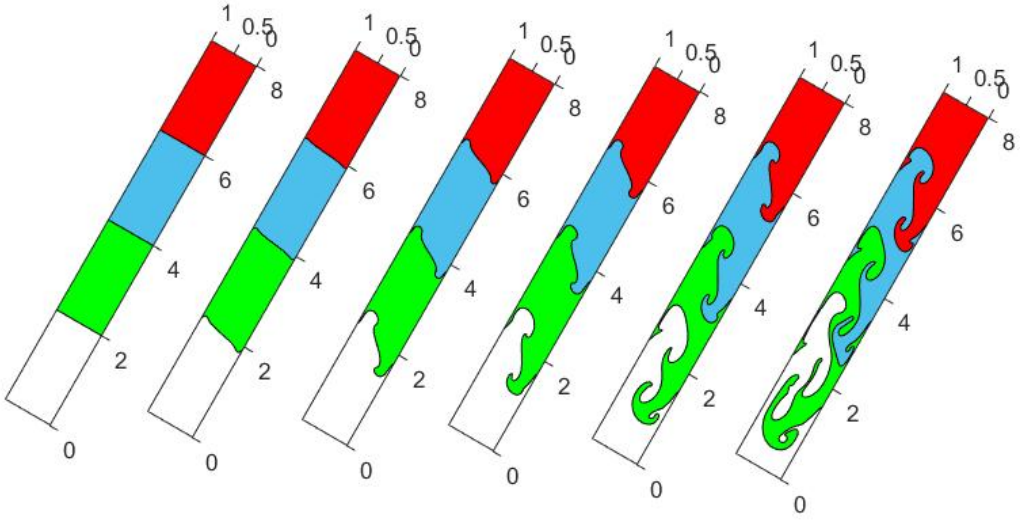
A tutorial for multi-phase flow simulation: Conservative Allen-Cahn phase-field fluid model of multi-phase fluid mixing in a tilted channel

This tutorial aims to introduce a simple and practical numerical framework for multi-phase incompressible and immiscible fluid simulation. The physical problem raised in this tutorial is N-component (N>=3) fluid mixing in a tilted channel. The mathematical model for capturing multi-phase fluid interfaces is the N-component conservative Allen-Cahn (CAC) equation, the fluid dynamics is governed by the incompressible Navier-Stokes equations.

The multi-component CAC equation is solved with a temporally first-order accurate semi-implicit operator splitting method. The incompressible Navier-Stokes equations with variable density and viscosity ratios are solved with a temporally first-order accurate pressure projection method. The governing equations are discretized in space with standard finite difference method. We describe the mathematical models and numerical implementations in details. **The C (for computation) and Matlab (for post-treatment) codes of a four-component case are provided.**

*It is worth noting that the model and algorithm described in this tutorial can be modified to simulate various multi-phase incompressible and immiscible fluids. The interested readers can modify this basic code according to their research requirements.

The numerical results are posted here:



Governing equations:

We consider the N-component incompressible, immiscible fluids in a two-dimensional inclined channel. In the gravitational field, we set the heavier fluids initially on the top of lighter fluids, with fluids 1 and N being the heaviest and lightest fluids, respectively. The angle between the channel and the horizontal direction is θ. L and H are the length and width of the channel, respectively. x and y denote the horizontal and vertical coordinates, respectively. The schematic diagram of the initial fluid distribution is shown in Fig. (1).

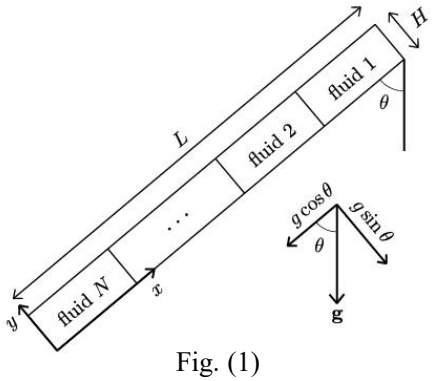


Fig. (1)

Let $c = (c_1, c_2, \cdots, c_N)$ be a vector-valued phase field function in which each order parameter c_k denotes mole fraction of the kth component in the mixture. Clearly, the total mole fractions satisfy the quality $c_1 + c_2 + \cdots + c_N = 1$. To study the multiphase fluid flow properties, we couple the incompressible NS equations and the multi-component CAC equations as follows:

$$\rho(\boldsymbol{c})\left(\frac{\partial \boldsymbol{u}}{\partial t} + \boldsymbol{u}\cdot\nabla\boldsymbol{u}\right) = -\nabla p + \nabla\cdot\left(\eta(\boldsymbol{c})(\nabla\boldsymbol{u} + (\nabla\boldsymbol{u})^T\right) + \rho(\boldsymbol{c})\boldsymbol{g}, \tag{1}$$

$$\nabla\cdot\boldsymbol{u} = 0, \tag{2}$$

$$\frac{\partial c_k(\boldsymbol{x},t)}{\partial t} + \nabla\cdot(c_k\boldsymbol{u}) = \Delta c_k + \frac{\alpha(\boldsymbol{c}(\boldsymbol{x},t))c_k(\boldsymbol{x},t)}{\epsilon^2} - \frac{f\,(c_k)}{\epsilon^2} + G(\boldsymbol{c}(\boldsymbol{x},t))\beta_k(t). \tag{3}$$

The boundary condition for velocity field is $\boldsymbol{u} = 0$, and linear boundary conditions are imposed for order parameter c_k,for $k = 1, 2\cdots, N$.

The specific symbol definitions are as follows: $\boldsymbol{u}$ is the fluid velocity, p is the pressure, $\boldsymbol{g} = (-gsin\theta, -gcos\theta)$ is the gravity term and g is the gravitational acceleration. ρ_k and η_k are the density and viscosity for the kth incompressible fluid, respectively. $\rho(\boldsymbol{c})$ is the variable density and $\eta(\boldsymbol{c})$ is the variable viscosity, where they are defined as $\rho(\boldsymbol{c}) = \sum_{k=1}^{N}\rho_k c_k$ and $\eta(\boldsymbol{c}) = \sum_{k=1}^{N}\eta_k c_k$, respectively. $F(c_k) = 0.25c_k^2(1-c_k)^2$ is the Helmholtz free-energy density for c_k, $\epsilon > 0$ is the gradient energy coefficient, $f(c_k) = F'(c_k) = c_k(c_k - 0.5)(c_k - 1)$, $\alpha(\boldsymbol{c}(\boldsymbol{x},t)) = \sum_{k=1}^{N} f\,(c_k)$. To make the AC equation conserve the total mass for each c_k, we introduce $G(\boldsymbol{c}(\boldsymbol{x},t))\beta_k(t)$ [18]. $G(\boldsymbol{c}(\boldsymbol{x},t))\beta_k(t)$ is a space-time dependent Lagrange multiplier, where $G(\boldsymbol{c}(\boldsymbol{x},t)) = \sum_{k=1}^{N}\sqrt{F\,(c_k)}$, $\beta_k(t) = \frac{\int_\Omega (f(c_k)-\alpha(\boldsymbol{c})c_k)d\boldsymbol{x} - \epsilon^2\int_{\partial\Omega}\nabla c_k\cdot\boldsymbol{n}ds}{\epsilon^2\int_\Omega G(\boldsymbol{c})d\boldsymbol{x}}$, where $\boldsymbol{n}$ is the unit normal vector to $\partial\Omega$.

We can proof that the total mass of each c_k is conserved:

$$\begin{aligned}\frac{d}{dt}\int_\Omega c_k d\boldsymbol{x} &= \int_\Omega \frac{\partial c_k}{\partial t} d\boldsymbol{x} \\ &= \int_\Omega \left(-\frac{f(c_k)}{\epsilon^2} + \Delta c_k + \frac{\alpha(\boldsymbol{c})c_k}{\epsilon^2} + G(\boldsymbol{c})\beta_k(t) - \nabla\cdot(c_k\boldsymbol{u}) \right) d\boldsymbol{x} \\ &= -\int_\Omega \left(\frac{f(c_k) - \alpha(\boldsymbol{c})c_k}{\epsilon^2} \right) d\boldsymbol{x} + \int_{\partial\Omega} \nabla c_k \cdot \boldsymbol{n} ds \\ &+ \beta_k(t)\int_\Omega G(\boldsymbol{c}) d\boldsymbol{x} - \int_{\partial\Omega} c_k u dy + \int_{\partial\Omega} c_k v dx \\ &= 0\end{aligned}$$

To obtain the dimensionless form of the Eqs.(1)-(3), the characteristic quantities of length (L_c), velocity (U_c), density (ρ_c), viscosity (η_c) are defined as $L_c = H$, $U_c = \sqrt{gH}$, $\rho_c = \rho_N$, $\eta_c = \eta_N$, respectively. Then the dimensionless variables are as follows:

$$\begin{aligned}&\boldsymbol{x}' = \frac{\boldsymbol{x}}{L_c}, \quad \boldsymbol{u}' = \frac{\boldsymbol{u}}{U_c}, \quad t' = \frac{tU_c}{L_c}, \quad \rho' = \frac{\rho}{\rho_c}, \\ &p' = \frac{p}{\rho_c U_c^2}, \quad \eta' = \frac{\eta}{\eta_c}, \quad \boldsymbol{g}' = \frac{\boldsymbol{g}}{g}.\end{aligned}$$

By substituting these variables into Eqs.(1)-(3) and drop the primes, we have

$$\rho(\boldsymbol{c})\left(\frac{\partial \boldsymbol{u}}{\partial t} + \boldsymbol{u}\cdot\nabla\boldsymbol{u}\right) = -\nabla p + \frac{1}{Re}\nabla\cdot\left(\eta(\boldsymbol{c})(\nabla\boldsymbol{u} + (\nabla\boldsymbol{u})^T\right) + \frac{\rho(\boldsymbol{c})}{Fr}\boldsymbol{g}, \tag{4}$$

$$\nabla\cdot\boldsymbol{u} = 0, \tag{5}$$

$$\frac{\partial c_k(\boldsymbol{x},t)}{\partial t} + \nabla\cdot(c_k\boldsymbol{u}) = \frac{\Delta c_k}{Pe} + \frac{\alpha(\boldsymbol{c}(\boldsymbol{x},t))c_k(\boldsymbol{x},t)}{\epsilon^2 Pe} - \frac{f(c_k)}{\epsilon^2 Pe} + \frac{G(\boldsymbol{c}(\boldsymbol{x},t))\beta_k(t)}{Pe}, 1 \le k \le N. \tag{6}$$

The dimensionless parameters are the Reynolds number, $Re = \rho_c U_c L_c/\eta_c$, the Froude number, $Fr = U_c/\sqrt{gL_c}$, and the Péclet number, $Pe = U_c L_c/\mu_c$.

By using the operator splitting approach, Eq. (6) is split into the following three simpler problems:

$$\frac{\partial c_k}{\partial t} = \frac{\Delta c_k}{Pe} + \frac{\alpha(\boldsymbol{c})c_k}{\epsilon^2 Pe} - \nabla\cdot(c_k\boldsymbol{u}), \tag{7}$$

$$\frac{\partial c_k}{\partial t} = -\frac{f(c_k)}{\epsilon^2 Pe}, \tag{8}$$

$$\frac{\partial c_k}{\partial t} = \frac{G(\boldsymbol{c})\beta_k(t)}{Pe}. \tag{9}$$

Numerical method:

Let the computational domain $\Omega = [a, b] \times [c, d]$ be partitioned into a uniform mesh with mesh spacing h. The center is located at $(x_i, y_j) = (a+(i-0.5)h, c+(j-0.5)h)$, for $i = 1, 2, \cdots, N_x, j = 1, 2, \cdots, N_y$, where N_x, N_y represent the numbers of cells in the $x-$ and $y-$ directions, respectively. Then the mesh vertices are denoted as $(x_{i+1/2}, y_{j+1/2}) = (a+ih, c+jh)$. Let T be the total time and N_t be the total number of time steps, then the uniform time step is defined to be $\Delta t = T/N_t$. We use Harlow and Welch's staggered marker-and-cell mesh [25] to define the velocities $u_{i+1/2,j}$,$v_{i,j+1/2}$ at $(x_{i+1/2}, y_j)$ and $(x_i, y_{j+1/2})$, respectively, and the pressure at the center of the grid (x_i, y_j). Let p_{ij}^n and $c_{k,ij}^n$ be the approximations of $p(x_i, y_j, n\Delta t)$ and $c_k(x_i, y_j, n\Delta t)$, respectively. Figure (**??**) shows the schematic illustration of MAC grid.

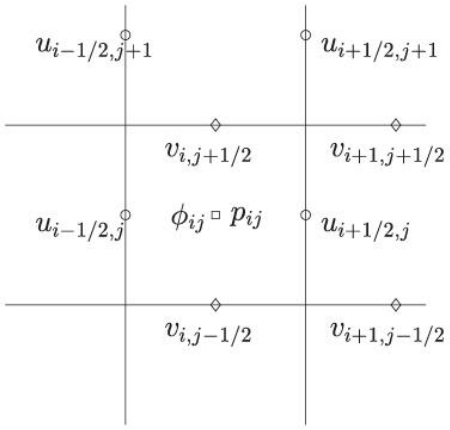


Fig 2. MAC grid

At the beginning of each time step, given $\boldsymbol{u}^n$,$\boldsymbol{c}^n$and p^n, we want to find $\boldsymbol{u}^{n+1}$, $\boldsymbol{c}^{n+1}$ and p^{n+1}. The main procedures are as follows:

Step 1. Initialize u^0 to be the divergence-free velocity field and $\boldsymbol{c}^0$. Note that we only need to initialize c_1, c_2, $\cdots$, c_{N-1} since $\sum_{k=1}^{N} c_k = 1$.

Step 2. Without the pressure gradient term, we solve the intermediate velocity field, $\tilde{\boldsymbol{u}}$. The discretization form of the equation is

$$\frac{\tilde{\boldsymbol{u}} - \boldsymbol{u}^n}{\Delta t} = \frac{1}{\rho^n Re} \nabla_d \cdot \left[\eta^n \left(\nabla_d \boldsymbol{u}^n + (\nabla_d \boldsymbol{u}^n)^T \right) \right] + \frac{1}{Fr} \boldsymbol{g} - (\boldsymbol{u} \cdot \nabla_d \boldsymbol{u})^n . \tag{10}$$

We get $\tilde{u}_{i+\frac{1}{2},j}$ and $\tilde{v}_{i,j+\frac{1}{2}}$:

$$\begin{aligned}
\tilde{u}_{i+1/2,j} = {} & u_{i+1/2,j}^n - \Delta t \left(u u_x + v u_y \right)_{i+1/2,j}^n - \frac{\Delta t}{Fr} sin\theta \\
& + \Delta t \frac{\left(2(\eta_{i+1,j}(u_{i+3/2,j}^n - u_{i+1/2,j}^n) - \eta_{i,j}(u_{i+1/2,j}^n - u_{i-1/2,j}^n) \right)}{h^2 \rho_{i+1/2,j}^n Re} \\
& + \Delta t \frac{\eta_{i+1/2,j+1/2}(u_{i+1/2,j+1}^n - u_{i+1/2,j}^n) - \eta_{i+1/2,j-1/2}(u_{i+1/2,j+1}^n - u_{i+1/2,j-1}^n)}{h^2 \rho_{i+1/2,j}^n Re} \\
& + \Delta t \frac{\eta_{i+1/2,j+1/2}(v_{i+1,j+1/2}^n - v_{i,j+1/2}^n) - \eta_{i+1/2,j-1/2}(v_{i+1,j-1/2}^n - v_{i,j-1/2}^n)}{h^2 \rho_{i+1/2,j}^n Re},
\end{aligned}$$

$$\tilde{v}_{i,j+1/2} = v^n_{i,j+1/2} - \Delta t\,(uv_x + vv_y)^n_{i,j+1/2} - \frac{\Delta t}{Fr}cos\theta$$
$$+ \Delta t\frac{\Big(2(\eta_{i,j+1}(v^n_{i,j+3/2} - v^n_{i,j+1/2}) - \eta_{i,j}(v^n_{i,j+1/2} - v^n_{i,j-1/2})\Big)}{h^2\rho^n_{i,j+1/2}Re}$$
$$+ \Delta t\frac{\eta_{i+1/2,j+1/2}(v^n_{i+1,j+1/2} - v^n_{i,j+1/2}) - \eta_{i-1/2,j+1/2}(v^n_{i,j+1/2} - v^n_{i-1,j+1/2})}{h^2\rho^n_{i,j+1/2}Re}$$
$$+ \Delta t\frac{\eta_{i+1/2,j+1/2}(u^n_{i+1/2,j+1} - u^n_{i+1/2,j}) - \eta_{i-1/2,j+1/2}(u^n_{i-1/2,j+1} - u^n_{i-1/2,j})}{h^2\rho^n_{i,j+1/2}Re}, \qquad (11)$$

where $\eta_{i+1/2,j+1/2} = \frac{\eta_{i,j+1}+\eta_{i,j}+\eta_{i+1,j}+\eta_{i+1,j+1}}{4}$,$\eta_{i+1/2,j-1/2} = \frac{\eta_{i,j}+\eta_{i,j-1}+\eta_{i+1,j}+\eta_{i+1,j-1}}{4}$,$\eta_{i-1/2,j+1/2} = \frac{\eta_{i-1,j}+\eta_{i-1,j+1}+\eta_{i,j}+\eta_{i,j+1}}{4}$,$\rho^n_{i+1/2,j} = \frac{\rho_{i,j}+\rho_{i+1,j}}{2}$,$\rho^n_{i,j+1/2} = \frac{\rho_{i,j+1}+\rho_{i,j}}{2}$ and the advection terms,$(uu_x + vu_y)^n_{i+1/2,j}$ and $(uv_x + vv_y)^n_{i,j+1/2}$ are processed as:

$$(uu_x + vu_y)^n_{i+1/2,j} = u^n_{i+1/2,j}\bar{u}^n_{x_{i+1/2,j}} + \frac{v^n_{i,j-1/2} + v^n_{i+1,j-1/2} + v^n_{i,j+1/2} + v^n_{i+1,j+1/2}}{4}\bar{u}^n_{y_{i+1/2,j}},$$
$$(uv_x + vv_y)^n_{i,j+1/2} = \frac{u^n_{i-1/2,j} + u^n_{i-1/2,j+1} + u^n_{i+1/2,j} + u^n_{i+1/2,j+1}}{4}\bar{v}^n_{x_{i,j+1/2}} + v^n_{i,j+1/2}\bar{v}^n_{y_{i,j+1/2}}.$$

We compute $\bar{u}^n_{x_{i+1/2,j}}$ and $\bar{u}^n_{y_{i+1/2,j}}$ with the second-order ENO method [26]. We first set intermediate variables k_1, k_2, d_1, d_2, then we have

$$k_1 = \begin{cases} i & \text{if } u^n_{i+1/2,j} + u^n_{i+3/2,j} \geq 0, \\ i+1 & \text{else,} \end{cases}$$

$$k_2 = \begin{cases} j & \text{if } v^n_{i,j+1/2} + v^n_{i+1,j+1/2} \geq 0, \\ j+1 & \text{else,} \end{cases}$$

$$d_1 = \begin{cases} \frac{u_{k+1/2,j}-u_{k-1/2,j}}{h} & \text{if } |\frac{u_{k+1/2,j}-u_{k-1/2,j}}{h}| \leq |\frac{u_{k+3/2,j}-u_{k+1/2,j}}{h}|, \\ \frac{u_{k+3/2,j}-u_{k+1/2,j}}{h} & \text{else,} \end{cases}$$

$$d_2 = \begin{cases} \frac{u_{i+1/2,k}-u_{i+1/2,k-1}}{h} & \text{if } |\frac{u_{i+1/2,k}-u_{i+1/2,k-1}}{h}| \leq |\frac{u_{i+1/2,k+1}-u_{i+1/2,k}}{h}|, \\ \frac{u_{i+1/2,k+1}-u_{i+1/2,k}}{h} & \text{else.} \end{cases}$$

So

$$\bar{u}^n_{x_{i+1/2,j}} = \begin{cases} u^n_{i+1/2,j} + \frac{h*d_1}{2} & \text{if } u^n_{i+1/2,j} + u^n_{i+3/2,j} \geq 0, \\ u^n_{i+3/2,j} - \frac{h*d_1}{2} & \text{else,} \end{cases}$$

$$\bar{u}^n_{y_{i+1/2,j}} = \begin{cases} u^n_{i+1/2,j} + \frac{h*d_2}{2} & \text{if } v^n_{i,j+1/2} + v^n_{i+1,j+1/2} \geq 0, \\ u^n_{i+1/2,j+1} - \frac{h*d_2}{2} & \text{else.} \end{cases}$$

$\bar{v}^n_{x_{i,j+1/2}}$ and $\bar{v}^n_{y_{i,j+1/2}}$ are computed similarly.

Step 3. Solve p^{n+1}. With the application of

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

$$\nabla_d\cdot\boldsymbol{u}^{n+1}=0, \tag{13}$$

we get a Poisson equation for the pressure at $(n+1)$ time step:

$$\nabla_d\cdot\left(\frac{1}{\rho^n}\nabla_d p^{n+1}\right)=\frac{1}{\Delta t}\nabla_d\cdot\tilde{\boldsymbol{u}}. \tag{14}$$

We discreticize Eq. (14) as

$$\begin{aligned}\nabla_d\cdot\left(\frac{1}{\rho^n}\nabla_d p^{n+1}\right)=&\frac{\frac{1}{\rho^n_{i+1/2,j}}p^{n+1}_{i+1,j}+\frac{1}{\rho^n_{i-1/2,j}}p^{n+1}_{i-1,j}+\frac{1}{\rho^n_{i,j+1/2}}p^{n+1}_{i,j+1}+\frac{1}{\rho^n_{i,j-1/2}}p^{n+1}_{i,j-1}}{h^2}\\&-\frac{\frac{1}{\rho^n_{i+1/2,j}}+\frac{1}{\rho^n_{i-1/2,j}}+\frac{1}{\rho^n_{i,j+1/2}}+\frac{1}{\rho^n_{i,j-1/2}}}{h^2}p^{n+1}_{ij},\end{aligned} \tag{15}$$

$$\nabla_d\cdot\tilde{\boldsymbol{u}}_{ij}=\frac{\tilde{u}_{i+1/2,j}-\tilde{u}_{i-1/2,j}}{h}+\frac{\tilde{v}_{i,j+1/2}-\tilde{v}_{i,j-1/2}}{h}. \tag{16}$$

The boundary conditions for pressure are

$$\boldsymbol{n}\cdot\nabla_d p^{n+1}=\boldsymbol{n}\cdot\left(-\rho^n\frac{\boldsymbol{u}^{n+1}-\boldsymbol{u}^n}{\Delta t}-\rho^n(\boldsymbol{u}\cdot\nabla_d\boldsymbol{u})^n+\frac{1}{Re}\Delta_d\boldsymbol{u}^n+\frac{\rho^n}{Fr}\boldsymbol{g}\right),$$

where $\boldsymbol{n}$ is the unit normal vector to $\partial\Omega$.

The periodic boundary condition to vertical boundaries and no slip boundary condition to the top and bottom domain are imposed, which leads to

$$\boldsymbol{n}\cdot\nabla_d p^{n+1}=\boldsymbol{n}\cdot\frac{\rho^n}{Fr}\boldsymbol{g},\quad \text{i.e. } \frac{\partial p}{\partial y}=-\frac{\rho^n}{Fr}cos\theta,\quad \text{at } y=0\quad \text{and}\quad y=L_y.$$

To get the unique solution of Poisson equation, we need to update the pressure [27, 28] as

$$p^{n+1}_{ij}=p^{n+1}_{ij}-\frac{1}{N_xN_y}\sum_{i=1}^{N_x}\sum_{j=1}^{N_y}p^{n+1}_{ij}. \tag{17}$$

We use a multigrid method[29] to solve Eq. (14) with boundary conditions(17).

Step 4. Solve the divergence-free velocities u^{n+1} and v^{n+1}.

$$u^{n+1}_{i+1/2,j}=\tilde{u}_{i+1/2,j}-\frac{\Delta t}{\rho^n_{i+1/2,j}h}\left(p_{i+1,j}-p_{ij}\right),\quad v^{n+1}_{i,j+1/2}=\tilde{v}_{i,j+1/2}-\frac{\Delta t}{\rho^n_{i,j+1/2}h}\left(p_{i,j+1}-p_{ij}\right). \tag{18}$$

Step 5. Solve the multi-component CAC equations.

We only need to solve c^{n+1}_1,c^{n+1}_2,$\cdots$,c^{n+1}_{N-1} since $\sum_{k=1}^{N}c^{n+1}_k=1$.First a semi-implicit method is applied to solve Equation (7),for $k=1,2,\cdots,N-1$:

$$\frac{c^{n+1,1}_{k,ij}-c^n_{k,ij}}{\Delta t}=\frac{\Delta_d c^{n+1,1}_{k,ij}}{Pe}+\frac{\alpha\left(\boldsymbol{c}^n_{ij}\right)c^n_{k,ij}}{\epsilon^2 Pe}-\nabla_d\cdot\left(c_k\boldsymbol{u}\right)^n_{ij}, \tag{19}$$

where $\Delta_d c^n_{ij}=(c^n_{i-1,j}+c^n_{i+1,j}+c^n_{i,j-1}+c^n_{i,j+1}-4c^n_{i,j})/h^2$.

Next, we use the method of separation of variables to solve Eq. (8) and the numerical solution follows as

$$c_{k,ij}^{n+1,2} = 0.5 + \frac{c_{k,ij}^{n+1,1} - 0.5}{\sqrt{e^{\frac{-\Delta t}{2\epsilon^2 Pe}} + \left(2c_{k,ij}^{n+1,1} - 1\right)^2 \left(1 - e^{\frac{-\Delta t}{2\epsilon^2 Pe}}\right)}}, for k = 1, 2, \cdots, N-1. \tag{20}$$

Finally, the discretization form of Eq. (9) is given below: for $k = 1, 2, \cdots, N-1$,

$$\frac{c_{k,ij}^{n+1} - c_{k,ij}^{n+1,2}}{\Delta t} = \frac{G\left(\boldsymbol{c}_{ij}^{n+1,2}\right)\beta_k^{n+1,2}}{Pe}. \tag{21}$$

To guarantee that the law of mass conservation holds for each c_k:

$$\begin{aligned}\sum_{i=1}^{N_x}\sum_{j=1}^{N_y} c_{k,ij}^{n+1} &= \sum_{i=1}^{N_x}\sum_{j=1}^{N_y} c_{k,ij}^{n+1,2} + \Delta t \frac{\beta_k^{n+1,2}}{Pe}\sum_{i=1}^{N_x}\sum_{j=1}^{N_y} G\left(\mathbf{c}_{ij}^{n+1,2}\right) \\ &= \sum_{i=1}^{N_x}\sum_{j=1}^{N_y} c_{k,ij}^{n},\end{aligned}$$

we take

$$\frac{\beta_k^{n+1,2}}{Pe} = \frac{\sum_{i=1}^{N_x}\sum_{j=1}^{N_y}\left(c_{k,ij}^{n} - c_{k,ij}^{n+1,2}\right)}{\Delta t \sum_{i=1}^{Nx}\sum_{j=1}^{Ny} G\left(\boldsymbol{c}_{ij}^{n+1,2}\right)}.$$

Substituting the calculated $\frac{\beta_k^{n+1,2}}{Pe}$ into Eq. (21), we can calculate c_k^{n+1}, for $k = 1, 2, \cdots, N-1$. Then we get $c_N^{n+1} = 1 - \sum_{i=1}^{N-1} c_k^{n+1}$.

These complete the calculations in one time step.

Note: Although the surface tension is absent in present simulation, we still provide the multi-component surface tension formulation is provided. For some details of multi-component continuous surface tension model, please refer to

Comput. Methods Appl. Mech. Engrg. 198 (2009) 3105–3112

Contents lists available at ScienceDirect

Comput. Methods Appl. Mech. Engrg.

journal homepage: www.elsevier.com/locate/cma

A generalized continuous surface tension force formulation for phase-field models for multi-component immiscible fluid flows

Junseok Kim *

Department of Mathematics, Korea University, Seoul 136-701, Republic of Korea

The multi-component surface tension is defined as:

$$\mathbf{SF}(\mathbf{c}) = \sum_{i=1}^{N-1}\left(\sum_{j=i+1}^{N}\frac{\sigma_{ij}}{2}\left[\mathbf{sf}(c_i) + \mathbf{sf}(c_j)\right]\delta(c_i, c_j)\right).$$

*** It is worth noting that the surface tension is necessary if you want to simulate droplets.**

Reference:

[18] Kim, J. and H.G. Lee, A new conservative vector-valued AllenCahn equation and its fast numerical method. Computer physics communications, 2017. 221: p. 102-108.

[25] Harlow FH, Welch JE. Numerical calculation of time-dependent viscous incompressible flow of fluid with free surface. Physics of Fluids 1965; 8:21822189.

[26] C. Shu, S. Osher, Efficient implementation of essentially non-oscillatory shock capturing schemes II, J. Comput. Phys. 83 (1989) 32–78.

[27] M.O. Deville, P.F. Fischer, E.H. Mund, High Order Methods for Incompressible Fluid Flow. Cambridge University Press, Cambridge, 2002.

[28] J.A. Escobar-Vargas, P.J. Diamessis, C.F. Van Loan, The numerical solution of the pressure Poisson equation for the incompressible NavierStokes equations using a quadrilateral spectral multidomain penalty method. J. Comp. Phys. (2011)

[29] Trottenberg U, Oosterlee C, Schller A. Multigrid. Academic press: London, 2001.

For some details of linear multigrid algorithm please refer to

J. KSIAM Vol.19, No.2, 103–121, 2015 http://dx.doi.org/10.12941/jksiam.2015.19.103

[SPECIAL SECTION : **Software and Education**]

NUMERICAL IMPLEMENTATION OF THE TWO-DIMENSIONAL INCOMPRESSIBLE NAVIER–STOKES EQUATION

YONGHO CHOI, DARAE JEONG, SEUNGGYU LEE AND JUNSEOK KIM†

DEPARTMENT OF MATHEMATICS, KOREA UNIVERSITY, SEOUL 136-713, REPUBLIC OF KOREA

The implementations of linear multigrid are as follows:

2.2. **Linear multigrid V-cycle algorithm.** In this section we describe the algorithm of the linear multigrid method for solving the discrete system in Eq. (2.7). In order to explain clearly the steps taken during a single V-cycle, we focus on a numerical solution on a 8×8 mesh. We define discrete domains, $\Omega_3,\ \Omega_2,\ \Omega_1$, and Ω_0, where

$$\Omega_k = \{(x_{k,i} = (i - 0.5)h_k, y_{k,j} = (j - 0.5)h_k) | 1 \leq i, j \leq 2^{k+1} \text{ and } h_k = 2^{3-k}h\}.$$

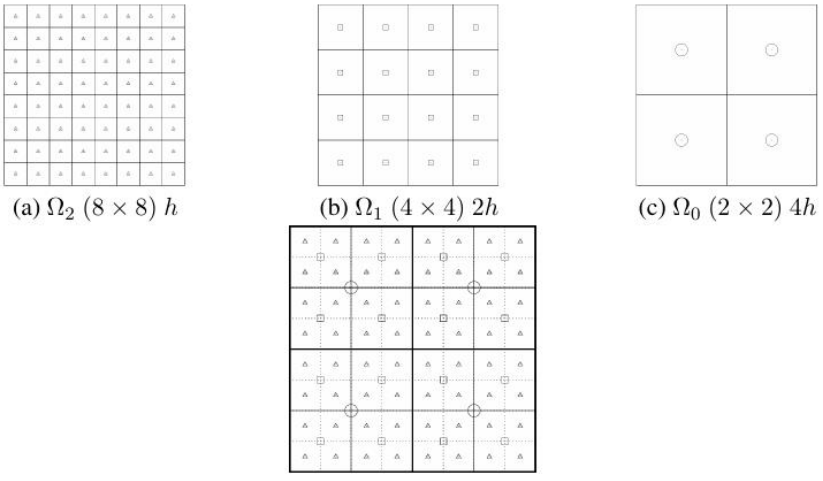
Ω_{k-1} is coarser than Ω_k by a factor of 2. The multigrid solution of the discrete Eq. (2.7) makes use of a hierarchy of meshes ($\Omega_3,\ \Omega_2,\ \Omega_1$, and Ω_0) created by successively coarsening the original mesh, Ω_3 as shown in Fig. 2. A pointwise Gauss–Seidel relaxation scheme is used as the smoother in the multigrid method. The algorithm of the multigrid method for solving Eq. (2.7) is as follows. We rewrite the Eq. (2.7) by

$$L_3(p^{n+1}_{3,ij}) = f_{3,ij} \text{ on } \Omega_3, \tag{2.17}$$

where

$$L_3(p^{n+1}_{3,ij}) = \Delta_d p^{n+1}_{3,ij} \text{ and } f_{3,ij} = \frac{1}{\Delta t}\nabla_d \cdot \tilde{\mathbf{u}}^n_{3,ij}.$$

Given the numbers, ν_1 and ν_2, of pre- and post- smoothing relaxation sweeps, an iteration step for the multigrid method using the V-cycle is formally written as follows [26]. That is, starting an initial condition p^0_3, we want to find p^n_3 for $n = 1, 2, \cdots$. Given p^n_3, we want to find the p^{n+1}_3 solution that satisfies Eq. (2.7). At the very beginning of the multigrid cycle the solution from the previous time step is used to provide an initial guess for the multigrid procedure. First, let $p^{n+1,0}_3 = p^n_3$.



(a) Ω_2 (8×8) h (b) Ω_1 (4×4) $2h$ (c) Ω_0 (2×2) $4h$

(d)

FIGURE 2. (a), (b), and (c) are a sequence of coarse grids starting with $h = L/N_x$. (d) is a composition of grids, $\Omega_2,\ \Omega_1$, and Ω_0.

Multigrid cycle

$$p^{n+1,m+1}_k = MGcycle(k, p^{n+1,m}_k, L_k, f_k, \nu_1, \nu_2).$$

That is, $p_k^{n+1,m}$ and $p_k^{n+1,m+1}$ are the approximations of p_k^{n+1} before and after an MGcycle. Now, define the MGcycle.

Step 1) Presmoothing

$$\bar{p}_k^{n+1,m} = SMOOTH^{\nu_1}(p_k^{n+1,m}, L_k, f_k),$$

means performing ν_1 smoothing steps with the initial approximation $p_k^{n+1,m}$, source terms f_k, and a $SMOOTH$ relaxation operator to get the approximation $\bar{p}_k^{n+1,m}$. Here, we derive the smoothing operator in two dimensions.

Now we derive a Gauss–Seidel relaxation operator. First, we rewrite Eq. (2.17) as

$$p_{k,ij}^{n+1} = \left[-f_{k,ij} + \frac{p_{i+1,j}^{n+1} + p_{i-1,j}^{n+1} + p_{i,j-1}^{n+1} + p_{i,j-1}^{n+1}}{h^2}\right] \Bigg/ \left(\frac{4}{h^2}\right). \tag{2.18}$$

Next, we replace $p_{k,\alpha\beta}^{n+1}$ in Eq. (2.18) with $\bar{p}_{k,\alpha\beta}^{n+1,m}$ if ($\alpha < i$) or ($\alpha = i$ and $\beta \le j$), otherwise with $p_{k,\alpha\beta}^{n+1,m}$, i.e.,

$$\bar{p}_{k,ij}^{n+1,m} = \left[-f_{k,ij} + \frac{p_{i+1,j}^{n+1,m} + \bar{p}_{i-1,j}^{n+1,m} + p_{i,j+1}^{n+1,m} + \bar{p}_{i,j-1}^{n+1,m}}{h^2}\right] \Bigg/ \left(\frac{4}{h^2}\right). \tag{2.19}$$

Therefore, in a multigrid cycle, one smooth relaxation operator step consists of solving Eq. (2.19) given above for $1 \le i \le 2^{k-3}N_x$ and $1 \le j \le 2^{k-3}N_y$.

Step 2) Coarse grid correction

• Compute the defect: $\bar{d}_k^m = f_k - L_k(\bar{p}_k^{n+1,m})$.

• Restrict the defect and $\bar{p}_k^m$: $\bar{d}_{k-1}^m = I_k^{k-1}\bar{d}_k^m$

The restriction operator I_k^{k-1} maps k-level functions to $(k-1)$-level functions.

$$\begin{aligned} d_{k-1}(x_i, y_j) = I_k^{k-1} d_k(x_i, y_j) = \frac{1}{4}[&d_k(x_{i-\frac{1}{2}}, y_{j-\frac{1}{2}}) + d_k(x_{i-\frac{1}{2}}, y_{j+\frac{1}{2}}) \\ &+ d_k(x_{i+\frac{1}{2}}, y_{j-\frac{1}{2}}) + d_k(x_{i+\frac{1}{2}}, y_{j+\frac{1}{2}})]. \end{aligned}$$

• Compute an approximate solution $\hat{p}_{k-1}^{n+1,m}$ of the coarse grid equation on Ω_{k-1}, i.e.,

$$L_{k-1}(p_{k-1}^{n+1,m}) = \bar{d}_{k-1}^m. \tag{2.20}$$

If $k = 1$, we use a direct or fast iteration solver for Eq. (2.20). If $k > 1$, we solve Eq. (2.20) approximately by performing k-grid cycles using the zero grid function as an initial approximation:

$$\hat{v}_{k-1}^{n+1,m} = MGcycle(k-1, 0, L_{k-1}, \bar{d}_{k-1}^m, \nu_1, \nu_2).$$

• Interpolate the correction: $\hat{q}_k^{n+1,m} = I_{k-1}^k \hat{q}_{k-1}^{n+1,m}$. Here, the coarse values are simply transferred to the four nearby fine grid points, i.e., $q_k(x_i, y_j) = I_{k-1}^k q_{k-1}(x_i, y_j) = q_{k-1}(x_{i+\frac{1}{2}}, y_{j+\frac{1}{2}})$ for the i and j odd-numbered integers.

• Compute the corrected approximation on Ω_k

$$p_k^{m,\,\text{after } CGC} = \bar{p}_k^{n+1,m} + \hat{q}_k^{n+1,m}.$$

Step 3) Postsmoothing: $p_k^{n+1,m+1} = SMOOTH^{\nu_2}(p_k^{m,\,\text{after } CGC}, L_k, f_k)$.

This completes the description of a MGcycle. Then, for unique solution, we redefine the pressure using Eq. (2.9) as follows:

$$p_{ij}^{n+1,m+1} = p_{ij}^{n+1,m+1} - \frac{1}{N_x N_y}\sum_{i=1}^{N_x}\sum_{j=1}^{N_y} p_{ij}^{n+1,m+1}. \tag{2.21}$$

One MGcycle step stops if the consequence error $\|p^{n+1,m+1} - p^{n+1,m}\|_\infty$ is smaller than a given tolerance, where

$$\|p\|_\infty = \max_{1 \le i \le N_x, 1 \le j \le N_y} |p_{ij}|.$$