FLIP 2D solver

A. Sztrajman

University of Buenos Aires

C. Batty, E. Grinspun

Columbia University

Most fluids in computer graphics simulations are incompressible. That is, they obey the incompressible Navier-Stokes equations [1]

$$\frac{\partial \vec{u}}{\partial t} + \vec{u} \cdot \nabla \vec{u} + \frac{1}{\rho} \nabla p = \vec{g} + \nu \nabla \cdot \nabla \vec{u} \tag{1}$$

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

In order to solve these equations, we've used an Eulerian approach over a 2D MAC[1] grid. In this grid configuration, each physical magnitude is stored at a different location within each cell, as seen in figure 1.

The intermediate values of the physical magnitudes are obtained by simple bilinear interpolation, or optionally by the more elaborate Catmull-Rom[2] interpolation.

The first step to solve equation 1 is to split it in the following set of differential equations:

$$\frac{\partial \vec{u}}{\partial t} = -\vec{u} \cdot \nabla \vec{u} \tag{3}$$

$$\frac{\partial \vec{u}}{\partial t} = -\frac{1}{\rho} \nabla p \tag{4}$$

$$\frac{\partial \vec{u}}{\partial t} = \vec{g} \tag{5}$$

$$\frac{\partial \vec{u}}{\partial t} = \nu \nabla \cdot \nabla \vec{u} \tag{6}$$

We'll use different methods to deal with each term, and sum up the solutions to obtain the actual velocity field $\vec{u}$.

- The advection equation 3 is solved using a semi–Lagrangian approach.
- Solving the pressure equation 4 along with the incompressibility condition 2 involves the inversion of the 5–point Laplacian matrix. This is accomplished by an implementation of the PCG (Preconditioned Conjugate Gradient) method by R. Bridson [3]. The pressure step implies the projection of the velocity field into a divergence free configuration.

[1] Marker and Cell

[2] Equivalent to cubic Hermite spline interpolation.

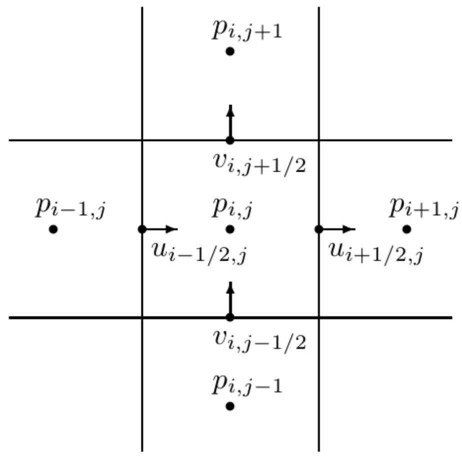

Figure 1 *Detail of single cell in a 2D MAC grid. Velocity components u and v are located in the faces of the cell, while pressure p is located in the center.*

- The gravity force is added by simple first order finite differences over equation 5.
- The PCG solver can also be used to deal with equation 6, given that it involves the inversion of the Laplacian matrix. Nevertheless, in this implementation we've considered only inviscid fluids.

Level set

In order to keep track of the air-liquid surface, we've implemented a level set representation of the interface. In this approach, the interface is implicitly described by means of a function $\psi(\vec{r})$. Its value is defined in the cell centers along the grid as the signed distance to the interface's closest point:

$$\psi(\vec{r}) = \min_{\vec{r}_p \in S} |\vec{r} - \vec{r}_p|$$

Thus, the interface is defined as the set of points where $\psi(\vec{r}) = 0$. By convention, ψ is taken to be negative inside the surface (in the liquid), and positive outside.

The time evolution of the surface results from the simple advection of the function ψ in the velocity field:

$$\frac{\partial \psi}{\partial t} = -\vec{u} \cdot \nabla \psi$$

Unfortunately, advecting a signed distance field doesn't in general preserve the signed distance property [1]. Thus, we generally need to periodically recalculate signed distance.

Since we are advecting fields beyond the interface, we need an extrapolation scheme for the velocity field from the surface interior to the rest of the grid. For this purpose, we've used constant extrapolation (i.e. setting a quantity at a point outside the fluid to the value at the closest point on the fluid surface), although we later developed an implementation of Aslam's higher-order generalizations [2].

Finally we've implemented the ghost fluid method, as laid out by Gibou et al. [4], in order to take into account the surface geometry in the pressure projection step (and improve over the voxelized treatment). In this approach, we temporarily[3] scale each cell's pressure by the fraction of cell filled with liquid (which is 0 for cells outside the surface, 1 inside, and values in between for cells that lie in the interface). This fraction is estimated by means of a linear interpolation of the signed distance function ψ.

References

[1] R. BRIDSON, *Fluid simulation for computer graphics*, A. K. Peters (2008).

[2] T. D. ASLAM, *A partial differential equation approach to multidimensional extrapolation*, **J. Comp. Phys.** 193 (2004), 349-355.

[3] `http://www.cs.ubc.ca/~rbridson/`

[4] F. GIBOU et al. *A second-order-accurate symmetric discretization of the Poisson equation on irregular domains*, **J. Comp. Phys.** 176 (2002), 205-227.

[3] Only for the purpose of the pressure projection step.