for too large of an area.

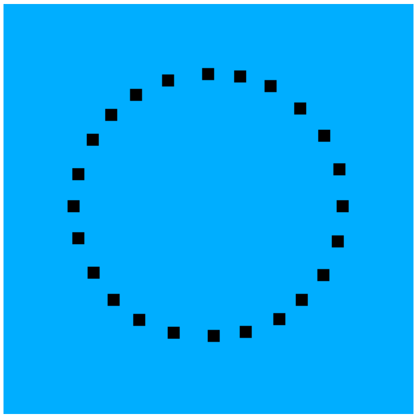

Figure 4. A Group of Wave Particles After Subdivision. Each particle from Figure 3 has divided into three new particles; thus each is responsible for a smaller region of space.

To correctly represent the behavior of wavefronts, wave particles supported two basic behaviors, the first of which is subdivision. Wavefronts expand, meaning that the wave particles representing the wavefront would move apart over time. As the wave particle method constructs waves by the summation of each wave particle's cosine distribution, the separation of particles would lead to gaps appearing in the wavefront as it spreads out [Figure 3]. To counter this, the wave particles subdivide into three new particles when the distance between neighboring particles is greater than half the radius of each particle [Figure 4]. Each new particle has an amplitude 1/3 that of its original particle, thus maintaining the same height of the wavefront. Rather than keeping track of the position of the other particles, each wave particle simply assumes the locations of the other wave particles in its wavefront, using both its dispersion angle and the distance it has traveled thus far. This removes the expensive requirement for particles to communicate.

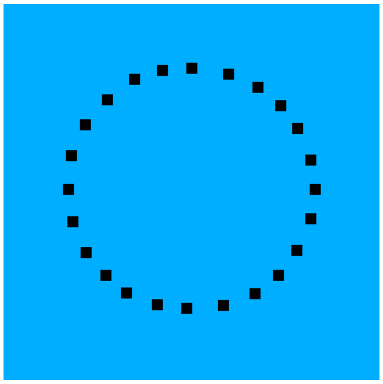

Figure 5. Before Collision. This shows how particles would behave without an object to collide with.

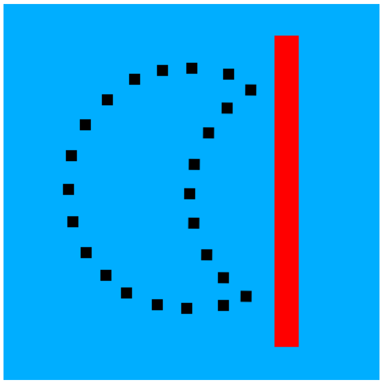

Figure 6. After Collision. This shows how particles would behave with an object to collide with.

The second behavior wave particles support is the ability to interact with obstacles. When a wavefront collides with an obstacle, it is reflected off that obstacle with respect to the angle of collision [Figure 5], [Figure 6]. To represent this, when a wave particle experiences a collision with an obstacle, the wave particle has its birth position reflected over the collision plane. As a wave particle's position is updated based on its birth position, this process will lead to the wave particle reflecting off the collision surface correctly. As wavefronts are represented simply as many wave particles, as each of its wave particle reflects, the wavefront is correctly reflected as well.

SECTION III

METHODS

In this work, we extend the wave particles approach with two new behaviors that more correctly represent wavefront behavior. These new behaviors focus on realistic behavior when wavefronts interact with obstacles, both by more correctly representing the splitting of a wavefront as a subsection of it is reflected by an obstacle, as well as representing diffraction of wavefronts around obstacles.

Particle Splitting

The first behavior we add is the ability of wave particles to split into new particles of a smaller radius. In the previous work, all wave particles were assumed to have the same radius. When a particle needed to subdivide, and thus assumed that the distance to its neighbor particles was too great, the particle would subdivide into three new particles of the same radius. However, there is a problem with requiring all particles to have the same radius. If the particle collided within half the radius of the particle from the end of the obstacle, a portion of the wave particle would extend out from the obstacle into open water, while a portion of the particle would interact directly with the obstacle. In the case of a wave particle partially interacting with an obstacle, if the center of the particle collided with the obstacle, then the entire wave particle would be reflected, thus leading to water reflecting that should instead simply move past the obstacle. In the case that the center of the particle did not collide, the entire particle would move past the obstacle, leading to portions of the wavefront passing through an obstacle that should have been reflected.

To help more realistically model wavefront behavior when interacting with obstacles, our method allows for wave particles to now have an arbitrary radius. Instead of a wave particle representing a cosine distribution of uniform radius around a center point, a wave particle's distribution is now scaled by two values, a particle radius and a wavefront radius. In the direction of the wave particles' propagation, the cosine distribution is scaled to the wavefront radius. In the direction perpendicular to the propagation, the particle is scaled to the particle radius. This allows particles of arbitrary radius to make up the same wavefront while maintaining a uniform wavefront depth.

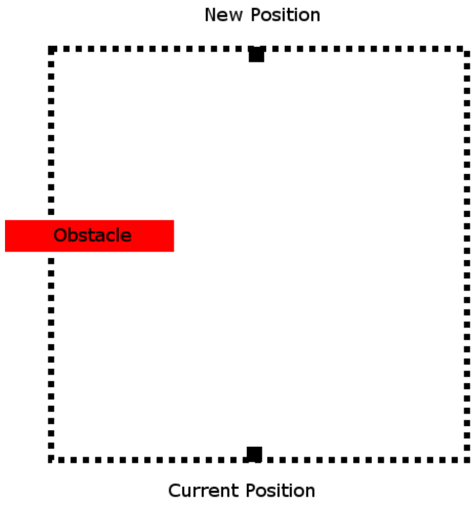

Figure 7. Particle Splitting - Step 1. A single particle's region of space covered in one time step. This region is checked for any obstacle end ponints.

Particle splitting works in the following manner. First, the region a wave particle is responsible for in the next timestep is calculated and tested against obstacle end points [Figure 7]. If an end point is detected in the region, the distance is found from this end point to each edge

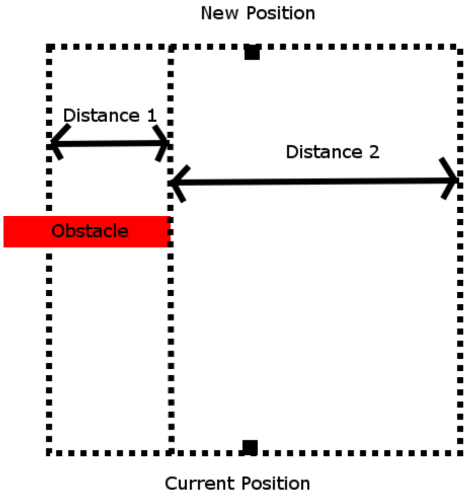

Figure 8. Particle Splitting - Step 2. The region from step 1 is divided into two new regions, each with a new distance that is not required to be equal.

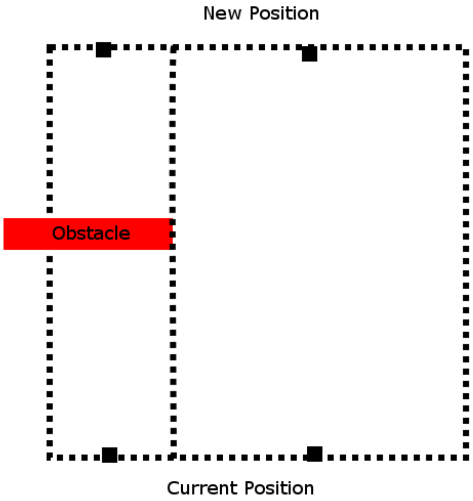

Figure 9. Particle Splitting - Step 3. The old particle is deleted and two new particles are generated, each responsible for one of the two regions calculated in step 2.

of the particle’s region not perpendicular to its direction of travel. This splits the region covered by the particle into two new regions, one on either side of the obstacle end point [Figure 8]. We then generate two new particles, each responsible for one of these two new regions of space [Figure 9]. Although similar to subdivision of particles, the motivation for generating new particles is very different. In subdivision, the new particles are necessary as one particle is no longer able to correctly represent the entire space it is responsible for. In particle splitting, the original particle is still able to fully represent its area in size. However, a single particle is no longer able to correctly represent the behavior of the water it represents, as a portion of the water should reflect and a portion should continue forward, this driving the need for two new particles, each representing a different behavior of the water. As the two new particles are taking the place of a particle that could fully represent its region of space, the new particles must be scaled to a maximum amplitude of the value of the original particle’s wave distribution at each new particle’s location. Without this, we would have the incorrect behavior of a single cosine distribution with one location with a maximum value of one splitting into two new cosine distributions with two locations with a value of one, and thus misrepresenting the region covered by the original particle.

Error Bound

As we are using two new cosine distributions to represent a single distribution, it is inevitable that there is error in the representation of the final wavefront. We calculate a bound for this error in the following manner. Using an original particle located at 0.0 along an axis, we calculate the maximum error when replacing the original particle with two new particles as the difference between the height of the original particle at x, and the summed heights of the two newly generated particles at x. For our tests, we maximize over x values ranging from -1.0 to 1.0 with a step size of 0.02. As this depends greatly on the split location, we run our analysis from a split position of -0.99 to 0.99, iterating with a precision of 0.01. We then iterate over the possible center values of each new particle, c_1 and c_2, as well as the possible max amplitudes of each particle, a_1 and a_2. For our analysis, we iterate these variables with a precision of 0.02. Finally, for each combination of variables for a single split position, we select the best values for c_1, c_2, a_1, and a_2 such that the maximum error is minimized. We then perform the same analysis but minimizing the average error rather than maximum error.

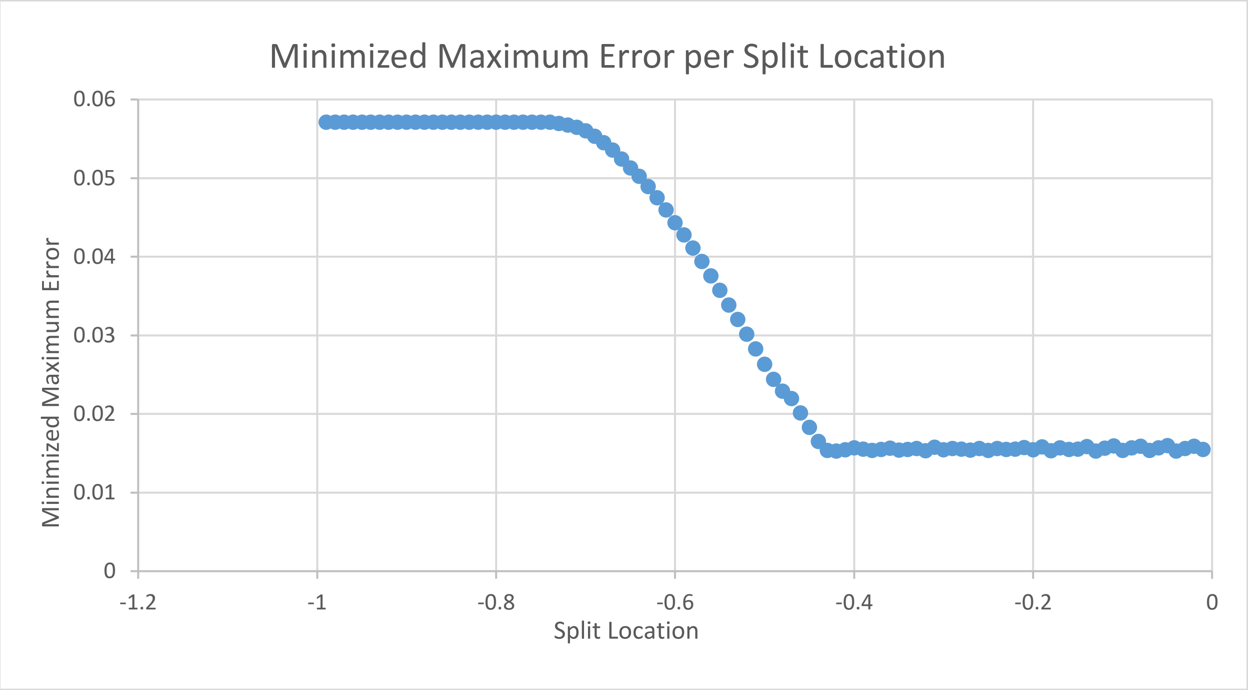

Figure 10. Minimized Maximum Error for every Split Location. The minimized maximum error for every tested split location is plotted. We see that the error holds steady at 5.7% from a split location of -0.99 to -0.7. Then, we see a downward trend in error from a split at -0.7 to -0.4. Finally, the error then holds steady at 0.5% from -0.4 to 0.0.

When minimizing the maximum error across the possible split locations in our analysis, the worst case maximum error was 5.7101%. [Figure 10] shows the behavior of the minimized maximum error as split location changes. We can see from the figure that as split location moves from -0.99 toward 0, the maximum error stays constant at about 5.7%. However, from split locations -0.7 to -0.4, we see a steady drop in the maximum error from 5.7% to 1.3%. The error then stays constant at about 1.3% from a split value of -0.4 to 0.0.

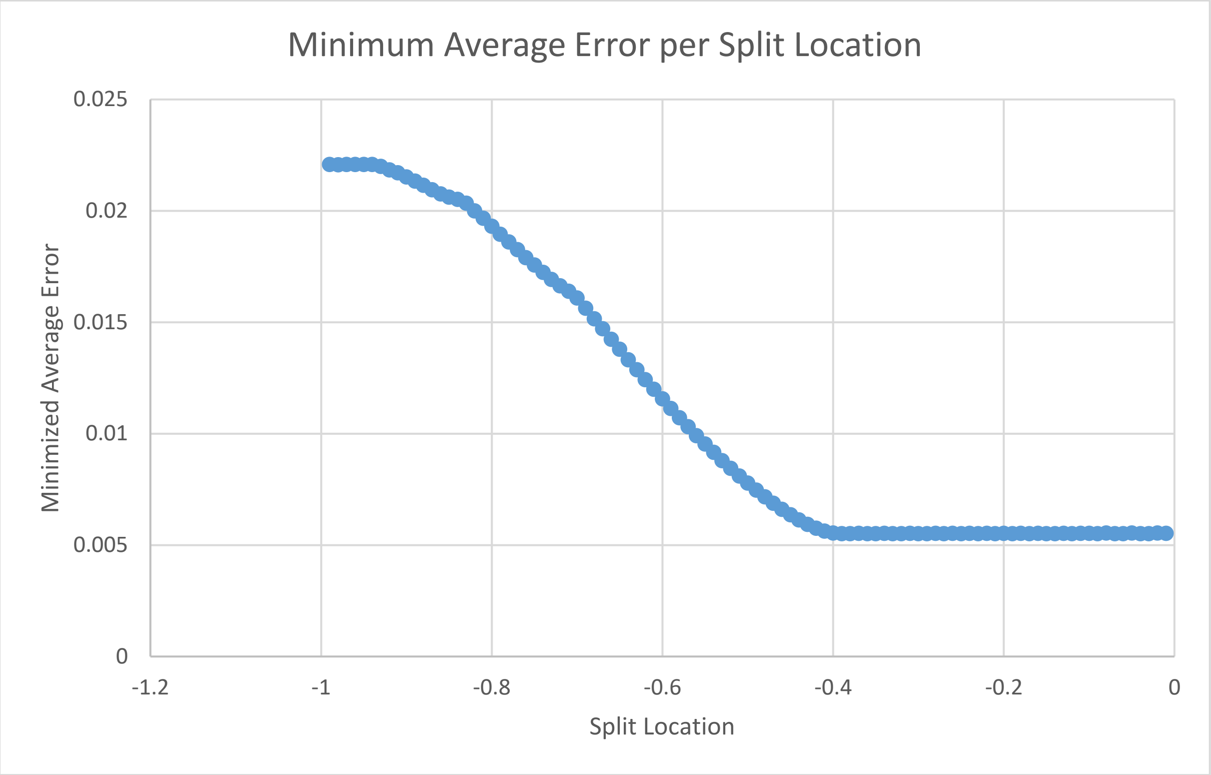

Figure 11. Minimum Average Error for every Split Location. The minimized average error for every tested split location is plotted. We see that there is a downward trend in error from a split at -0.99 to -0.4, then the error holds stead at 0.5% from -0.4 to 0.0.

We see similar behavior when minimizing the average error across the possible split locations. We found then when minimizing for average error, the worst case average error is 2.2083%. [Figure 11] shows the behavior of the minimized average error as split location changes. Unlike the minimized maximum error, the minimized average error does not really have a plateau of constant error values toward a split length of -1. Instead, the maximum average error drops steadily from a split location of -0.99 to about -0.4. However, like the minimized maximum error, the minimized average error holds steady at 0.5% error from a split length of -0.4 to 0.0.

Diffraction

The second behavior our method adds is the ability of wave particles to generate a diffraction effect when interacting with obstacles. When a wavefront passes by an obstacle, a bending effect known as diffraction appears. This bending effect extends from the wavefront as it passes an obstacle in a semi-circular shape, right up to the obstacle. In the original method, when wave particles passed by an obstacle, they simply continued and no diffraction pattern was generated. In our method, we simulate this diffraction.

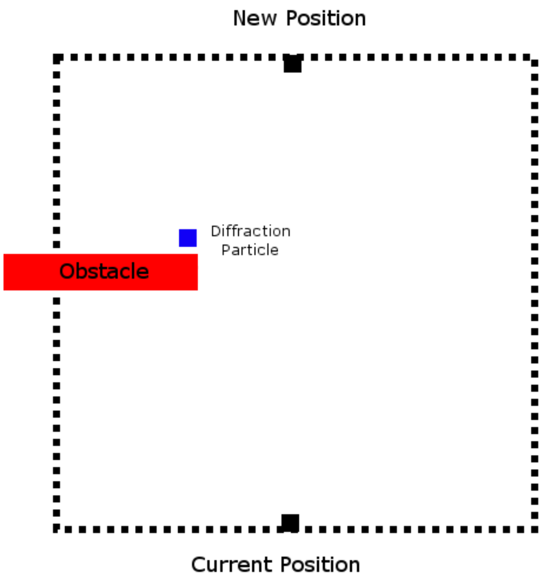

Figure 12. Diffraction - Step 1. Similar to splitting, the region a particle is responsible for is calculated. If an obstacle is detected, we generate a new diffraction particle.

We simulate diffraction in the following manner. First, for each particle, the region of space it is responsible for in the next timestep is calculated, just like for particle splitting [Figure 12]. Then, if an obstacle end point is detected in this region, a new diffraction particle is generated at the location of the obstacle endpoint, but slightly offset in the direction of the current wave particle’s travel.

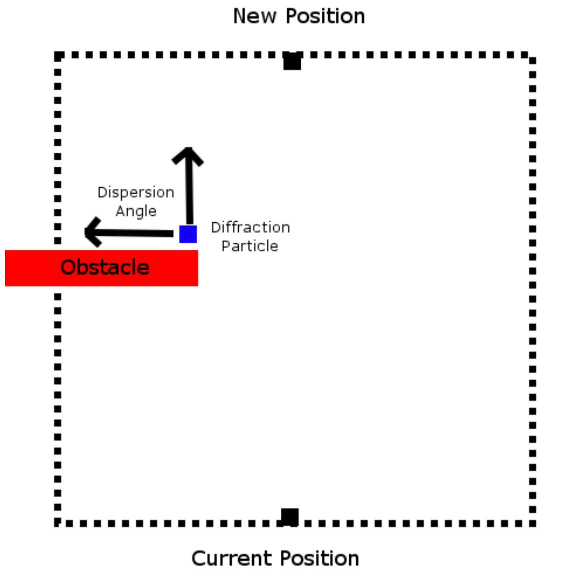

Figure 13. Diffraction - Step 2. The particle generated in step 1 is offset slightly from the obstacle in the direction of the original particles movement. In addition, the new particle's dispersion angle is set to that between the original particle's direction, and the obstacle's direction.

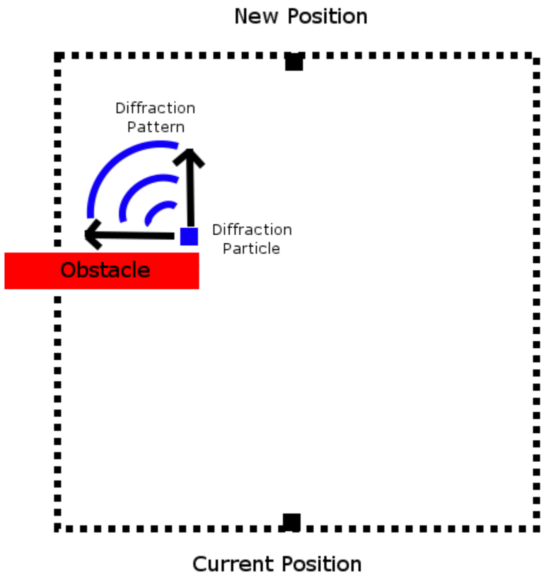

Figure 14. Diffraction - Step 3. Finally, this new particle is added to the simulation and begins to propagate and subdivide over time, leading to a modeled diffraction patter.

The reason for this offset is to avoid the case where this new diffraction particle would simply collide with the obstacle over and over, as it would if the new particle was placed at exactly the endpoint of the obstacle. The new diffraction particle is then set to have a dispersion angle of the angle between the original particle's direction of travel, and the direction from the endpoint in question to the other endpoint of the obstacle [Figure 13]. The motivation for this is that a full diffraction pattern is to be generated, extending from the edge of the original wavefront after the obstacle collision, right up to the obstacle itself. The diffraction particle, after being spawned, would rapidly subdivide into enough particles to correctly represent the space [Figure 14].

It is important to note, however, that our diffraction method assumes that there is a full diffraction pattern when a wavefront passes an obstacle. This is not always the case. For instance, when a wavefront passes through a slit made of two obstacles, the behavior of the diffraction pattern depends on the size of the slit and the wavelength of the wavefront. If the slit is equal to or smaller than the wavelength, a full diffraction effect results. This is the effect our implementation assumes to always be the case. However, if the size of the slit is larger than the wavelength, there is only a slight bending of the wavefront as it passes by the obstacle. Instead of extending completely to the obstacle, this bend only extends partially toward the obstacle, moving toward the obstacle as the slit size narrows. Our method does not model this behavior.

SECTION IV

EXPERIMENTATION AND RESULTS

Experiments were performed on a pc setup with a i7-4790k processor, as well as a Nvidia GTX 970 graphics card and 16 GB of ram. The code was written in C++ and used the DirectX 11 API.

Experimentation

To test our implementation of particle splitting, we compared our method with the original method in the same environment. The test scenario used involved generation of a parallel wave traveling toward an obstacle. A portion of this wave would collide and be reflected off the obstacle, while a portion would continue past the obstacle. This environment is set up such that there are two places where this behavior happens. We are focused on the behavior of the particles and generated wavefront right at the edges of the obstacle. We are looking to see if an incorrect amount of water is reflected off or continues to propagate past the object, or if there is correct behavior.

Similarly, to test our implementation of diffraction, we compared our method with the original method in the same environment. The test scenario used involved generation of a parallel wave traveling toward an obstacle. The obstacle forms a single sided barrier, and thus we expect to see a diffraction pattern extending from the wavefront that crosses past the barrier back to the barrier itself. For this test, we are focused primarily on the curved diffraction effect.

Results

First, we discuss the results of testing our method for correct particle splitting.

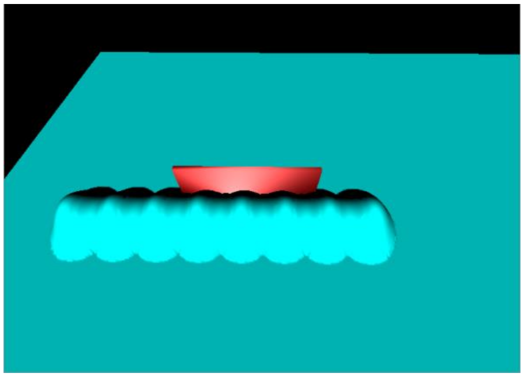

Figure 15a. Original Obstacle Collision.

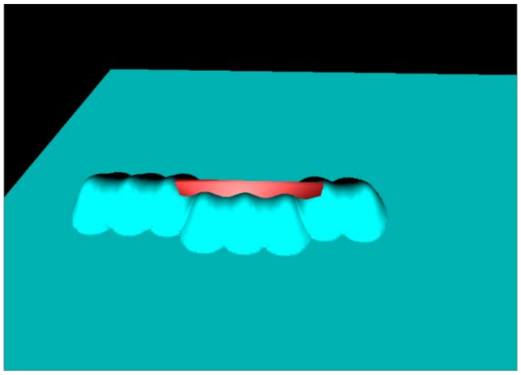

Figure 15b. Original Obstacle Collision

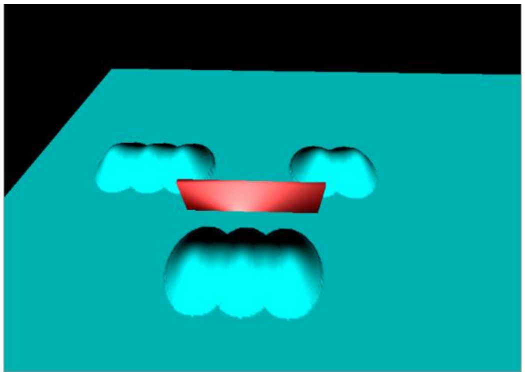

Figure 15c. Original Obstacle Collision. In these figures, we can see the wavefront collide with the obstacle. However, more water moves past the obstacle than should be allowed.

In the case of the original method, it is easy to see that there was unrealistic behavior as an incorrect amount of water was reflected when the wavefront interacted with the edge of the obstacle [Figure 15]. As the wave travels toward the obstacle, we expect for water exactly striking the obstacle to be reflected. Instead, we see that the entire wave particles at the edge of the obstacle, due to their centers not colliding with the obstacle, entirely pass through the obstacle in an unrealistic manner. The result is that less water was reflected by the obstacle than there should have been.

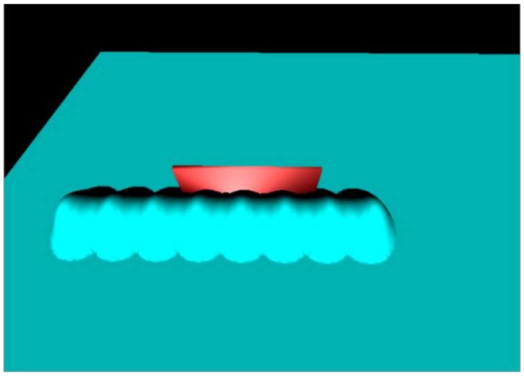

Figure 16a. Our Obstacle Collision.

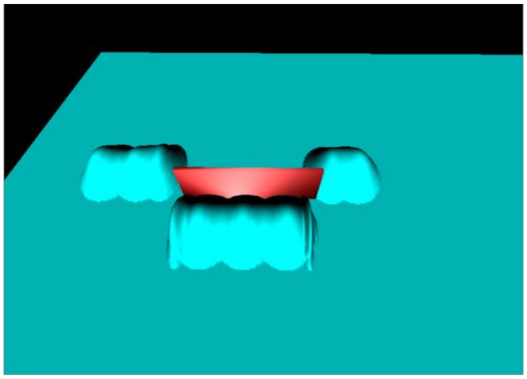

Figure 16b. Our Obstacle Collision.

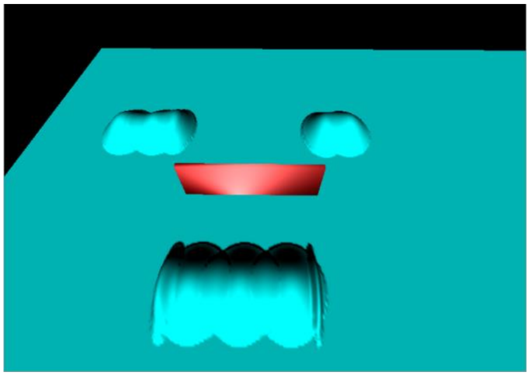

Figure 16c. Our Obstacle Collision. In these figures, we see the wavefront collide with an obstacle. However, unlike the original method, our method splits the particles on the obstacle boundary, providing a hard cut in the wave at the obstacle boundary.

We then compare this with our splitting method. Like the original method, the test of our method starts with the wavefront striking the obstacle. However, unlike the original method, our method causes the wave particles at the edge of the obstacles to split into particles of smaller radii [Figure 16]. The resulting particles then behave differently than the original particle. Instead of the entire particle's amount of water continuing past the obstacle, we have one new particle reflects, while the other particle continues past the obstacle. Together, these two particles represent the same amount of water in the one particle they came from. Thus, we see that our method correctly models the behavior of waves as they pass and obstacle.

Now, we discuss the results of testing our diffraction method.

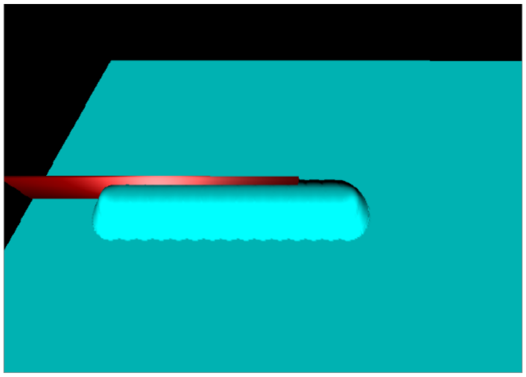

Figure 17a. Original Method (No diffraction)..

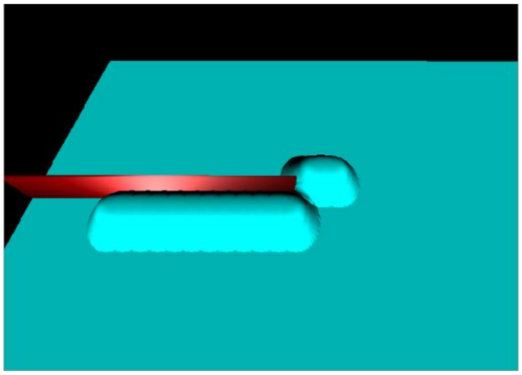

Figure 17b. Original Method (No diffraction).

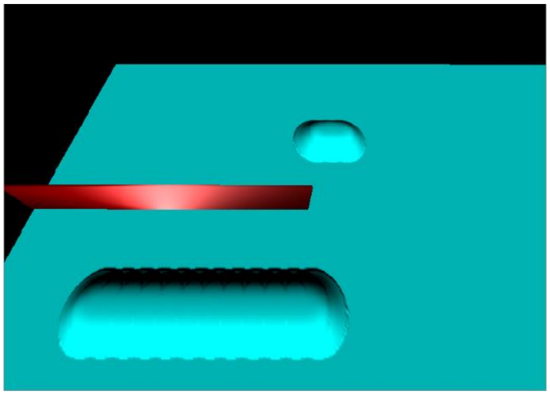

Figure 17c. Original Method (No diffraction). A portion of a wavefront passes an obstacle. Upon reaching the other side of the obstacle however, the wavefront simply continues. There is no attempt to model the diffraction of the wave.

In the case of the original method, we can see that as a wavefront strikes an obstacle, a section of the wavefront moves past the obstacle [Figure 17]. However, once the wave moves past the obstacle itself, the wavefront simply continues and no diffraction pattern is generated. This behavior looks unrealistic as we expect a curved diffraction pattern to be generated from the wavefront to the obstacle, extending outward from the obstacle.

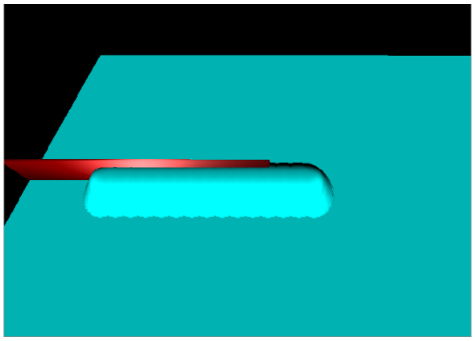

Figure 18a. Our Diffraction Method.

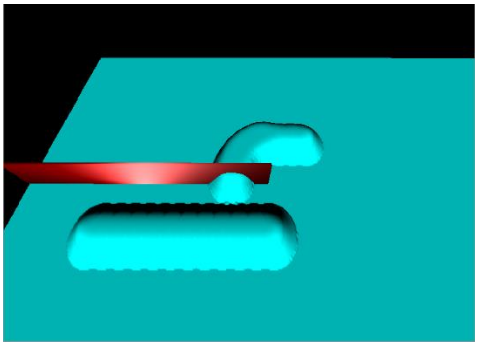

Figure 18b. Our Diffraction Method.

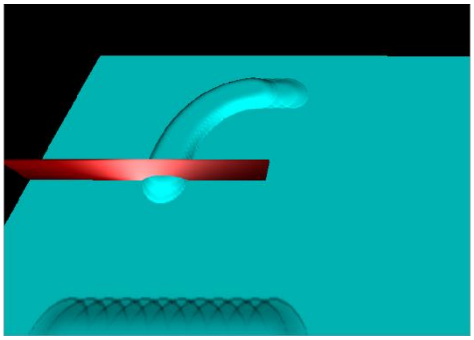

Figure 18c. Our Diffraction Method. A portion of a wavefront passes an obstacle. Upon reaching the other side of the obstacle, a diffraction pattern is generated that extends to the obstacle, outward from the obstacle's end point.

In the case of our diffraction method, we can see that the wavefront collides with the obstacle in the same way as the original method. In addition, a portion of this wavefront continues past the obstacle in the same way as the original [Figure 18]. However, our method correctly generates a diffraction pattern that extends from the wavefront passing by the obstacle, all the way to the obstacle itself. This is a correct diffraction pattern, modeled by a semi-circle that covers the entire region of space extending from the wave particle closest to the obstacle that moves past the obstacle, to the obstacle.

It is important to note two odd behaviors in the experiment. First, there is slight bleed-over of the particles through the obstacle in the narrow direction. This artifact is due to our use of infinitely narrow obstacles when rendering. In practical use cases, obstacles will be wider and thus this artifact will not appear. In addition, there is a height difference between the diffraction wavefront and the original wavefront. This is due to our method spawning multiple particles to represent the same diffraction wavefront as multiple particles require splitting at the edge of an obstacle. Thus all the particles sum to a greater height than desired. This is a limitation of our diffraction detection and is discussed more in future work.

SECTION V

CONCLUSION

We have introduced two extensions of the wave particle method to simulate more correct water surface wave behavior. One extension now allows for correct wave on obstacle collisions during which a wave particle collides with the end of an obstacle. The second extension simulates water wave diffraction when a particle passes by the edge of an obstacle. In doing so, we have shown that wave particles can successfully be extended to support more realistic interaction with an environment, primarily through more correct collisions with obstacles as well as modeling full diffraction patterns.

Future work

One main area we see for future work would be to separate the diffraction and particle splitting methods into different parts of the particle's update, similarly to how subdivision and collision are separate from each other. Currently, due to a particle's region being the trigger for both particle splitting and diffraction, detection of diffraction must take place before a particle at the edge of an obstacle is split. If not, the new particle's regions would not contain an obstacle endpoint, and thus diffraction would never occur. One direction we believe might be promising would be to change obstacle detection to allow colliding particles to push through the object and form diffraction particles. While this would not support every possible combination of obstacles, it would remove the dependency on diffraction tracking the particles region, while also supporting more complex obstacles than our method currently allows.

Another area of future work would involve adding support for slight bending diffraction effects, and thus removing the assumption of full diffraction during obstacle interactions. This extension would require a few new steps. First, the wavelength of a wavefront would need to be tracked. However, this is essentially the wavefront size parameter in the new particle splitting extension, so this should be trivial. The more complex feature that is required would be a better method of detecting obstacles, specifically the detection of regions between obstacles. As the bending diffraction effect is dependent on knowing the distance between two obstacles, in order to support this bending of wavefronts, the distance between obstacles must be detected. We are unsure if there is a trivial solution to this problem, at least one that stays true to the wave particle idea of discretizing the wave rather than the environment. In addition, more advanced obstacle diffraction for diffraction particle spawning could remove the error in generating multiple diffraction particles to represent a single diffraction effect. A better obstacle detection method would detect the main wave particle that is being split and only allow this main particle to generate a diffraction particle, correcting the problem of multiple diffraction waves stacking to form a slightly thicker wavefront seen in our experimentation.

A final area of future work would involve adding support for wave breaking. This extension would involve tracking the distance of the final water surface, and thus the wave particle plane, to the ground surface under the water. When this distance has reduced below a certain threshold, the wave would then break. In addition, the extended heightmap would need to be modified to allow the heightmap to join in two different points, as is the real behavior when a wave breaks. An approach similar to Chentanez and Müller’s hybrid approach would likely be a good starting point (Chentanez & Müller, 2011b), using extended wave particles that track the distance to the floor beneath the water to simulate large, open bodies of water when the wave isn’t breaking, then switching to use of a more detail specific simulation when and where the simulation enters a state where waves should break.

REFERENCES

Chen, J. X., & Lobo, N. da V. (1995). Toward Interactive-Rate Simulation of Fluids with Moving Obstacles Using Navier-Stokes Equations. *Graphical Models and Image Processing*, *57*(2), 107–116. https://doi.org/10.1006/gmip.1995.1012

Chentanez, N., & Muller, M. (2010). Real-time simulation of large bodies of water with small scale details. *Proceedings of the 2010 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, 197–206. https://doi.org/10.2312/SCA/SCA10/197-206

Chentanez, N., & Müller, M. (2011a). Real-time Eulerian water simulation using a restricted tall cell grid. *ACM Transactions on Graphics*, *30*(4), 1. https://doi.org/10.1145/2010324.1964977

Chentanez, N., & Müller, M. (2011b). Real-time Eulerian water simulation using a restricted tall cell grid. *ACM Transactions on Graphics*, *30*(4), 1. https://doi.org/10.1145/2010324.1964977

Foster, N., & Fedkiw, R. (2001). Practical animation of liquids. *Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques - SIGGRAPH ’01*, (August), 23–30. https://doi.org/10.1145/383259.383261

Foster, N., & Metaxas, D. (1997). Controlling fluid animation. *Proceedings Computer Graphics International*, 178–188. https://doi.org/10.1109/CGI.1997.601299

Fournier, A., & Reeves, W. T. (1986). A Simple Model of Ocean Waves. *SIGGRAPH 86 Proceedings of the 13th Annual Conference on Computer Graphics and Interactive Techniques*, *20*(4), 75–84. https://doi.org/10.1145/15922.15894

Irving, G., Guendelman, E., Losasso, F., & Fedkiw, R. (2006). Efficient simulation of large bodies of water by coupling two and three dimensional techniques. *ACM Transactions on Graphics*, *25*(3), 805. https://doi.org/10.1145/1141911.1141959

Mastin, G. A., Watterberg, P. A., & Mareda, J. F. (1987). Fourier Synthesis of Ocean Scenes. *IEEE Computer Graphics and Applications*, *7*(3), 16–23. https://doi.org/10.1109/MCG.1987.276961

Peachey, D. (1986). Modeling Waves and Surf. *ACM Siggraph Computer Graphics*, *20*(4), 65–74. https://doi.org/http://doi.acm.org/10.1145/15922.15893

Schachter, B. (1980). Long crested wave models. *Computer Graphics and Image Processing*, *12*(2), 187–201. https://doi.org/10.1016/0146-664X(80)90011-8

Stam, J. (1999). Stable Fluids. *Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques*, 121–128. https://doi.org/10.1145/311535.311548

Treuille, A., Lewis, A., & Popović, Z. (2006). Model reduction for real-time fluids. *ACM Transactions on Graphics*, *25*(3), 826. https://doi.org/10.1145/1141911.1141962

Ts’o, P. Y., & Barsky, B. A. (1987). MODELING AND RENDERING WAVES: WAVE-TRACING USING BETA-SPLINES AND REFLECTIVE AND REFRACTIVE TEXTURE MAPPING. In *ACM Transactions on Graphics* (Vol. 6, pp. 191–214). https://doi.org/10.1145/35068.35070

Yuksel, C., House, D. H., & Keyser, J. (2007). Wave particles. *ACM Transactions on Graphics, 26*(3), 99. https://doi.org/10.1145/1276377.1276501