Fluid Simulation

Rina Fumoto

Introduction

Fluid simulation has been widely used in CGI, including VFX and animation as well as aeronautics, mechanics and construction. This report will first review relevant literature, explore different techniques and identify some existing solutions that have been used for fluid simulation. Then, it will introduce algorithms and implementation methods used for this project.

Background

Definitions

According to White (2011), fluid is defined as an aggregation of molecules that moves and deforms continuously when shear stress is applied. There are two types of fluids, liquids and gases. There are smaller spaces between molecules for liquid so it is more difficult to compress compared to gases. Some important properties of fluids are pressure, density and viscosity.

Pressure is the compression stress at a point in a static fluid. The pressure on the small area in a large mass of fluid can be calculated by the force caused by the surrounding fluid divided by the area (Rajput 2015). The pressure differences advect fluid (Russano and Avelino 2019).

The density of a fluid can be calculated by its mass divided by its volume (Rajput 2015). Density in liquids is nearly constant which means the liquid is almost incompressible (White 2011). The density of water is about 1000kg/m3.

Viscosity is a measure of how well a fluid resists flow. It determines the rate of shearing strain generated by the sliding motion of each layer in the fluid (White 2011). Fluid with low viscosity such as air moves easily and fluid with a high viscosity like honey is hard to move. Kinematic viscosity is defined as the ratio between the dynamic viscosity and density of the fluid.

Tracking Methods

There are three approaches to track fluid flow, which are Lagrangian, Eulerian and hybrid approaches.

The Lagrangian approach is a particle-based approach. This approach tracks individual points in space that store quantities, such as velocity, density and temperature in time (Salomonsson 2011). The advantages of this approach are the tracking accuracy and less storage cost. Also, the equations are relatively simple as it is similar to rigid body dynamics (Parent 2012). However, it is difficult to track the relationship between the points and work with the spatial derivatives (Salomonsson 2011).

The Eulerian approach is a grid-based approach. This approach observes fluid properties at fixed points in space. As the points are fixed in space, it is easier to track the relationship between the points and handle the spatial derivatives but it requires large storage to sample the whole domain. (Salomonsson 2011)

Hybrid approaches combine the advantages of particle-based approach and grid-based approach. These approaches compute quantities on a grid and advect the velocity field using particles. There are two popular methods called Particle-in-Cell (PIC) and Fluid Implicit Particle (FLIP). PIC was introduced by Evans, Harlow, and Bromberg in 1957. This method first transfers particle velocities to a grid and calculate the quantities. After the calculation, the velocities are interpolated to particles from the grid and particles are advected. This method suffers excessive numerical diffusion due to double interpolation. FLIP, which was introduced by Brackbill and Ruppel in 1986, solves this problem by calculating the change in velocity and

adding it to the existing particle velocities. There are more advanced methods, such as Rigid Particle-in-Cell (RPIC) and Affine Particle-in-Cell (APIC) (Jiang et al. 2015).

Navier-Stokes Equations

A fluid with closely constant density and temperature is described by a velocity field and a pressure field according to Stam (1999). These quantities vary with both position and time. When the initial velocity and pressure is known, the changing of these quantities over a time step can be calculated by the incompressible Navier-Stokes equations:

$$\nabla \cdot \boldsymbol{u} = 0 \tag{1}$$

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

where $\boldsymbol{u}$ is the velocity, $\boldsymbol{v}$ denotes the kinematic viscosity of the fluid, ρ is the density, p is the pressure and $\boldsymbol{f}$ is an external force. The symbol ∇ is the vector of spatial partial derivatives, $\nabla = \left(\frac{\delta}{\delta x}, \frac{\delta}{\delta y}, \frac{\delta}{\delta z}\right)$ in three dimensions.

Equation 1 describes the conservation of mass. This equation is called the incompressibility condition which ensures that the volume of the fluid is constant. Equation 2 describes the conservation of momentum. This equation is a vector equation for three components of the velocity field in three dimensions as shown below:

$$\begin{aligned}
\frac{\partial u}{\partial t} &= -(\boldsymbol{u} \cdot \nabla)u - \frac{1}{\rho}\nabla p + \boldsymbol{v}\nabla^2 u + f_x \\
\frac{\partial v}{\partial t} &= -(\boldsymbol{u} \cdot \nabla)v - \frac{1}{\rho}\nabla p + \boldsymbol{v}\nabla^2 v + f_y \\
\frac{\partial w}{\partial t} &= -(\boldsymbol{u} \cdot \nabla)w - \frac{1}{\rho}\nabla p + \boldsymbol{v}\nabla^2 w + f_z
\end{aligned} \tag{3}$$

These equations correspond to the Newton's second law of motion $\boldsymbol{F} = m\boldsymbol{a}$.

Solving the Equations

The above equations are too complex to be solved in a single step. To make it simpler, the equations can be split into four steps as presented by Stam (1999). The four steps are external forces, advection, diffusion and projection.

The Navier-Stokes equations take forces that influence the flow externally into account. The forces can be gravity, wind or any forces caused by other objects. Adding external forces is important to simulate natural phenomena acting on the fluid (Marek 2007). The advection step computes how particles of fluid are transported with the velocity field. There are different approaches to compute advection, such as Lagrangian, Semi-Lagrangian, Eulerian, FLIP and PIC. The diffusion step solves the viscosity that influences the velocity. After the previous steps, the velocity field is not divergence-free. The projection step makes the resulting field divergence-free and the fluid incompressible. This step also enforces the boundary conditions to make sure the fluid does not leak into solid objects.

In many cases for VFX, viscosity can be dropped from the equation. The Navier-Stoke equations without the viscosity term are called Euler equations:

$$\begin{aligned}
\nabla \cdot \boldsymbol{u} &= 0 \\
\frac{\partial \boldsymbol{u}}{\partial t} &= -(\boldsymbol{u} \cdot \nabla)\boldsymbol{u} - \frac{1}{\rho}\nabla p + \boldsymbol{f}
\end{aligned} \tag{4}$$

The material derivative describes how a quantity of a fluid particle changes in time as it moves through a velocity field:

$$\frac{D\boldsymbol{u}}{Dt} = \frac{\partial \boldsymbol{u}}{\partial t} + u\frac{\partial \boldsymbol{u}}{\partial x} + v\frac{\partial \boldsymbol{u}}{\partial y} + w\frac{\partial \boldsymbol{u}}{\partial z}$$
$$= \frac{\partial \boldsymbol{u}}{\partial t} + (\boldsymbol{u} \cdot \nabla)\mathbf{u} \tag{5}$$

Using the material derivative, the Euler equations can be written in a compact form:

$$\frac{D\boldsymbol{u}}{Dt} = -\frac{1}{\rho}\nabla p + \boldsymbol{f} \tag{6}$$

MAC Grid

Marker and Cell (MAC) grid introduced by Harlow and Welch (1965) is used for storing the quantities on a grid. It is called a staggered grid because it stores different quantities at different locations. As can be seen in Figure 1, pressure is stored at the centre of each cell and velocity is split into components and stored at the centres of cell faces.

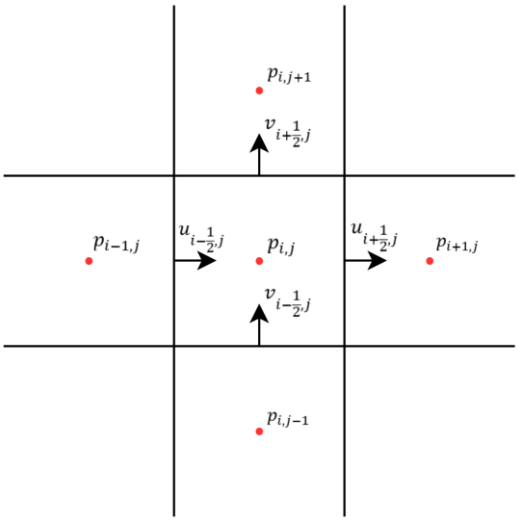

Figure 1: The two-dimensional MAC grid

Using the MAC grid allows calculating the derivative of the velocity field at cell centres more accurately. For example, the derivative of velocity u at the centre of cell i,j can be estimated by central differencing:

$$\left(\frac{\partial u}{\partial x}\right)_{i,j} \approx \frac{u_{i+\frac{1}{2},j} - u_{i-\frac{1}{2},j}}{\Delta x} \tag{7}$$

where Δx is the width of the grid cell.

Implementation

Initialisation

In the initialisation step, particles need to be created on the grid. Zhu and Bridson (2005) suggested to generate 8 (2x2x2) particles per grid cell randomly in 3-dimensions. As this project is in 2-dimensions, 4 (2x2) particles per grid cell are generated.

Particles to Grid

Each velocity point on the grid takes the weighted average of the particle velocities that are in the square of twice the grid cell width centred on the point (Zhu and Bridson 2005). For this project, a simple bilinear

hat function is used for a robust and fast computation (Salomonsson 2011). For example, the velocity u at $i+\frac{1}{2}, j$ can be calculated with the following equations:

$$u_{i+\frac{1}{2},j} = \frac{\sum_p u_p k(\boldsymbol{x}_p - \boldsymbol{x}_{i+\frac{1}{2},j})}{\sum_p k(\boldsymbol{x}_p - \boldsymbol{x}_{i+\frac{1}{2},j})} \tag{8}$$

$$k(x,y) = h\left(\frac{x}{\Delta x}\right) h\left(\frac{y}{\Delta y}\right) \tag{9}$$

$$h(r) = \begin{cases} 1-r, & 0 \le r < 1 \\ 1+r, & -1 \le r < 0 \\ 0, & otherwise \end{cases} \tag{10}$$

Add Forces

External forces can be added easily with a simple Euler method:

$$\boldsymbol{u} = \boldsymbol{u} + \boldsymbol{f}\Delta t \tag{11}$$

Project

In the projection step, the velocity is updated by the following equation:

$$\boldsymbol{u}^{n+1} = \boldsymbol{u}^n - \Delta t \frac{1}{\rho} \nabla p \tag{12}$$

so that it satisfies the incompressibility condition (Equation 1) and boundary condition (Braley and Sandu 2010):

$$\boldsymbol{u}^{n+1} \cdot \hat{n} = \boldsymbol{u}^n_{solid} \cdot \hat{n} \tag{13}$$

Using the MAC grid, Equation 12 can be written as:

$$\begin{aligned} u^{n+1}_{i+\frac{1}{2},j} &= u^n_{i+\frac{1}{2},j} - \Delta t \frac{1}{\rho} \frac{p_{i+1,j} - p_{i,j}}{\Delta x} \\ v^{n+1}_{i,j+\frac{1}{2}} &= v^n_{i,j+\frac{1}{2}} - \Delta t \frac{1}{\rho} \frac{p_{i,j+1} - p_{i,j}}{\Delta x} \end{aligned} \tag{14}$$

Also, Equation 1 can be written as:

$$\nabla \cdot \boldsymbol{u}^{\boldsymbol{n+1}} = \frac{u^{n+1}_{i+\frac{1}{2},j} - u^{n+1}_{i-\frac{1}{2},j}}{\Delta x} + \frac{v^{n+1}_{i,j+\frac{1}{2}} - u^{n+1}_{i,j-\frac{1}{2}}}{\Delta x} = 0 \tag{15}$$

The following equation can be formed from Equation 14 and 15 to solve the unknown pressure values:

$$\Delta t \frac{1}{\rho} \left(\frac{4p_{i,j} - p_{i+1,j} - p_{i,j+1} - p_{i-1,j} - p_{i,j-1}}{\Delta x^2} \right) = -\left(\frac{u_{i+\frac{1}{2},j} - u_{i-\frac{1}{2},j}}{\Delta x} + \frac{v_{i,j+\frac{1}{2}} - v_{i,j-\frac{1}{2}}}{\Delta x} \right) \tag{16}$$

This is an approximation of the following Poisson problem (Birger and Hedblom 2012):

$$\Delta t \frac{1}{\rho} \nabla \cdot \nabla p = \nabla \cdot \boldsymbol{u} \tag{17}$$

Two boundary conditions are used, Dirichlet and Neumann. Dirichlet boundary condition says that the value of the quantity at the boundary is specified directly. The flow only exists within the fluid boundary so the pressures outside the boundary are set to 0. Neumann boundary condition says the values of the derivative is applied at the border. The velocity at the border of the static solid becomes 0 which means that the flow disappears when it reaches solid.

For example, when the cell on the right is solid, $u^{n+1}_{i+\frac{1}{2},j} = u_{solid}$. From Equation 14, $p_{i+1,j}$can be solved as follows:

$$p_{i+1,j} = p_{i,j} + \frac{\rho \Delta x}{\Delta t}(u_{i+\frac{1}{2},j} - u_{solid}) \tag{18}$$

When the cell below is air, $p_{i,j+1} = 0$. By replacing $p_{i+1,j}$ and $p_{i,j+1}$in Equation 16, the following equation is formed:

$$\Delta t \frac{1}{\rho}\left(\frac{3p_{i,j} - p_{i-1,j} - p_{i,j-1}}{\Delta x^2}\right) = -\left(\frac{u_{i+\frac{1}{2},j} - u_{i-\frac{1}{2},j}}{\Delta x} + \frac{v_{i,j+\frac{1}{2}} - v_{i,j-\frac{1}{2}}}{\Delta x}\right) + \left(\frac{u_{i+\frac{1}{2},j} - u_{solid}}{\Delta x}\right) \tag{19}$$

As can be seen, the pressure for the air cell and solid cell is removed from the equation. Also, the coefficient in front of $p_{i,j}$ is reduced by one. It can be said that the coefficient of $p_{i,j}$ is the number of non-solid neighbour cells.

The Poisson problem can be expressed using a linear equation:

$$\boldsymbol{Ap} = \boldsymbol{b} \tag{20}$$

where $\boldsymbol{A}$ is the coefficient matrix, p is the unknown pressures and b is the negative divergence of every cell. It can be solved using Preconditioned Conjugate Gradient, Multi-Grid or Jacobi iterative technique. The conjugate gradient solver in the Eigen library is used for this project.

Grid to Particles

The FLIP method uses the changes in velocities computed on the grid to update the particle velocities. The changes are transferred to particles using the hat function mentioned earlier (Equation 9, 10):

$$u_p = u_p + \frac{\sum_{i,j} \Delta u_{i,j} k(\boldsymbol{x}_p - \boldsymbol{x}_{i,j})}{\sum_{i,j} k(\boldsymbol{x}_p - \boldsymbol{x}_{i,j})} \tag{21}$$

Advection

After the update of particle velocities, the particles can be moved using the forward Euler method. Runge-Kutta 2nd order method with 5 sub-steps can be used for higher accuracy as used by Zhu and Bridson (2005).

Conclusion

This report covered the basic concepts of the fluid simulation and the implementation focused on FLIP simulation. There are many ways to simulate fluid and more aspects, such as temperature, density, viscosity and colour can be added to the simulation. I would like to continue working on this topic and try different methods and different aspects in the future.

References

Birger, C. and Hedblom, 2012. A., PIC/FLIP fluid simulation with anisotropic surface reconstruction.

Brackbill, J.U. and Ruppel, H.M., 1986. FLIP: A method for adaptively zoned, particle-in-cell calculations of fluid flows in two dimensions. Journal of Computational physics, 65(2), pp.314-343.

Braley, C. and Sandu, A., 2010. Fluid simulation for computer graphics: A tutorial in grid-based and particle-based methods. Virginia Tech, Blacksburg.

Evans, M.W., Harlow, F.H. and Bromberg, E., 1957. The particle-in-cell method for hydrodynamic calculations. LOS ALAMOS NATIONAL LAB NM.

Harlow, F.H. and Welch, J.E., 1965. Numerical calculation of time-dependent viscous incompressible flow of fluid with free surface. The physics of fluids, 8(12), pp.2182-2189.

Jiang, C., Schroeder, C., Selle, A., Teran, J. and Stomakhin, A., 2015. The affine particle-in-cell method. ACM Transactions on Graphics (TOG), 34(4), pp.1-10.

Marek, S., 2007 Grid-based Realtime Fluid Dynamics

Parent, R., 2012. Computer animation: algorithms and techniques. Newnes

Rajput, R.K., 2015. A textbook of fluid mechanics and hydraulic machines. S. Chand Publishing.

Russano, E. and Avelino, E. F., 2019. Computational Fluid Mechanics and Dynamics for Scientists. Ashland: Arcler Press.

Salomonsson, F., 2011. PIC/FLIP Fluid Simulation Using Block-Optimized Grid Data Structure.

Stam, J., 1999, July. Stable fluids. In Proceedings of the 26th annual conference on Computer graphics and interactive techniques (pp. 121-128).

White, F. M., 2011. Fluid mechanics 7th edition. Boston, Mass, WCB/McGraw-Hill.

Zhu, Y. and Bridson, R., 2005. Animating sand as a fluid. ACM Transactions on Graphics (TOG), 24(3), pp.965-972.