Rhie-Chow interpolation in OpenFOAM[1]

Fabian Peng Kärrholm

Department of Applied Mechanics
Chalmers Univesity of Technology
Göteborg, Sweden, 2006

[1]Appendix from *Numerical Modelling of Diesel Spray Injection and Turbulence Interaction*, by Fabian Peng Kärrholm, Chalmers University

Rhie-Chow interpolation in OpenFOAM

OpenFOAM is fairly new and open source, and not everything is well documented. Therefore the intent of the authour is to extend the documentation, and at the same time share some of the "extra" knowledge that has been gathered during the course of this work. One very important aspect of CFD is Rhie-Chow correction, and it will therefore be described here. Rhie-Chow correction is absolutely necessary for flow simulations using a colocated grid, since it remove oscillations in the solutions. The oscillations occur if the pressure gradient does not depend on the pressure in adjacent cells, and thus allowing a jigsaw pattern.

Some basic information about the notation in OpenFOAM is needed to describe how it works. Partial Differential Equations in OpenFOAM are solved by setting up the appropriate matrix system:

$$\texttt{fvm::operation(coefficient,U)} \tag{1}$$

This means that the solver is to solve for `U`. `coefficient` can vary in space, or be a constant. Operations like this one will be used many times throughout the section, and it is important to understand the nomenclature.

Rhie and Chow is most often seen as a correction proportional to the difference between the pressure gradient at the face and the interpolated pressure gradient at the face. For example from Ferziger & Peric [1], for a cell face, the velocity is corrected by:

$$u_j = \overline{u_j} - \Delta \overline{\left(\frac{1}{A_p^{u_j}}\right)} \left(\frac{\partial p}{\partial x_j} - \overline{\left(\frac{\partial p}{\partial x_j}\right)} \right) \tag{2}$$

where the overbar indicates interpolation, and Δ is related to the mesh size. OpenFOAM does not have such an explicit term in the equations, which makes it difficult to see how the Rhie-Chow correction is applied. OpenFOAM applies a correction what some would call 'in the spirit of Rhie and Chow'. If we take a simple incompressible flow as an example, in vector form the equations would be:

$$\frac{\partial \mathbf{U}}{\partial t} + (\mathbf{U} \cdot \nabla)\mathbf{U} - \nabla \cdot (\nu \nabla \mathbf{U}) = -\frac{1}{\rho} \nabla p \tag{3}$$

If we consider the UEqn.H in icoFoam, the convection term is slightly different.

```
fvVectorMatrix UEqn                                  (4)
(
   fvm::ddt(U)
+  fvm::div(phi, U)
−  fvm::laplacian(nu, U)
);
```

Equation 4 has no right hand side, and there is a field named `phi` (ϕ) instead of the $\mathbf{U^T}$-term. ϕ is (for incompressible flows) the volume velocity flux defined on the faces of each cell, and it is used because OpenFOAM can utilize the Gauss theorem, which is frequently used in applied mathematics, and defines the transform of a volume integral into a surface integral.

$$\begin{aligned}\int_V \nabla\cdot(\mathbf{U}\boldsymbol{\Upsilon})\,dV &= \int_S (\mathbf{U}\boldsymbol{\Upsilon})_f\cdot\hat{\mathbf{n}}\,dS = \\ =\sum_i \mathbf{U}_{f,i}\boldsymbol{\Upsilon}_{f,i}\cdot\mathbf{Sf}_i &= \sum_i \mathbf{U}_{f,i}\phi_i \end{aligned} \tag{5}$$

where

$$\phi = \boldsymbol{\Upsilon}_f\cdot\mathbf{Sf} \tag{6}$$

Υ is the velocity that will be held constant when the equation for pressure is solved, while $\mathbf{U}$ is the vector velocity that will be solved for. It is important to note the difference in the subscript when the surface integral is introduced; subscript f indicates that the term should be evaluated on the face. ϕ is defined as the scalar product of the cell face velocity and the cell face normal (Eq. 6). The magnitude of the cell face normal is the cell face area.

We can now connect OpenFOAM's equation to its analytical counterpart:

$$\int_V \nabla \cdot (\boldsymbol{\Upsilon}\mathbf{U})\, dV \Longrightarrow \texttt{fvm::div(phi,U)} \tag{7}$$

It is important to note which part of the left hand side is turned into `phi` and which is solved for. In this case we solve for U, and $\boldsymbol{\Upsilon}$ is related to `phi` via equation 6. There is also a difference in that $\boldsymbol{\Upsilon}$ is a field evaluated in the cell centres, while `phi` is defined on the surfaces.

We are now ready to return to equation 4. It does not include the pressure gradient, which is a part of the separation of pressure and velocity. The code adds the influence of p to U in another way, which is described below.

Recalling from [1], the discretized momentum equation would be this matrix system:

$$\mathcal{A}[\mathbf{U}] = \mathcal{H} - \nabla[p] \tag{8}$$

$\mathcal{A}$ and $\mathcal{H}$ are discretisation operators, decomposed from the momentum equation. They are created by issuing the commands `UEqn.A()` and `UEqn.H()`, respectively. [.] denotes the numerical approximation of the corresponding variable. This system is solved by dividing by $\mathcal{A}$, which results in:

$$[\mathbf{U}] = \frac{\mathcal{H}}{\mathcal{A}} - \frac{1}{\mathcal{A}}\nabla[p] \tag{9}$$

Equation 9 cannot be solved at this stage, since we have not updated the pressure yet. If the problem of interest includes transport of a scalar property, such as enthalpy or the mass fraction of a chemical compound, we need a predictor for $\mathbf{U}$, which is the predictor is calculated from the old pressure:

$$\texttt{solve(UEqn == - fvc::grad(p));} \tag{10}$$

This is called a momentum predictor of $\mathbf{U}$. So far only the momentum equation has been used, but we also have the continuity equation, which is used to create an equation for pressure. The first term on the right hand side (equation 10 is needed for the pressure equation, and given a special notation:

$$\texttt{U}^{*}\texttt{ = UEqn.H()/UEqn.A();} \tag{11}$$

The velocity from 11 does not satisfy continuity, and is lacking influence of pressure. To create the equation for pressure, we take the divergence of equation 9:

$$\nabla \cdot [\mathbf{U}] = \nabla \cdot ([\mathbf{U}^{*}]) - \nabla \cdot \left(\frac{1}{\mathcal{A}}\nabla[p]\right) \tag{12}$$

The left hand side is zero for incompressible flows, since it represents the correct velocity. 12 is then:

$$\nabla \cdot ([\mathbf{U}^*]) = \nabla \cdot \left(\frac{1}{\mathcal{A}} \nabla [p] \right) \tag{13}$$

The left hand side will be treated explicitly, since we now need to find the pressure, and thus keep the velocity constant. In OpenFOAM, this is done by utilizing `fvc` instead of `fvm`. The velocity $\mathbf{U}^*$ is replaced by the velocity flux ϕ, since the face velocities will be used to evaluate the term. The resulting equation is then:

```
surfaceScalarField phi =
          fvc::interpolate(U) & mesh.Sf()         (14)
volScalarField rUA = 1.0/UEqn.A();                (15)
fvScalarMatrix pEqn                               (16)
(
  fvm::laplacian(rUA, p) == fvc::div(phi)
);
```

What cannot be seen in this formulation is that OpenFOAM again uses the Gauss theorem. Thus, it is not necessary to calculate a second derivative of p, but only a first derivative. This derivative is needed on the cell faces, and it is evaluated by using the cell centre values of the pressure.

When the PISO loop is finished, the velocity is corrected with the correct pressure gradient.[2]

```
U -= rUA*fvc::grad(p);                            (17)
```

The gradient in equation 17 is again evaluated using the Gauss theorem, and therefore no gradient calculation is necessary. Instead, only the pressure on the cell face is needed.

[2]Actually this is computed at the end of each PISO loop, and the boundary conditions are corrected. The code used as example here is `icoFoam`, but the method is the same in most of the other codes available in OpenFOAM.

To summarize, there are four important points regarding the Rhie-Chow correction in OpenFOAM.

1. `phi` does not include any effect of pressure when solving the continuity equation
2. `rUA` does not include any effect of pressure when solving for continuity and final velocity corrector
3. The Laplacian-term of p uses the value of the gradient of p on the cell face. The gradient is calculated using neighbouring cells, and not neighbouring faces
4. The gradient-term of p is calculated from the cell face values of p.

This method gives an oscillation-free velocity in line with Rhie-Chow, even though there is no explicit Rhie-Chow correction.

Variable definitions and dimensionless numbers

$$S = \nabla\mathbf{U} : \mathrm{dev}(\mathrm{sym}(\nabla\mathbf{U})) \tag{18}$$

$$f = -0.3\mathrm{e}^{-\left(min\left[\left(\frac{\rho k^2}{\mu\varepsilon}\right)^2, 50\right]\right)} \tag{19}$$

$$R = \eta\frac{-\frac{\eta}{\eta_0}+1}{\beta\eta^3+1} \tag{20}$$

$$\eta = \sqrt{(|S|)}\frac{k}{\varepsilon} \tag{21}$$

$$\mathrm{Re}_d = \frac{\rho_d|\mathbf{u}_{rel}|r}{\mu_l} \tag{22}$$

$$We = \frac{\rho|\mathbf{u}_{rel}|^2 r}{\sigma} \tag{23}$$

$$\mathrm{Co} = \frac{\mathbf{U}\cdot\mathbf{d}}{|\mathbf{d}|^2\Delta t} \tag{24}$$

$$\mathrm{Sc} = \frac{\nu}{D_v} \tag{25}$$

Acknowledgements

Thanks to Hrvoje Jasak for providing comments and changes.

References

[1] J.H. Ferziger & M. Peric. *Computational Methods for Fluid Dynamics.* Springer, 1999.