LAB REPORT: LAB 6

TNM079, MODELING AND ANIMATION

Viktor Sjögren
viksj950@student.liu.se

Monday 7th June, 2021

Abstract

We discuss fluid simulation in computer graphics using the Navier-Stokes equations. In the context of CG, very often we simulate incompressible fluids which is what is also done in this lab. This means that criterion's are set in place to ensure a static density for a volume. Alongside this, the fluid simulated in this context is a low viscosity fluid close to water. This means for the the Navier-Stokes equations that viscosity term is removed. Implementation of external forces, boundary condition and projection step is done in this lab and theory behind it is discussed in a summarized manner.

1 Background

Fluid simulation is a huge application in computer graphics that allows for mediums such as water to be simulated and visualised virtually. One of the most common method to achieve satisfactory results is the Navier-Stokes equations, which is what was used in this lab. Navier-Stokes defines how the flow of fluid changes over time, where the flow is represented as a vector field. To keep the complexity down, every fluid that is modelled is required to be incompressible, meaning the density is constant. The Navier-Stokes equations in which the flow is incompressible are as:

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

$$\nabla \cdot \boldsymbol{V} = 0 \tag{2}$$

where $\boldsymbol{F}$ represents external forces and $(\boldsymbol{V} \cdot \nabla)\boldsymbol{V}$ is the self-advection term. The $v\nabla^2\boldsymbol{V}$ term describes viscosity, the $\frac{\nabla p}{\rho}$ term handles the pressure and density and is a part of maintaining the fluid incompressible, which is also true for the $\nabla \cdot \boldsymbol{V} = 0$ constraint.

To better understand and work with the equations a technique called *operator splitting* makes it so that the equations can be solved term for term. The terms and in which order they are solved are as follows:

$$\boldsymbol{V}_0 \xrightarrow{(\boldsymbol{V}\cdot\nabla)\boldsymbol{V}} \boldsymbol{V}_1 \xrightarrow{\boldsymbol{F}} \boldsymbol{V}_2 \xrightarrow{v\nabla^2\boldsymbol{V}} \boldsymbol{V}_3 \xrightarrow{\nabla\cdot\boldsymbol{V}, \frac{\nabla p}{\rho}} \boldsymbol{V}_{\nabla t}$$

$\boldsymbol{V}_0$ correspond to the initial vector field at time t_0 while $\boldsymbol{V}_{\nabla t}$ is the vector field at a later time $t_0 + \nabla t$. Due to the limitations of the lab, water was the sole focus to simulate, meaning viscosity is negligible and therefore the viscosity term is removed (Euler equations).

To go from $\boldsymbol{V}_0$ to $\boldsymbol{V}_1$ the self-advection term needs to be solved:

$$\frac{\partial \boldsymbol{V}_1}{\partial t} = -(\boldsymbol{V}_0 \cdot \nabla)\boldsymbol{V}_0 \tag{3}$$

One method used to solve this is *characteristics method*, which in short interpolates between $\boldsymbol{V}_0$ at a location where a *zero-mass* particle was ∇t time units ago, and $\boldsymbol{V}_1$ to approximate the results.

The results is an approximation of the motion of the velocity field along itself.

Looking at the external force term $\boldsymbol{F}$:

$$\frac{\partial \boldsymbol{V}_2}{\partial t} = \boldsymbol{F} \tag{4}$$

which can be solved using first order Euler time integration where the vector field $\boldsymbol{V}_2$ is created from $\boldsymbol{V}_1$. The resulting term then results in:

$$\boldsymbol{V}_2 = \boldsymbol{V}_1 + \Delta t \cdot \boldsymbol{F} \tag{5}$$

The last term encapsulates what is responsible for the incompresibility criterion, $\nabla \cdot \boldsymbol{V} = 0$. This term is referred to as the projection step. The idea is to use *Helmholtz-Hodge decomposition* to divide the vector field $\boldsymbol{V}_2$ into a curl free part $\boldsymbol{V}_{cf}$ and a divergence free part $\boldsymbol{V}_{df}$. The curl free field can be constructed as a gradient of some scalar field q.

$$\boldsymbol{V}_2 = \boldsymbol{V}_{\nabla t} + \nabla q \tag{6}$$

which then can be assigned to the pressure field p as in equation (1). This would by subtracting it generate the final equation as:

$$\boldsymbol{V}_{\nabla t} = \boldsymbol{V}_2 - \frac{\nabla p}{\rho} \tag{7}$$

where $\boldsymbol{V}_{\nabla t}$ represents the divergence free vector field, and $p = q = 1$. q is unknown in (6) but by applying the divergence operator to both sides and then simplifying, q can be formed as:

$$\nabla \cdot \boldsymbol{V}_2 = \nabla^2 q \tag{8}$$

The left side representing the divergence of this equation can then be discretized using a central difference scheme. The divergence operator describes whether a point in the vector field has flow going outwards or inwards, in other words if the point is a sink or a source. The central difference scheme discretizes in the following manner:

$$\begin{aligned}\nabla \cdot \boldsymbol{V}_{i,j,k} = {} & \frac{u_{i+1,j,k} - u_{i-1,j,k}}{2\Delta x} + \\ & \frac{v_{i,j+1,k} - v_{i,j-1,k}}{2\Delta y} + \\ & \frac{w_{i,j,k+1} - w_{i,j,k-1}}{2\Delta z}\end{aligned} \tag{9}$$

in this case, u, v and w are the corresponding x, y and z components of the vectors in $\boldsymbol{V}$.

The right side of equation (8) can be discretized using the Laplacian operator, becoming:

$$\begin{aligned}\nabla^2 \cdot q_{i,j,k} = {} & \frac{q_{i+1,j,k} - 2q_{i,j,k} + q_{i-1,j,k}}{(\Delta x)^2} + \\ & \frac{q_{i,j+1,k} - 2q_{i,j,k} + q_{i,j-1,k}}{(\Delta y)^2} + \\ & \frac{q_{i,j,k+1} - 2q_{i,j,k} + q_{i,j,k-1}}{(\Delta z)^2}\end{aligned} \tag{10}$$

With the laplacian operator, a description of each voxel's material exchange between one and another is given. Using a uniform grid also means that (10) can be simplified and reduced to a common denominator.

The tasks in this lab involved implementing the external force term and the projection step. In *Fluidsolver.cpp* the external force term was integrated into the code structure as described in equation (5).

The projection step was implemented the same class, where the divergence field of $\boldsymbol{V}$ was computed as seen in equation (9).

Furthermore, due to the nature of simulating fluids there will be scenarios where there is a solid next to fluid voxels, in which case we do not want the fluid to flow into these spaces. To ensure that this is the behavior of the fluid, a condition by the name of *Dirichlet boundary condition* was implemented. In short, we localize points where flow is directed outwards where a solid is present and then alter these to be directed instead alongside the solid. No tampering with the flow is done with fluid if the point is not next to a solid.

2 Results

To see the fluid simulation we use a fluid-like levelset with, as discussed earlier, low viscosity to mimic water. Around this levelset a box shape that is defined as a solid is placed in the scene.

With the Dirichlet boundary condition in place, no fluid passes the border of the solid box and throughout Figure 1(a) to 1(f) the results of the fluid simulation can be seen. At the start the box is getting filled up by a stream of water in the center, which then falls down and creates a wave effect which goes out to the sides as seen in Figure 1(a) through 1(d). Figure 1(e) and 1(f) showcases that the fluid takes some time to settle completely and wobbles just like we'd expect from something that tries to mimic low-viscosity fluid like water.

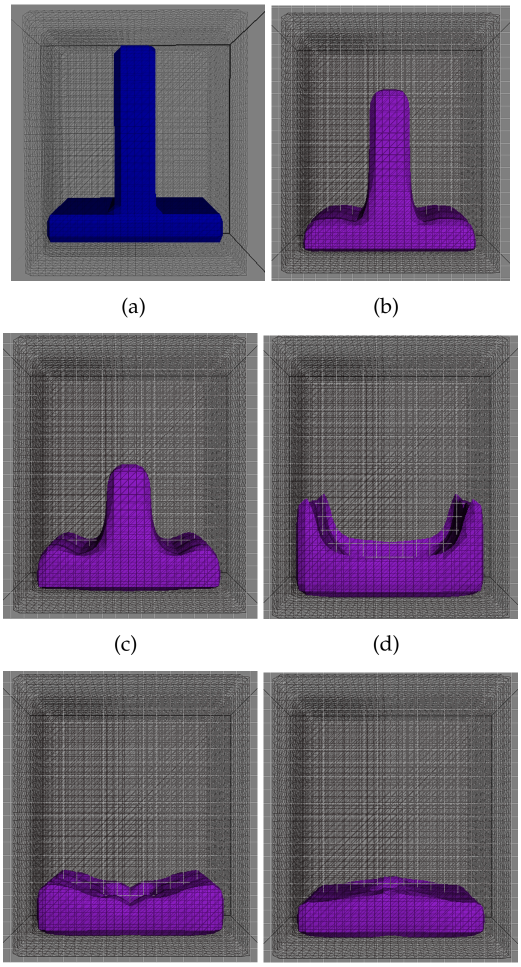
In Figure 2, the same setup as before is seen but without the box that stopped the fluid from leaking or in this case falling indefinitely downwards This is expected as nothing is blocking the liquid on the bottom and therefore in Figure 2(d) the fluid is almost but all under the bounding box from falling.

3 Lab partner and grade

The lab was made in collaboration with Algot Sandahl. The report scope and content is aimed at grade **3**.

References

[1] M.E. Dieckmann. *Lectures 13-15* , tnm079, 2021.



(a) (b) (c) (d) (e) (f)

Figure 1: Fluid simulation using a solid box to encapsulate a levelset mimicking water. Time is set to a stable timestep each propagation step

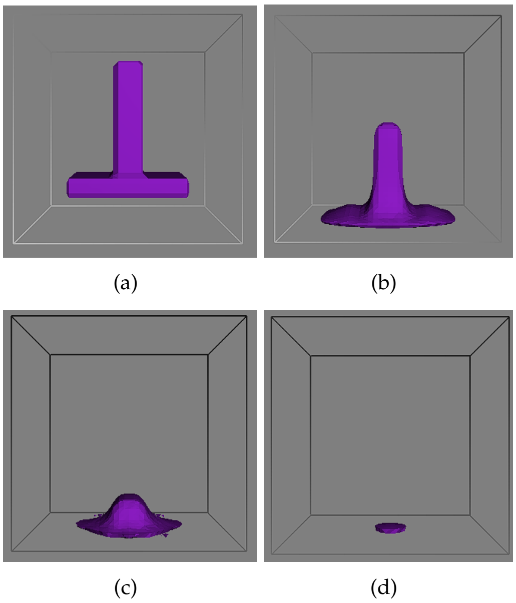


Figure 2: Fluid simulation using only a fluid levelset. Time is set to a stable timestep each propagation step