LAB REPORT: LAB 6

TNM079, MODELING AND ANIMATION

Alma Linder
almli825@student.liu.se

Thursday 8th May, 2025

Abstract

The work of this lab investigates fluid simulations with level set surfaces. Under the assumption that the fluid is incompressible, the Navier-Stokes equations can be used to compute the components of fluid properties. For realistic simulation it is also important that the fluid breaks off as in the physical world when hitting a solid object. To this end, the Dirichlet and Neumann boundary conditions come in handy. Discarding the viscosity, and implementing boundary conditions, projection for pressure compensation, and an external force for gravity, a somewhat realistic block-like fluid simulation was created. The fluid managed to spread, oscillate and progressively ease its movements. The volume changes and the effects of the individual terms of the Navier-Stokes equations are also showcased.

This laboratory work aims for `grade 3`.

1 Background

The Navier-Stokes equation describes how a fluid behaves over time based on the properties of an external force F, pressure p, viscosity and self-advection $(\mathbf{V} \cdot \nabla)\mathbf{V}$. If the viscosity term is ignored, Euler's fluid equations in Equation 1 are obtained, with $\mathbf{V}$ denoting the velocity field of the fluid, and ρ representing the fluid density. Because the fluid is incompressible its volume and mass should be preserved and so it should not flow through a solid object, hence the condition $\nabla \cdot \mathbf{V} = 0$.

Thanks to operator splitting, each term in the Euler equations can be solved for separately, added to $\mathbf{V}$ and then sent to the next term, as Figure 1 illustrates for those terms from the Euler equations actually implemented in the lab. In Figure 1 $\mathbf{V} \cdot n = 0$ is the Dirichlet boundary condition.

$$\begin{cases} \frac{\partial \mathbf{V}}{\partial t} = F + (\mathbf{V} \cdot \nabla)\mathbf{V} - \frac{\nabla p}{\rho}, \\ \nabla \cdot \mathbf{V} = 0 \end{cases} \tag{1}$$

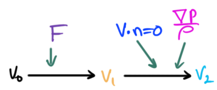


Figure 1: Operator splitting. Each fluid component influence computed separately and then added to the fluid's velocity field. $\mathbf{V_0}$ is the initial state of $\mathbf{V}$.

The first term implemented was F, which in this lab was designed to act as the gravitational force in the system. Since F is external (comes from the world) the grid points in which we sample its value firstly had to be converted to world coordinates [1]. Then it was possible to compute the effects of F on $\mathbf{V_0}$ in discrete terms with Equation 2, which adds F to the initial velocity field $\mathbf{V_0}$ at the next stable time step Δt, given that the current grid position is in the fluid (not in solid surface object or an empty voxel cell). We have now obtained $\mathbf{V_1}$ in Figure 1.

[1]We still work with uniformly spaced grids creating voxels in 3D space indexed by integers, as for level set surfaces.

$$\mathbf{V_1} = \mathbf{V_0} + \Delta t \cdot F \tag{2}$$

The Dirichlet boundary condition $\mathbf{V} \cdot n = 0$ states that there is no flow through a solid surface whose normal is n. Therefore, this condition allows all solutions to the differential equations before and after projection to be zero for points along the surface.

In practice, this meant checking if the adjacent points to the current point (i, j, k) in any direction is a solid. If that was the case, along with the condition that $\mathbf{V}$ is carrying information in that same direction, the $\mathbf{V}$-component in that direction was set to zero. Figure 2 displays this concept, where vectors in $\mathbf{V}$ adjacent to the surface become flattened out, *"gluing"* themselves onto the surface and thus tracking it.

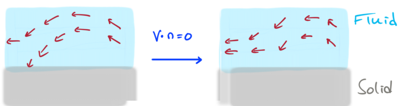


Figure 2: Setting vectors in the velocity field $\mathbf{V}$ adjacent to a solid to zero effectively suppresses the fluid's motion in that region.

By definition, the volume and mass of an incompressible fluid are preserved at all times (i.e divergence-free $\nabla \cdot \mathbf{V} = 0$). This property is desired since tracking density (mass per volume unit) changes in a fluid is more challenging. Moreover, applying external forces or advection to a fluid usually upsets it in such a way that the in- and outflow of a region (in this case a voxel) no longer are identical, creating sources (fluid fans out) and sinks (fluid swallowed down a drain), effectively setting $\nabla \cdot \mathbf{V} \neq 0$ and producing volume loss.

The pressure term $\frac{\nabla p}{\rho}$ in Equation 1 is thus used to compensate the loss of volume external forces can introduce by changing so that the right amount of fluid flows in and out of a voxel maintaining the $\nabla \cdot \mathbf{V} = 0$-criterion. However, even though the pressure term is introduced to the Euler equations to enforce incompressibility, the fluid does not become truly incompressible. This is because of approximations in the projection computations, which will be discuss below.

In reality, the pressure term is computed by projecting $\mathbf{V_1}$ (only external force applied) from Equation 2 onto $\mathbf{V_1}$'s divergence-free component. This is possible since Helmholtz-Hodge decomposition states that any vector field can be split into a divergence-free component $\mathbf{V_1}_{df}$ and a curl-free component $\mathbf{V_1}_{cf}$ (the curl-free component has most likely divergence), according to Equation 3.

$$\mathbf{V_1} = \mathbf{V_1}_{df} + \mathbf{V_1}_{cf} \tag{3}$$

Put in other words, we seek a scalar pressure field p that applied to $\mathbf{V_1}$ removes the curl-free component of $\mathbf{V_1}$ and conserves the divergence-free component. Replacing p with q, assuming $\rho = 1$ and using the Helmholtz-Hodge decomposition, $\mathbf{V_2}$ can be expressed as $\mathbf{V_2} = \mathbf{V_1} - \nabla q$. Taking the divergence on both sides results in the Poisson equation in Equation 4, where term $\mathbf{V_1}$ cancels since $\nabla \cdot \mathbf{V_1} = 0$ for incompressible fluids as discussed above.

$$\nabla \cdot \mathbf{V_2} = \nabla^2 q \Longleftrightarrow b = Ax \tag{4}$$

Firstly, b (the divergence of $\mathbf{V_2}$) was found in its discrete form by using centered differentiation for points (i, j, k) on the grid, in all directions x, y and z of the velocity field. (u, v, w) designate the (x, y, z) components of $\mathbf{V_1}$. This scheme is presented in Equation 5 (uniform grid).

$$\begin{aligned} \nabla \cdot \mathbf{V_2}_{i,j,k} = & \frac{u_{i+1,j,k} - u_{i-1,j,k}}{2\Delta x} + \\ & \frac{v_{i,j+1,k} - v_{i,j-1,k}}{2\Delta x} + \\ & \frac{w_{i,j,k+1} - w_{i,j,k-1}}{2\Delta x} \end{aligned} \tag{5}$$

Secondly, matrix A had to be found. This was done by rewriting the centered discrete differentiation scheme for Ax $(= \nabla^2 q_{i,j,k})$ into vector notation, as is done in Equation 6. For this to work, the current cell in the grid is assumed to contain fluid and hence marked with a 1. Then all neighboring cells to the current cell in all three dimensions were checked and marked. 1 was used for fluid or empty cells, and 0 was used to mark cells in solids, since the fluid should not be able to penetrate the solids and thus have no velocity there according to the Neumann boundary condition $\frac{\partial \mathbf{V}}{\partial n} = 0$. For every neighboring cell contained in solid, the current (center) cell value was incremented by 1, as contribution to our sought pressure should be dependent on how close to a solid any (center) cell (i, j, k) is.

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

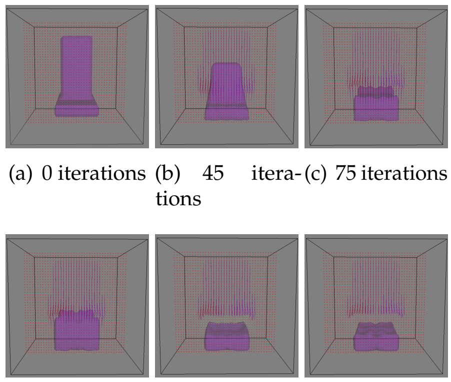
Once A has been obtained it is possible to solve for the divergence of the pressure $x = q$. This was also done with a central differentiation scheme in the discrete grid space (same logic as in Equation 5). In practice, the computations described above to solve for q were iterated until the solution for q reached an acceptable threshold ϵ.

Finally, because of the additive nature of the Helmholtz-Hodge decomposition, subtracting the divergence of $x = q$ (for the curl-free component) from $\mathbf{V_1}$ is enough to only save $\mathbf{V_1}$'s divergence-free component, hence obtaining our final fluid velocity field $\mathbf{V_2}$ in Figure 1.

2 Results

250 simulation steps were run on a cube structure, and Figure 3 shows the original cube and the results for iterations $45, 75, 100, 170$ and 250. For iterations 0 to 100 the effects of the external gravitational force are clearly visible, as it pulls the fluid body downwards to the bottom of the solid cube in which it is contained. The velocity field for this interval visualizes these effects in Figures 3(a)-3(d), where the topmost half of vectors in the field increasingly grow in the downwards direction (negative y-direction). The bottom half of the velocity field fans out, instead hinting where the fluid is going; i.e spreading out along the cube floor in positive and negative z-direction.

Continuing the simulation, the fluid is accurately stopped by the solid boundaries of the cube and bounces back inwards again, creating an oscillating pattern. The velocity vectors around 170 iterations nicely illustrate the opposite motion as the fluid bounces back to form a ridge. Finally, at $200 - 250$ iterations the system's forces begin to dissipate, as the shorter velocity vectors in Figure 3(f) demonstrate.



(a) 0 iterations (b) 45 iterations (c) 75 iterations

(d) 100 iterations (e) 170 iterations (f) 250 iterations

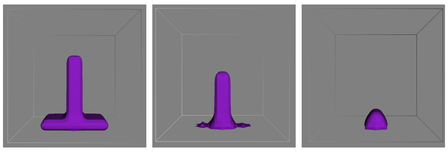
Figure 3: Fluid simulation on a cube structure. Snapshots at 45, 75, 100, 170 and 250 iterations. Velocity field shown in red.

Table 1 details the volume changes of the fluid over the sequence in Figure 3. Unsurprisingly, the volume decreases. The volume loss can be attributed to the fact that the solution for $p = q$ is an approximation, as the algorithm stops at an "*acceptable*" threshold ϵ. Additionally, as always, discretization leads approximate values as defined grid locations are sampled. However, Table 1 showcases the pressure's ability to keep the loss relatively small (unlike when there is no preservation present, see Figure 4) - especially as the number of iterations become larger - and even.

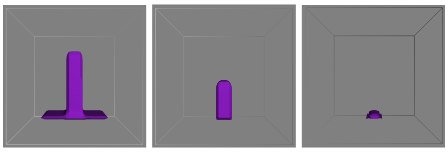
Table 1: Volume change of the fluid over the sequence in Figure 3.

#iter.	**0**	**45**	**75**	**100**	**170**	**250**
Vol.	0.401	0.386	0.346	0.307	0.279	0.239

Figure 4 visualizes the consequences of not implementing boundaries for adjacent solids, or projection for volume preservation. Visibly in 4(a) through 4(c), the fluid just falls through the solid boundary box if the Dirichlet boundary condition is removed. If instead the pressure is ignored - as in Figures 4(d) to 4(f) - the fluid rapidly looses volume, collapsing almost completely already around 75 iterations.



(a) 35 iterations (b) 50 iterations (c) 75 iterations



(d) 30 iterations (e) 55 iterations (f) 75 iterations

Figure 4: The effects of removing the Dirichlet boundary condition (top), and removing the volume preservation from pressure (bottom).