Understanding DPMFoam/MPPICFoam

Jeroen Hofman

March 18, 2015

In this document I intend to clarify the flow solver and at a later stage, the particle-fluid and particle-particle interaction forces as implemented in DPMFoam in OpenFoam 2.3.x.

1 The Flow Solver

In this section we first set up the incompressible Navier-Stokes equations for the gas flow solver, then we describe the derivation of the pressure equation in semi-discretized form, which is used in the PIMPLE algorithm employed by OpenFOAM in the last section.

1.1 Incompressible Navier-Stokes

We start with the incompressible Navier-Stokes equation for laminar flow with gas volume fraction ϵ, constant viscosity μ, gas density ρ, gravity g, stress tensor $\boldsymbol{\tau}$ and momentum transfer term $\mathbf{F}$:

$$\rho\frac{\partial\epsilon\mathbf{u}}{\partial t} + \rho\nabla\cdot(\epsilon\mathbf{u}\mathbf{u}) - \rho\nabla\cdot\epsilon\boldsymbol{\tau} = -\nabla p + \rho\epsilon\mathbf{g} - \mathbf{F} \tag{1}$$

where the stress tensor with unit tensor $\boldsymbol{\delta}$ and kinematic viscosity $\nu = \frac{\mu}{\rho}$ is given by:

$$\boldsymbol{\tau} = \nu\left(\nabla\mathbf{u} + (\nabla\mathbf{u})^T\right) + \frac{2}{3}\nu\left(\nabla\cdot\mathbf{u}\right)\boldsymbol{\delta} \tag{2}$$

The interphase momentum transfer term $\mathbf{F}$ contains both momentum transfer due to drag and buoyancy [2], details will be given in section 1.2. Therefore it is not fully a source term since the drag is proportional to $\mathbf{u}$. When dividing by the density and defining the kinematic pressure as $P = \frac{p}{\rho}$ we obtain:

$$\frac{\partial\epsilon\mathbf{u}}{\partial t} + \nabla\cdot(\epsilon\mathbf{u}\mathbf{u}) - \nabla\cdot\epsilon\boldsymbol{\tau} = -\nabla P + \epsilon\mathbf{g} - \frac{\mathbf{F}}{\rho} \tag{3}$$

We also define the continuity equation as:

$$\frac{\partial \epsilon}{\partial t} + \nabla \cdot (\epsilon \mathbf{u}) = 0 \tag{4}$$

1.2 Momentum transfer

In this section we look at different ways of defining momentum transfer to either the particle or the fluid phase. By switching from the discrete particle description to a continuum description we try to link the different ways of defining the momentum transfer term since regardless of definition of $\mathbf{F}$ equation 3 the resulting PDEs should be equal.

We will first look at the way the momentum transfer term is usually defined as a combination of buoyancy and drag for particles in a computational cell:

$$\mathbf{F} = \sum_p \mathbf{f} = \frac{1}{V} \sum_p \left(\mathbf{f}_{\text{drag}} - V_p \nabla p_i \right) = \mathbf{F}_{\text{drag}} - \frac{1}{V} \sum_p V_p \nabla p_i \tag{5}$$

where V is the volume of the computational cell, V_p the volume of the particle and ∇p_i the locally averaged pressure gradient. In a continuum description we can write $\frac{1}{V} \sum_p V_p \nabla p_i = (1 - \epsilon) \nabla p$ and rewrite equation 3:

$$\frac{\partial \epsilon \mathbf{u}}{\partial t} + \nabla \cdot (\epsilon \mathbf{u}\mathbf{u}) - \nabla \cdot \epsilon \boldsymbol{\tau} = -\epsilon \nabla p + \epsilon \mathbf{g} - \frac{\mathbf{F}_{\text{drag}}}{\rho} \tag{6}$$

Another way of incorporating buoyancy in stead of via the pressure gradient is by explicitly adding a local acceleration term plus an adjustment to the drag ([6], section 2.5.1):

$$\mathbf{F} = \sum_p \mathbf{f} = \frac{1}{V} \sum_p \left(\frac{\mathbf{f}_{\text{drag}}}{\epsilon} - \rho V_p \left(\mathbf{g} - \frac{D\mathbf{u}}{Dt} \right) \right) = \frac{\mathbf{F}_{\text{drag}}}{\epsilon} + \frac{1}{V} \sum_p \rho V_p \left(\mathbf{g} - \left[\frac{D\mathbf{u}}{Dt} \right]_p \right) \tag{7}$$

where $\left[\frac{D\mathbf{u}}{Dt} \right]_p$ is the locally averaged fluid velocity at the location of particle p. Using that the gravity field is uniform and we use $\frac{1}{V} \sum_p \rho V_p \mathbf{g} = (1 - \epsilon) \rho \mathbf{g}$ we take this term out of the interaction term and we can write equation 3 as:

$$\frac{\partial \epsilon \mathbf{u}}{\partial t} + \nabla \cdot (\epsilon \mathbf{u}\mathbf{u}) - \nabla \cdot \epsilon \boldsymbol{\tau} = -\nabla P + \mathbf{g} - \frac{\mathbf{F}_{\text{drag}}}{\epsilon \rho} - \frac{1}{V} \sum_p V_p \left[\frac{D\mathbf{u}}{Dt} \right]_p \tag{8}$$

Furthermore realizing that, by using continuity in equation 4 and using product rules, we have:

$$\frac{\partial \epsilon \mathbf{u}}{\partial t} + \nabla \cdot (\epsilon \mathbf{u}\mathbf{u}) = \epsilon \frac{D\mathbf{u}}{Dt} = \epsilon \frac{\partial \mathbf{u}}{\partial t} + \epsilon \mathbf{u} \cdot \nabla \mathbf{u} \tag{9}$$

and using that in the continuum description we have $\frac{1}{V}\sum_p V_p \left[\frac{D\mathbf{u}}{Dt}\right]_p = (1-\epsilon)\frac{D\mathbf{u}}{Dt}$ we obtain a second form of the momentum equation:

$$\frac{D\mathbf{u}}{Dt} - \nabla\cdot\epsilon\boldsymbol{\tau} = -\nabla p + \mathbf{g} - \frac{\mathbf{F}_{\text{drag}}}{\epsilon\rho} \tag{10}$$

Expressions 6 and 10 should be the same when the latter is multiplied by ϵ, however there are clearly not because of the treatment of the stress term. I have spent quite some time digging into this problem and it turns out this is a very subtle problem arising from the precise definitions of both fluid and solid equations in a continuum description. See for instance [7], where the authors show that many (well-cited) papers even mess this up. Let us first consider how to fix this problem:

1. Omitting the viscuous stress is the approach that is taken in the MP-PIC papers [8], this will make equations 6 and 10 equal. Arguably this is valid as the driving factors are often drag and particle pressure, and especially for gases the viscosity is very small.

2. The approach applied in both [7] and [6] is changing the ansatz of the derivation, namely the momentum equation; equation 3. Similar to accounting for pressure in both the gas phase momentum equation and the particle momentum transfer term, we can instead also split the viscuous stress in this way. We then obtain:

$$\frac{\partial\epsilon\mathbf{u}}{\partial t} + \nabla\cdot(\epsilon\mathbf{u}\mathbf{u}) - \nabla\cdot\boldsymbol{\tau} = -\nabla P + \epsilon\mathbf{g} - \frac{\mathbf{F}}{\rho} \tag{11}$$

$$\mathbf{F} = \mathbf{F}_{\text{drag}} - \frac{1}{V}\sum_p V_p\left(\nabla p_i + \nabla\cdot\boldsymbol{\tau}_i\right) \tag{12}$$

Combining the above equations and using the continuum description we end up with:

$$\epsilon\frac{D\mathbf{u}}{Dt} - \epsilon\nabla\cdot\boldsymbol{\tau} = -\epsilon\nabla p + \epsilon\mathbf{g} - \frac{\mathbf{F}_{\text{drag}}}{\rho} \tag{13}$$

Similarly if we use equation 11 we can follow the same steps when incorporating buoyancy via the local acceleration as we did before and then we obtain the following replacement of equation 10:

$$\frac{D\mathbf{u}}{Dt} - \nabla\cdot\boldsymbol{\tau} = -\nabla p + \mathbf{g} - \frac{\mathbf{F}_{\text{drag}}}{\epsilon\rho} \tag{14}$$

Now we see that the approaches do line up. Some comments can be made here: first of all in this approach we end up with the phase fraction ϵ outside the divergence term of the viscuous stress. Arguably this is valid when $\nabla\epsilon \approx 0$ since $\nabla\cdot\epsilon\boldsymbol{\tau} = \epsilon\nabla\cdot\boldsymbol{\tau} + \boldsymbol{\tau}\cdot\nabla\epsilon$, however this is clearly not the case in for instance fluidized beds. Furthermore a simple derivation using control volumes and flux evaluation will not give you an answer to this question. Instead, one can theoretically derive the momentum equations (in a continuum framework) in which there are two approaches ([9], chapter 3): the Jackson approach, in which the viscuous stress

has ϵ outside the divergence term, but also includes $(1-\epsilon)\nabla\cdot\boldsymbol{\tau}$ in the particle momentum equations; and secondly the Ishii approach, in which ϵ is inside the divergence term of the vicuous stress but not included in the particle equations. These approaches (also known as model A and model B in literature) cause a high amount of confusion and often are wrongly implemented in for instance TFM, however the difference can be subtle as again viscuous stresses are deemed not so important. If we omit viscuous stress the models are the same. The confusion is further enhanced when the particle phase is discrete, since we introduce an extra layer of complexity.

Returning to the key-point of this report, namely understanding OpenFOAM, it seems to use a mixed approach. The momentum equation is as equation 3 and it uses the locally averaged fluid velocity as substitute for the pressure gradient, so eventually equation 8 is used, where the drag $\mathbf{F}_{\text{drag}}$ is defined as:

$$\frac{1}{(1-\epsilon)V}\sum_p V_p\beta\left(\mathbf{u}_p-\mathbf{u}|_p\right) \tag{15}$$

where β is defined according to the Ergun-Wen-Yu drag model [9], $\mathbf{u}_p$ is the particle velocity and $\mathbf{u}|_p$ is the fluid velocity interpolated to the particle position.

1.3 Discretization

There are many choices regarding discretization of the terms in equation 3, which we will not discuss in depth here. We only highlight that when looking at the convection term $\nabla\cdot(\epsilon\mathbf{uu})$ we see that the discretized system would be non-linear. A possible solution would be to solve this system using non-linear solvers but this is in general quite expensive and impractical. A second solution would be to linearize the convection term. Using the discretized version of Gauss' theorem [1] we can write for a control volume V_P with faces f centered around a point P, where all variables are defined at cell centres:

$$\begin{aligned}\int_{V_P}\nabla\cdot(\epsilon\mathbf{uu})dV &= \sum_f \mathbf{S}\cdot(\epsilon\mathbf{uu})_f \\ &= \sum_f \mathbf{S}\cdot(\epsilon\mathbf{u})_f\mathbf{u}_f \\ &= \sum_f F\mathbf{u}_f\end{aligned} \tag{16}$$

where we have defined $F=\mathbf{S}\cdot(\epsilon\mathbf{u})_f$ as the (phase) mass face flux and values are assumed to be interpolated from cell centers[1]. F has to satisfy continuity, which is equation 4.

[1] We make the assumption that $(\mathbf{ab})_f=\mathbf{a}_f\mathbf{b}_f$, which is not necessarily true for interpolated values, but often assumed to make interpolations easier, see for instance the discussion in [3]

By linearizing the convection term it follows that we define the fluxes F as a time-lagged quantity, i.e. we assume we have an existing velocity flux field F satisfying equation 4. We can now write the discretized version of the convection term as a linear combination of the velocity:

$$\int_{V_P} \nabla \cdot (\epsilon \mathbf{u}\mathbf{u}) dV = \sum_f F \mathbf{u}_f \approx a_P \mathbf{u}_P + \sum_N a_N \mathbf{u}_N \tag{17}$$

where the coefficients a_N and a_P contain the known fluxes F and the velocity at the face is assumed to be interpolated from the control volume and neighboring control volumes. This implicitly also assumes the phase fraction ϵ to be known when discretizing $\mathbf{u}$.

1.4 Derivation of the pressure equation

Since equation 3 can be discretized in a linear fashion following the treatment of the convection term above, we can semi-discretize the momentum equation. $\mathbf{F}$ can be split as a combination of a contribution to the velocity and a source term, i.e. $\mathbf{F} = f_P \mathbf{u_P} + \mathbf{f}$. Details of the split were discussed in section 1.2. For now it is sufficient that the split can be made. The semi-discretized equation has the following form, combining the diagonal coefficients in one coefficient a_P:

$$\begin{aligned}
\left(a'_P - \frac{f_P}{\rho}\right) \mathbf{u}_P &= -\sum_N a_N \mathbf{u}_N + a_0 \mathbf{u}^0_P + \mathbf{g} + \frac{\mathbf{f}}{\rho} - \nabla P \\
a_P \mathbf{u}_P &= \mathbf{H}(\mathbf{u}) - \nabla P \\
\mathbf{u}_P &= \frac{\mathbf{H}(\mathbf{u})}{a_P} - \frac{\nabla P}{a_P}
\end{aligned} \tag{18}$$

where the subscript 0 indicates the previous value. $\mathbf{H}(\mathbf{u})$ contains all off-diagonal contributions of the linear system, including source terms.

The semi-discretized version of the continuity equation (equation 4) can be written as:

$$\frac{\partial \epsilon}{\partial t} + \sum_f \mathbf{S} \cdot (\epsilon \mathbf{u})_f = 0 \tag{19}$$

where we assume some explicit discretization of the temporal term, not shown here. We first express the velocity of the cell face as:

$$\mathbf{u}_f = \left(\frac{\mathbf{H}(\mathbf{u})}{a_P}\right)_f - \left(\frac{\nabla P}{a_P}\right)_f \tag{20}$$

where ∇P, $\mathbf{H}(\mathbf{u})$ and a_P are interpolated to the cell faces. Interpolation of ∇P is performed by simple differencing of both control volumes adjacent to the face, i.e. $(\nabla P)_f \approx \frac{P_P - P_E}{\Delta x}$, where control volumes with centres P and E are adjacent to face f and Δx is the distance between P and E. By

directly calculating the gradient of pressure on the cell faces by the pressure in adjacent cells and not interpolating the gradient itself, checkerboard problems for the pressure are avoided.

By combining equation 19 and 20 we obtain the pressure equation:

$$\sum_f \mathbf{S} \cdot \epsilon_f \left(\frac{\nabla P}{a_P} \right)_f = \frac{\partial \epsilon}{\partial t} + \sum_f \mathbf{S} \cdot \epsilon_f \left(\frac{\mathbf{H}(\mathbf{u})}{a_P} \right)_f \tag{21}$$

The flux F is defined as:

$$F = \mathbf{S} \cdot (\epsilon \mathbf{u})_f = \mathbf{S} \cdot \epsilon_f \left[\left(\frac{\mathbf{H}(\mathbf{u})}{a_P} \right)_f - \left(\frac{\nabla P}{a_P} \right)_f \right] \tag{22}$$

This flux is guaranteed to be conservative by construction.

1.5 Rhie-Chow interpolation

It is well-known that solving the incompressible Navier-Stokes equations on a collocated grid gives rise to the so-called checkerboard problem when using central differencing, where the solutions for velocity and pressure field are no longer unique (up to a constant in the case of pressure), see [4], chapter 7.5. A well-known cure to this problem is the so called Rhie-Chow interpolation procedure, where the face velocity is adjusted as:

$$\mathbf{u}_f = \overline{\mathbf{u}}_f - A * \overline{\left(\frac{1}{a_P} \right)_f} \left[\nabla P_f - \overline{\nabla P_f} \right] \tag{23}$$

The overline denotes interpolated values. In other words, the interpolated velocity is corrected by the difference in the gradient of the pressure at the cell face and the interpolated gradient. It can be shown that this correction is proportional to the third derivative of the pressure, corresponding to a fourth order correction in the continuity equation.

We will now show something similar happens when deriving the pressure equation as we did in the previous section. Consider again equation 18 where we explicitly state that the gradient of pressure is evaluated at point P:

$$\mathbf{u}_P = \frac{\mathbf{H}(\mathbf{u})}{a_P} - \frac{(\nabla P)_P}{a_P} \tag{24}$$

By defining the cell-faced velocity as in equation 20 and combining with the above we have:

$$\mathbf{u}_f = \left(\mathbf{u}_P + \frac{(\nabla P)_P}{a_P} \right)_f - \left(\frac{\nabla P}{a_P} \right)_f \tag{25}$$

Writing out the interpolation operation with $a_f = \frac{1}{2}(a_P + a_E)$ we get an expression that is very similar to equation 23:

$$\begin{aligned}
\mathbf{u}_f &= \frac{1}{2}\left(\mathbf{u}_P + \mathbf{u}_E\right) + \frac{1}{2a_f}\left[(\nabla P)_P + (\nabla P)_E\right] - \frac{1}{a_f}\frac{P_P - P_E}{\Delta x} \\
&= \frac{1}{2}\left(\mathbf{u}_P + \mathbf{u}_E\right) + \frac{1}{2a_f}\left[\frac{P_{EE} - P_P}{2\Delta x} + \frac{P_E - P_W}{2\Delta x}\right] - \frac{1}{a_f}\frac{P_P - P_E}{\Delta x} \\
&= \frac{1}{2}\left(\mathbf{u}_P + \mathbf{u}_E\right) + \frac{1}{4a_f\Delta x}\left[P_{EE} - 3P_E + 3P_P - P_W\right] \qquad (26)
\end{aligned}$$

the last term on the right hand side of the above equation is indeed the discretization of the third derivative of the pressure, the same as the Rhie Chow interpolation.

1.6 PIMPLE in OpenFOAM

We are now ready to describe the combined PISO + SIMPLE algorithm that is used in OpenFOAM. The following steps can be identified:

1. Set up the discretized equation for $\mathbf{u}$ on the grid without any source terms, i.e. we set up a coefficient matrix according to the first line of equation 18, excluding the source terms.
2. Relax this equation.
3. Solve equation 18 with an initial pressure field estimate P^{guess} to obtain the momentum predictor $\mathbf{u}^*$ at the cell centres. P^{guess} is usually chosen equal to the pressure field in the previous time-step.
4. Using the predictor $\mathbf{u}^*$ we can assemble the operator $\mathbf{H}(\mathbf{u}^*)$ and interpolate both a_P and $\mathbf{H}(\mathbf{u}^*)$ to the cell faces.
5. Solve the pressure equation (equation 21) on the cell faces.
6. Update the flux field at the cell faces using the obtained pressure (equation 22).
7. Relax the pressure.
8. Update the velocity at the cell centres using the obtained pressure (equation 18).
9. Update the boundary conditions for consistency.

Steps 1 - 9 can be repeated any number of times. This, together with the relaxation, is the essence of the SIMPLE algorithm [5]. The momentum correction (steps 4 - 9) can also be repeated any number of times, this is the essence of the PISO algorithm [5]. OpenFOAM defines the PIMPLE (SIMPLE + PISO) algorithm applying both loops, including relaxation. Combining the methods in general has proven to give faster convergence.

2 Particle contributions in MPPICFoam

2.1 Particle-Particle forces

3 MPPICFoam/DPMFoam code

3.1 Parcel Evolution

3.2 Flow Solving

References

[1] http://powerlab.fsb.hr/ped/kturbo/OpenFOAM/docs/HrvojeJasakPhD.pdf

[2] Andrews, M.J., O'Rourke, P.J., 1996, *The multiphase particle-in-cell (MP-PIC) method for dense particulate flows*, Int. J. Multiphase Flow 22, 379-402.

[3] http://www.cfd-online.com/Forums/openfoam-programming-development/83638-phi-peqn-flux-vs-linearinterpolate-u-mesh-sf.html#post289622

[4] Ferziger, J. H. and Peric, M., Computational Methods for Fluid Dynamics, 2nd ed., Springer-Verlag (2001)

[5] Versteeg H.K., Malalasekera W., *An introduction to computational fluid dynamics, the finite volume method (2nd ed.)*, 2007.

[6] http://gradworks.umi.com/33/03/3303878.html

[7] Z. Y. ZHOU, S. B. KUANG, K. W. CHU and A. B. YU (2010). Discrete particle simulation of particle-fluid flow: model formulations and their applicability. Journal of Fluid Mechanics, 661, pp 482-510 doi:10.1017/S002211201000306X

[8] O'Rourke, P.J., Snider, D.M., 2010, *An improved collision damping time for MP-PIC calculations of dense particle flows with applications to polydisperse sedimenting beds and colliding jet particles*, Chemical Engineering Science 65, 6014-6028.

[9] Wachem, B., *Derivation, implementation, and validation of computer simulation models for gas-solid fluidized beds*, PhD thesis of our group, 2000.