LAB REPORT: LAB 6

TNM079, MODELING AND ANIMATION

Iris Kotsinas
iriko934@student.liu.se

Monday 13th September, 2021

Abstract

This is the sixth and final lab in a series of labs in the course TNM079 Modeling and Animation at Linköping University. The report covers the topic of simple fluid simulation systems using the Navier-Stokes equations. The use of simulated smoke, water and fire has become common in modern special effects and is widely implemented in computer graphics.

1 Introduction

This lab studied how to solve the Navier-Stokes equations using Stable Fluids approach. This lab focuses on simulating water, but simulation of fire and smoke can also be achieved using the same techniques. In order to simulate fire, a diffusion term is required, which is ignored in this lab. The Poisson equation is however quite similar to the projection step used in the implmentation.

2 Background

The well-known Navier-Stokes equations describe how a fluid flow changes over time. The flow is described by $\mathbf{V}$, a vector field. A vector field is an entity that assigns a vector to every point in space. The equations for incompressible flow are

$$\frac{\partial \mathbf{V}}{\partial t} = \mathbf{F} + v\nabla^2\mathbf{V} - (\mathbf{V}\cdot\nabla)\mathbf{V} - \frac{\nabla p}{\rho} \quad (1)$$

$$\nabla \cdot \mathbf{V} = 0 \quad (2)$$

where $\mathbf{V}$ is the velocity field, $\mathbf{F}$ is the external force term, ρ the constant density and p the pressure field.

With the use of a technique called operator splitting, one can solve the Navier-Stokes equations one term at a time. The temporary field $\mathbf{V}_1$ can be calculated from $\mathbf{V}_0$ by solving for the self-advection term, $(\mathbf{V}\cdot\nabla)\mathbf{V}$. Then $\mathbf{V}_2$ can be calculated from $\mathbf{V}_1$ by addition of the surface force, $\mathbf{F}$. $\mathbf{V}_3$ can be calculated from $\mathbf{V}_2$ by solving for the diffusion term, $v\nabla^2\mathbf{V}_1$. Lastly, $\mathbf{V}_{\Delta t}$ can be calculated from $\mathbf{V}_3$ by projecting the velocity field onto its divergence free part, which includes the terms $[\frac{\Delta p}{\rho}]$ and $\nabla \cdot \mathbf{V}$. The terms are together responsible for maintaining the incompressibility of the solution.

In this lab, since focus is on simulating water which has a low viscosity, the diffusion term is ignored. The Navier-Stokes equations without viscosity is called the Euler equations and the algorithm for solving the Euler equation is

$$\mathbf{V}_0 \xrightarrow{(\mathbf{V}\cdot\nabla)\mathbf{V}} \mathbf{V}_1 \xrightarrow{\mathbf{F}} \mathbf{V}_2 \xrightarrow{\frac{\nabla p}{\rho},\nabla\cdot\mathbf{V}} \mathbf{V}_{\Delta t} \quad (3)$$

In Navier-Stokes equations, the self-advection term is the non-linear $-(\mathbf{V}\cdot\nabla)\mathbf{V}$ term. It is an important property that this term is non-linear, since it allows Navier-Stokes equations to model vortices or swirls

in the fluid. The self-advection term can be interpreted as *the motion of the velocity field along itself* and unconditional stability is achieved by modeling this term as pure advection instead. This is true if V had not been time dependent, but since it is the self-advection can be determined by solving the following partial differential equation

$$\frac{\partial \mathbf{V}_1}{\partial t} = -(\mathbf{V}_0 \cdot \nabla)\mathbf{V}_0 \tag{4}$$

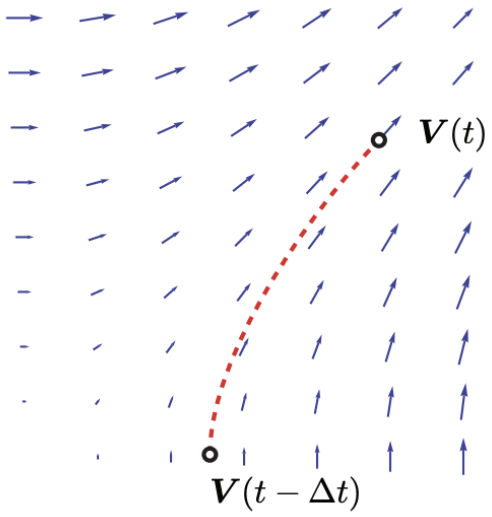

Figure 1: Backwards tracing from the stable fluids approach [1].

This calculated $\mathbf{V}_1$ by following the streamlines of $\mathbf{V}_0$. With the velocity field $\mathbf{V}_0$, $\mathbf{V}_1$ can be created by using the velocity of $\mathbf{V}_0$ at a position which corresponds to the position of a zero-mass particle Δt time-units ago (see Figure 1. This method relies on interpolation. The choice of interpolant is therefore important to the quality of the solution. In this lab simple trilinear interpolation was used, which is fast and easy to implement but results in numerical viscosity.

The external force, $\mathbf{F}$, is equal to the time derivative of $\mathbf{V}_2$, and can therefore be solved through a straightforward approach with Euler time integration (see Equation 5 and 6).

$$\frac{\mathbf{V}_2 - \mathbf{V}_1}{\Delta t} = \mathbf{F} \rightarrow \tag{5}$$

$$\mathbf{V}_2 = \mathbf{V}_1 + \Delta t \cdot \mathbf{F} \tag{6}$$

The last step in the Navier-Stokes equations is to enforce incompressibility. The term $\nabla \cdot \mathbf{V} = 0$ is responsible for this. A divergence free vector field is volume conserving and therefore incompressible. This is done by applying the projection step to $\mathbf{V}_2$ using *Helmhotz-Hodge decomposition* which says that each vector field $\mathbf{V}$ can be divided into a *curl free* part $\mathbf{V}_{cf}$ and a *divergence free* part $\mathbf{V}_{df}$ (see Equation 7).

$$\mathbf{V}_2 = \mathbf{V}_{df} + \mathbf{V}_{cf} \tag{7}$$

The term $\mathbf{V}_{\Delta t}$ is defined as divergence free, and therefore set to equal $\mathbf{V}_{df}$. The gradient of a scalar field, q, is always curl free. This allows us to rewrite the equation to

$$\mathbf{V}_{\Delta t} = \mathbf{V}_2 - \nabla q \tag{8}$$

The divergence operator, $\nabla \cdot$, is used to estimate value of q, which results in the cancellation of the divergence free part $\mathbf{V}_{\Delta t}$ in Equation 7, and we get Equation 12.

$$\nabla \cdot \mathbf{V}_2 = \nabla^2 q \tag{9}$$

The divergence operator is discretized using central differencing scheme

$$\begin{aligned} \nabla \cdot \mathbf{V}_{i,j,k} = \frac{u_{i+1,j,k} - u_{i-1,j,k}}{2\Delta x} + \\ \frac{v_{i,j+1,k} - v_{i,j-1,k}}{2\Delta y} + \\ \frac{w_{i,j,k+1} - w_{i,j,k-1}}{2\Delta z} \end{aligned} \tag{10}$$

where u, v and w are the x, y and z components of the vectors in the vector field $\mathbf{V}$. The

computation and calculation of $\nabla^2 q$ would be rather complex, but can fortunately be reduced to

$$\nabla^2 q_{i,j,k} = \frac{1}{\Delta x^2} \begin{bmatrix} 1 & 1 & 1 & -6 & 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} q_{i+1,j,k} \\ q_{i-1,j,k} \\ q_{i,j+1,k} \\ q_{i,j,k} \\ q_{i,j-1,k} \\ q_{i,j,k+1} \\ q_{i,j,k-1} \end{bmatrix} \tag{11}$$

In an abstract description it can be described as

$$\mathbf{A}x = b \tag{12}$$

where $\mathbf{A} = \nabla^2$, $x = q$ and $b = \nabla \cdot \mathbf{V}_2$. By finding the *Poisson inverse matrix* of $\mathbf{A}$, the equation can be solved, or by using an iterative parallel algorithm.

There are two boundary conditions that need to be taken into account and consider when solving the Poisson equation. The first condition is the *Dirichlet* boundary, which states that there can be no flow, in or out, of the boundary surface to which $\mathbf{n}$ is normal (see Equation 13). In other words, it stops the fluid from flowing into solid objects.

$$\mathbf{V} \cdot \mathbf{n} = 0 \tag{13}$$

The second boundary is the *Neumann* boundary condition

$$\frac{\partial \mathbf{V}}{\partial \mathbf{n}} = 0 \tag{14}$$

It stops any flow along the normal $\mathbf{n}$ of a boundary, solid, surface and is required to Equation 13.

3 Results

The lab was sectioned into three steps. First, the external forces function was implemented. It iterates over every voxel with fluid and sets a new velocity value at (i, j, k) (see Equation 5). Then, the Dirichlet boundary condition was enforced to stop the flow from going through a solid surface. The function checks whether the voxels with fluid have solid voxel neighbors. If it does, the part of the velocity field in that direction is set to zero.

Then the projection function was implemented to solve the inverse Poisson matrix A^{-1}. At position (i, j, k) the divergence of the velocity field is calculated with central differencing. The Neumann boundary condition is maintained by checking the neighboring voxels, to determine whether they are solid. A translated matrix is obtained with seven entries, where each entry corresponds to a specific neighbor to the voxel (i, j, k) and is equal to 1 if it is solid or 0 if it is not. The entry in the middle is equal to the negative sum of all neighbors. All entries are divided by $\frac{1}{\Delta x^2}$.

The new velocity field at voxel (i, j, k) is calculated by using central differencing on the approximated q in the projection step. It is then subtracted from the original velocity.

The fluid simulation can be studied in Figures 2 to 6.

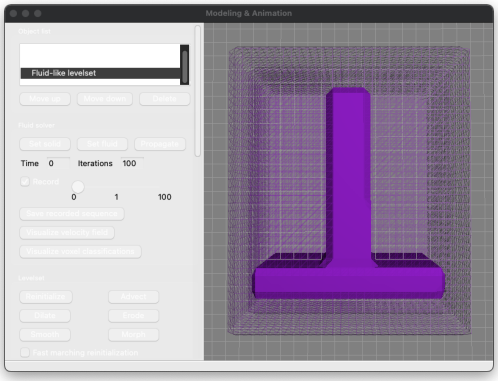

Figure 2: Fluid simulation.

4 Lab partner and grade

My lab partner was Viktor Tholén, and I am aiming for grade 3.

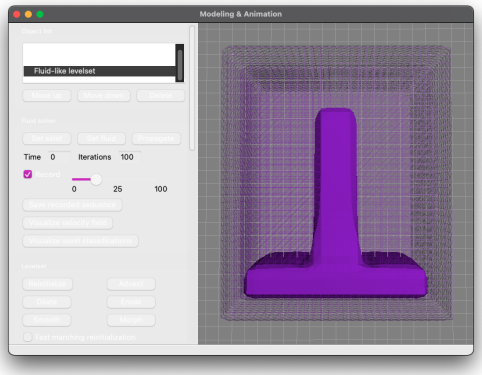

Figure 3: Fluid simulation.

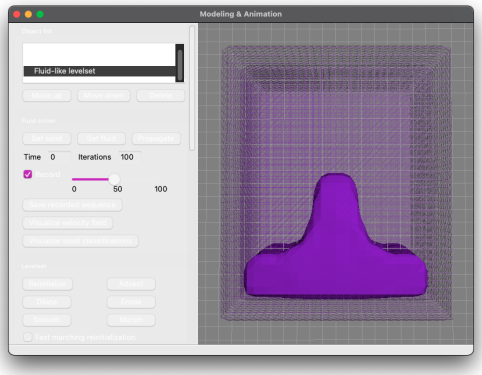

Figure 4: Fluid simulation.

References

[1] Emma Broman Mark Eric Dieckmann, Robin Skånberg. *Fluid simulation*. Modeling and animation, lab 6, 2021.

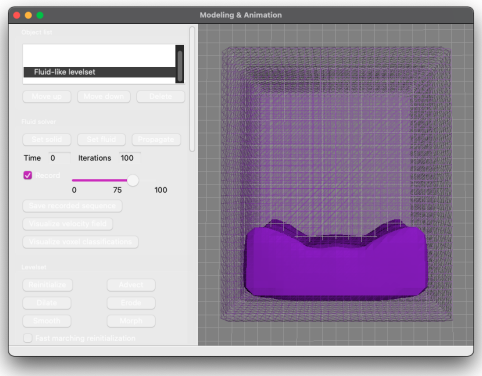

Figure 5: Fluid simulation.

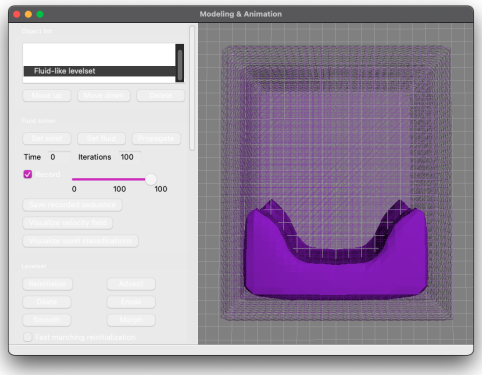

Figure 6: Fluid simulation.