2D Fluid Simulator

Abdullah Alrashdan
aalrashdan@berkeley.edu
University of California, Berkeley
Berkeley, California, USA

Ishan Amin
ishanthewizard@berkeley.edu
University of California, Berkeley
Berkeley, California, USA

Kaitlyn Chen
kaitlynchen@berkeley.edu
University of California, Berkeley
Berkeley, California, USA

Zehan Ma
zehanma@berkeley.edu
University of California, Berkeley
Berkeley, California, USA

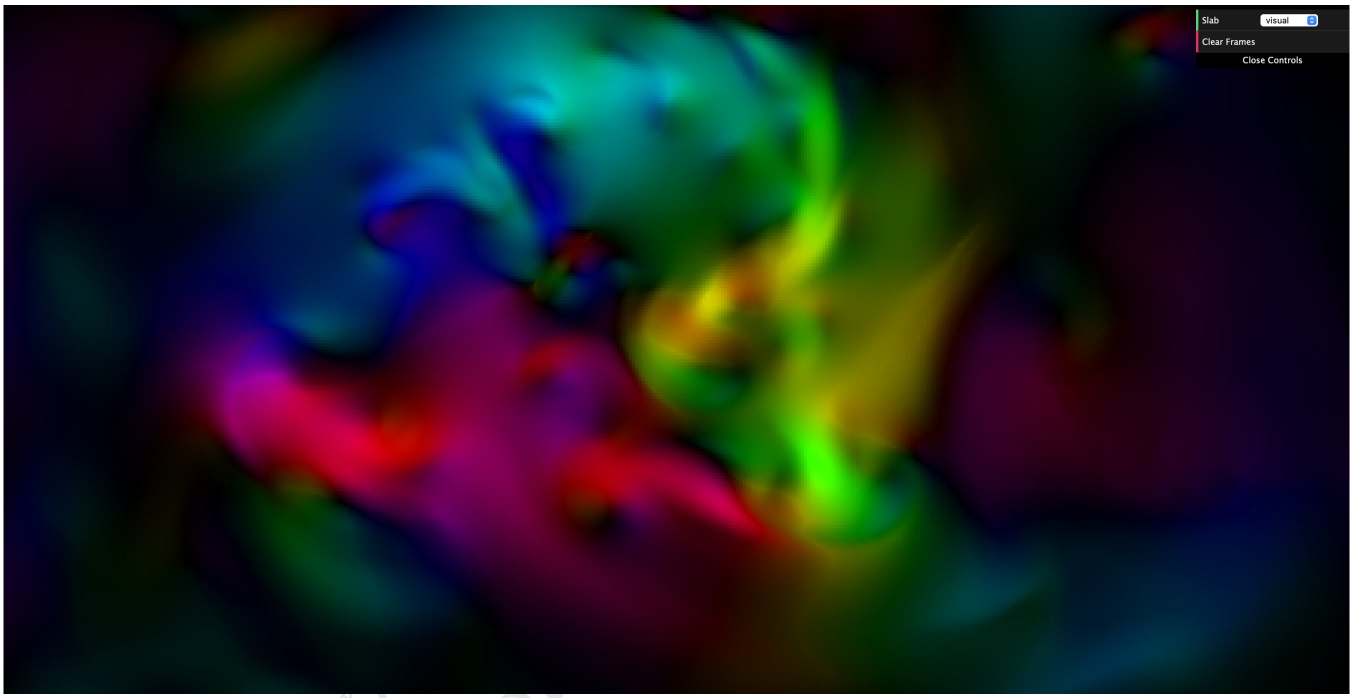

Figure 1: 2D Fluid Simulator UI

Abstract

In this project, our aim is to build an interactive rapid 2D fluid simulator using computational fluid dynamics. The fluid simulator is able to simulate a given fluid's velocity, pressure, temperature, and buoyancy. The fluid can be anything from gaseous smoke, air, to water or honey. The interactive interface let's the user toggle and initiate the fluid simulation by just clicking on the 2D grid domain. The fluid is simulated using appealing graphics such as colorful velocity vector fields and pressure gradients to illustrate a realistic physics phenomena. In addition, the user is able to create arbitrary visual obstacles with a press of a button to visualize how the fluid interacts with them.

Keywords

Fluid, Computer Graphics, Simulation.

1 Introduction

Fluid Simulation software is a critical tool that designers, engineers and researchers use to study the behavior of fluids. Simulating a fluid to understand the effects of it when interacting with mediums, structures and flowability became a fundamental practice in manufacturing cars, airplanes or even building structures. This tool allows engineers to evaluate the fluid physics and effects prior to developing hardware, which can substantially minimize costs and accelerate production. However, such a tool can be tricky to develop given its computational cost and energy consumption. For many years, physicists and engineers developed multiple mathematical approaches to simulate realistic behavior of fluids. Each mathematical approach makes certain assumptions that aims to reduce the computational expense while not compromising the integrity of the mathematical solver.

1.1 Problem Statement

Current Fluid Simulation software lacks the flexibility and the rapidness of simulating fluids while maintaining a realistic visualization. In addition, many commercial available software requires costly licensing purchase and occupies massive memory when installed.

1.2 Goals and Objectives

Develop a rapid 2D fluid simulator that allows ease of visualizing many fluids characteristics such as velocity, pressure, temperature, and buoyancy. In addition, a user is able to place obstacles while simulating the fluid to visualize realsitic interactions between the fluid and placed obstacles. The fluid simulator user interface should be interactive and doesn't require installation on a computer.

2 Technical Overview

In order to simulate the fluid, a mathematical approach must be adopted and algorithmically developed. We assume that our fluid is incompressible, and homogenous. An incompressible fluid is one in which the volume of any portion remains unchanged over time. A homogeneous fluid has a density, , that is uniform across all spatial locations. Together, these conditions imply that is constant both in space and in time. Although idealized, these assumptions are widely used in fluid dynamics and still provide an excellent approximation for many real fluids, such as water and air.

Our simulation uses a uniform Cartesian grid with spatial coordinates x = (x,y) and time t. We describe the fluid by its velocity field u(x,t) and a scalar pressure field p(x,t), each depending on space and time. Given u(x,0) and p(x,0), the fluid's evolution is determined by the incompressible Navier–Stokes equations:

2.1 Mathematical Approach

2.1.1 *Navier-Stokes.*

$$\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla)\mathbf{u} = -\frac{1}{\rho}, \nabla p + \nu, \nabla^2 \mathbf{u} + \mathbf{F} \tag{1}$$

$$\nabla \cdot \mathbf{u} = 0 \tag{2}$$

$$\nabla^2 \phi_{i,j} = \frac{\phi_{i+1,j} + \phi_{i-1,j} + \phi_{i,j+1} + \phi_{i,j-1} - 4, \phi_{i,j}}{\Delta x^2} \tag{3}$$

$$(\mathcal{P} \circ \mathcal{F} \circ \mathcal{D} \circ \mathcal{A})(\mathbf{u}(t))$$

$$q(\mathbf{x}, t + \Delta t) = q(\mathbf{x} - \mathbf{u}(\mathbf{x}, t), \Delta t, t) \tag{4}$$

$$\frac{\partial q}{\partial t} = \nu, \nabla^2 q \tag{5}$$

$$(\mathbf{I} - \nu, \Delta t, \nabla^2), q(t + \Delta t) = q(t) \tag{6}$$

$$x_{i,j}^{(k+1)} = \frac{b_{i,j} + \beta(x_{i+1,j}^{(k)} + x_{i-1,j}^{(k)} + x_{i,j+1}^{(k)} + x_{i,j-1}^{(k)})}{\alpha} \tag{7}$$

2.1.2 *Helmholtz-Hodge Decomposition.*

$$\mathbf{w} = \mathbf{u} + \nabla p \tag{8}$$

$$\mathbf{u} = \mathbf{w} - \nabla p \tag{9}$$

$$\nabla \cdot \mathbf{w} = \nabla \cdot \mathbf{u} + \nabla^2 p \tag{10}$$

$$\nabla^2 p = \nabla \cdot \mathbf{w} \tag{11}$$

2.1.3 *Advection.*

2.1.4 *Viscous Diffusion.*

2.2 Algorithm Development

2.2.1 *Implementation.*

- `Fragment Programs:`
- `Slabs:`
- `Shaders:`

3 Simulations

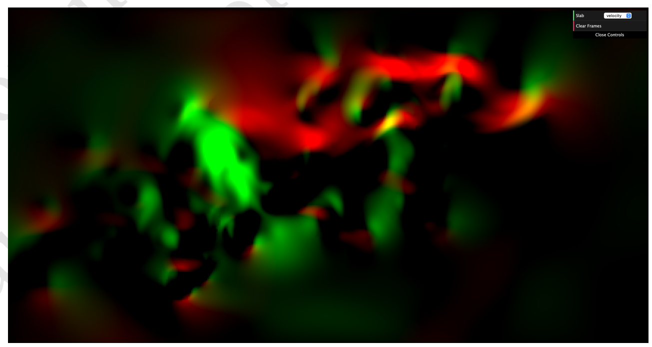

Figure 2: Velocity Simulations

3.0.1 *Velocity Simulations.*

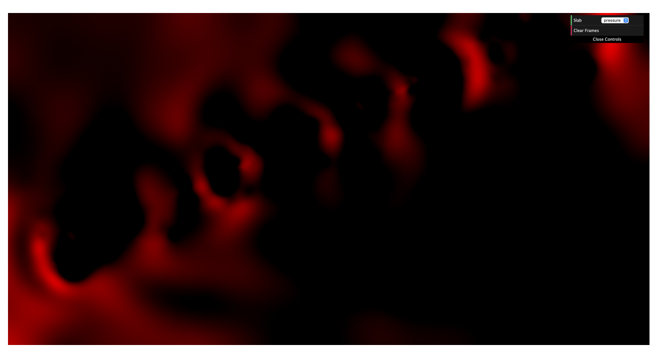

Figure 3: Pressure Simulations

3.0.2 *Pressure Simulations.*

3.0.3 *Temperature Simulations.*

3.0.4 Buoyancy Simulations.

3.0.5 Selected Scinarios and cases.

4 Discussion

5 Conclusion

5.0.1 Summary.

5.0.2 Lessons Learned.

6 Future Work

References