The THINC-WLIC VOF method for PHOENICS

THINC METHOD

In the following, we will present a simple and practical scheme for capturing moving interface in the multi-fluid simulation. By using hyperbolic tangent function, we can devise a conservative, oscillation-less and smearing-less scheme which is called THINC (Tangent of Hyperbola for INterface Capturing) scheme. This scheme shows competitive accuracy compared to most existing methods without any geometry reconstruction. Multi-dimensional computing is conducted by WLIC (weighted line interface calculation) method.

Original THINC scheme:

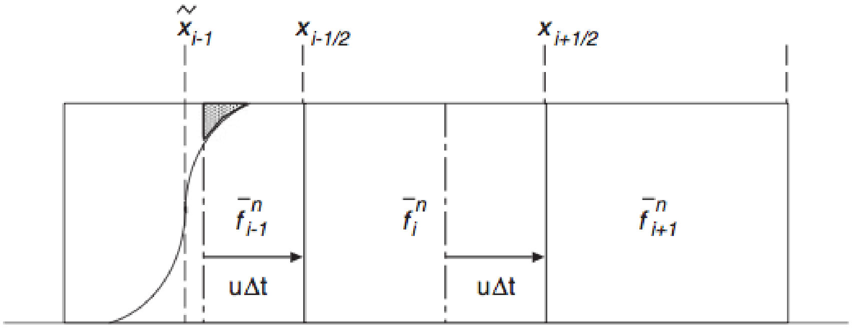
The VOF function is based on the following advection equation:

$$\frac{\partial \phi}{\partial \mathrm{t}} + \nabla \cdot (u\phi) - \phi \nabla \cdot u = 0$$

Where u is velocity field. The VOF function f has a value between 0 and 1. For simplicity, the THINC scheme is started from basic one-dimension:

$$\frac{\partial \phi}{\partial \mathrm{t}} + \frac{\partial}{\partial x}(u\phi) - \phi \frac{\partial u}{\partial x} = 0$$

Where t refers to the time, x the special coordinate, u the advection speed and f the transported quantity.



The VOF has its solution bounded 0 and 1 which a moving interface in one dimension can be represented by a jump shown in the figure. Here, we employ $\bar{\phi}_i^n = \frac{1}{\Delta x}\int_{x_{i-(1/2)}}^{x_{i+(1/2)}} \phi(x,t^n)dx$ to denote the cell-average value of the numerical solution to equation, which is defined at *i*th cell over $[x_{i-(1/2)}, x_{i+(1/2)}]$ at the *n*th time step ($t = t^n$). It is obvious that hyperbolic tangent is the simplest continuous function which itself has the step-jump distribution property. We use the piecewise modified hyperbolic tangent function as:

$$F_i(x) = \frac{\alpha}{2}\left(1 + \gamma tanh\left(\beta\left(\frac{x-x_{i-(1/2)}}{\Delta x_i} - \tilde{x}_i\right)\right)\right).$$

The parameter α, β, γ are important parameters for determining the quality of the numerical solution and will be discussed later.

Given α, β, γ, the only unknown in the equation is the middle point of the transition jump in the hyperbolic tangent function $\tilde{x}_i$, which is computed from the cell-integrated average $\bar{\phi}_i^n$ as $\frac{1}{\Delta x}\int_{x_{i-(1/2)}}^{x_{i+(1/2)}} F_i(x)dx$=$\bar{\phi}_i^n$.

After the piecewise interpolation functions $F_i(x)$ has constructed for all mesh cells, the VOF function f is updated by the following formulation:

$$\bar{\phi}_i^{n+1} = \bar{\phi}_i^n - \frac{\left(g_{i+\left(\frac{1}{2}\right)} - g_{i-\left(\frac{1}{2}\right)}\right)\Delta t}{\Delta x_i} + \bar{\phi}_i^n \frac{\left(u_{i+\left(\frac{1}{2}\right)} - u_{i-\left(\frac{1}{2}\right)}\right)\Delta t}{\Delta x_i},$$

where $g_{i+\left(\frac{1}{2}\right)}$ denotes the flux boundary $x = x_{i+(1/2)}$ during $, t^{n+1} - t^n$ and is computed as

$$g_{i+\left(\frac{1}{2}\right)} = \begin{cases} -\int_{t^n}^{t^{n+1}} F_i\left(x_{i+\left(\frac{1}{2}\right)} - u_{i+\left(\frac{1}{2}\right)}(t - t^n)\right) dt, & if\ u_{i+\left(\frac{1}{2}\right)} \geq 0 \\ \int_{t^n}^{t^{n+1}} F_{i+1}(x_{i+\left(\frac{1}{2}\right)} - u_{i+\left(\frac{1}{2}\right)}(t - t^n)) dt, & otherwise \end{cases}$$

The way to determine α, β, γ as follows.

Parameter γ depends on the slope orientation of the jump, and is determined as

$$\gamma = \begin{cases} 1, & if\ \phi_{i+1}^n \geq \phi_{i-1}^n \\ -1, & otherwise \end{cases}$$

In order to have the interpolation function $F_i(x)$ bounded between ϕ_{i-1}^n and ϕ_{i+1}^n, parameter α is determined as

$$\alpha = \begin{cases} \phi_{i+1}^n, & if\ \phi_{i+1}^n \geq \phi_{i-1}^n \\ \phi_{i-1}^n, & otherwise \end{cases}$$

Parameter β determines the steepness of the jump in the interpolation function. A larger β leads to a steeper jump in the interpolation reconstruction and thus a less numerical dissipation. However, a large β gives a steep interface jump but tends to wrinkle an interface. According to the numerical test, $\beta = 3.5$ seems to be proper choice. The multi-dimensional problem is conducted by direction splitting method.

Practical implementation in PHOENICs is done in the subroutine GXSURF.FOR.

The original THINC-WLIC method is fully explicit. In PHOENICS, using the SIMPLEST method enables us however to use an implicit method and to use or not the conservative form.

$$\frac{\partial \Phi}{\partial t} + \nabla \cdot \left(\Phi \vec{V} \right) = 0$$

Setting NONCONS=.TRUE., the following equation will be solved instead:

$$\frac{\partial \Phi}{\partial t} + \nabla \cdot \left(\Phi \vec{V} \right) - \Phi \nabla \cdot \vec{V} = 0$$

Note: Since there is an iterative process, the conservative form should work fine as well.

We do not use directly the Hyperbolic tangent. We use the parameters α, β, γ as described above and introduce a weighting parameter ω_i defined in each spatial direction as follow (for example in the x direction):

$$\omega_i = \frac{n_x}{|n|}$$

Where n_x is the normal in x direction and |n| the module of the normal. The flux $g_{i+\left(\frac{1}{2}\right)}$ using the above condition will be computed as:

$$g_{i+\left(\frac{1}{2}\right)} = 0.5 * [1.0 - \alpha * \frac{dx}{u * dt * \beta} * \log\left(\frac{a_4}{a_5}\right)]$$

Where : $a_1 = e^{\beta(2\Phi - 1.0)/\alpha}$,$a_3 = e^{\beta}$, $x_c = \left(\frac{0.5}{\beta}\right) * \log \frac{(a_3 - a_1)a_3}{a_1 * a_3 - 1.0}$,

$$a_4 = \cosh(\beta * \left(\gamma - \frac{udt}{dx} - x_c\right)), \ a_5 = \cosh(\beta * (\gamma - x_c))$$

And finally, $g_{i+\left(\frac{1}{2}\right)} = g_{i+\left(\frac{1}{2}\right)} * \omega_i + \Phi_i * (1.0 - \omega_i)$

Note that to be able to use the flux in gxsurf.for, the flux is divided by U*dt (This is what make it useable as SEM and the other VOF methods).

If we use directly the following time discretization:

$$\frac{C^{n+1}-C^{n}}{dt}+((u_{e}C_{e})^{n+1}-(u_{w}C_{w})^{n+1})=\frac{C^{n+1}-C^{n}}{dt}+(u_{e}^{n+1}C_{e}^{n+1}-u_{w}^{n+1}C_{w}^{n+1})=0$$

Then, the THINC scheme does not work well. To make it work in an efficient manner (due to the implicit iterative process), we use a Crank-Nicolson to compute the flux $g_{i+\left(\frac{1}{2}\right)}$. $g_{i+\left(\frac{1}{2}\right)}=\frac{{g^{n+1}}_{i+\left(\frac{1}{2}\right)}+{g^{n}}_{i+\left(\frac{1}{2}\right)}}{2}$. This is not published yet since no publish paper has used the SIMPLE type algorithm and the THINC algorithm to solve for the color function.

Reference

Feng Xiao, Satoshi Ii, Chungang Chen, “Revisit to the THINC scheme: A simple algebraic VOF algorithm”, Journal of Computational Physics 230 (2011) 7086–7092

K. Yokoi, Efficient implementation of THINC scheme: a simple and practical smoothed VOF algorithm, Journal of Computational Physics. 226 (2007) 1985–2002.