Simulating 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 investigated 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 possibilities has made it hard to code realistic fluid motion. Several different methods to approaching this problem have been researched in this field.

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 stores the heights of each point and then graphed. As for actual physics, the most common method for determining velocities of fluids is the Navier-Stokes equation, but this project used a variation on the shallow water equations for added simplicity.

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

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. This equation is not linear



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. (N. Foster and R. Fedkiw, "Practical Animation of Liquids")



because the depth "d" depends on the height "h", and this inevitably causes problems. Therefore, we need to create a linear matrix with similar equations.

$$Ah_i(n) = 2h_i(n-1) - h_i(n-2)$$

Figs. 3/4. Shows the final useful equation and the tridiagonal linear matrix. When solved, the equation produces a matrix containing the heights of every point in the one-directional wave. (Michael Kass, Gavin Miller, "Rapid, stable fluid dynamics for computer graphics")

This is the process for a wave in one direction. To create a two directional wave, I merely repeated the process for both the X and Y axes.

A =



Results and Conclusions

The expected results of this project were 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 was realistic ripples and waves on the surface of the fluid. In it's final version, the program correctly displays the height field in 3D, and is able to rotate, scale, and pan with light effects. The shallow water equations is coded correctly, and the implementation and application are very nearly correct. The program, when run, is capable of displaying a wave moving in either one or two directions across the height field. However, because the implementation is not quite perfect, the body of water slowly loses volume from it's starting point, but this does not affect the overall quality of the look of the wave generated.



 $\cdot \cdot \cdot e_{n-3} \quad f_{n-3}$

 $f_{n-3} e_{n-2} f_{n-2}$ $f_{n-2} e_{n-1}$