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

Jacob Dominy

January 28, 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.

**Keywords:** Computational Fluid Dynamics, Computer Graphics, OpenGL, Navier-Stokes, Saint Venant

# 1 Introduction

# 2 Introduction Research

Computer graphics have found many uses nowadays. It's used for animated movies, video games, and simulation software. As technology has progressed over the last few years, however, the quality of this graphic design has been increased dramatically. Photorealistic animations are now commonplace. However, it is no longer enough to merely look realistic, they must now act realistic as well. 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. The Navier-Stokes equations, first proposed in 1822 provide a way of understanding motion in incompressible fluids. The Saint Venant equations are also used, and these are based on the Navier-Stokes equations but applied in a context that does not compensate for depth.

## 2.1 Objectives

There has been a great deal of research into the field of fluid dynamics and the rules that govern fluids. Our objective is to determine how these rules can be applied in a simulated environment. Because this is a very broad field of study, this research will focus only on liquid fluids in small, standing, and shallow contexts. This way we can isolate only a few variables to be concerned with. In doing this, we can ignore the factors of flow and currents, like those found in rivers and oceans, making the scope of the project more specified.

# 3 Goal

The goal of this project is to create a program that can model the motion, such as ripples and waves, of a small, shallow body of water when disturbed in three dimensions. It will allow for user input to dictate the nature and starting place of the motion, as well as for the changing of certain variables that can change how the fluid operates. The program will allow for the view of the fluid to be changed.

### 3.1 Design Criteria

There will be several elements that I will incorporate into my project. The first will obviously be the window displaying the current 3 Dimensional model. I will also incorporate mouse and keyboard inputs to controls things such as the view and orientation of the model and the zooming in and out of it. In addition, several menus will be used to control key variables in the project. Light effects will be added and reflection and refraction of that light is planned.

### 3.2 Procedure and Scope

In order to understand the physics behind this project, I will continue to do in depth research into what has already been learned by others about the subject and I will also discuss it with the resident physics teachers at Thomas Jefferson. I will most likely also have to do my own hands-on research on the subject of waves. I will then, from this research, discern the best way to model this motion in the form of mathematical equations and use them to manipulate the 3-D models in my program. Because of the complexity of the general subject area, this project will only focus itself on shallow contexts and not try to recreate currents or large bodies of water.

### 3.3 Expected Results

The goal of this project is a very ambitious one, so while I do have high hopes for this project, I am uncertain of it's feasibility. My measure of success will be if I can have a running simulation program that accepts all appropriate inputs and shows a basic system of using these inputs in the changing of its modeling of fluids. I will most definitely need all the time I can get for working on this project to complete it. If a student next year would like to continue in the same direction as I, I would suggest the modeling of currents such as those found in rivers and oceans.

### 3.4 Related Work

The field of fluid dynamics is not limited to only liquids, and others have performed research regarding the flow of air and other gases [3]. This sort of project has a number of uses in areas such as aerodynamics and wind tunnel simulations.

# 4 Problem to Solve

## 4.1 Fluid Representation

Before even thinking about any fluid dynamics equations, we must first consider how to represent the fluid in a computer generated environment. In the case of liquids, one of the most popular options is to use a particle system, in which each particle represents one water molecule and acts independently of the others. Another, similar method is to create a large 3-Dimensional grid, called a voxel field, and have each cell act as a small section of the body of water. These methods are very intensive on computer resources, as they require calculations for a very large number of different objects. These will not be used in this project, as they are far too complicated for our needs. Instead we will use what is called a hight field. A height field is a 2-Dimensional matrix of points that represent points on the surface of our liquid body. This field is displayed on the XZ-plane to the viewer, and each point in the grid has a height value, which is then reflected on the Y-axis. This method is the most reasonable for our purposes, as our shallow body of water is not concerned with the flow of water below the surface.

## 4.2 Computational Fluid Dynamics

After deciding on how to represent liquid in 3-D space, we must then figure out how to make it move. The most common method is the use of the Navier-Stokes equations, discovered in the 19th century. These equations take into account variables such as pressure, gravity, viscosity, and density to describe velocity vectors for every point in a body of liquid. This equation is applied in many different ways depending on the context and often combined with other mathematical ideas. Because our situation only concerns a shallow body of water and thus we only care about its motion on the surface, the Navier-Stokes equation can be simplified to disregard the motion underneath the surface of the body of water. Our program will use this equation and apply it to every point in the height matrix which will in turn resolve that point's height and display it accordingly.

## 4.3 Assumptions

To properly derive an equation that we can properly translate into code for our program, there are several assumptions that we must first make. The first is that the the only thing used is a height map to represent the surface of the water. This means that although we do not concern ourselves with any water underneath the surface, that also means that the motion on the surface is severely limited. It cannot splash; for that, we must use a particle system. The second assumption is that we can ignore the vertical velocity of the water points. The third and final assumption is that any given point represents a column of water under it, and that the horizontal velocity at any point is constant through that column.

## 4.4 Implementation

In order to properly implement the equation into the program, after we have made our assumptions, we are left with a form of the Navier-Stokes equation as shown:

$$\frac{\partial u}{\partial t} + g\frac{\partial s}{\partial x} = 0$$

$$\frac{\partial s}{\partial t} + \frac{\partial}{\partial x}(ud) = 0$$

Figure 1: Version of the Navier-Stokes equation using Newton's Second Law and the Law of Conservation of Mass[5].

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

Figure 2: After calculations and integration, we are left with a single equation that solves for the vertical acceleration of a single point on the height field[5]. Represents the vertical acceleration for any point in the height field.

# 5 Results

The current results are far from the expected ones. So far, the Navier-Stokes equation is implemented, but the values it returns are very far from being correct. The variables cannot be changed yet, as I am leaving that for after the implementation is corrected.

# References


[1] N. Foster and R. Fedkiw, "Practical Animation of Liquids", Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques, pp. 23-30, August 2001.

[2] Janghee Kim, Deukhyun Cha, Byungjoon Chang, Bonki Koo, Insung Ihm, "Practical animation of turbulent splashing water", Proceedings of the 2006 ACM SIGGRAPH/Eurographics symposium on Computer animation, September 02-04, 2006, Vienna, Austria.

[3] Bryan M. Klingner, Bryan E. Feldman, Nuttapong Chentanez, James F. O'Brien, "Fluid animation with dynamic meshes", ACM SIGGRAPH 2006 Papers, July 30-August 03, 2006, Boston, Massachusetts.

[4] Michael Kass, Gavin Miller, "Rapid, stable fluid dynamics for computer graphics", Proceedings of the 17th annual conference on Computer graphics and interactive techniques, pp.49-57, September 1990, Dallas, TX, USA.

[5] Jeff Lander, "Research on the Rhine: Reflections on Water Simulation", Game Developer, pp. 1-4, January 2000.