Understanding DPMFoam/MPPICFoam

Jeroen Hofman
Multiscale modeling for multiphase reactors
Department of chemical engineering
Eindhoven University of Technology

May 20, 2015

In this document I intend to clarify the MPPICFoam flow solver. It is by no means complete and it's composed of several separate reports I wrote, so there might be discontinuities in the description (and I don't have time to rewrite the whole thing). It doesn't only contain things specific to MP-PIC, but it also includes remarks on the more general solving of the fluid-phase by using PIMPLE and employing Rhie-Chow interpolation. In general it includes clarifications I stumbled upon while trying to understand the solver.

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:

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

In MPPICFoam another way is employed; another way of incorporating buoyancy 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. In the tutorials covering MPPICFoam/DPMFoam the local acceleration term is usually turned off (which is wrong in my opinion), it can be turned on by including the `pressureGradientForce` in the list of particle

forces. 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 (if you turn it on that is), 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 $\nabla \cdot (\epsilon\mathbf{u}\mathbf{u})$ 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:

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

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.

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:

[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]

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

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:

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

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 distribution function

In MP-PIC we work with a particle distribution function $f(\mathbf{x}_p, \mathbf{u}_p, m_p, t)$ which follows the transport equation in phase space:

$$\frac{\partial f}{\partial t} + \nabla_{\mathbf{x}} \cdot (f\mathbf{u}_p) + \nabla_{\mathbf{v}} \cdot (f\mathbf{A}_p) = 0 \tag{27}$$

where $\mathbf{A}_p$ is the particle acceleration and given by $\mathbf{A}_p = \frac{\rho_p \beta}{\epsilon(1-\epsilon)} \left([\mathbf{u}]_p - \mathbf{u}_p\right) + \mathbf{g}\left(1 - \frac{\rho}{\rho_p}\right) + \frac{\rho}{\rho_p}\left[\frac{D\mathbf{u}}{Dt}\right]_p - \frac{1}{\rho_p \theta}\left[\nabla \tau_p\right]_p$, where τ_p is the particle stress expression (see below). Contrary to what is stated in the MP-PIC papers this is not the Liouville equation, which is a special case of equation 27 where forces are considered conservative ($\nabla_{\mathbf{v}} \cdot \mathbf{A}_p = 0$) and velocity is divergence free ($\nabla_{\mathbf{x}} \cdot \mathbf{u}_p = 0$). In this special case the phase-space volume (but not its shape) occupied by a cloud of particles is preserved over time. Equation 27 is instead the generalized form of the phase continuity equation and hence conservation of particle numbers. In MP-PIC we define parcels, consisting of N_p particles with identical mass, position and velocity, where N_p is kept constant in time. The equation is assuming no collisions take place; forces arising from collisions can be added to the right hand side of the equation.

The phase continuity equation can be derived from equation 27. First note that the particle volume fraction $\theta = 1 - \epsilon$ is defined as:

$$\theta = \int \int f \frac{m_p}{\rho_p} \mathrm{d}m_p \mathrm{d}\mathbf{u}_p \tag{28}$$

Using the above identity we can take moments of equation 27 that produces the continuum formulation of the particle phase. If we multiply the Liouville equation by m_p and integrate over m_p and $\mathbf{u}$ we obtain (assuming f is smooth and finite):

$$\frac{\partial}{\partial t}\int\int f m_p \mathrm{d}m_p \mathrm{d}\mathbf{u}_p + \nabla_{\mathbf{x}} \cdot \left(\int\int f \mathbf{u}_p m_p \mathrm{d}m_p \mathrm{d}\mathbf{u}_p \right) + \int\int m_p \nabla_{\mathbf{v}} \cdot (f\mathbf{A}_p) \mathrm{d}m_p \mathrm{d}\mathbf{u}_p = 0$$

$$\frac{\partial \theta \rho_p}{\partial t} + \nabla_{\mathbf{x}} \cdot (\theta \rho_p \overline{\mathbf{u}}_p) + \int m_p \left[f\mathbf{A} \right]_{\mathbf{u}_p - \infty}^{\mathbf{u}_p \infty} \mathrm{d}m_p = 0$$

$$\frac{\partial \theta \rho_p}{\partial t} + \nabla_{\mathbf{x}} \cdot (\theta \rho_p \overline{\mathbf{u}}_p) = 0 \tag{29}$$

where the cell-averaged velocity is defined as:

$$\overline{\mathbf{u}}_p = \frac{1}{\theta \rho_p} \int\int f \mathbf{u}_p m_p \mathrm{d}m \mathrm{d}\mathbf{u} \tag{30}$$

Further details of similar derivations can be found in any of the MP-PIC papers [2][10]. It can be shown in a similar way as above that when deriving the continuum form of the momentum equations for the particles, it contains a term that accounts for the kinematic stress arising from local granular temperature. Since we are explicitly tracking particles we do not need to be worried about having accurate closures for these types of expressions, they are implied by solving Lagrangian particle tracks. Another benefit of using f instead of θ, as is common in for instance the Two Fluid Model, is the explicit mass dependency. This is also reflected in the original reason to develop the MP-PIC model, namely being able to include a particle size distribution for modelling multiphase flow.

2.2 Particle update equations

Given a parcel containing N_p particles at time n with velocity $\mathbf{u}_p^n$ at position $\mathbf{x}_p^n$ we can update its position and velocity according to integration of $\mathbf{A}_p$:

$$\mathbf{x}_p^{n+1} = \mathbf{x}_p^n + \Delta t \mathbf{u}_p^{n+1} \tag{31}$$

$$\mathbf{u}_p^{n+1} = \frac{\mathbf{u}_p^n + \Delta t \left[\frac{\rho_p \beta}{\epsilon(1-\epsilon)} [\mathbf{u}]_p^n + \mathbf{g}\left(1 - \frac{\rho}{\rho_p}\right) + \frac{\rho}{\rho_p} \left[\frac{D\mathbf{u}}{Dt} \right]_p^n - \frac{1}{\rho_p \theta} \left[\nabla \tau_p^n \right]_p \right]}{1 + \Delta t \frac{\rho_p \beta}{\epsilon(1-\epsilon)}} \tag{32}$$

The particle stress expression is an extension of the model of Harris and Crighton [11]:

$$\tau_p^n = \frac{P_s \theta^{n,\beta}}{\max[\theta_{cp} - \theta^n, \gamma(1-\theta^n)]} \tag{33}$$

where P_s, β and γ are constants and θ_{cp} is the close pack particle volume fraction. Since this stress model is defined in terms of the local particle volume fraction, τ_p^n is necessarily a cell-based quantity and should be interpolated back to the particle position, hence the brackets. The second term in the denominator is there to prevent divergence of the particle stress close to close-pack, as well as allowing slight over-packing. Exceeding close packing can occur due to shifting and rearranging of granular material. The gradient of the particle stress tends to be either dominate or insignificant

due to the non-linearity of the particle stress. It is also highly sensitive to changes in θ. Therefore we split the velocity update in two parts[2]:

$$\mathbf{u}_p^{n+1} = \mathbf{u}_p^{n'} + \mathbf{u}_{p\tau} \tag{34}$$

$$\mathbf{u}_p^{n'} = \frac{\mathbf{u}_p^n + \Delta t \left[\frac{\rho_p \beta}{\epsilon(1-\epsilon)} [\mathbf{u}]_p^n + \mathbf{g} \left(1 - \frac{\rho}{\rho_p} \right) + \frac{\rho}{\rho_p} \left[\frac{D\mathbf{u}}{Dt} \right]_p^n \right]}{1 + \Delta t \frac{\rho_p \beta}{\epsilon(1-\epsilon)}} \tag{35}$$

$$\mathbf{u}_{p\tau} = -\frac{\Delta t}{\rho_p \theta^{n'}} \left[\nabla \tau_p^{n'} \right]_p \tag{36}$$

where the factor $\Delta t \frac{\rho_p \beta}{\epsilon(1-\epsilon)}$ in the denominator of $\mathbf{u}_{p\tau}$ is neglected. Neglecting this term is relatively safe since the particle pressure gradient force is either dominant or insignificant compared to the other forces. We can subsequently define an intermediate position:

$$\mathbf{x}_p^{n'} = \mathbf{x}_p^n + \Delta t \mathbf{u}_p^{n'} \tag{37}$$

Particle volume fractions are implied as follows:

$$\theta^n = \theta(\mathbf{x}_p^n) \quad \theta^{n'} = \theta(\mathbf{x}_p^{n'}) \quad \theta^{n+1} = \theta(\mathbf{x}_p^{n+1}) \tag{38}$$

There are two approaches of calculating the particle pressure, by using either implicit or explicit volume fractions. Both methods are implemented in OpenFOAM and discussed below.

2.3 Particle pressure gradient force via explicit volume fraction

In the explicit method we use the intermediate velocity $\mathbf{u}_p^{n'}$ to compute the intermediate positions $\mathbf{x}_p^{n'}$ of the particles. Given the intermediate positions we calculate the intermediate particle volume fraction $\theta^{n'}$. Given $\theta^{n'}$, we compute $\mathbf{u}_{p\tau}$. There are problems with the particle stress model since it does not takes into account the particle velocity relative to the velocity of the dense region. For instance, a particle moving near a close-pack region but moving in opposite direction of the close-pack will still experience a very strong force due to the large gradient in pressure, while in a realistic physical sense the particle should be moving unrestricted. Also particles moving together in a close-packed region experience high particle stress, but physically the movement should be restricted by the relative mean movement of the surrounding particles (i.e. particles in the cell). Incorporating these changes will adjust the particle stress for direction κ [10], where e is an elastic restitution factor and $\overline{\mathbf{U}}_p$ is the locally averaged particle velocity (i.e. the average particle velocity in the computational cell):

[2] we neglect the second term in the denominator in the particle stress since the scaling is highly non-linear, in OpenFOAM it is technically very difficult to include this term.

$$
\begin{aligned}
&\nabla\tau_p \leq 0 \\
&\quad \mathbf{u'}_{p\tau_\kappa} = \min[\mathbf{e}_\kappa \cdot \mathbf{u}_{p\tau}, (1+e)(\overline{\mathbf{U}}_p - \mathbf{u}_p^{n'}) \cdot \mathbf{e}_\kappa] \\
&\quad \mathbf{u}_{p\tau_\kappa} = \max[\mathbf{u'}_{p\tau_\kappa}, 0]
\end{aligned} \tag{39}
$$

$$
\begin{aligned}
&\nabla\tau_p > 0 \\
&\quad \mathbf{u'}_{p\tau_\kappa} = \max[\mathbf{e}_\kappa \cdot \mathbf{u}_{p\tau}, (1+e)(\overline{\mathbf{U}}_p - \mathbf{u}_p^{n'}) \cdot \mathbf{e}_\kappa] \\
&\quad \mathbf{u}_{p\tau_\kappa} = \min[\mathbf{u'}_{p\tau_\kappa}, 0]
\end{aligned} \tag{40}
$$

The crucial assumption in this model is that the intermediate particle volume fraction $\theta^{n'}$ is a good enough approximation of θ^{n+1}. However, since the particle stress changes significantly for small changes in θ near close-pack regions, this assumption is often not valid, as we will show later in the 3D gravity dominated sedimentation test.

2.4 Particle pressure gradient force via implicit volume fraction

Again we can calculate $\theta^{n'}$ from the velocities (and corresponding positions $\mathbf{x}_p^{n'}$) given by equation 35. However in the implicit model we do not assume $\theta^{n'}$ is a good approximation for θ^{n+1} and instead we try to solve for θ^{n+1} implicitly. OpenFOAM solves the following equation for θ^{n+1}:

$$
\frac{\theta^{n+1} - \theta^{n'}}{\Delta t} - \nabla \cdot \left(\frac{\Delta t}{\rho_p} \frac{\partial \tau_p^{n'}}{\partial \theta^{n'}} \nabla \theta^{n+1} \right) = 0 \tag{41}
$$

After solving this equation the particle velocities are updated by $-\frac{\Delta t}{\theta^{n+1}\rho_p}\frac{\partial \tau_p^{n'}}{\partial \theta^{n'}}\nabla\theta^{n+1}$ by interpolating to the local particle position.

The explanation for this equation is given in the MP-PIC papers. Let us first consider the definition (in the discrete sense) of the particle volume fraction for a cell i:

$$
\theta_i^n = \frac{1}{V_{\text{cell}}} \sum_p \frac{m_p N_p}{\rho_p} S_i(\mathbf{x}_p^n) \tag{42}
$$

where $S_i(\mathbf{x}_p^n)$ is the interpolation operator. If the interpolation operator is a differentiable function, the particle volume fraction can be Taylored around the particle positions and equation 41 is obtained [2]. In OpenFOAM however the interpolation operator is a nearest-neighbour operator, i.e. a particle only contributes to the particle volume fraction of a cell i if $\mathbf{x}_p^n$ is contained in cell i. Such an operator is not differentiable and hence a Taylor approximation of equation 42 will not work.

We can however still sketch an idea of the proof of equation 41. For a cell i we can consider the continuity equation at $n+1$, which should approximately be satisfied if we are using the cell-averaged particle velocity $\overline{\mathbf{u}}_{i,p}^{n+1}$:

$$\frac{\partial \theta_i^{n+1}}{\partial t} + \nabla \cdot (\theta_i^{n+1} \overline{\mathbf{u}}_{i,p}^{n+1}) = 0 \tag{43}$$

In order to make the equation less implicit we approximate the particle volume fraction in the divergence term by $\theta_i^{n'}$ and split the velocity according to equation 34.

$$\frac{\partial \theta_i^{n+1}}{\partial t} + \nabla \cdot (\theta_i^{n'} \overline{\mathbf{u}}_{i,p}^{n'}) + \nabla \cdot (\theta_i^{n'} \overline{\mathbf{u}}_{i,p\tau}^{n'}) = 0 \tag{44}$$

The error that is made is of the form $\nabla \cdot \left((\theta_i^{n+1} - \theta_i^{n'}) \overline{\mathbf{u}}_{i,p}^{n+1} \right)$ and is second order in space. Now if we further assume that at n' the continuity equation is also satisfied we get:

$$\frac{\partial \theta_i^{n+1}}{\partial t} - \frac{\partial \theta_i^{n'}}{\partial t} + \nabla \cdot (\theta_i^{n'} \overline{\mathbf{u}}_{i,p\tau}^{n'}) = 0 \tag{45}$$

Next we enter the expression for $\overline{\mathbf{u}}_{i,p\tau}^{n'}$ from equation 36. Since we use the cell-averaged velocity, we also use the particle stress gradient of the cell:

$$\frac{\partial \theta_i^{n+1}}{\partial t} - \frac{\partial \theta_i^{n'}}{\partial t} + \nabla \cdot \left(\frac{\Delta t}{\rho_p} \nabla \tau_{i,p}^{n+1} \right) = 0 \tag{46}$$

The last step is to linearise the particle stress term in terms of θ_i^{n+1} by using the product rule:

$$\frac{\partial \theta_i^{n+1}}{\partial t} - \frac{\partial \theta_i^{n'}}{\partial t} + \nabla \cdot \left(\frac{\Delta t}{\rho_p} \frac{\partial \tau_{i,p}^{n'}}{\partial \theta_i^{n'}} \nabla \theta_i^{n+1} \right) = 0 \tag{47}$$

which is equation 41.

2.5 Collisional forces in MP-PIC

In MP-PIC collisional forces are described by two components: the collisional return-to-isotropy force and the collisional damping force. The first one describes the relaxation of an anisotropic velocity distribution towards an isotropic Gaussian distribution, the second force describes the damping of relative particle motions, i.e. the relaxation of particle velocities towards the mass-averaged velocity. Both terms are described by adding a source term to the MP-PIC particle phase-space continuity equation:

$$\frac{\partial f}{\partial t} + \nabla_{\mathbf{x}} \cdot (f \mathbf{u}_p) + \nabla_{\mathbf{v}} \cdot (f \mathbf{A}_p) = \frac{f_G - f}{\tau_G} + \frac{f_D - f}{\tau_D} \tag{48}$$

where we have defined $f_G(\mathbf{x}_p, \mathbf{u}_p, m_p, t) = N(\mathbf{x}, m, t)G(\mathbf{u}_p, \overline{\mathbf{u}}, \sigma^2(m))$ where $N(\mathbf{x}_p, m, t) = \int f(\mathbf{x}_p, \mathbf{u}_p, m_p, t)\mathrm{d}\mathbf{u}_p$ is the particle size distribution function and $G(\mathbf{u}_p, \overline{\mathbf{u}}, \sigma^2(m_p))$ is the Gaussian velocity distribution. In other words, f_G represents a particle distribution with a perfect Gaussian velocity distribution. Furthermore we have $f_D(\mathbf{x}_p, \mathbf{u}_p, m_p, t) = N(\mathbf{x}_p, m_p, t)\delta(\mathbf{u}_p - \overline{\mathbf{u}}_p)$, and it represents a particle distribution with all velocities equal to the mass-averaged particle velocity.

The particle distribution function converges over time to these two distributions proportional to the inverse of the relaxation times τ_G and τ_D. An expression for τ_G and τ_D, which is highly dependent on the spherical geometry of the particles, is given by:

$$\frac{1}{\tau_G} = \frac{8\sqrt{2}}{5\pi}\frac{\theta}{r_{32}^3}\frac{\int\int f(r+r_{32})^4(\mathbf{u}_p - \overline{\mathbf{u}})^2\mathrm{d}m_p\mathrm{d}\mathbf{u}_p}{\int\int f(r+r_{32})^2\sqrt{(\mathbf{u}_p - \overline{\mathbf{u}})^2}\mathrm{d}m_p\mathrm{d}\mathbf{u}_p}g_0(\theta)\eta(2-\eta) \tag{49}$$

$$\frac{1}{\tau_D} = \frac{8\sqrt{2}}{3\pi}\frac{\theta}{r_{32}^3}\frac{\int\int f(r+r_{32})^4(\mathbf{u}_p - \overline{\mathbf{u}})^2\mathrm{d}m_p\mathrm{d}\mathbf{u}_p}{\int\int f(r+r_{32})^2\sqrt{(\mathbf{u}_p - \overline{\mathbf{u}})^2}\mathrm{d}m_p\mathrm{d}\mathbf{u}_p}g_0(\theta)\eta(1-\eta) \tag{50}$$

where r_{32} is the particle Sauter mean radius, r the particle radius, $g_0(\theta)$ the radial distribution function and $\eta = \frac{1+e}{2}$ is dependent on the coefficient of restitution e. Details can be found in the MP-PIC paper [14]. The expressions for both relaxation times are nearly the same, save a multiplicative factor and the dependency on the coefficient of restitution. The larger the velocity anomaly of the particle, the smaller τ_G and τ_D.

Both relaxation times depend on the restitution coefficient as shown in figure 1. If the restitution coefficient is 1, τ_D becomes infinite as perfectly elastic particles will never converge to the mass-averaged velocity. τ_G on the other hand is smallest when the particles behave perfectly elastic, so convergence to a isotropic Gaussian distribution is faster for more elastic particles.

In the discrete world we can write the local particle distribution function f in a cell at time n as the sum over all parcels p in that cell (each parcel containing N_p particles):

$$f^n(\mathbf{x}_p, \mathbf{u}_p, m_p) \approx \sum_p N_p\delta(\mathbf{x} - \mathbf{x}_p^n)\delta(\mathbf{u} - \mathbf{u}_p^n)\delta(m - m_p) \tag{51}$$

After some derivations it can be shown that applying the left hand side of the source term in equation 48 alters the local particle distribution function by:

$$f^n(\mathbf{x}_p, \mathbf{u}_p, m_p) \approx \sum_p N_p\delta(\mathbf{x} - \mathbf{x}_p^n)\delta(m - m_p)\left(\delta(\mathbf{u} - \mathbf{u}_p^n)\exp\left(-\frac{\Delta t}{\tau_G}\right)\right. \tag{52}$$

$$\left. + G(\mathbf{u}_p, \overline{\mathbf{u}}, \sigma^2(m))\left(1 - \exp\left(-\frac{\Delta t}{\tau_G}\right)\right)\right) \tag{53}$$

Now consider the updated velocity of equation 34, $\mathbf{u}_p^{n+1}$. The description of adjusting this velocity is now fairly straightforward:

1. For each particle p, choose a uniform random number X between [0,1]. If $X < \exp\left(-\frac{\Delta t}{\tau_G}\right)$, then $\mathbf{u}_p^{n+1}$ is unadjusted.
2. If $X \geq \exp\left(-\frac{\Delta t}{\tau_G}\right)$, sample a velocity from $G(\mathbf{u}, \overline{\mathbf{u}}, \sigma^2(m))$ and adjust the velocity to $\mathbf{u}_p^{n+1} + G(\mathbf{u}, \overline{\mathbf{u}}, \sigma^2(m))$

Since we are considering a finite number of particles this procedure will not conserve mean velocity and velocity variance (granular temperature) in the cell. The adjusted velocity distribution is adjusted such that the mean velocity and granular temperature are preserved, details can be found in the paper.

A straightforward implementation of the collision damping force (the second term on the right hand side of 48) would imply splitting the computational parcel in two separate parcels; one part of size $N_p \exp(-\delta t/\tau_D)$ with original velocity $\mathbf{u}_p$ and one part of size $N_p(1 - \exp(-\delta t/\tau_D))$ with mass-averaged velocity $\overline{\mathbf{u}}_p$. Since this is impractical instead the velocity is updated by a linear combination of the two:

$$\mathbf{u}_p^{n+1} = \psi \mathbf{u}_p^n + (1 - \psi)\overline{\mathbf{u}}_p \tag{54}$$

where $\psi = \exp(-\delta t/2\tau_D)$ such that the velocity variance of the above equation is equal to the variance of the particles if they were split.

2.6 Determining the coefficient of restitution

The relaxation times depend on the coefficient of restitution e. This is a single coefficient incorporating both normal and tangential restitution, as well as friction. Jenkins and Zhang [13] suggested a way of computing this effective restitution e in terms of normal restitution e_n, the coefficient of tangential restitution β_0 and friction coefficient μ:

$$e = e_n + \frac{1}{2}a_1(\mu_0) + \frac{1}{2}a_2(\mu_0)\frac{b1(\mu_0)}{b2(\mu_0)} \tag{55}$$

$$\mu_0 = \frac{7}{2}\frac{\mu(1 + e_n)}{(1 + \beta_0)} \tag{56}$$

Typical values for glass beads of $\mu = 0.1$, $\beta_0 = 0.33$ and $e_n = 0.97$ give $e = 0.85$. The sensitivity of the model to e will be investigated further in the single-spouted fluidized bed measurements.

3 Wall effects

The last subject so far untouched in the reporting about the mechanics of MP-PIC is the parcel-wall interaction. In the MP-PIC papers wall-interactions are not mentioned. In the commercial MP-PIC software BARRACUDA the wall restitution coefficients are mentioned but the implementation is

not specified. In OpenFOAM the wall interaction is defined by two coefficients; the coefficient of normal restitution e_t and the specularity coefficient μ'. Consider a particle with normal velocity $\mathbf{u}^0_{p,n} = \mathbf{u}^0_p \cos\phi$ and tangential velocity $\mathbf{u}^0_{p,t} = \mathbf{u}^0_p \sin\phi$ impacting a wall, where ϕ is the impact angle relative to the wall normal vector, then OpenFOAM defines the velocity after impact as:

$$\mathbf{u}_{p,n} = -e_n \mathbf{u}^0_p \cos\phi \tag{57}$$

$$\mathbf{u}_{p,t} = (1-\mu')\mathbf{u}^0_p \sin\phi \tag{58}$$

When the specularity coefficient is 0, the tangential velocity after the wall collision is the same as before the collision. When the coefficient is 1, the particle will be aligned after collision with the wall normal vector. When the coefficient is 2, the particle will move back along the trajectory from where it came from. The specularity coefficient can be approximated by looking at the way hard-sphere collisions are defined in DPM in terms of μ, β_0 and e_n

$$\mathbf{u}_{p,n,DPM} = -e_n \mathbf{u}^0_p \cos\phi \tag{59}$$

$$\mathbf{u}_{p,t,DPM} = \begin{cases} u^0_p \sin\phi(1-\mu(1+e_n)\cot\phi) & \text{if} \quad \phi > \tan^{-1}\mu_0 \\ u^0_p \sin\phi(1-\frac{2}{7}(1+\beta_0)) & \text{if} \quad \phi < \tan^{-1}\mu_0 \end{cases} \tag{60}$$

where μ_0 is defined in equation 56. We can now estimate the specularity coefficient by equating both representations of the velocity vector and solving for the unknown μ'. Specifically, we solve:

$$\int_0^{\pi/2} \left(\frac{\mathbf{u}_{p,t}}{\mathbf{u}^0_{p,t}} - \frac{\mathbf{u}_{p,t,DPM}}{\mathbf{u}^0_{p,t}} \right) \cos\phi \mathrm{d}\phi = 0 \tag{61}$$

Using typical values for glass beads [15] where $\mu = 0.3$, $\beta_0 = 0.33$ and $e_n = 0.97$ we obtain $\mu' = 0.357748$. The sensitivity of the model to μ' will be discussed further with the single-spouted fluidized bed measurements.

4 MPPICFoam/DPMFoam code

The above sections should cover all that is needed to understand the MPPICFoam/DPMFoam code in OpenFOAM, at least on a global level. Many intricacies of the solver itself, in terms of 'corrections' and 'adjustments' (as they are called in OpenFOAM) are not discussed here, as I think they add little understanding specific to either DPM or the MP-PIC model.

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.

[10] Snider, D.M., 2001, *An incompressible three-dimensional multiphase particle-in-cell model for dense particle flows*, J. Comp. Physics 170, 523-549.

[11] Harris, S.E., Crighton, D.G., 1988, *Solutions, solitary waves and voidage disturbances in gas-fluidized beds*, J. Fluid Mech. 195, 437.

[12] Maureen S. van Buijtenen, Willem-Jan van Dijk, Niels G. Deen, J.A.M. Kuipers, T. Leadbeater, D.J. Parker, Numerical and experimental study on multiple-spout fluidized beds, Chemical Engineering Science, Volume 66, Issue 11, 1 June 2011, Pages 2368-2376.

[13] Jenkins, James T. and Zhang, Chao, Kinetic theory for identical, frictional, nearly elastic spheres, Physics of Fluids (1994-present), 14, 1228-1235 (2002)

[14] O'Rourke, P.J., Snider, D.M., 2012, *Inclusion of collisional return-to-isotropy in the MP-PIC method*, Chemical Engineering Science 80, 39-54.

[15] Christoph Goniva, Christoph Kloss, Niels G. Deen, Johannes A.M. Kuipers, Stefan Pirker, Influence of rolling friction on single spout fluidized bed simulation, Particuology, Volume 10, Issue 5, October 2012, Pages 582-591.

[16] Farzam Fotovat, Jamal Chaouki, Jeffrey Bergthorson, The effect of biomass particles on the gas distribution and dilute phase characteristics of sandbiomass mixtures fluidized in the bubbling regime, Chemical Engineering Science, Volume 102, 11 October 2013, Pages 129-138.

[17] Justin M. Weber, Ky J. Layfield, Dirk T. Van Essendelft, Joseph S. Mei, Fluid bed characterization using Electrical Capacitance Volume Tomography (ECVT), compared to CPFD Software's Barracuda, Powder Technology, Volume 250, December 2013, Pages 138-146.

[18] Cheng Chen, Joachim Werther, Stefan Heinrich, Hai-Ying Qi, Ernst-Ulrich Hartge, CPFD simulation of circulating fluidized bed risers, Powder Technology, Volume 235, February 2013, Pages 238-247.

[19] Yongshi Liang, Yongmin Zhang, Tingwen Li, Chunxi Lu, A critical validation study on CPFD model in simulating gassolid bubbling fluidized beds, Powder Technology, Volume 263, September 2014, Pages 121-134.

[20] Jos Leboreiro, Gustavo G. Joseph, Christine M. Hrenya, Revisiting the standard drag law for bubbling, gas-fluidized beds, Powder Technology, Volume 183, Issue 3, 21 April 2008, Pages 385-400.

[21] Fei Li, Feifei Song, Sofiane Benyahia, Wei Wang, Jinghai Li, MP-PIC simulation of CFB riser with EMMS-based drag model, Chemical Engineering Science, Volume 82, 12 September 2012, Pages 104-113.

[22] E.M. Ryan, D. DeCroix, R. Breault, W. Xu, E.D. Huckaby, K. Saha, S. Dartevelle, X. Sun, Multi-phase CFD modeling of solid sorbent carbon capture system, Powder Technology, Volume 242, July 2013, Pages 117-134.

[23] N.A. Patankar, D.D. Joseph, Modeling and numerical simulation of particulate flows by the EulerianLagrangian approach, International Journal of Multiphase Flow, Volume 27, Issue 10, October 2001, Pages 1659-1684.

[24] Stefan Radl, Sankaran Sundaresan, A drag model for filtered EulerLagrange simulations of clustered gasparticle suspensions, Chemical Engineering Science, Volume 117, 27 September 2014, Pages 416-425, ISSN 0009-2509.

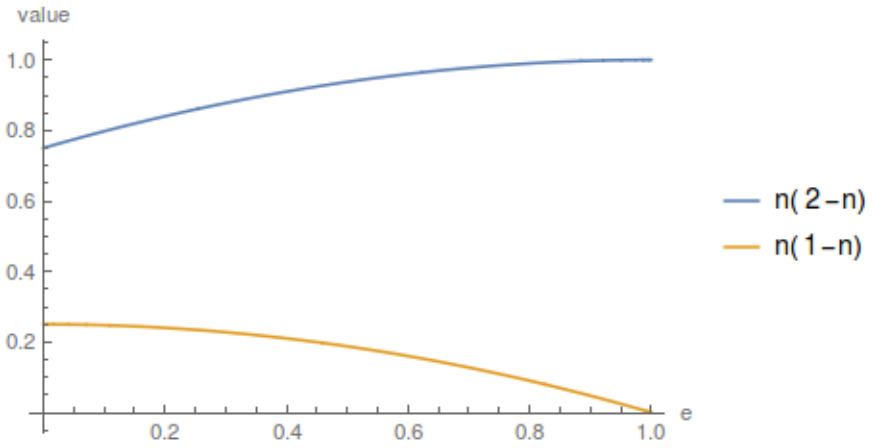

Figure 1: Dependency of restitution coefficient $\eta(1-\eta)$ and $\eta(2-\eta)$ on the restitution coefficient e.