2D Raindrop Simulation report

Lihui Ji

July 27, 2021

1 Finite Volume Method

1.1 Overview

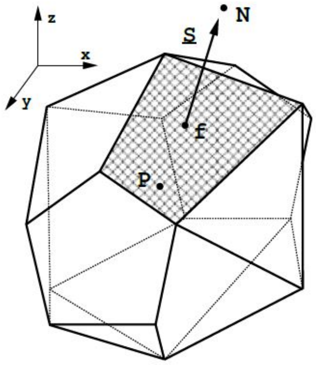

Figure 1: Control Volume in FVM

In finite volume method, the domain is divided to small control volumes. The value is stored in the center point of control volume (P in Figure 1). Each cell has neighbor cells which shares one face. One face can be shared by only two cells. The face (f) belongs to only one cell, which is the face owner (P), the cell that shares the face is denoted as neighbor (N). The face vector is perpendicular to it, pointing outside of owner. The cell can be arbitrary shape. Structured grid means the grids are orthogonal to each other, opposite to unstructured grid. (Santiago Márquez Damián)

Our raindrop simulation grids are structured grids. The center of cell is defined by:

$$V_P \mathbf{x}_P = \int_{V_P} \mathbf{x} dV \tag{1}$$

The PDE is integrated over control volume (so that we can track advection/diffsion between cells) and a small time step (Δt)

$$\int_t^{t+\Delta t} (\int_{V_P} (PDE_{LHS} - PDE_{RHS}))dVdt = 0 \tag{2}$$

1.1.1 Gauss's theorems

The Gauss's theorems will be used through procedure:

$$\int_V \nabla \cdot \mathbf{a} dV = \oint_{\partial V} \mathbf{a} \cdot d\mathbf{S} \tag{3}$$

$$\int_V \nabla \phi dV = \oint_{\partial V} \phi dS \tag{4}$$

$$\int_V \nabla \mathbf{a} dV = \oint_{\partial V} \mathbf{a} \otimes dS \tag{5}$$

1.1.2 Discretisation of spacial terms

The volume integral (Recall the definition of cell center):

$$\int_{V_P} \phi(\mathbf{x}) dV = \phi_P V_P \tag{6}$$

The termis under divergence operator:

$$\int_{V_P} \nabla \cdot \mathbf{a} dV = \oint_{\partial V_P} \mathbf{a} \cdot d\mathbf{S} = \sum_f (\int_f \mathbf{a} \cdot d\mathbf{S}) = \sum_f \mathbf{S} \cdot \mathbf{a}_f \tag{7}$$

The discretisation of convection term (where $\mathbf{F}$ is mass flux through the face:

$$\int_{V_P} \nabla \cdot (\rho \mathbf{U} \phi) dV = \sum_f \mathbf{S} \cdot (\rho \mathbf{U})_f \phi_f = \sum_f F \phi_f \tag{8}$$

Note in above equation, ϕ_f is the the flux value in the face, which should be calculated by interpolation between the owner and neighbour cell point.

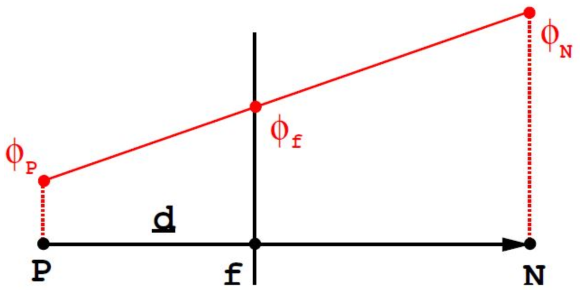

Figure 2: Face Interpolation

$$\phi_f = f_x \phi_P + (1 - f_x)\phi_N, \quad f_x = \frac{\overline{fN}}{\overline{PN}} \tag{9}$$

This is central differencing scheme, which is second order accurate. The problem is CD causes unphysical oscillations in the solution for convection-dominated problems, thus violating the boundedness of solution (Patankar, S.V.'s book.

An alternative discretization is upwind differencing (UD):

$$\phi_f = \begin{cases} \phi_P & \text{if } F \geqslant 0 \\ \\ \phi_N & \text{if } F < 0 \end{cases} \tag{10}$$

Blended differencing (BD) is an attempt to preserve both boundedness and accuracy:

$$\phi_f = (1-\gamma)(\phi_f)_{UD} + \gamma(\phi_f)_{CD} \tag{11}$$

There are different interpolation options in OpenFOAM, currently we use the upwind scheme in raindrop simulation.

The discretisation of diffusion term is:

$$\int_{V_P} \nabla \cdot (\rho\Gamma_\phi \nabla\phi) dV = \sum_f \mathbf{S} \cdot (\rho\Gamma_p hi \nabla\phi)_f = \sum_f (\rho\Gamma_\phi)_f \mathbf{S} \cdot (\nabla\phi)_f \tag{12}$$

For non-orthogonal grids, the face vector is decomposed to two vectors, one is parallel with while the other is perpendicular to the $\nabla\phi$ direction.

$$\mathbf{S} \cdot (\nabla\phi)_f = \Delta \cdot (\nabla\phi)_f + \mathbf{k} \cdot (\nabla\phi)_f = |\mathbf{S}| \frac{\phi_N - \phi_P}{|d|} + \mathbf{k} \cdot (\nabla\phi)_f \tag{13}$$

There are various ways to decompose non-orthogonal face vector, such as minimum correction approach, minimum correction approach, orthogonal correction approach, orthogonal correction approach and over-relaxed approach. The following figure demonstrates minimum correction approach.

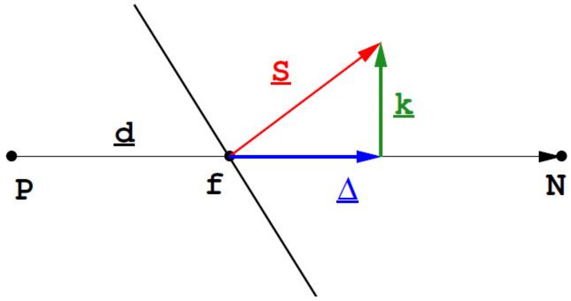

Figure 3: Minumum Correction Approach

1.1.3 Discretisation of temporal terms

There are three time schemes, forward Euler (explicit), backward Euler (implicit) and Crank Nicholson. Currently, we use the first order Euler implicit scheme as default:

$$\frac{\partial \phi}{\partial t} = \frac{\phi^n - \phi^0}{\Delta t} \tag{14}$$

where ϕ^0 is the previous value that has been calculated and ϕ^n is the value to be calculated in the current time.

1.2 Discretize Navier Stokes Equation

The velocity advection term (nonlinearity term) can be treated as follows:

$$\nabla \cdot (\mathbf{UU}) = \sum_f \mathbf{S} \cdot (\mathbf{U})_f (\mathbf{U})_f = \sum_f F \mathbf{U}_f \tag{15}$$

Rather than iterate over non-linear terms, the solver neglects the non-linearity effects. If the time step is short enough, the change between between consecutive solutions will be small and it is therefore possible to lag the non-linearity without any significant effect. With such procedure, the Navier-Stokes equation can be discretized to a system of algebraic equations:

$$a_P \mathbf{U}_P^n + \sum_N a_N \mathbf{U}_N^n = R_P \tag{16}$$

The above equation system has two unknown variables, velocity U and pressure P. The pressure equation is derived by continuity (incompressible flow). We begin with:

$$a_P \mathbf{U}_P = \mathbf{H(U)} - \nabla p \tag{17}$$

The $H(U)$ consistes of two parts: the "transport part" from neighboring cells and the "source part" including the transient term:

$$\mathbf{H(U)} = -\sum_f a_N \mathbf{U} + \frac{U^0}{\Delta t} \tag{18}$$

THe discretised form of the continuity equation is:

$$\nabla \cdot \mathbf{U} = \sum_f \mathbf{S} \cdot \mathbf{U}_f = 0 \tag{19}$$

Eqn. (17) is used to express $\mathbf{U}$:

$$\mathbf{U}_P = \frac{\mathbf{H(U)}}{a_p} - \frac{1}{a_P} \nabla p \tag{20}$$

Velocities on the cell face are expressed as the face interpolate of Eqn. (20):

$$\mathbf{U}_f = \left(\frac{\mathbf{H(U)}}{a_P} \right)_f - \left(\frac{1}{a_P} \right) (\nabla p)_f \tag{21}$$

Plug Eqn. (21) into Eqn. (19), the following form of the pressure equation is obtained:

$$\nabla \cdot \left(\frac{1}{a_P} \nabla p \right) = \sum_f \mathbf{S} \left(\frac{\mathbf{H}(\mathbf{U})}{a_P} \right)_f \tag{22}$$

1.3 Volume of Fluid Method

Two immiscible fluids are considered as one effective fluid throughout the domain, the physical properties of which are calculated as weighted averages based on the distribution of the liquid volume fraction, thus being equal to the properties of each fluid in their corresponding occupied regions and varying only across the interface,

$$\rho = \rho_l \gamma + \rho_g (1 - \gamma) \tag{23}$$

$$\mu = \mu_l \gamma + \mu_g (1 - \gamma) \tag{24}$$

where ρ_l and ρ_g are densities of liquid and gas, respectively. The model makes use of the two-fluid Eulerian model for two-phase flow, where phase fraction equations are solved separately for each individual phase; hence the equations for each of the phase fractions can be expressed as

$$\frac{\partial \gamma}{\partial t} + \nabla \cdot (\mathbf{U}_l \gamma) = 0 \tag{25}$$

$$\frac{\partial (1 - \gamma)}{\partial t} + \nabla \cdot [\mathbf{U}_g (1 - \gamma)] = 0 \tag{26}$$

Assuming that the contributions of the liquid and gas velocities to the evolution of the free surface are proportional to the corresponding phase fraction, and defining the velocity of the effective fluid in a VOF model as a weighted average:

$$\mathbf{U} = \gamma \mathbf{U}_l + (1 - \gamma) \mathbf{U}_g \tag{27}$$

Eqn. (25) can be rearranged and used as an evolution equation for the phase fraction γ,

$$\frac{\partial \gamma}{\partial t} + \nabla \cdot (\mathbf{U} \gamma) + \nabla \cdot [\mathbf{U}_r \gamma (1 - \gamma)] = 0 \tag{28}$$

where $\mathbf{U}_r = \mathbf{U}_l - \mathbf{U}_g$ is the vector of relative velocity, designated as the "compression velocity". This term is manually added and has no meaning in the continuum formation but is suitable to compress the interface in the discrete formation, especially when the interface is not sharp enough. The sharp means the fraction changes suddenly near the interface (ideally over one or two computational cells) (Ubbink and Issa).

In order to ensure stability of the solution procedure, the calculations are performed using a self-adapting time step which is adjusted at the beginning of the time iteration loop based on Courant number. Also, temporal subcycling is common in VOF-based methods that the convergence and stability of the solution procedure are very sensitive with respect to the equation for phase fraction.

Berberović and Hinsberg describes the compression terms in details. For the discretization of the compression term, the relative velocity at cell faces, formulated based on the maximum velocity magnitude at the interface region and its direction, is determined from the gradient of phase fraction as follows:

$$\mathbf{U}_{r,f} = \mathbf{n}_f min\left[C_\gamma \frac{|\phi|}{|\mathbf{S}_f|}, max\left(\frac{|\phi|}{|\mathbf{S}_f|}\right)\right] \tag{29}$$

where ϕ is face volume flux, and $\mathbf{n}_f$ is face unit normal flux, calculated at cell faces in the interface region using the phase fraction gradient at cell faces,

$$\mathbf{n}_f = \frac{(\nabla\gamma)_f}{|(\nabla\gamma)_f + \delta_n|} \cdot \mathbf{S}_f \tag{30}$$

A stabilization factor δ_n is used, which accounts for nonuniformity of the grid,

$$\delta_n = \frac{\epsilon}{\left(\frac{\sum_N V_i}{N}\right)^{1/3}} \tag{31}$$

where N is the number of computational cells and ϵ is a small parameter, set to 10^{-8} here.

The intensity of the free surface compression is controlled by the constant C_γ, which yields no contribution if set to zero, a conservative compression if the value is one and enhanced compression for values greater than one.

1.4 Solving processes

1.4.1 PISO Loop

In order to achieve a property coupling between velocity and pressure in interFoam is necessary to adapt the PISO loop to the momentum equation for interphase solver and derive a new pressure equation. Starting with momentum equation:

$$\frac{\rho\mathbf{U}}{\partial t} + \nabla \cdot (\rho\mathbf{U}\mathbf{U}) - \nabla \cdot (\mu\nabla\mathbf{U}) - (\nabla\mathbf{U}) \cdot \nabla\gamma = -\nabla p_d - \mathbf{g} \cdot \mathbf{x}\nabla\rho + \sigma_\kappa\nabla\gamma \tag{32}$$

Then the discretized form of momentum equation is obtained:

$$a_P^{\mathbf{u}}\mathbf{u}_P = \mathbf{H}(\mathbf{u}) - \nabla p_d - \mathbf{g} \cdot \mathbf{x}\nabla\rho + \sigma_\kappa\nabla\gamma \tag{33}$$

Isolating the velocity at cell centers:

$$\mathbf{u}_P = [a_P^{\mathbf{u}}]^{-1}\{[\mathbf{H}(\mathbf{u}) - \nabla p_d - \mathbf{g} \cdot \mathbf{x}\nabla\rho + \sigma_\kappa\nabla\gamma] - \nabla p_d\} \tag{34}$$

Replacing this velocity in continuity equation is possible to assemble a Poisson equation for pressure p_d:

$$\nabla \cdot \{[a_P^{\mathbf{u}}]^{-1}\nabla p_d\} = \nabla \cdot \{a_P^{\mathbf{u}}]^{-1}[\mathbf{H}(\mathbf{u}) - \nabla p_d - \mathbf{g} \cdot \mathbf{x}\nabla\rho + \sigma_\kappa\nabla\gamma]\} \tag{35}$$

Finally it is necessary to obtain fluxes that obeys continuity, it is achieved by:

$$F = -(a_P^{\mathbf{u}})^{-1}\mathbf{s}_f \cdot \nabla p_d + (a_P^{\mathbf{u}})^{-1}\mathbf{s}_f \cdot [\mathbf{H}(\mathbf{u}) - \mathbf{g} \cdot \mathbf{x}\nabla\rho + \sigma_\kappa\nabla\gamma] \tag{36}$$

1.4.2 Transient Solution Procedure

1. Set up the initial conditions for all field values.

2. Start the calculation of the new time-step values.

3. Assemble and solve the momentum predictor equation with the available face fluxes.

4. Go through the PISO loop until the tolerance for pressure-velocity system is reached. At this stage, pressure and velocity fields for the current time-step are obtained, as well as new set of conservative fluxes.

References

[1] Santiago Márquez Damián, *Description and utilization of interFoam multiphase solver*

[2] Edin Berberović, Nils P. van Hinsberg, Suad Jakirlić, Ilia V. Roisman, and Cameron Tropea, *Drop impact onto a liquid layer of finite thickness: Dynamics of the cavity evolution.*

[3] Patankar, S.V., *Numerical Heat Transfer and Fluid Flow, 1981.*

[4] Fabian Peng Karrholm, *Rhie-Chow interpolation in OpenFOAM.*

[5] O. Ubbink and R. I. Issay, *A Method for Capturing Sharp Fluid Interfaces on Arbitrary Meshes.*