Fluid Simulation with Bubbles

CSE 528: Computer Graphics
&
CSE 530: Geometric Foundations

Project Report

Apurva Kumar
~109751733~
Stony Brook University

Dec 4th 2014
# CONTENTS

Acknowledgements .................................................................................................................. iii

Overview .................................................................................................................................. 1

Introduction ............................................................................................................................. 2

The Equations of Fluids .......................................................................................................... 2
  Eulerian Approach ................................................................................................................. 2
  MAC Grid ............................................................................................................................. 3
  Simulation Loop .................................................................................................................... 3

Level Sets ............................................................................................................................... 5

Bubbles ..................................................................................................................................... 5

Conclusion & Future Work ...................................................................................................... 7

References ............................................................................................................................... 8
I am a complete novice to physics based fluid simulations. This project would not have been possible without the countless resources and open-source material available on the internet.

I would specifically like to point out that the 2007 SIGGRAPH Fluid Simulation course notes served as the basis of the necessary theory for fluid simulations, from which I further built upon by referring to numerous research papers and tutorials on the internet.

I would also like to thank Prof. Hong Qin for giving me the opportunity to pursue this project on fluid simulation. As an avid gamer and movie enthusiast, I was always fascinated by special effects and graphics animations. Through the courses offered on Graphics and Geometric Foundations, it gave me the initial push to delve deep into the field of Fluid Simulations and develop a basic working prototype.
OVERVIEW

The goal of this project was to implement an Eulerian grid based Fluid Simulator using level sets. The second part of the project was to come up with a framework for integrating bubble generation and dynamics in this Eulerian grid.

This report documents the work done over the Fall 2014 semester at Stony Brook University (SBU), as part of the project for the Graphics and Geometric Foundations courses. This report shall give a brief introduction into fluid simulations and the equations & concepts implemented as part of the project.

Apart from the coding aspect, the project involved an extensive survey of papers and theory in the field of fluid simulation & bubble dynamics.
INTRODUCTION

Close-up scenes of fluid phenomena such as stormy oceans, curly rising smokes and droplet splashes are amongst the most spectacular visual effects both in the real life and in the special effects industry. Photographers, movie makers and game developers all try their best to catch these moments of beauty. It is obvious that the realistic fluid animation is getting more and more demanding as people have higher and higher requirements on the visual effects of movies and games. However, the extreme complexity of fluid dynamics renders it impossible for the artists to animate fluid effects frame by frame. Thus, physically-based methods are now becoming the widely used techniques for generating realistic fluid animations.

Physically-based methods model the dynamics of fluids by solving the governing equations. Although the Navier-Stokes equations are proposed hundreds years ago to depict the fluid phenomena, the general closed form solutions remain undiscovered. With the development of the computer technology, various numerical methods are applied in approximating the Navier-Stokes solutions. Computer graphics mainly focuses on generating plausible visual effects and usually trade accuracy for speed. In addition to modeling the dynamics of fluids, rendering is another important issue in fluid animation. Various fluid phenomena demonstrate disparate visual effects. Thus, different rendering methods are adopted according to the categories and representations of fluids.

THE EQUATIONS OF FLUIDS

The fluid flow simulation is governed by the famous incompressible Navier-Stokes equations, a set of partial different equations that are supposed to hold throughout the fluid. They are usually written as:

\[
\frac{\partial u}{\partial t} + u \cdot \nabla u + \frac{1}{\rho} \nabla p = f + \nu \nabla \cdot \nabla u
\]

\[
\nabla \cdot u = 0
\]

where \(u\) is the velocity, \(\nu\) is the viscosity, \(\rho\) is the density, \(p\) is the pressure and \(g\) is the body force. In the project, I have considered a simple incompressible fluid hence the viscosity term can be omitted. Also the external force comprises of only gravity.

Eulerian Approach

The Eulerian approach looks at fixed points in space and sees how the fluid quantities measured at those points change with time. The above equations are discretized with the grids. The finite difference methods are used to solve the equations numerically.
MAC Grid

The most popular way is to store the scalars, such as pressures, level set values and temperatures at the center of each grid and to store the vectors, such as velocities at the faces of each grid cell. This staggered configuration of MAC grid benefits from its unbiased and second order accurate central difference scheme.

![2D MAC grid](image)

**Fig 1:** 2D MAC grid

Simulation Loop

Operator splitting techniques are used to break the Navier-Stokes equation into separate terms for advection, external force and pressure. Each term is then solved separately and integrated with the next to simulate the fluid.

**Advection**

\[ \frac{Du}{Dt} = 0 \]

The velocity is advected through the velocity field for a time interval \( \Delta t \) based on the pressure values. Using the standard bilinear/trilinear interpolation in semi-Lagrangian advection quickly smears out sharp features in the fields. Hence, a Catmull-Rom spline interpolation has been used to estimate values at a point \( x \) between \( x_i \) and \( x_{i+1} \).

**Body Forces**

\[ \frac{\partial u}{\partial t} = g \]

In this step, the velocity of every grid cell is updated by adding the constant gravitational force at every time step.
Projection (pressure/incompressibility)

\[
\frac{\partial u}{\partial t} + \frac{1}{\rho} \Delta p = 0 \quad \text{s.t.} \quad \nabla \cdot u = 0
\]

In this step, pressure quantities are calculated at each cell while enforcing the incompressibility condition. The above equations on solving comes out to:

\[
\frac{\Delta t}{\rho} \left( 4p_{i,j} - p_{i+1,j} - p_{i,j+1} + p_{i-1,j} - p_{i,j-1} \right) = - \left( \frac{u_{i+1/2,j} - u_{i-1/2,j}}{\Delta x} + \frac{v_{i,j+1/2} - v_{i,j-1/2}}{\Delta x} \right)
\]

We thus have a large system of linear equations for the unknown pressure values. It can be represented as a large coefficient matrix, A, times a vector consisting of all pressure unknowns, p, equal to a vector consisting of the divergences in each fluid grid cell, d.

\[ Ap = d \]

This system is solved by implementing a Modified Incomplete Cholesky Conjugate Gradient (MICCG) using a preconditioner.

Boundary Conditions

Boundary conditions need to be enforced to ensure that the fluid stays within the predefined grid container. If we have a particle trajectory which strayed outside the fluid boundaries due to numerical error, we find the closest point that is on the boundary of the fluid region, and interpolate the quantity from the fluid values stored on the grid near there.

**Fig 2: Simulation Loop**
**LEVEL SETS**

A “level set” is simply an implicit surface whose function $\varphi$ is represented with values sampled on a grid. The interior of the water is the set where $\varphi(x) < 0$ and the exterior (air) is the set where $\varphi(x) > 0$. Level sets are attractive for water simulation since they can very easily model smooth surfaces and further easily provide information such as whether a point $x$ is inside the fluid by interpolating $\varphi$ at $x$ from nearby grid points and check its sign.

Given the level set signed distance at each grid center, we can compute the surface of the water by evaluating $\varphi(x) = 0$. This is done by implementing a marching squares approach so as to be able to draw a smooth liquid interface.

At every time step, we now have to advect the level set. This is done by solving:

$$\frac{D\varphi}{Dt} = 0$$

**BUBBLES**

In real fluids, we often observe bubbles rising and floating on the surface. Bubble cluster has a complex but characteristic structure, which can change both in geometry and in topology due to the surrounding flow and film rupture. This results in a very interesting and impressive visual impact, and also is a great challenge for computer graphics to model the structure and dynamics of bubbles. Many researchers in computer graphics carried out studies to achieve a realistic dynamic bubble effect, but a general model that can handle the full effect of dynamic bubbles is not yet available.

In my project, to simplify matters, bubbles are considered as incompressible, spherical particles in the Eulerian grid.

**Distance to Surface ($\varphi$)**

The bubbles formed and propagated can classified based on the distance to the surface as:

- Air ($\varphi > r$)
- Surface ($-r \leq \varphi \leq r$)
- Liquid ($\varphi < r$)

**Drag & Buoyancy**

Each bubble experiences a drag force based on the velocity fields in the position on the grid. Every bubble also experiences an upward buoyant force $f_{\text{buoyancy}}$ proportional to its volume.

**Merging & Bursting**

Bubbles which come close to each other combine to form bigger bubbles. If a bubble grows too big, it bursts immediately. Bubbles which escape the liquid ie. Air bubbles are instantly removed. Bubbles on the surface last for a stipulated lifetime before bursting. Bubbles within the liquid remain until they get converted to surface or air bubbles.
**Fig 3:** Bubbles simulated in my program from two sources in (a) calm water, (b) moving water

**Fig 4:** Bubble simulation in my program showing the grid points and the velocity field within the fluid
CONCLUSION & FUTURE WORK

I was successful in simulating Eulerian grid based water with simple bubble dynamics in 2D. The original plan was to implement the framework in 3D. Migrating my current code to a third dimension as far as generating vector fields on the grid is concerned should not be a big issue. However, due to lack of time & technical resources and the tough nature of coding level sets for fluid visualization in 3D, I was unable to do so. Considering the limited timeframe of a semester, starting from absolutely no knowledge of fluid simulation to understand and develop a working prototype in 2D, I am pretty satisfied.

Possible future extensions include extending the system to 3D and including more complex physics such as: incompressible flow, bubble deformation and complex foam formation.
REFERENCES


+ Numerous websites and tutorials which helped understand the mathematical concepts involved and approaches towards coding them.