# Methods of Simulation Fluid Motion in a Shallow Context in 3-Dimensions

Jacob Dominy
Computer Systems Lab 2009-2010

## Abstract

As computer graphics become more advanced and realistic, it becomes necessary to learn how to recreate real-life events in a virtual environment. The events that have proved most problematic in this regard are those that occur in nature. In this project I will investigate techniques to automate simple, shallow fluid motion found in everyday life.

## Background

Recreation of the motion of fluids has proved to be an enduring conundrum for graphic designers. This is because all computer graphics are based on the combined use of many 3-Dimensional objects. This makes modeling solids very simple, but very difficult when it comes to fluids. A deep understanding of physics is also required to recreate the motion of fluids as they are dictated by a large set of rules in nature. Because there are countless different situations and conditions that occur in nature, encompassing all of these possiblities has made it hard to code realistic fluid motion. Several different methods to approaching this problem have been researched in this field.

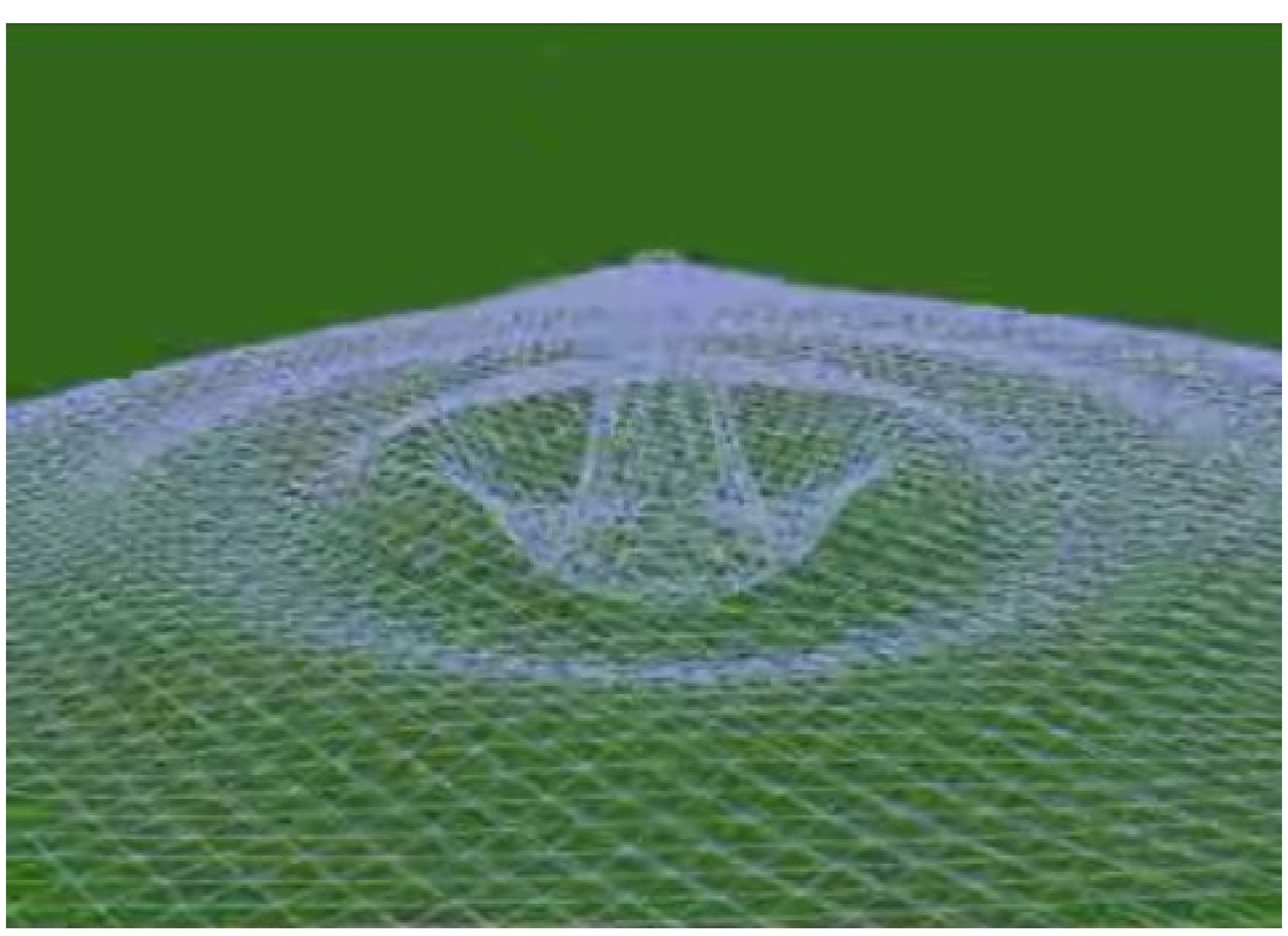

Fig. 1. An example program using a height field to simulate water ripples. Program uses a modified sine equation to create waves that look similar to fluids, but does not actually recreate its motion.
http://www.youtube.com/watch?v=TcVrEZ0i_u0

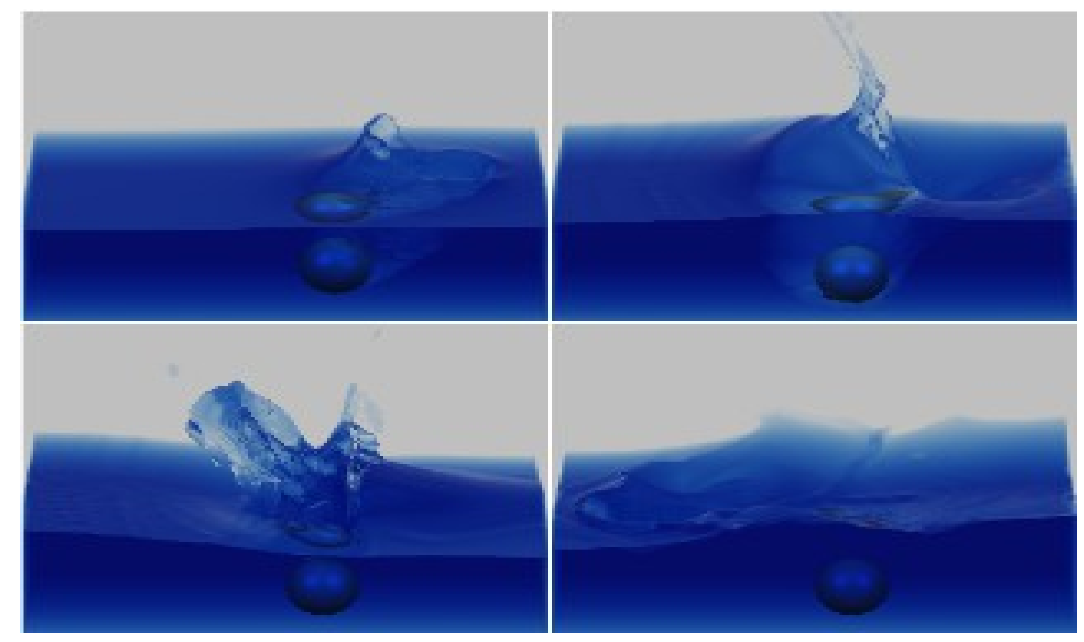

Fig. 2. Screenshots of a simulated ball splashing into water. Simulation uses a combination of a voxel field and and a particle system to create the splashing effect for a more realistic experience.

## Discussion

There are several different methods of representing fluids in a simulated environment. A body of water can be simulated using a particle system or a 3-Dimensional grid of cells. The program then will calculate the forces and velocities in all directions on every single particle or cell. However, for our purposes, this will most likely prove too complicated and too hard on computer resources. A more practical method, especially in a shallow body environment, is to use a hight field. This is a 2-D matrix that is then calculated for accelerations, velocities and the heights of each point are determined and then graphed. As for actual physics, the most common method for determining velocities of fluids is the Navier-Stokes equation.

$$\frac{\partial^2 s}{\partial t^2} = -g\left[\frac{d_{x-1} + d_x}{2(\Delta x)^2}\right](s_x - s_{x-1}) + g\left[\frac{d_x + d_{x+1}}{2(\Delta x)^2}\right](s_{x+1} - s_x)$$

Fig. 3. A simplification and discretization of the Navier-Stokes equation that evaluates for the vertical acceleration of any point on the height field.

This is not all we need however. In the first generation of my program, I only used this equation and derived the velocity and displacement from this, however, after further research, we do not need either of those things. We need to continue to apply this to our program. After using this equation on every point, we can find the height of any point by using the two heights previous to it.

## Results and Conclusions

The expected results of this project will be the successful application of concepts of fluid dynamics to computer graphics in the context of a small, standing, shallow body of water. With these methods applied, the expected output will be realistic ripples and waves on the surface of the body. As of now, the program correctly displays the height field in 3D, and is able to rotate, scale, and pan with light effects and a button to reset the user's view. The Navier-Stokes equation is coded correctly, and the implementation and application is fixed more, and is closer to the way it is supposed to be. The program, when run, displays a wave in moving one direction across the height field. However, the wave only lasts a few iterations before the heights become too large to be realistic. This is a problem that will be easily fixed as the problem is relatively simple.