Documentation of open-source MFIX–QMOM software for gas-solids flows [1]

A. Passalacqua
albertop@iastate.edu
Iowa State University - Department of Chemical and Biological Engineering
Ames, IA, USA

R. O. Fox
rofox@iastate.edu
Iowa State University - Department of Chemical and Biological Engineering
Ames, IA, USA

January 9, 2012

[1] Please refer to this document as: A. Passalacqua, R. O. Fox Documentation of open-source MFIX–DEM software for gas-solids flows, From URL https://mfix.netl.doe.gov/documentation/qmomk_doc_2012-1.pdf

Table of Contents

1 Introduction

This document provides a concise description of the implementation of the quadrature method of moments developed in Fox (2008); Fox and Vedula (2009) for the solution of the Boltzmann equation and of its coupling with a fluid solver, following Passalacqua et al. (2010). The reader is invited to refer to these two references for the theoretical background of the method and for the its derivation.

2 Limitations of the implementation

Currently the implementation of QMOM for the solution of kinetic equations into MFIX is affected by the following limitations:

- The implementation reflects the method proposed by Fox (2008), which solves the Boltzmann equation for point particles. As a consequence, the code is capable of simulating only dilute flows, with volume fractions below 4%, to avoid the formation of regions with volume fractions that exceeds the particle packing limit [1]. The extension of the method to solve the Boltzmann-Enskog equation for finite-size particles and limited mean free path is presented in Fox and Vedula (2009), but currently not implemented into MFIX.
- The multi-dimensional quadrature algorithm implemented into MFIX-QMOM is not stable for restitution coefficients below 0.95. The problem has been addressed by developing the conditional quadrature method of moment (Yuan and Fox, 2011).
- Only first-order kinetic fluxes are available for the moment transport equations. Extension to higher-order methods can be found in Vikas et al. (2011).
- The space discretization is assumed to be uniform in each direction (Δx, Δy and Δx are supposed to be constant, but can be different). The extension to non-uniform grids is trivial, and the adoption of unstructured grids with arbitrarily shaped cells is illustrated in Vikas et al. (2011).
- The cut-cell approach is not supported.
- The drag model is hard-coded. The Wen and Yu (1966) is adopted.
- Walls are assumed to be specularly reflective, with user-defined particle-wall restitution coefficient.
- Verification and validation studies have been performed in closed domains (only wall boundary conditions) or in periodic domains (walls and periodic boundary conditions). The implementation of other boundary conditions has not been tested.
- The implementation is not parallelized (no SMP or DMP support).

It is worth noticing that the limitation listed above are specific to the current implementation into MFIX and not to the quadrature-based moment method itself, as clarified in the cited references.

[1] This value has to be considered as an indication of the average volume fraction in the system, and it depends on the actual flow conditions.

3 Governing Equations

4 The Mathematical Model

In this section the governing equations of the fluid and particle phases solved when the QMOM algorithm is used are briefly presented.

4.1 Fluid-phase governing equations

The behavior of the fluid phase is described by the classical continuity and momentum equations solved in multi-fluid models (Drew, 1971; Syamlal et al., 1993; Gidaspow, 1994; Enwald et al., 1996). The fluid continuity equation has the form

$$\frac{\partial}{\partial t}\left(\alpha_{\mathrm{f}}\rho_{\mathrm{f}}\right) + \nabla\cdot\left(\alpha_{\mathrm{f}}\rho_{\mathrm{f}}\mathbf{U}_{\mathrm{f}}\right) = 0, \tag{1}$$

and the fluid momentum equation is given by

$$\frac{\partial}{\partial t}\left(\alpha_{\mathrm{f}}\rho_{\mathrm{f}}\mathbf{U}_{\mathrm{f}}\right) + \nabla\cdot\left(\alpha_{\mathrm{f}}\rho_{\mathrm{f}}\mathbf{U}_{\mathrm{f}}\otimes\mathbf{U}_{\mathrm{f}}\right) = \nabla\cdot\left(\alpha_{\mathrm{f}}\boldsymbol{\tau}_{\mathrm{f}}\right) - \alpha_{\mathrm{f}}\nabla p + \alpha_{\mathrm{f}}\rho_{\mathrm{f}}\mathbf{g} + \mathbf{M}_{\mathrm{fp}}, \tag{2}$$

where α_{f}, ρ_{f}, $\mathbf{U}_{\mathrm{f}}$ are, respectively, the fluid-phase volume fraction, density and mean velocity, $\mathbf{M}_{\mathrm{fp}}$ is the momentum exchange term due to the drag between the fluid and particle phases, and $\mathbf{g}$ is the gravitational acceleration vector.

For incompressible fluids, the fluid pressure p is used to satisfy the continuity equation. The fluid phase is assumed to be Newtonian, and its stress tensor τ_{f} is given by

$$\tau_{\mathrm{f}} = \mu_{\mathrm{f}}\left(\nabla\mathbf{U}_{\mathrm{f}} + \left(\nabla\mathbf{U}_{\mathrm{f}}\right)^{\mathrm{T}}\right) - \frac{2}{3}\mu_{\mathrm{f}}\left(\nabla\cdot\mathbf{U}_{\mathrm{f}}\right)\mathbf{I}, \tag{3}$$

where μ_{f} is the fluid dynamic viscosity and $\mathbf{I}$ the unit tensor.

4.2 Particle-phase governing equations - Monodisperse case

The particle phase is described assuming that particles are smooth, non-cohesive spheres. As a consequence, its governing equation is represented by a kinetic equation for the particle number density function $f\left(t,\mathbf{x},\mathbf{v}\right)$, defined so that $f d\mathbf{x} d\mathbf{v}$ is the average number of particles with velocity between $\mathbf{v}$ and $\mathbf{v}+d\mathbf{v}$ and position between $\mathbf{x}$ and $\mathbf{x}+d\mathbf{x}$, at time t. The form of the kinetic equation is (Chapman and Cowling, 1961; Cercignani et al., 1994; Struchtrup, 2005)

$$\frac{\partial f}{\partial t} + \mathbf{v}\cdot\frac{\partial f}{\partial\mathbf{x}} + \frac{\partial}{\partial\mathbf{v}}\cdot\left(f\frac{\mathbf{F}}{m_{\mathrm{p}}}\right) = \mathbb{C}, \tag{4}$$

where $\mathbb{C}$ represents the rate of change in the number density function due to binary collisions between the particles, and $\mathbf{F}$ is the force acting on each particle, which includes gravity and drag.

The collision term $\mathbb{C}$ can be described using the Bhatnagar-Gross-Krook collision operator (Bhatnagar et al., 1954):

$$\mathbb{C} = \frac{1}{\tau_{\mathrm{c}}}\left(f_{\mathrm{es}} - f\right), \tag{5}$$

where τ_{c} is the collision time and f_{es} is the equilibrium distribution function, extended to account for inelastic collisions:

$$f_{\mathrm{es}} = \frac{N}{\left[\det\left(2\pi\boldsymbol{\lambda}\right)\right]^{1/2}}\exp\left(-\frac{1}{2}(v_{\mathrm{p},i} - U_{\mathrm{p},i})\boldsymbol{\lambda}^{-1}(v_{\mathrm{p},j} - U_{\mathrm{p},j})\right), \tag{6}$$

where $\boldsymbol{\lambda}^{-1}$ is the inverse of the the matrix $\boldsymbol{\lambda}$, defined by

$$\boldsymbol{\lambda} = \gamma\omega^2\Theta\mathbf{I} + \left(\gamma\omega^2 - 2\gamma\omega + \mathbf{I}\right)\boldsymbol{\sigma} \tag{7}$$

with $\gamma = 1/\,\mathrm{Pr}$, and $\omega = (1+e)/2$, being N the number density of particles (zero-order moment), $\mathbf{U}_\mathrm{p}$ the mean particle velocity (first-order moment), e the restitution coefficient, Θ_p the granular temperature, and $\boldsymbol{\sigma}$ the velocity covariance matrix. In this work $\gamma = 1$, being $\mathrm{Pr} = 1$ in the standard BGK model (Struchtrup, 2005).

A more complete description of the collisional process is achieved by adopting the complete Boltzmann collision integral Fox and Vedula (2009). The moments of the hard-sphere Boltzmann collision integral can be written in the form (Fox and Vedula, 2009; Passalacqua et al., 2011):

$$C^\gamma_{ijk} = \frac{6g_0}{d_\mathrm{p}} \int_{\mathbb{R}^3}\int_{\mathbb{R}^3} gI_{ijk}\left(\omega, \mathbf{v}_1, \mathbf{g}\right) f\left(\mathbf{v}_1\right) f\left(\mathbf{v}_2\right) \mathrm{d}\mathbf{v}_1 \mathrm{d}\mathbf{v}_2, \tag{8}$$

where

$$I_{ijk} = \frac{1}{\pi g}\int_{S^+} \left[\left(v'_{1,1}\right)^i \left(v'_{1,2}\right)^j \left(v'_{1,3}\right)^k - \left(v_{1,1}\right)^i \left(v_{1,2}\right)^j \left(v_{1,3}\right)^k\right] |\mathbf{g}\cdot\mathbf{n}|\,\mathrm{d}\mathbf{n}, \tag{9}$$

$\mathbf{g} = \mathbf{v}_1 - \mathbf{v}_2$ the relative velocity vector with magnitude g, $\mathbf{n}$ the unit vector along the line containing the two colliding particles centers, $\mathbf{v}'_1 = \mathbf{v}_1 - \omega(\mathbf{g}\cdot\mathbf{n})\mathbf{n}$, and $\omega = (1+e)/2$.

4.3 The quadrature method of moments

In this work a set of twenty moments W^3 of f up to the third order defined by

$$\begin{aligned} W^3 = (&M^0, M^1_1, M^1_2, M^1_3, M^2_{12}, M^2_{13}, M^2_{22}, M^2_{23}, M^2_{33},\\ &M^3_{111}, M^3_{112}, M^3_{113}, M^3_{122}, M^3_{123}, M^3_{133}, M^3_{222},\\ &M^3_{223}, M^3_{233}, M^3_{333}), \end{aligned}$$

is considered, where the superscripts represent the order of the corresponding moment (Fox, 2008). Each moment is defined through integrals of the distribution function as

$$\begin{aligned} M^0 &= \int f d\mathbf{v}, \qquad M^1_i = \int v_i f d\mathbf{v},\\ M^2_{ij} &= \int v_i v_j f d\mathbf{v}, \qquad M^3_{ijk} = \int v_i v_j v_k f d\mathbf{v}. \end{aligned} \tag{10}$$

Note that the particle-phase volume fraction α_p and mean particle velocity $\mathbf{U}_\mathrm{p}$ are related to these moments by

$$\alpha_\mathrm{p} = V_\mathrm{p} M^0 \tag{11}$$

and

$$\rho_\mathrm{p}\alpha_\mathrm{p} U_{\mathrm{p},i} = m_\mathrm{p} M^1_i, \tag{12}$$

where $m_\mathrm{p} = \rho_\mathrm{p} V_\mathrm{p}$ is the mass of a particle with density ρ_p and volume $V_\mathrm{p} = \pi d^3_\mathrm{p}/6$. In this work, m_p is constant. Likewise, the particle temperature is defined in terms of the trace of the particle velocity covariance matrix, which is found from M^2_{ij} and lower-order moments. By definition, $\alpha_\mathrm{f} + \alpha_\mathrm{p} = 1$ and this relation must be accounted for when solving a fully coupled system for the fluid and particle phases.

The application of the definition of the moments to both sides of Eq. 4 allows the moment transport equations to be derived. If the force acting on each particle is divided in two components,

one due to the drag and the other due to gravity, the set of twenty transport equations, one for each moment in W^3 is given by

$$\begin{aligned}
\frac{\partial M^0}{\partial t} + \frac{\partial M_i^1}{\partial x_i} &= 0, \\
\frac{\partial M_i^1}{\partial t} + \frac{\partial M_{ij}^2}{\partial x_j} &= A_i^1, \\
\frac{\partial M_{ij}^2}{\partial t} + \frac{\partial M_{ijk}^3}{\partial x_k} &= C_{ij}^2 + A_{ij}^2, \\
\frac{\partial M_{ijk}^3}{\partial t} + \frac{\partial M_{ijkl}^4}{\partial x_l} &= C_{ijk}^3 + A_{ijk}^3,
\end{aligned} \tag{13}$$

where A_i^1, A_{ij}^2 and A_{ijk}^3 are the source terms due to the acceleration acting on each particle, while C_{ij}^2 and C_{ijk}^3 are those due to the collision operator. It is worth noting that the force term only affects the moments of order higher than zero, because it is assumed that the number density of the particles is conserved. Also, the collision term only influences the moments of order higher than one, because of the assumption that collisions do not change the particle number density (no aggregation and breakage phenomena), and do not influence the mean momentum of the particle phase. In general, the conservative equations for the particle phase needed for coupling with the fluid phase are found by multiplying the expressions in Eq. 13 by m_p. For simplicity, hereinafter we will assume that all of the velocity moments have been multiplied by V_p, so that the zero order moment corresponds to the particle phase volume fraction $M^0 = \alpha_\mathrm{p}$.

The set of transport Eqs. 13 is not closed, because each equation contains the spatial fluxes of the moments of order immediately higher, and the source terms due to the drag force and to collisions. As a consequence, closures have to be provided for these terms. In quadrature-based moment methods Gaussian quadrature formulas are used to provide closures to the source terms in the moment transport equations by introducing a set V_β of β weights n_α and abscissas $\mathbf{U}_\alpha$, which are determined from the moments of the distribution function using an inversion algorithm, and approximating the distribution function with a sum of Dirac delta functions:

$$f(\mathbf{v}) = \sum_{\alpha=1}^{\beta} n_\alpha \delta\left(\mathbf{v} - \mathbf{U}_\alpha\right). \tag{14}$$

In the following discussion we will consider a set of $\beta = 8$ weights and abscissa V_8 per each velocity component, which are obtained by considering two quadrature nodes in each direction of velocity phase space. The inversion algorithm to obtain V_8 from the set of moments W^3 is explained in detail in Fox (2008). Once the weights and abscissas are known, the moments can be computed as a function of the quadrature weights and abscissas by approximating the integrals in Eq. 10 with summations:

$$\begin{aligned}
M^0 &= \sum_{\alpha=1}^{\beta} n_\alpha, \qquad & M_i^1 &= \sum_{\alpha=1}^{\beta} n_\alpha U_{\alpha i}, \\
M_{ij}^2 &= \sum_{\alpha=1}^{\beta} n_\alpha U_{\alpha i} U_{\alpha j}, \qquad & M_{ij}^3 &= \sum_{\alpha=1}^{\beta} n_\alpha U_{\alpha i} U_{\alpha j} U_{\alpha k}.
\end{aligned} \tag{15}$$

The source terms due to drag and gravity are computed as

$$\begin{aligned} A_i^1 &= \sum_{\alpha=1}^{\beta} n_\alpha \left(\frac{F_{i\alpha}^{\mathrm{D}}}{m_{\mathrm{p}}} + g_i \right), \\ A_{ij}^2 &= \sum_{\alpha=1}^{\beta} n_\alpha \left[\left(\frac{F_{i\alpha}^{\mathrm{D}}}{m_{\mathrm{p}}} + g_i \right) U_{j\alpha} + \left(\frac{F_{j\alpha}^{\mathrm{D}}}{m_{\mathrm{p}}} + g_j \right) U_{i\alpha} \right], \\ A_{ijk}^3 &= \sum_{\alpha=1}^{\beta} n_\alpha \left[\left(\frac{F_{i\alpha}^{\mathrm{D}}}{m_{\mathrm{p}}} + g_i \right) U_{j\alpha} U_{k\alpha} + \left(\frac{F_{j\alpha}^{\mathrm{D}}}{m_{\mathrm{p}}} + g_j \right) U_{k\alpha} U_{i\alpha} \right. \\ &\quad \left. + \left(\frac{F_{k\alpha}^{\mathrm{D}}}{m_{\mathrm{p}}} + g_k \right) U_{i\alpha} U_{j\alpha} \right], \end{aligned} \tag{16}$$

where the drag force terms are computed as

$$\mathbf{F}_\alpha^{\mathrm{D}} = \frac{m_{\mathrm{p}}}{\tau_\alpha^{\mathrm{D}}} \left(\mathbf{U}_{\mathrm{f}} - \mathbf{U}_\alpha \right) = K_{\mathrm{fp},\alpha}^{\mathrm{QMOM}} \left(\mathbf{U}_{\mathrm{f}} - \mathbf{U}_\alpha \right), \tag{17}$$

with the drag time for each abscissa given by

$$\tau_\alpha^{\mathrm{D}} = \frac{4 d_{\mathrm{p}} \rho_{\mathrm{p}}}{3 \alpha_{\mathrm{f}} \rho_{\mathrm{f}} C_{\mathrm{D}}(\mathrm{Re}_{\mathrm{p}\alpha}, \alpha_{\mathrm{f}}) \left| \mathbf{U}_{\mathrm{f}} - \mathbf{U}_\alpha \right|}. \tag{18}$$

The particle Reynolds number for each abscissa is defined by

$$\mathrm{Re}_{\mathrm{p}\alpha} = \frac{\rho_{\mathrm{f}} d_{\mathrm{p}} \left| \mathbf{U}_{\mathrm{f}} - \mathbf{U}_\alpha \right|}{\mu_{\mathrm{f}}}, \tag{19}$$

and the drag coefficient C_{D} is provided by the Schiller and Naumann (1935) correlation, modified to account for moderately dense flows ($\alpha_{\mathrm{f}} > 0.8$) as in Wen and Yu (1966):

$$C_{\mathrm{D}}(\mathrm{Re}_{\mathrm{p}}, \alpha_{\mathrm{f}}) = \frac{24}{\alpha_{\mathrm{f}} \mathrm{Re}_{\mathrm{p}}} \left[1 + 0.15 (\alpha_{\mathrm{f}} \mathrm{Re}_{\mathrm{p}})^{0.687} \right] \alpha_{\mathrm{f}}^{-2.65}. \tag{20}$$

It is worth observing that in the evaluation of the force term, the relative velocity vector $\mathbf{U}_{\mathrm{f}} - \mathbf{U}_\alpha$ is defined as a function of the quadrature abscissas $\mathbf{U}_\alpha$, instead of the mean particle velocity. As a consequence, the drag time and the drag force have different values for each quadrature node. Note also that for small Re_{p} (i.e., the Stokes flow limit), the drag coefficient reduces to $C_{\mathrm{D}}(\mathrm{Re}_{\mathrm{p}}, \alpha_{\mathrm{f}}) = 24/(\alpha_{\mathrm{f}}^{3.65} \mathrm{Re}_{\mathrm{p}})$ and τ_α^{D} will be the same for all abscissas.

For collisions, the source terms in the moment transport equations are given by

$$\begin{aligned} C_{ij}^2 &= \frac{\alpha_{\mathrm{p}}}{\tau_{\mathrm{c}}} \left(\lambda_{ij} - \sigma_{ij} \right), \\ C_{ijk}^3 &= \frac{1}{\tau_{\mathrm{c}}} \left(\Delta_{ijk} - M_{ijk}^3 \right). \end{aligned} \tag{21}$$

For hard-sphere collisions, the collision time is defined by

$$\tau_{\mathrm{c}} = \frac{\pi^{1/2} d_{\mathrm{p}}}{12 \alpha_{\mathrm{p}} g_0 \Theta^{1/2}}, \tag{22}$$

with the granular temperature Θ defined in terms of the moments by

$$\Theta = \frac{1}{3} \left(\sigma_{11} + \sigma_{22} + \sigma_{33} \right), \tag{23}$$

where

$$\sigma_{11} = \frac{M_{11}^2}{M^0} - \left(\frac{M_1^1}{M^0}\right)^2,$$
$$\sigma_{22} = \frac{M_{22}^2}{M^0} - \left(\frac{M_2^1}{M^0}\right)^2, \tag{24}$$
$$\sigma_{33} = \frac{M_{33}^2}{M^0} - \left(\frac{M_3^1}{M^0}\right)^2.$$

In Eq. 22, g_0 is the radial distribution function, which depends on α_p, and is used to account for the increased collision frequency in moderately dense flows. In this work, we use the model proposed by Carnahan and Starling (1969):

$$g_0 = \frac{1}{1-\alpha_\mathrm{p}} + \frac{3\alpha_\mathrm{p}}{2\left(1-\alpha_\mathrm{p}\right)^2} + \frac{\alpha_\mathrm{p}^2}{2(1-\alpha_\mathrm{p})^3}. \tag{25}$$

An important point for obtaining a stable solution to the moment transport equations is represented by the closure provided for the moment spatial fluxes. These fluxes are represented by the second term on the left-hand side of Eq. 13, and are computed according to their kinetic definition (Perthame, 1990; Desjardin et al., 2008; Fox, 2008). First each moment involved in the expression for the fluxes is decomposed into two contributions, as shown in Eq. 26 for the zero-order moment, whose spatial flux involves the first-order moments:

$$M_i^1 = \int_{-\infty}^{0} v_i \left(\int f dv_j dv_k\right) dv_i + \int_{0}^{+\infty} v_i \left(\int f dv_j dv_k\right) dv_i. \tag{26}$$

The integrals are then approximated using the Dirac-delta representation of the distribution function f leading to

$$M_i^1 = \sum_{\alpha=1}^{\beta} n_\alpha \mathrm{min}\left(0, U_{i\alpha}\right) + \sum_{\alpha=1}^{\beta} n_\alpha \mathrm{max}\left(0, U_{i\alpha}\right). \tag{27}$$

In a similar manner, the decomposition is applied to all other moments, to compute the fluxes as a function of the weights and abscissas. This procedure to evaluate the spatial fluxes is essential to ensure the realizability of the set of moments by means of the quadrature approximation, or, in other words, that the set of weights and abscissas actually represent a real distribution function. It is worth to notice that the third-order moment spatial flux depends on fourth-order moments, which are not provided by the solution of the transport equations (13). Closures for M_{ijkl}^4 are obtained in terms of the quadrature representation of the distribution function as (Fox, 2008)

$$M_{ijkl}^4 = \sum_{\alpha=1}^{\beta} n_\alpha U_{\alpha i} U_{\alpha j} U_{\alpha k} U_{\alpha l}. \tag{28}$$

The explicit closure for the third-order moments spatial fluxes is implicit in equation 30 and 31.

5 Implementation details

5.1 Solution of the moment transport equations

The moment transport equations (Eqs. 13) are discretized according to the finite-volume technique, using a second-order Runge-Kutta scheme for time integration. Before proceeding with the

description of the QMOM solver, it is worth reiterating that the moment transport equations are rescaled so that the zero-order moment M^0 represents the particle-phase volume fraction instead of the number density. This operation is important to ensure the stability and accuracy of the quadrature-inversion algorithm, which would be compromised by the high round-off error caused by the computation in terms of the number density. As noted earlier, the scaling of the equations is simply performed by multiplying them by the particle volume V_p, and by modifying the collision and drag terms accordingly. After this rescaling, the collision time τ_c and drag time τ_α^D are unchanged. The rescaled weights n_α can then be interpreted as representing the volume fraction of the corresponding abscissa.

The steps in the QMOM solver for the solution of the moment transport equations can be summed up as follows:

1. Initialize weights and abscissas in V_8.
2. Compute the moments in W^3 using Eq. 15. Note that it is not necessary to initialize the moments directly, because they can be computed as a consequence of the specified weights and abscissas.
3. Advance the moments in W^3 over a half time step $\Delta t/2$, and, using a time-split procedure:
   - account for the spatial fluxes,
   - account for collisions,
   - account for the body and drag forces acting on particles,
   - apply boundary conditions.
4. Apply the inversion algorithm to the new set of moments to compute the updated weights and abscissas.
5. Recompute the moments from the weights and abscissas using Eq. 15, performing the projection step (Fox, 2008), necessary to ensure that the transported moments are consistent with their quadrature representation.
6. Advance the moments over a half time step $\Delta t/2$ and repeat the same operations performed from step 3 to 5 for the full time step.
7. Repeat from step 3.

The time step used in the QMOM solver is evaluated on the basis of the collision time τ_c, the drag time τ^D, and the Courant number based on the maximum abscissas in the whole computational domain to ensure the stability of the solution. The key steps in the algorithm are illustrated in detail in the following subsections. Note that, due to hyperbolic nature of the moment transport equations, it is theoretically possible to use a CFL number of unity without losing stability. Thus the QMOM solver will be particularly efficient for flows where the collision time does not control the time step (i.e. Sufficiently large Knudsen number). In such cases, the efficiency of the gas-phase flows solver will be critical for the overall efficiency.

5.2 Computation of the moment spatial fluxes

The moment spatial fluxes are computed as a function of the quadrature weights and abscissas, following their kinetic definition, as discussed earlier. To explain how the computation of the fluxes

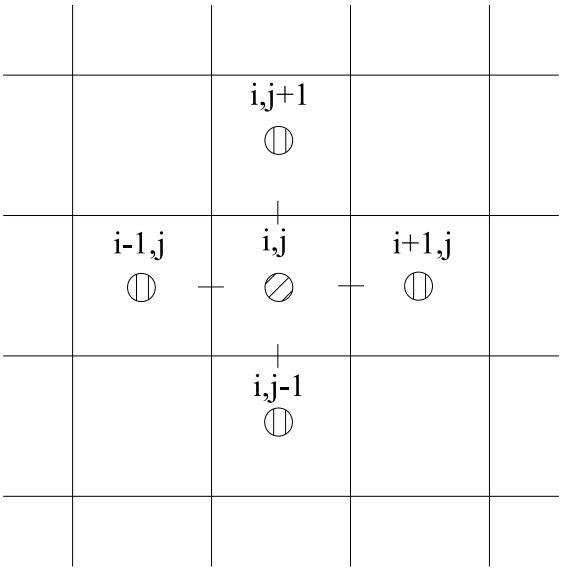

Figure 1: Schematic representation of a computational cell to illustrate how the moment spatial fluxes are computed.

is performed, let us consider the computational cell in Fig. 1, and introduce four sets of weights and abscissas: $V^-_{\beta,l}$, $V^+_{\beta,l}$, $V^-_{\beta,r}$, $V^+_{\beta,r}$. These sets of weights and abscissa are found by interpolating the cell-centered values of the weights and abscissa on the faces of the computational cell. If we consider the horizontal direction in Fig. 1 and adopt a first-order scheme, we have

$$\begin{aligned}
V^-_{\beta,l} &= \left\{ n^-_{\alpha,l} = n^{i-1,j}_\alpha ; \mathbf{U}^-_{\alpha,l} = \mathbf{U}^{i-1,j}_\alpha \right\}, \\
V^+_{\beta,l} &= \left\{ n^+_{\alpha,l} = n^{i,j}_\alpha ; \mathbf{U}^+_{\alpha,l} = \mathbf{U}^{i,j}_\alpha \right\}, \\
V^-_{\beta,r} &= \left\{ n^-_{\alpha,r} = n^{i,j}_\alpha ; \mathbf{U}^-_{\alpha,r} = \mathbf{U}^{i,j}_\alpha \right\}, \\
V^+_{\beta,r} &= \left\{ n^+_{\alpha,r} = n^{i+1,j}_\alpha ; \mathbf{U}^+_{\alpha,r} = \mathbf{U}^{i+1,j}_\alpha \right\}.
\end{aligned} \tag{29}$$

At this point the two Riemann fluxes at the left and right cell faces, G_l and G_r, are computed as follows:

$$\begin{aligned}
G_{1,l} = \sum_{\alpha=1}^{\beta} n^-_{\alpha,l} \max\left(0, U^-_{1\alpha l}\right) \begin{pmatrix} 1 \\ U^-_{i\alpha l} \\ U^-_{i\alpha l} U^-_{j\alpha l} \\ U^-_{i\alpha l} U^-_{j\alpha l} U^-_{k\alpha l} \end{pmatrix} \\
+ \sum_{\alpha=1}^{\beta} n^+_{\alpha,l} \min\left(0, U^+_{1\alpha l}\right) \begin{pmatrix} 1 \\ U^+_{i\alpha l} \\ U^+_{i\alpha l} U^+_{j\alpha l} \\ U^+_{i\alpha l} U^+_{j\alpha l} U^+_{k\alpha l} \end{pmatrix},
\end{aligned} \tag{30}$$

$$G_{1,r} = \sum_{\alpha=1}^{\beta} n^{-}_{\alpha,r} \max\left(0, U^{-}_{1\alpha r}\right) \begin{pmatrix} 1 \\ U^{-}_{i\alpha r} \\ U^{-}_{i\alpha r} U^{-}_{j\alpha r} \\ U^{-}_{i\alpha r} U^{-}_{j\alpha r} U^{-}_{k\alpha r} \end{pmatrix} + \sum_{\alpha=1}^{\beta} n^{+}_{\alpha,l} \min\left(0, U^{+}_{1\alpha r}\right) \begin{pmatrix} 1 \\ U^{+}_{i\alpha r} \\ U^{+}_{i\alpha r} U^{+}_{j\alpha r} \\ U^{+}_{i\alpha r} U^{+}_{j\alpha r} U^{+}_{k\alpha r} \end{pmatrix}, \quad (31)$$

and the net flux in the horizontal direction is then computed as

$$G_1 = G_{1,r} - G_{1,l}. \quad (32)$$

Using a similar procedure, it is possible to compute the moment spatial fluxes in each direction, thereby obtaining the flux vector $\mathbf{G} = \{G_1, G_2, G_3\}$. The moments are then updated as a consequence of their spatial fluxes by solving the ODE

$$\frac{dW^3}{dt} = \mathbf{G} \quad (33)$$

in each computational cell of the domain under consideration.

5.3 Computation of force contributions

The contributions to the evolution of the moments in W^3 due to the forces acting on each particle are directly computed, operating on the weights and abscissas of the quadrature approximation, by solving a set of two ODEs:

$$\begin{aligned} \frac{dn_\alpha}{dt} &= 0, \\ \frac{dU_{i\alpha}}{dt} &= \frac{F_{i\alpha}}{m_\mathrm{p}} + g_i, \end{aligned} \quad (34)$$

written considering that the body and drag forces do not affect the quadrature weights because they do not change the number of the particles, and only influence the abscissas.

5.4 Contribution of collisions

Collisions are accounted for by resolving the differential equation for the change in the moments due to collisions:

$$\frac{dW^3}{dt} = C(W^3), \quad (35)$$

where $C(W^3)$ is provided by Eq. 21.

5.5 Boundary conditions

The boundary conditions for the moment transport equations can be specified either in terms of the moments, or in terms of the weights and abscissas of the quadrature. The latter approach is often more convenient due to its simplicity. In this work periodic and wall-reflective boundary conditions are considered. Periodic boundary conditions, where H is the length of the system in the periodic direction, are specified in terms of the quadrature weights and abscissas as

$$V_{\beta,0} = V_{\beta,H}, \quad (36)$$

where $V_{\beta,0}$ and $V_{\beta,H}$ are the set of weights and abscissas in each cell of the two periodic boundaries of the computational domain. Once the weights and abscissas are set, the moments at the periodic boundaries can be computed by means of Eq. 15.

Specularly reflective walls, with particle-wall restitution coefficient e_w, are described by

$$\begin{pmatrix} n_\alpha \\ U_{1,\alpha} \\ U_{2,\alpha} \\ U_{3,\alpha} \end{pmatrix}_{i=0} = \begin{pmatrix} n_\alpha / e_\mathrm{w} \\ U_{1,\alpha} \\ -e_\mathrm{w} U_{2,\alpha} \\ U_{3,\alpha} \end{pmatrix}_{i=1}, \tag{37}$$

which is written considering a planar wall perpendicular to the second direction of the reference frame, located at position $i = 0$, where $i = 1$ represents the computational cell neighboring the wall. Note that other types of boundary conditions used in Lagrangian simulations (e.g. diffuse walls) can be easily accommodated using quadrature.

5.6 Extension to poly-disperse systems

In the case of poly-disperse systems, a kinetic equation is considered for each particle phase:

$$\frac{\partial f_i}{\partial t} + \mathbf{v}_i \cdot \frac{\partial f_i}{\partial \mathbf{x}} + \frac{\partial}{\partial \mathbf{v}_i} \cdot \left(f_i \frac{\mathbf{F}_i}{m_{\mathrm{p},i}} \right) = \mathbb{C}_{ij}, \tag{38}$$

where $\mathbb{C}_{ij}$ is the collisional rate of change of f_i due to collisions between particles of species i and j. It is worth noticing that such a term incorporates the momentum exchange terms between particles of different species, which does not require any explicit closure in QMOM.
In the current implementation, if multiple dispersed phases are defined:

- A set of twenty transport equations for the moments of each specie is solved.
- The Boltzmann collision integral is used.

A detailed description of the moment closures used in the poly-disperse case is available in Fox and Vedula (2009).

6 Tutorials

7 MFIX–QMOM user input variables

The setup for a simulation with MFIX-QMOM consists in the following steps:

1. Disable the solution of the multi-fluid equations:
   - Dispersed phase momentum equations
   - Granular temperature equations
   - Species transport equations
2. Specify the following parameters:
   - Activate QMOM:
     - `QMOMK = .FALSE.`

- Specify the wall boundary condition for particles.
  - `QMOMK_WALL_BC_TYPE = 'SPECULAR_REFLECTIVE'`

  Currently the only option is `SPECULAR_REFLECTIVE`.
- Specify the collision model:
  - `QMOMK_COLLISIONS = 'BGK'`

  Two options are available:
  - `BGK`
  - `Boltzmann`

  The BGK collision model is the default for mono-disperse simulations. The Boltzmann collision operator is automatically used when more than one dispersed phase is present.
- Specify the order of integration for the collision integral (used only for Boltzmann):
  - `QMOMK_COLLISIONS_ORDER = 0`

  Possible values are `0` (first order, default), and `1` (second order).
- Specify if two-way coupling with the fluid phase has to be considered:
  - `QMOMK_COUPLED = .TRUE.`
- Specify the CFL condition for QMOM internal stepping:
  - `QMOMK_CFL = 0.4`

  The default value is 0.4.

7.1 Gas-particle flow in a vertical channel: Mono-disperse case

The setup required to simulate the flow in a vertical channel (0.1 x 1m), with particles is reported below. The domain is represented by a channel, periodic in the flow direction, with two walls. A constant mass flux for the fluid phase is imposed. Particles are mono-disperse ($d_p = 252.9 \times 10^{-6}$), with a density of 1500 kg/m^3 and an initial volume fraction equal to 0.001. A simulation with the BGK model is performed, using specular-reflective wall boundary conditions.

```
!Run control
! Run-control section
DESCRIPTION     = 'Channel_flow_with_mono-disperse_particles'
RUN_TYPE        = 'NEW'
UNITS           = 'SI'
TIME            = 0.0                       !start time
DT_MIN          = 1.0E-06
RUN_NAME        = "Channel_QMOM_alpha_0_001"
DT              = 1.0e-5
DT_MAX          = 1.0
DT_FAC          = 0.95
TSTOP           = 5.0

CYCLIC_Y_PD = .TRUE.
DELP_Y = 3000.0
FLUX_G = 2.42

! Numerical settings
DISCRETIZE = 9*2
leq_it = 9*200
NORM_S = 0
NORM_G = 0
TOL_RESID_TH = 1.0e-2

!Under-relaxation
!Pressure
UR_FAC(1) = 0.6

!Solids volume fraction
UR_FAC(2) = 0.4

! Excluding particle phase momentum  equations - Not necessary using QMOM

MOMENTUM_X_EQ(1) = .FALSE.
```

```
 MOMENTUM_Y_EQ(1) = .FALSE.
 MOMENTUM_Z_EQ(1) = .FALSE.
 GRANULAR_ENERGY  = .FALSE.
 ENERGY_EQ        = .FALSE.              !do not solve energy eq
 SPECIES_EQ       = .FALSE.     .FALSE.  !do not solve species eq
 !CLOSE_PACKED    = .FALSE.

 GRAVITY = 9.81

 ! Setting the number of dispersed phases
 MMAX = 1

 ! Gas phase properties
 MU_g0                  =     1.73E-4
 RO_g0                  =     1.2

 ! Particle phase properties
 RO_s                   =     1500
 D_p0                   =     252.9E-06
 C_e                    =     1
 e_w                    =     1.0
 EP_star                =     0.37     !void fraction at minimum fluidization

 Phi                    =     28.0     !angle of internal friction
 phi_w                  =     28.0
 PHIP                   =     0.0

 ! Geometry definition
 COORDINATES            = 'cartesian'
 XLENGTH                = 0.1
 IMAX                   = 20
 YLENGTH                = 1.0
 JMAX                   = 200
 NO_K                   = .TRUE.

 ! Initial conditions
 IC_X_w(1)              =      0.0
 IC_X_e(1)              =      0.1
 IC_Y_s(1)              =      0.0
 IC_Y_n(1)              =      1.0

 IC_EP_g                =      0.999

 IC_theta_m(1,1)        =      0.1

 IC_U_g                 =      0.0        ! x-dir gas velocity
 IC_V_g                 =      0.0        ! y-dir gas velocity
 IC_W_g                 =      0.0
 IC_U_s(1,1)            =      0.0
 IC_V_s(1,1)            =      0.0
 IC_W_s(1,1)            =      0.0
 IC_P_star              =      0.0

! Boundary conditions
 BC_X_w(1)              =      0.0
 BC_X_e(1)              =      0.0
 BC_Y_s(1)              =      0.0
 BC_Y_n(1)              =      1.0

 BC_X_w(2)              =      0.1
 BC_X_e(2)              =      0.1
 BC_Y_s(2)              =      0.0
 BC_Y_n(2)              =      1.0

 BC_TYPE(1)             = "NSW"
 BC_TYPE(2)             = "NSW"

 BC_Uw_g(1) = 0.0
 BC_Vw_g(1) = 0.0
 BC_Ww_g(1) = 0.0

 BC_Uw_g(2) = 0.0
 BC_Vw_g(2) = 0.0
 BC_Ww_g(2) = 0.0

! QMOMB Input
  QMOMK = .TRUE.
  QMOMK_COLLISIONS = 'BGK'
  QMOMK_COUPLED = .TRUE.
  QMOMK_COLLISIONS_ORDER = 0
  QMOMK_CFL = 0.1

! End QMOMB

! Output Control

  RES_DT                =     0.01
  SPX_DT = .01 .01 .01 .01 .01 .01 .01 .01 .2      .1E+03 .1E+03
  OUT_DT                = 20.0             !write text file BUB01.OUT
                                           !   every 0.1s
  NLOG                  = 10               !write logfile BUB01.LOG
                                           !every 25 time steps
```

```
  FULL_LOG                = .TRUE.          !display residuals on screen

  CALL_USR                = .TRUE.

!SPX_DT values determine how often SPx files are written.  Here BUB01.SP1, which
!contains void fraction (EP_g), is written every 0.01 s, BUB01.SP2, which contains
! gas and solids pressure (P_g, P_star), is written every 0.1 s, and so forth.
!
        ! EP_g  P_g        U_g   U_s   ROP_s      T_g   X_g
        !       P_star     V_g   V_s               T_s   X_s      Theta    Scalar
        !                  W_g   W_s

!  The decomposition in I, J, and K directions for a Distributed Memory Parallel machine

  NODESI = 1    NODESJ = 1    NODESK = 1

!  Sweep Direction

LEQ_SWEEP(1) = 'ISIS'
LEQ_SWEEP(2) = 'ISIS'
LEQ_SWEEP(3) = 'ISIS'
LEQ_SWEEP(4) = 'ISIS'
LEQ_SWEEP(5) = 'ISIS'
LEQ_SWEEP(6) = 'ISIS'
LEQ_SWEEP(7) = 'ISIS'
LEQ_SWEEP(8) = 'ISIS'
LEQ_SWEEP(9) = 'ISIS'
```

7.2 Gas-particle flow in a vertical channel: Bi-disperse case

The setup required to simulate the flow in a vertical channel (0.032 x 0.3m), with particles is reported below. The domain is represented by a channel, periodic in the flow direction, with two walls. A constant mass flux for the fluid phase is imposed. Particles are bi-disperse ($d_{p,1} = 120 \times 10^{-6}$, $d_{p,2} = 185 \times 10^{-6}$), with a density of 2400 kg/m^3 and an initial volume fraction equal to 0.001. A simulation with the Boltzmann model is performed, using specular-reflective wall boundary conditions.

```
! Run-control section
  RUN_NAME                = 'Channel_QMOM_Bidisperse'
  DESCRIPTION             = 'Channel flow with bi-disperse particles'
  RUN_TYPE                = 'NEW'
  UNITS                   = 'SI'
  TIME                    = 0.0
  TSTOP                   = 20.0
  DT                      = 1.0E-5
  DT_MAX                  = 1.0E-1
  DT_MIN                  = 1.0E-8
  DT_FAC                  = 0.9
  DETECT_STALL            = .TRUE.

  MOMENTUM_X_EQ(1)     = .FALSE.
  MOMENTUM_Y_EQ(1)     = .FALSE.
  GRANULAR_ENERGY      = .FALSE.

  MOMENTUM_X_EQ(2)     = .FALSE.
  MOMENTUM_Y_EQ(2)     = .FALSE.

  ENERGY_EQ               = .FALSE.
  SPECIES_EQ              = .FALSE.     .FALSE.     .FALSE.

  MAX_NIT                   = 200

  CYCLIC_Y_PD             = .TRUE.
  DELP_Y                  = 400.0
  FLUX_G                  = 1.2

 ! Numerical settings
  DISCRETIZE             = 9*2
  leq_it                 = 9*200

! Geometry Section
  COORDINATES             = 'cartesian'
  XLENGTH                 = 0.032
  IMAX                    = 25
  YLENGTH                 = 0.30
  JMAX                    = 60
  NO_K                    = .TRUE.

! Gas-phase Section

  MU_g0                   = 1.8E-5
  RO_g0                   = 1.2
```

```
! Solids-phase Section
  MMAX                      = 2
  RO_s(1)                   = 2400
  RO_s(2)                   = 2400
  D_p0(1)                   = 120.0E-06
  D_P0(2)                   = 185.0E-06

  e                         = 1.0
  Phi                       = 30.0
  EP_star                   = 0.390
  C_f                       = 0.1
  PHIP = 0.0

! Initial Conditions Section

  IC_X_w(1)                 =  0.0
  IC_X_e(1)                 =  0.032
  IC_Y_s(1)                 =  0.0
  IC_Y_n(1)                 =  0.30

  IC_theta_m(1,1)           =  0.01
  IC_theta_m(1,2)           =  0.01

  IC_EP_g(1)                =  0.975

  IC_ROP_s(1,1)             =  30
  IC_ROP_s(1,2)             =  30

  IC_U_g(1)                 =  0.0
  IC_V_g(1)                 =  0.0

  IC_U_s(1,1)               =  0.0
  IC_V_s(1,1)               =  0.0

  IC_U_s(1,2)               =  0.0
  IC_V_s(1,2)               =  0.0

! Boundary Conditions Section
  BC_X_w(1)                 =  0.0
  BC_X_e(1)                 =  0.0
  BC_Y_s(1)                 =  0.0
  BC_Y_n(1)                 =  0.3

  BC_X_w(2)                 =  0.032
  BC_X_e(2)                 =  0.032
  BC_Y_s(2)                 =  0.0
  BC_Y_n(2)                 =  0.3

  BC_TYPE(1)                 = "NSW"
  BC_TYPE(2)                 = "NSW"

  BC_Uw_g(1) = 0.0
  BC_Vw_g(1) = 0.0

  BC_Uw_g(2) = 0.0
  BC_Vw_g(2) = 0.0

! QMOMK Input
  QMOMK = .TRUE.
  QMOMK_COLLISIONS = 'Boltzmann'
  QMOMK_COUPLED = .TRUE.
  QMOMK_COLLISIONS_ORDER = 0
  QMOMK_CFL = 0.1

!  Output Control
  OUT_DT                    = 10.
  RES_DT                    = 0.01
  NLOG                      = 25
  FULL_LOG                  = .true.

!SPX_DT values determine how often SPx files are written.  Here BUB02.SP1, which
!contains void fraction (EP_g), is written every 0.01 s, BUB02.SP2, which contains
! gas and solids pressure (P_g, P_star), is written every 0.1 s, and so forth.

        ! EP_g  P_g        U_g   U_s   ROP_s      T_g   X_g
        !       P_star     V_g   V_s              T_s   X_s      Theta    Scalar
        !                  W_g   W_s
  SPX_DT = 0.0001  0.0001   0.0001   0.0001  0.0001   100.  100.      0.01      0.01

!  Sweep Direction
  LEQ_SWEEP(1) = 'ISIS'
  LEQ_SWEEP(2) = 'ISIS'
  LEQ_SWEEP(3) = 'ISIS'
  LEQ_SWEEP(4) = 'ISIS'
  LEQ_SWEEP(5) = 'ISIS'
  LEQ_SWEEP(6) = 'ISIS'
  LEQ_SWEEP(7) = 'ISIS'
  LEQ_SWEEP(8) = 'ISIS'
  LEQ_SWEEP(9) = 'ISIS'

 NODESI=1 NODESJ=1 NODESK=1
```

References

Bhatnagar, P. L., Gross, E. P., Krook, M., 1954. A model for collisional processes in gases. I. Small amplitude processes in charged and neutral one-component systems. Phys. Rev. 94, 511–525.

Carnahan, N. F., Starling, K. E., 1969. Equation of state for nonattracting rigid spheres. J. Chem. Phys. 51 (2), 635 – 636.

Cercignani, C., Illner, R., Pulvirenti, M., 1994. The Mathematical Theory of Dilute Gases. Springer-Verlag.

Chapman, S., Cowling, T. G., 1961. The Mathematical Theory of Non-uniform Gases, 2nd Edition. Cambridge University Press.

Desjardin, O., Fox, R. O., Villedieu, P., 2008. A quadrature-based moment method for dilute fluid-particle flows. J. Comput. Phys. 227, 2524–2539.

Drew, D. A., 1971. Averaged equations for two-phase flows. Stud. Appl. Math. L (3), 205 – 231.

Enwald, H., Peirano, E., Almstedt, A. E., 1996. Eulerian two-phase flow theory applied to fluidization. Int. J. Multiphase Flow 22, 21–66.

Fox, R. O., 2008. A quadrature-based third-order moment method for dilute gas-particle flows. J. Comput. Phys. 227, 6313–6350.

Fox, R. O., Vedula, P., 2009. Quadrature-based moment model for moderately dense polydisperse gas-particle flows. Ind. Eng. Chem. Res. (49).

Gidaspow, D., 1994. Multiphase Flow and Fluidization. Academic Press.

Passalacqua, A., Fox, R. O., Garg, R., Subramaniam, S., 2010. A fully coupled quadrature-based moment method for dilute to moderately dilute fluid-particle flows. Chem. Eng. Sci. 65, 2267–2283.

Passalacqua, A., Galvin, J., Vedula, P., Hrenya, C. M., Fox, R. O., 2011. A quadrature-based kinetic model for dilute non-isothermal granular flows. Comm. Comp. Phys. (10), 216 – 252.

Perthame, B., 1990. Boltzmann type schemes for compressible Euler equations in one and two space dimensions. SIAM J. Num. Anal. 29 (1), 1–19.

Schiller, L., Naumann, A., 1935. A drag coefficient correlation. V. D. I. Zeitung 77, 318–320.

Struchtrup, H., 2005. Macroscopic Transport Equations for Rarefied Gas Flows. Springer, Berlin.

Syamlal, M., Rogers, W., O'Brien, T. J., 1993. MFIX Documentation Theory Guide. U.S. Department of Energy, Office of Fossil Energy, Morgantown Energy Technology Center, Morgantown, West Virginia.

Vikas, V., Wang, Z. J., Passalacqua, A., Fox, R. O., 2011. Realizable high-order finite-volume schemes for quadrature-based moment method. J. Comp. Phys. 230 (13), 5328 – 5352.

Wen, C. Y., Yu, Y. H., 1966. Mechanics of fluidization. Chem. Eng. Prog. S. Ser. 62, 100–111.

Yuan, C., Fox, R. O., 2011. Conditional quadrature method of moments for kinetic equations. J. Comp. Phys. 230 (22), 8216 – 8246.