

Project CFD–Lab

Free Surface Simulation

Group Number : 1

Krenz Lukas Reiser Andreas Kouroudis Ioannis

Supervisor
Menhorn Friedrich

August 1, 2017

Abstract

In industry there are many applications that require the simulation of a system of unmixable fluids with vastly different kinematic viscosity, separated by a discrete interface. This problem is commonly referred to as a free surface flow problem. In effect, the kinematic viscosity difference causes the first fluid, which is usually the industrially relevant one, to exert force on the second one, receiving a negligible reaction in return. The most common configuration of this sort is a liquid-gas system and is frequently encountered in applications such as dam breaking scenarios, material processes and wave simulations.

In this particular project, the phenomenon is simulated using a Lattice Boltzmann approach combined with volume of fluid method. To track the interface, the mass streamed in and out of the interface cells is calculated. We consider filled cells to have a volume equal to 1 and empty cells equal to 0. When a cell emptied, all the neighboring fluid cells became interface cells. Equally, when an interface cell was filled, all the neighboring empty cells became interface cells. The program was written in C++ utilizing omp parallelization.

1 Aspects of implementation

1.1 Boundary conditions

In addition to the previously implemented boundary conditions (Free slip, No slip, , etc) a new condition was implemented, namely a free surface boundary condition. The physical perspective of this condition is that the gas has a far greater kinematic viscosity and is consequently pushed without resistance by the liquid. Further, it is assumed that the gas pressure is atmospheric, which is used in place of density in the equilibrium function($\rho_A = 1$). If e_i from an interface cell points to an empty cell the following distribution is constructed:

$$f'_{\hat{i}}(x, t + \Delta t) = f_{\hat{i}}^{eq}(1, \mathbf{u}) + f_i^{eq}(1, \mathbf{u}) - f_i(\mathbf{x}, t) \quad (1)$$

However, constructing these distributions along only one side of the interface would result in a force imbalance. To rectify this, additional distributions have to be reconstructed during the streaming step.

If $(n * e_i) > 0$ f_i is reconstructed according to equation (1)

n is the vector normal to the interface and is computed as the central differences of the fluid fraction between neighboring cells

$$\mathbf{n} = \frac{1}{2} \begin{bmatrix} \epsilon(x_{j-1,k,l}) - \epsilon(x_{j+1,k,l}) \\ \epsilon(x_{j,k-1,l}) - \epsilon(x_{j,k+1,l}) \\ \epsilon(x_{j,k,l-1}) - \epsilon(x_{j,k,l+1}) \end{bmatrix} \quad (2)$$

1.2 gravity

As in our scenarios there is need of an accelerating force (gravity) the BGK lattice Boltzmann equation is transform according to (3)[TPR⁺06]

$$f_i(x, t + \Delta t) = (1 - \omega)f_i^{eq} + \omega f_i^{eq}(\rho(\mathbf{x}, t), \mathbf{u}(\mathbf{x}, t) + \frac{\mathbf{g}}{\omega}) \quad (3)$$

1.3 Mass exchange

In order to keep track of the interface two additional variables have to be stored for each cell, namely mass m and the density ρ . The mass exchanged between an interface and a fluid cell is found directly using their distributions.

$$\Delta m_i(x, t + \Delta t) = f_i(x + e_i, t) - f_i(x, t) \quad (4)$$

For interface-fluid cell the mass exchange must take into account the interface shared between them. During this step, there might still remain interface cells amidst either fluid or empty cells. These artifacts are corrected using the equation (5) in conjecture with the table ???. Since the mass lost from one cell is the mass gained from another, the mass exchange is a symmetric process.

$$\Delta m_i(x, t + \Delta t) = s_e \frac{e(x + \Delta t * e_i, t) + e(x, t)}{2} \quad (5)$$

standard cell at (x)	standard cell at ($x + e_i$)	no fluid nb at ($x + e_i$)	no empty nb at ($x + e_i$)
	$f_i(x_{nb}, t) - f(x, t)$	$f_i(x_{nb}, t)$	$-f_i(x, t)$
no fluid neighbors at (x)	$-f_i(x, t)$	$f_i(x_{nb}, t) - f(x, t)$	$-f_i(x, t)$
no empty neighbors at (x)	$f_i(x_{nb}, t)$	$f_i(x_{nb}, t)$	$f_i(x_{nb}, t) - f(x, t)$

Table 1: Interface Interface mass streaming s_e coefficient

1.4 Flag Update

After the mass calculation step, all the interface cells that have a density higher than 1 are assigned a temporal flag as Fluid cells, and their surrounding Empty cells become interface. Following the same principle, all previously interface cells with mass lesser or equal to zero become empty cells, and their surrounding fluid cells receive a temporal interface cell flag. The temporal flag serves to make the update order independent. As the former empty cells become interface they

are initialised with the average density and velocity of their surrounding cells, excluding empty ones, which would otherwise be included if the update was made at this point. In practice, to avoid constant emptying and filling of cells when they are close to changing their state, a tolerated offset is introduced, transforming the criterion in the following fashion.

$$M(x, t + \Delta t) > (1 + offset) * \rho(x, t + \Delta t) \quad (6)$$

$$M(x, t + \Delta t) < (0 - offset) * \rho(x, t + \Delta t) \quad (7)$$

In this case the used offset was 10^{-3} [Thü07]. Formerly empty cells that become interface cells are initialised with the average density and velocity of their surrounding cells.

1.5 Excess Mass

After the flagfield has been updated, excess or negative mass issues are tackled. All newly empty or filled cells possibly have mass greater than 1 or lesser than 0. This is corrected by distributing or taking mass from the nearby interface cells. The distribution is weighted according to the relative position of the cell direction to the normal vector of the current cell. Specifically:

$$\Delta m(x + e_i) = m^{ex} \frac{v_i}{v_{total}} \quad (8)$$

Where v_{total} is the sum of all weights and v_i is the weight of the neighboring cell calculated as follows:

for filled cells:

$$v_i = \begin{cases} \mathbf{n} * \mathbf{e}_i & \text{if } \mathbf{n} * \mathbf{e}_i > 0 \\ 0 & \text{if otherwise} \end{cases} \quad (9)$$

for empty cells

$$v_i = \begin{cases} -\mathbf{n} * \mathbf{e}_i & \text{if } \mathbf{n} * \mathbf{e}_i < 0 \\ 0 & \text{if otherwise} \end{cases} \quad (10)$$

1.6 Adaptive time step and turbulent model

To avoid instabilities and to acquire speedup during periods of rest an adaptive time step was used. If $|u_{max}| > \frac{5}{24}$ the new timestep becomes $\Delta t_n = \frac{4}{5} * \Delta t_{old}$. Consequently, all time related quantities must be updated as well to keep the particle distributions inside either the current or the neighbouring cells over the course of one time iteration. This is done according to the following equations:

$$s = \frac{\Delta t_n}{\Delta t_o} \quad (11)$$

$$\omega = \frac{1}{s(\frac{1}{\omega_o} - \frac{1}{2}) + \frac{1}{2}} \quad (12)$$

$$\rho_n = s(\rho_o - \rho_{med}) + \rho_{med} \quad (13)$$

$$u_n = su_o \quad (14)$$

$$g = s^2 * g_o \quad (15)$$

Time steps can also increase in periods of relative calmness. If $|u_{max}| < \frac{1}{6} * \frac{5}{4}$ the resize factor is $\frac{5}{4}$. The logic behind $\frac{1}{6}$ is that it is half the speed of sound. If a velocity increases beyond this threshold, f^{eq} will become negative. If $\omega < 1.99$, as this will cause negative equilibrium distributions and numerical instabilities. Some of the simulated scenarios generated a high degree of turbulence and quickly became unstable. To rectify this, a turbulence model was introduced. The local stress tensor is computed and subsequently used to modify the relaxation time according to the eddy viscosity. The Smagorinsky constant C is set to 0.03 [Thü07]. The non equilibrium stress tensor $\Pi_{\alpha\beta}$ is computed according to the equation (16)

$$\Pi_{\alpha,\beta} = \sum_{i=1}^{19} e_{i\alpha} e_{i\beta} (f_i - f_i^{eq}) \quad (16)$$

where α and β run over the three spatial direction. The intensity of the local stress tensor S is computed as

$$S = \frac{1}{6C^2} \left(\sqrt{\nu^2 + 18C^2} \sqrt{\sum_{\alpha=1}^3 \sum_{\beta=1}^3 \Pi_{\alpha,\beta}^2} - \nu \right) \quad (17)$$

and the modified relaxation time is given according to (18)

$$\tau_s = 3(\nu + C^2 S) + \frac{1}{2} \quad (18)$$

2 Results/Scenario

Two simple scenarios were chosen, and as can be seen produced very good physical results.

2.1 Dam Breaking

The dam breaking was simulated by having a large vertical mass of fluid at the start be released for one side of the geometry.

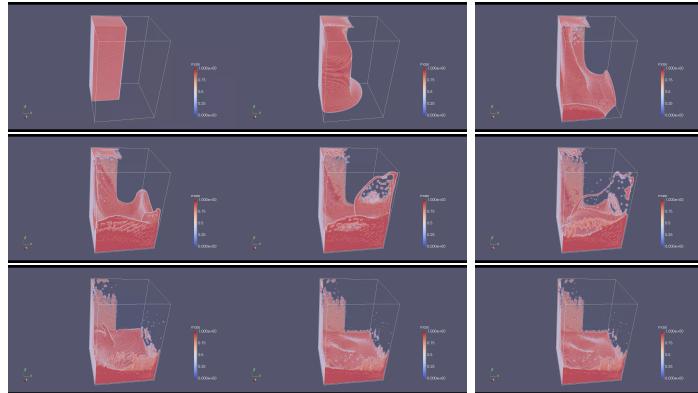


Figure 1: Breaking Dam

2.2 Falling Drop

A drop was left to fall in a pool of stationary water. As can be seen, the results look comparable to other sources and resemble the expected physical behaviour.

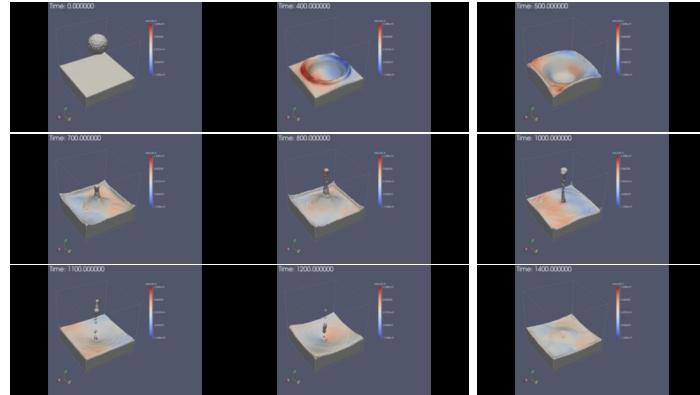


Figure 2: Falling Drop

References

- [Thü07] Nils Thürey. *Physically Based Animation of Free Surface Flows with the Lattice Boltzmann Method: Physikalische Animation Von Strömungen Mit Freien Oberflächen Mit Der Lattice-Boltzmann-Methode*. Verlag Dr. Hut, 2007.
- [TPR⁺06] Nils Thürey, Thomas Pohl, Ulrich Rüde, Markus Oechsner, and Carolin Körner. Optimization and stabilization of lbm free surface flow simulations using adaptive parameterization. *Computers & fluids*, 35(8):934–939, 2006.