

# Preserving the Volume of Fluid Using Multi-phase Flow Approach

Roman Ďurikovič \*

Faculty of Mathematics, Physics and Informatics, Comenius University, Slovakia

Katsuhiro Numata

Software Department, The University of Aizu, Japan

## Abstract

*Methods to animate multiphase fluids are rarely researched, although such phenomena are common in our daily lives. The main focus of this research is to develop a suitable method to create realistic animation of multiphase flows, i.e. flow of fluid mixtures. Our method is based on Navier-Stokes equations, a set of physical equations that describe motion of fluid. A Marker-and-Cell (MAC) grid is used to solve the equations on discrete computational domain. To simulate several fluids, we propose the modification of Volume-of-Fluid (VOF) representation of fluid and integrate it into the multiphase-fluid approach. A mixture of fluids is treated as a single fluid having variable density and viscosity. This scheme allows two or more fluids having different densities and viscosities to be simulated simultaneously.*

## 1 Introduction

We see fluid motions on a daily basis and the phenomena shows very complex behaviors. We often see rising bubbles or flow of muddy water, such flows involve several fluids. Making such animations are challenging tasks in computer graphics area. In this research, we adopted multiphase CFD (Computational Fluid Dynamics) technique to solve the problem.

Our approach is based on Volume-of-Fluid (VOF) method, and we extended it into multi-phase cases. The VOF method is very efficient and accurate scheme to track interfaces of fluids. The method takes special care in computing fluid fluxes to preserve sharp interface between fluids. The VOF scheme also enables fast interface construction. We directly construct fluid interfaces from VOF values using the marching cubes algorithm. The interfaces are composed of polygon meshes and rendered.

---

\*e-mail: roman.durikovic@fmph.uniba.sk. Also with University of Saint Cyril and Metod, Trnava, Slovakia.

## 1.1 Related Work

In CFD area, VOF method was first proposed by Hirt and Nichols [4], it provides a simple and economical way to track free boundaries, and preserves a fluids volume completely. In the method, a donor-acceptor procedure can completely conserve a fluid volume when the fluid passes through a donor cell to an acceptor cell. The VOF method was improved in engineering area by Fluid-attenuated inversion recovery (FLAIR) and Cubic Interpolated Propagation (CIP), and other methods developed to estimate fluid flux in more accurate ways.

There are a few researches to handle interaction of two fluids, or fluid and solid objects in computer graphics. Tanaka et al. [7] used VOF and CIP techniques to create fluid animations. They also integrated rigid body simulation into their fluid simulator. Hong and Kim [5] used the VOF method to animate bubbles in a liquid.

Carlson et al. [2] achieved to model high viscous, almost solid materials like a wax. The method avoids stability problems, which arises at high viscosities, and it can simulate melting effects by changing the viscosities. They also proposed rigid-fluid method [3]. The technique treats rigid bodies, as they were made of a fluid. Velocities inside solids are changed so that rigid motion is enforced in the region.

## 2 Motion of Multiple Fluids

The main contribution towards the goal of this research is to simulate mixtures of fluids, for example a gas and liquid. We represent each fluid by a VOF function. Assuming the fluids are incompressible, then the spatial distributions of fluids has the following relation:

$$\langle F \rangle = \sum F_n = 1.0,$$

where  $F_n$  is the VOF function of  $n$ th fluid, while fluid can have different densities and viscosities,  $\rho_n$  and  $\nu_n$ .

Each cell has two VOF values, the top row represents fraction of the first fluid and the bottom row corresponds to

the second fluid. The interface between two fluids lies in cells which contain both fluids. We treat a flow of a mixture of fluids as a flow of a single fluid which have variable density and viscosity. Then, the motion of the flow is calculated using density and viscosity weighted by VOF values  $\langle \rho \rangle = \sum(F_n \rho_n)$ ,  $\langle \nu \rangle = \sum(F_n \nu_n)$ .

## 2.1 Navier-Stokes Equations for Multiple Fluids

The fluid density and viscosity were assumed to be constant all over the fluid. In this case, there are two or more fluids, and they are treated as a fluid with variable density and viscosity. The equation describing the multiple flow dynamics then will be

$$\begin{aligned} \frac{\partial \mathbf{u}}{\partial t} &= -(\mathbf{u} \cdot \nabla) \mathbf{u} + \nabla \cdot (\langle \nu \rangle \nabla \mathbf{u}) - \frac{1}{\langle \rho \rangle} \nabla p + \frac{\mathbf{f}}{\langle \rho \rangle} \quad (1) \\ \nabla \cdot \mathbf{u} &= 0, \quad (2) \end{aligned}$$

Here the viscosity  $\langle \nu \rangle$  is placed inside the derivative. The conservation of mass can be written in the other way as

$$\nabla \cdot (\langle \rho \rangle \mathbf{u}) = -\frac{\partial \langle \rho \rangle}{\partial t}.$$

We can interpret the second equation as that total of mass flow out of a small region is negative of mass change of the region.

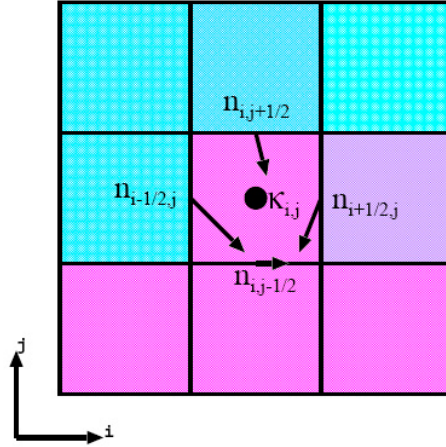
## 2.2 Simulation Steps

The simulation steps for the MAC method are summarized as:

1. Choose an appropriate time step size  $\Delta t$ .
2. Move the VOF values according to the current velocity field.
3. Set boundary velocities on faces of obstacle cells and similarly set Neumann boundary conditions for pressure on obstacle cells.
4. Calculate intermediate velocities using the Navier-Stokes equations without taking account the pressure term.
5. Calculate pressure field from the intermediate velocity field.
6. Make the intermediate velocity field mass-conserving to get correct velocities.
7. To generate the animation frame construct fluid surface from the VOF values.

## 3 Estimation of Surface Tension

Sometimes, the fluid interfaces behave like elastic films. The effect is caused by surface tension acting on the interfaces. We propose the idea of calculation the surface tension for multiple fluids. Accurate calculation of the effect needs well-defined fluid interfaces. Although our simulator doesn't consider actual shape of the interface, a model called continuum surface force (CSF) can be used to approximate the surface force without that information [1]. Method is based on volume force that converges to surface force as the grid size becomes smaller. We propose the extension of this idea



**Figure 1. Estimation of the normal and curvature at fluid surface.**

to multiple fluids as well. Volume force exerted by fluid  $n$  based on the VOF value,  $F_n$ , is written as

$$\mathbf{f}_n^{sv} = \sigma_n \kappa_n \nabla F_n,$$

where  $\sigma_n$  is surface tension coefficient, and  $\kappa_n$  is curvature of the fluids surface. Normal and curvature of the surface are calculated from VOF values as

$$\begin{aligned} \mathbf{n}_n &= \nabla F_n, \\ \kappa_n &= -(\nabla \cdot \hat{\mathbf{n}}_n), \end{aligned}$$

where  $\hat{\mathbf{n}}_n$  represents unit normal vector,  $\mathbf{n}_n/|\mathbf{n}_n|$ . Curvature  $\kappa_n$  is defined at center of a cell. However, as normal is a vector quantity, components of  $\hat{\mathbf{n}}_n$  are separately defined on faces of a cell, and they must be interpolated to obtain values at other positions, see Figure 1. For example,  $x$ -component of a normal vector defined on center of right face of a cell  $(i, j, k)$  will be

$$n_{i+1/2,j,k}^{nx} = \frac{F_{n,i+1,j,k} - F_{n,i,j,k}}{dx},$$

and its y-component is obtained by interpolation:

$$n_{i+1/2,j,k}^{ny} = \frac{n_{i,j-1/2,k}^{ny} + n_{i+1,j-1/2,k}^{ny} + n_{i,j+1/2,k}^{ny} + n_{i+1,j+1/2,k}^{ny}}{4}.$$

Now, we are almost ready to calculate the surface force  $\mathbf{f}_n^{sv}$  on each cell face. To get the curvature defined on a face between two cells we need to interpolate two cell-centered curvatures. Finally, volume forces for all fluids are accumulated and added to the external forces:

$$\mathbf{f} = \sum \mathbf{f}_n^{sv} + \mathbf{f}^{ext}.$$

## 4 Proposed Donor-Acceptor Method

A special care must be taken to preserve sharpness of fluid surfaces. A flux approximation technique called donor-acceptor method was used in the case of a single fluid [4]. The essential idea is to use multiple values in upstream/downstream, leftstream/rightstream, from-stream/backstream cells to approximate crude shape of surface in the cells, and then compute the flux using that information. First we must find the *donor* and *acceptor* cells that are determined by normal velocity  $\mathbf{u}$  on cell faces. The amount of  $F_n$  fluxed across a cell face in a single time step then given by

$$\begin{aligned} \nabla F_n &= \min(F_n^{AD}|V| + CF_n, F_n^D), \\ CF_n &= \max((1 - F_n^{AD})|V| - (\sum_i F_i^D - F_n^D), 0), \\ V &= \frac{u\Delta t}{\Delta x}, \end{aligned}$$

where  $F_n^D$  is  $F_n$  value of the donor cell, and the value of  $F_n^{AD}$  is determined by conditions described below. The above calculation is iterated over all faces between cells marked in MAC grid as *full*.

**Definition 1** *Isolated cell.* The donor cell is called isolated if there is fluid  $n$  that satisfies  $F_n^D > F_n^{neighbor}$  for all neighboring cells.

The value of  $F_n^{AD}$  is initially set to  $F_n^D$  if the donor cell is isolated or otherwise  $F_n^A$ . Our approach prevents a fluid in a donor cell to flow into the next cell until the donor cell is filled with the fluid, unless the fluid is isolated in the cell.

Figure 2 visually shows some examples of  $\Delta F_n$  and  $CF_n$ . Since velocity on the middle face is directed to the right cell, the left cell becomes *donor* and the right cell becomes *acceptor*. There are two fluids, namely *fluid1* and *fluid2*. Left cell contains more *fluid2* than *fluid1*. while right cell has more *fluid1* than the left cell. We assume that some of neighbor cells of the left cell are filled with *fluid2* therefore, the left cell is not isolated in this case, and we will use the

values of acceptor cell for  $F_n^{AD}$ . Vertical dashed line represents  $-V$  and minimum in the above equation works for the *fluid1*. It limits the flow of *fluid1* to the amount of the fluid in the left cell. The flux of the *fluid1*,  $\Delta F_1$ , is highlighted in the right bottom part of the left cell. For the *fluid2* the above equation makes value  $CF_2$  larger than 0. It accounts for the additional flux of the *fluid2* shown under the horizontal dashed line. It compensates the lack of the *fluid1*. At the end the fluids will flow through the face, and the hole process is iterated for all simulation cells.

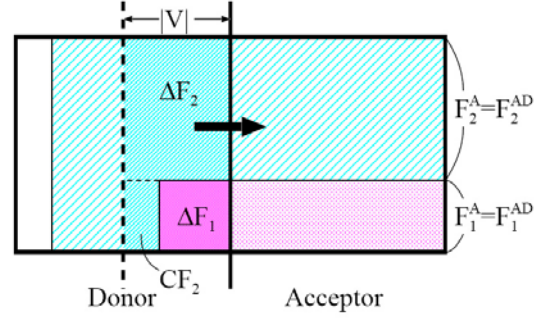


Figure 2. Illustration of  $\Delta F_n$  and  $CF_n$ .

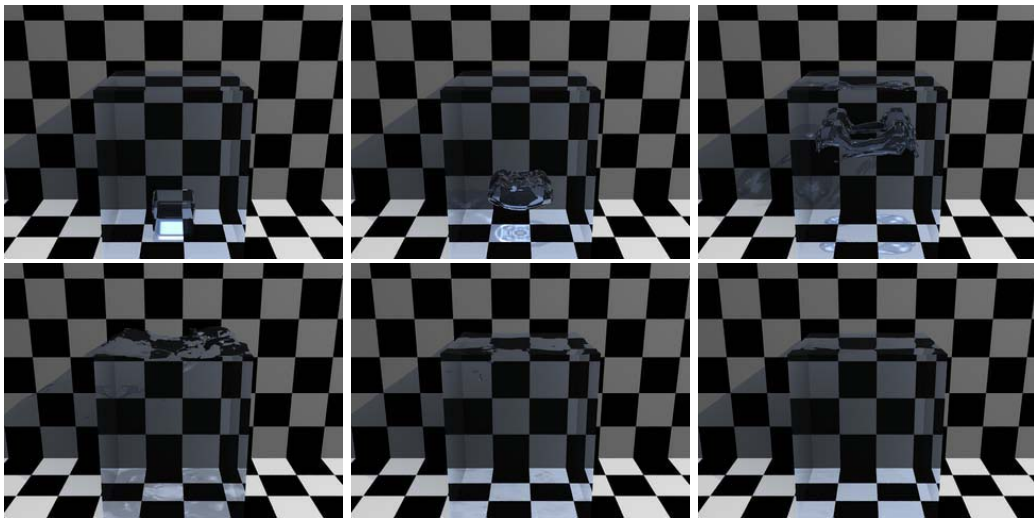
## 5 Results

The input to the simulator is the simulation environment defined by the size of simulation grid, fluids properties and placement of fluids and obstacles in grid cells. State of the simulation is calculated at every simulation time step and the polygonizer constructs the surfaces of fluids composed of triangles.

We demonstrate the implementation of proposed algorithms in our custom system on an example including two fluids, heavy and light one. Densities of the fluids in simulation are 1.0 and 0.1, respectively. The heavy fluid has viscosity of 1.0 and surface tension coefficient of 75, and values of the light fluid are 10.0 and 0, respectively. There is a gravity vector acting downward, (0.0, -980, 0.0). The size of the simulation grid cell is 0.5.

The simulation results were finally rendered by POV-Ray (Persistence of Vision Raytracer) [6], which added some optical effects using ray-tracing algorithm. All of the simulation examples are run on a PC with Athlon XP 1900+ and 1GB RAM.

In Figure 3 a block of the light fluid is at the bottom, and some of the upper cells are filled with the heavy fluid. The block behaves like a bubble in a liquid, changing its shape dynamically. First, it gets rounded, becomes ringed, and separates into small parts. When they reach to the surface, they disappear leaving small waves. Size of the simulation grid is  $24 \times 24 \times 24$  cells.



**Figure 3. Animation frames of a light fluid bubble rising in heavy fluid.**

## 6 Conclusion

The major advantage of simulation method is that the method can deal with many fluids simultaneously. The novel estimation of surface tension for multiple fluids and proposed extension of VOF method, prevents mass losses of the fluids during the simulation, completely.

Our simulation examples show many capabilities of our method. Animations of fluid interactions, and bubbles in liquid, are realistically simulated. The method is also applicable for fluid motions in which only single fluid plays a large part.

## Acknowledgments

This research was sponsored by grants from the EU-FP6-MC-040681- APCOCOS and VEGA 1/3083/06.

## References

- [1] J. U. Brackbill, D. B. Kothe, and C. Zemach. A continuum method for modeling surface tension. *J. Comput. Phys.*, 100(2):335–354, 1992.
- [2] M. Carlson, P. J. Mucha, R. I. B. V. Horn, and G. Turk. Melting and flowing. In *Proceedings of the ACM SIGGRAPH/Eurographics symposium on Computer animation*, pages 167–174, New York, 2002. ACM, ACM Press / ACM SIGGRAPH.
- [3] M. Carlson, P. J. Mucha, and G. Turk. Rigid fluid: animating the interplay between rigid bodies and fluid. *ACM Trans. Graph.*, 23(3):377–384, 2004.
- [4] C. W. Hirt and B. D. Nichols. Volume of fluid (vof) method for the dynamics of free boundaries. *Journal of Computational Physics*, 39:201–225, 1981.
- [5] J.-M. Hong and C.-H. Kim. Animation of bubbles in liquid. *Computer Graphics Forum*, 22(3):253–253, 2003.
- [6] W. page. Pov-ray - the persistence of vision raytracer. In <http://www.povray.org/>, Jan. 2006.
- [7] K. Tanaka, H. Ueki, and A. Kunimatsu. The cubic interpolated level set method for realistic fluid animation. In *SIGGRAPH '03: Proceedings of the SIGGRAPH 2003 conference on Sketches & applications*, pages 1–1, New York, NY, USA, 2003. ACM Press.