New VOF CFD Model Based on interFOAM (OpenFOAM) for Momentum Conservative High Viscosity Two Phase Flow using Least Distance Interpolation in Velocity and Flux

Dr. Daniel Ulrich Witte, LIST Technology AG, CH-4422 Arisdorf, Switzerland

Introduction

The possibility to accurately predict momentum and flow in two phase systems has been studied since a number of years. The high viscosity leads to large off-diagonal contributions in the momentum equation and therefore poor convergence of the overall iteration of the equation system. To overcome this problem, we propose to limit the interpolations to only the unknown fields velocity, volumetric content and pressure as much as possible and apply a unique interpolation scheme, in this case least distance, for all interpolations. This means that we get a geometric consistent form of the Navier Stokes Equation system where all balances are applied on cell center values and convective, gradients and similar terms on face positions.

Implemetation

To prove this conceptional idea, we have developed an improved interpolation of viscosity, which now is not interpolated itself, but is based on interpolation or gradient of velocity and volumetric content. Moreover, we have reformulated the deviatory stress that it can be treated implicitly to reduce the source term of the velocity side of the momentum equation to a minimum.

To replace interpolation of momentum matrix coefficients as in OpenFOAM official code, we apply a LU-discretization of the momentum equation and multiply by the inverted diagonal tensor field:

$$\vec{U} + \overline{\overline{D_m}}^{-1}\overline{\overline{L_m}}\vec{U}_{no} + \overline{\overline{D_m}}^{-1}\overline{\overline{U_m}}\vec{U}_{on} - \overline{\overline{D_m}}^{-1}\vec{F_p} = \overline{\overline{D_m}}^{-1}\left(\overrightarrow{S_{U,new}} + \overrightarrow{F_{g,source}} + \vec{F_\sigma}\right)$$

We then can derive the volumetric flux:

$$\varphi = \left(w\overrightarrow{U_o}^T + (1-w)\overrightarrow{U_n}^T\right)\overrightarrow{Sf}$$

We get a new pressure equation that requires a broader discretization of pressure compared to the standard OpenFOAM method by summing up the volumetric flux:

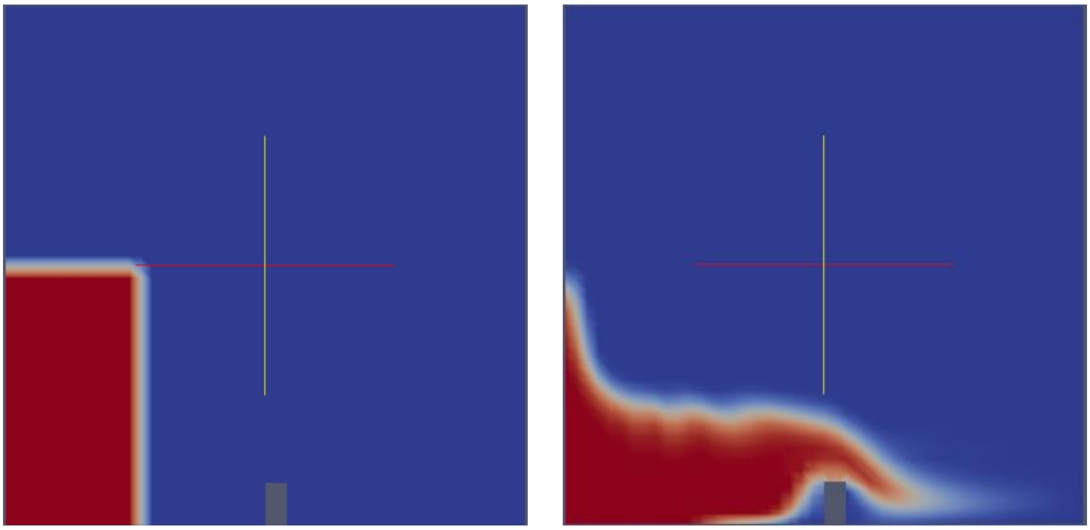
$$\left(\sum_{f_o}\varphi - \sum_{f_n}\varphi\right) = 0$$

We have implemented an extension of the addressing method to be able to cover those additional off-diagonal terms. With our method the off-diagonal velocity term in the pressure equation is explicit and fully accurate. The momentum equation may be a tensor vector equation system. We were able to discretize this term in velocity and thus create an extended linear equation system that covers both pressure and velocity in one single scheme. This scheme requires an entirely new matrix addressing method. We have developed a fast method that can be applied as a pre-processing tool to move as much as possible computational overhead upfront of the solving of the equation system itself.

The boundary conditions of volumetric content and velocity of OpenFOAM official version can be kept, whereas that of the pressure needs to be adapted. The OpenFOAM concept of “fix flux pressure” boundary condition requires a possibility of balancing force gradients at faces or walls. Since our momentum balance is solved at cell center only, this method does not work. We therefore use a zero gradient boundary condition for pressure. The wall contribution of the flux must be set to zero.

Results and Discussion

This system has been solved and results can be compared to standard OpenFOAM solutions. Our solution does not need smoothing and spurious events are eliminated. Since we have a single equation system for pressure and velocity, the solution converges much better and faster. We think that our solution should be accurate close to walls even without mesh refinement and independent on cell shape type. The solution predicts that high viscosity fluids try to avoid walls to avoid shear stress. This prediction seems plausible whereas OpenFOAM official code predicts a flow close to walls.



Outlook

We currently are developing a solver for sliding grids using an implicit interpolation on the sliding wall based on our current static mesh solver.