twoWayGPBEFoam: An open-source Eulerian QBMM solver for monokinetic bubbly flows

twoWayGPBEFoam: an open-source Eulerian QBMM solver for monokinetic bubbly flows

Dongyue Li[b,a,**], Daniele Marchisio[c,*], Christian Hasse[d], Dirk Lucas[b]

[a]*State Key Laboratory of Advanced Metallurgy, University of Science and Technology Beijing, China*
[b]*Helmholtz-Zentrum Dresden-Rossendorf, Dresden, Germany*
[c]*Politecnico di Torino, Torino, Italy*
[d]*Institute for Simulation of Reactive Thermo-Fluid Systems, Technische Universität Darmstadt, Darmstadt, Germany*

Abstract

`twoWayGPBEFoam` is an open-source mescoscopic Eulerian QBMM solver for monokinetic bubbly flows. The solver is implemented within the OpenFOAM software framework. Unlike the existing macroscopic two-fluid model (TFM) solver `twoPhaseEulerFoam`, it can predict the size segregation phenomenon and the size-conditional velocities of the disperse phase, although it will not be able to predict the particle trajectory crossing (PTC). On theoretical grounds, the evolution of the disperse phase in multiphase flows is dictated by the generalized population balance equation (GPBE), which can be transformed into moment transport equations and solved using the finite-volume method with higher-order realizable spatial-discretization schemes and time-integration schemes. In order to address the closure problem of the size-conditional spatial flux, the size-conditional velocities need to be modelled. In many previous works, these are assumed to be identical with the disperse phase velocity predicted by the TFM. In this work, the size-conditional velocities were modelled using the velocity polynomial approximation (VPA), for which the velocity polynomial coefficients (VPCs) can be obtained from the moments themselves. By carrying out several test cases with both one-way and two-way coupling, we show that the results predicted by our solver agree well with the analytical solutions and the existing experimental data.

*Primary corresponding author. Tel: +39 0110904622. Fax: +39 0110904699. Email: daniele.marchisio@polito.it

**Secondary corresponding author. Email: li.dy@dyfluid.com

Keywords: Computational fluid dynamics; Multiphase flow; Generalized population balance equation; Quadrature method of moments; OpenFOAM;

PROGRAM SUMMARY/NEW VERSION PROGRAM SUMMARY

Program title: `twoWayGPBEFoam` and `oneWayGPBEFoam`

Licensing provisions: GNU General Public License 3 (GPL)

Programming language: C++

Nature of problem: `twoWayGPBEFoam` and `oneWayGPBEFoam` have been developed to help investigate multiphase flows using the Eulerian QBMM. It provides an easily extended, parallelised, Eulerian QBMM environment.

Solution method: The continuous phase is solved by the Eulerian approach. The disperse phase is solved by the QBMM. These equations are one-way coupled in `oneWayGPBEFoam` and two-way coupled in `twoWayGPBEFoam`.

Additional comments including restrictions and unusual features: All appropriate methodological references are contained in the section entitled References.

1. Introduction

Multiphase flows are omnipresent in a variety of industrial applications. Knowledge about the flow can be gained by using computational fluid dynamics (CFD) models. In this work we are concerned with polydisperse multiphase flows made up of a disperse phase and a continuous phase [1]. In the past, different approches have been developed, namely the two-fluid model (TFM) [2] and the Eulerian-Lagrangian (E-L) method [3], as reported in Fig.1. In both the TFM and E-L methods, the continuous phase is described by mass, momentum and energy balance equations, leading to equations similar to the Navier-Stokes equations for single-phase flows. The difference lies in the modelling of the disperse phase, whose evolution is governed by the generalized population balance equation (GPBE). In the TFM, the disperse phase is modelled via a small sub-set of transport equations derived from

the GPBE, corresponding to the low-order moments (up to the first or second order) of the underlying number density functions (NDFs). The resulting closure problem is frequently overcome by using approximations; namely, it is assumed that all the elements of the disperse phase share the same mass (or size) and velocity. In the E-L approach, the motion of individual or groups of bubbles is simulated by solving the force balance on that bubble. The trajectories combined with the continuous phase each represent one of the possible realizations of the multiphase flow and, in the case of pseudo-stationary flows, averaged measurable quantities characteristic of the flow can be determined via the ergodic theorem and time averages. The TFM and E-L methods rose to great popularity in the past and have been employed for many different applications [4, 5, 6, 7, 8, 9, 10].

Although the TFM and E-L methods are the most popular CFD models in the field of multiphase flows, both of them have disadvantages. Due to the approximation adopted to overcome the closure problem (i.e. all elements of the disperse phase share the same mass/size and velocity), the TFM, with just two phases, loses the capability to predict polydispersity effects, and it cannot be applied to the cases in which the Knudsen number (Kn) of the elements of the disperse phase is larger than 0.1 [1, 11]. On the other hand, the E-L method, based on solving the GPBE via Lagrangian tracking, is characterized by large computational costs. Moreover, statistical noise cannot be avoided, and the results can be highly dependent on mesh size [12]. In order to alleviate the drawbacks of the TFM and E-L methods, another method can be employed. In this method, the evolution of the disperse phase is described via moments of the GPBE, which are solved with an alternative method, known as the quadrature-based moments method (QBMM) [1, 13]. In the QBMM, the GPBE is transformed to the moment transport equations, as with the standard TFM; however, now more moments are transported and an accurate closure strategy is assumed. It is worth mentioning that the manner in which moments are advected in space is crucial. When the velocity of the elements of the disperse phase is assumed to be known, the moment transport equations can be solved easily. The most simple and widely used assumption is that

all elements of the discrete phase have the same velocity. This approach is often known as CFD-PBE or CFD-PBM coupling [14, 15, 16, 17, 18, 19, 20, 21, 22]. It should be mentioned that QBMM is not the only way to solve the GPBE in this context. Another interesting approach is to use a class method, e.g. the MUSIG approach [23, 24, 25].

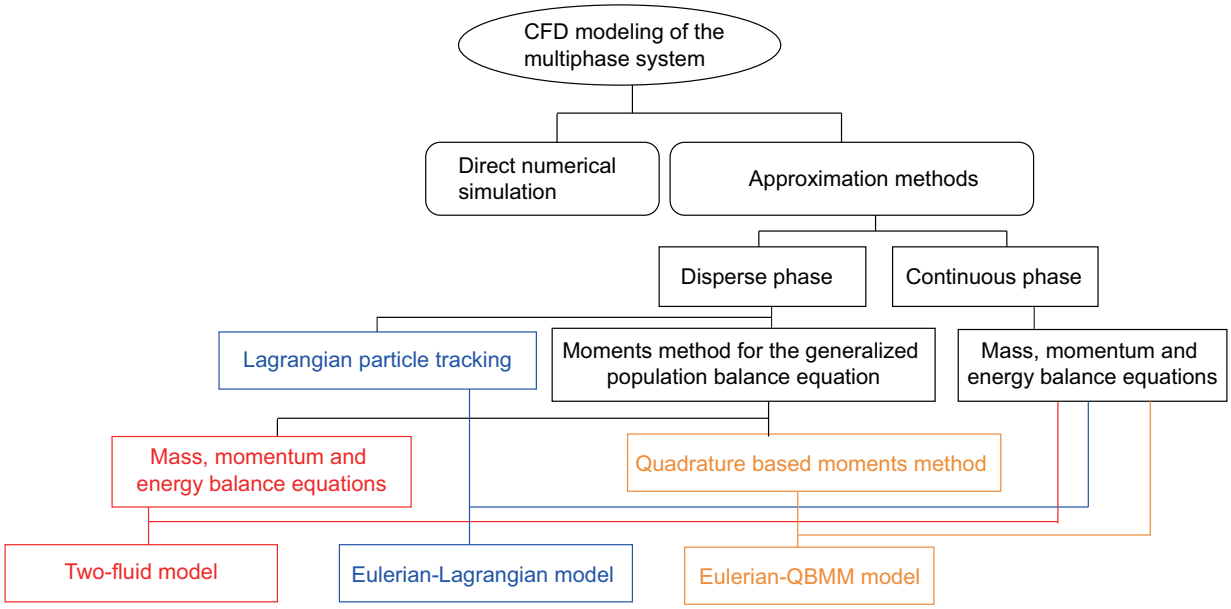


Figure 1: Algorithms for multiphase flows.

Traditional CFD-PBE coupling also has some drawbacks, the most important being that all the elements of the disperse phase move with the same velocity as mentioned above. Possible solution is to employ a moment/multi-fluid methods [26] or determines the bubble velocities from a linear interpolation between the averaged bubble velocity with the gas phase velocity [27, 28]. Another interesting solution is to treat both size/mass and velocity as internal coordinates in the GPBE. This is what was proposed in the original formulation of the direct quadrature method of moments (DQMOM) [29]. This method is similar to the inhomogenerous MUSIG model [23, 24, 25], and it has been employed to predict sprays and fluidized beds in other works [30, 31]. However, the disadvantage of this method is that other internal coordinates are difficult to include. Another previously proposed solution is to model the dependence of the NDF on velocity with presumed distributions (e.g., Gaussian

distribution), then transform the GPBE to the moment transport equations together with the conditional quadrature method of moments (CQMOM) [32] to obtain the multi-dimensional velocity abscissas. Other internal coordinates (e.g., particle size) can also be included in this method, and it has been employed to predict the PTC in gas-particle flows [33, 34, 35].

In this work we focus on a different method, which is numerically more stable and less complex. A relationship between the mass/size and velocity of the elements of the disperse phase is assumed. This relationship takes the form of an algebraic polynomial dependency resulting in a size-conditional velocity polynomial approximation (VPA) [1]. Using this method, the particle velocities can be calculated from the moments. By employing QMOM or EQMOM, the higher-order spatial unclosed mixed-moments flux can be updated, and the mixed-moments system can be solved. The essence of this method is that it can provide a continuous velocity distribution, which implies that the reconstructed particle velocity is continuous with respect to the particle size. Once the self-closed moments system is solved, it can be coupled with the governing equations of the continuous phase. Here, this method is labelled "Eulerian QBMM" with the "Eulerian" referring to the approach for the continuous phase and "QBMM" denoting the solution method for the discrete phase. It should be noted that the transported equations both for the continuous phase and disperse phase in the E-QBMM are also solved under the Eulerian framework. However, it is our intention to differentiate the E-QBMM with the classical macroscopic TFM, since only mesoscopic closure is employed for the disperse phase. With adequate closure (e.g., using CQMOM to model the velocity distribution instead of the VPA approach), the E-QBMM can be used to predict the PTC which can never be predicted by the TFM. In some other works, the "Eulerian" is somewhat neglected and only the "QBMM" is retained to highlight the mesoscopic closure [34, 35].

The algorithm and code discussed in the present form only apply to monokinetic bubbly flows. This is because the monokinetic assumption for bubbly flows is reasonable, since a unique relationship between the bubble velocity and size exists at small Stokes numbers,

which implies that only one bubble velocity for a certain bubble size will be observed at any point in the flow at a given time. Moreover, we neglect the coalescence and breakage process, which was proven to have no impact on our test cases' results, although including them is straightforward. On the other hand, additional kinetic terms from particulate kinetic theory need to be implemented if one wants to simulate multiphase particle flows. To the authors' knowledge, the Eulerian QBMM with the VPA has only been investigated by Yuan et al. [36], who employed the EQMOM with two primary nodes and three secondary nodes to simulate 2-D bubbly flows, but their source codes have not been made publicly available.

Firstly, we systematically verify and validate the specific formulation of the Eulerian QBMM to verify the numerical attributes, then implement the algorithms in the open-source CFD code OpenFOAM-5.x. Two open-source solvers are released under the GNU General Public License in a publicly available software repository that includes detailed tutorials. In the one-way coupled solver `oneWayGPBEFoam`, the effects of the disperse phase on the continuous phase were neglected. This solver can be used to simulate passive scalar transport problems. In the two-way coupled solver `twoWayGPBEFoam`, the disperse phase and the continuous phase are fully coupled via the momentum interface exchange terms. In contrast to uniform velocities, both solvers can be employed to solve the size-segregation problems due to their capability to predict the size-conditional velocity by employing the VPA. Meanwhile, the reconstructed particle velocity is continuous with respect to the particle size. One one-way test case and three two-way test cases were included to verify the algorithm and the implementation. The predicted results were compared with the analytical solutions and experimental data. The remainder of the document is structured as follows, in Section 2, the governing equations for the disperse phase and continuous phase are revisited. The fully coupled algorithm is comprehensively discussed. The solver's structure is described in Section 3. In Section 4, the one-way coupled code `oneWayGPBEFoam` and the fully two-way coupled code `twoWayGPBEFoam` are validated by different test cases. Finally, conclusions are drawn in Section 5.

2. Governing equations and the coupling procedure

2.1. Eulerian method for the continuous phase

The mass conservation equation for the continuous phase is described by the continuity equation [2]:

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

where α_{c} is the phase fraction of the continuous phase, ρ_{c} is the density, and $\mathbf{U}_{\mathrm{c}}$ is the velocity. The momentum conservation can be described by the volume-averaged momemtum equation [2]:

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

where $\mathbf{R}_{\mathrm{c}}$ represents the stress tensor, p is the average pressure, $\mathbf{g}$ is the gravity acceleration vector, and $\mathbf{M}_{\mathrm{d}}$ is the momentum interface exchange term, which will be discussed in next section. Readers are referred to other work for more information on the Eulerian method [37].

2.2. GPBE/QBMM for the disperse phase

In the GPBE, the internal coordinates include both active and passive variables, namely the velocities and sizes. Although the released codes in this work were implemented in 3D, in order to make the algorithm clearer, equations are presented in 2D. The 3D NDF with the 2D velocity components u and v and the diameter d is written as $n(t, x, y, u, v, d)$, in which the three internal coordinates are u, v, and d. The GPBE for the disperse phase can be written as [1]

$$\frac{\partial n}{\partial t}+\nabla_{\mathbf{X}} \cdot\left(\mathbf{U}_{\mathrm{d}} n\right)+\nabla_{\mathbf{U}_{\mathrm{d}}} \cdot(\mathbf{A} n)=S, \tag{3}$$

where $\mathbf{U}_{\mathrm{d}}=(u, v)$ is the transported velocity, S is the possible source term accounting for bubble collision, coalescence and breakup, which are neglected in this work, and $\mathbf{A}$ is the

acceleration term. Following [1], it is given by

$$\mathbf{A} = -\frac{1}{\rho_{\mathrm{d}}}\nabla p_{\mathrm{c}} + \mathbf{g} + \frac{1}{\tau}\left(\mathbf{U}_{\mathrm{c}} - \mathbf{U}_{\mathrm{d}}\right), \tag{4}$$

where ρ_{d} is the disperse phase density, p_{c} is the pressure of the continuous phase, $\mathbf{g}$ is the gravity acceleration vector and τ is the relaxation time. The GPBE equation reported in Eq. (3) is also the same fundamental equation for the disperse phase in the multiphase phase particle-in-cell (MP-PIC) method [38]. In order to obtain a numerical solution, Eq. (3) is often written in terms of the moments of the NDF, defined by

$$m_{j,k,i} = \int u^j v^k d^i n \mathrm{d}u\mathrm{d}v\mathrm{d}d. \tag{5}$$

Applying the transformation to Eq. (3) leads to the moment transport equations, which are solved in this work using the operator splitting procedure in this work [1]. In the first step of the operator splitting procedure, the convection part of Eq. (3) is solved,

$$\frac{\partial n}{\partial t} + \frac{\partial un}{\partial x} + \frac{\partial vn}{\partial y} = 0. \tag{6}$$

Eq. (6) is transformed into the following mixed-moments transport equations by the definition reported in Eq. (5):

$$\frac{\partial m_{j,k,i}}{\partial t} + \frac{\partial m_{j+1,k,i}}{\partial x} + \frac{\partial m_{j,k+1,i}}{\partial y} = 0. \tag{7}$$

Depending on the value of j, k, and i, Eq. (7) has a specific physical meaning. For example, if $i = j = k = 0$ then Eq. (7) becomes the transport equation of the total particle number density, defining the evolution in space and time of the number of particles per unit volume. Higher-order moments have other physical meanings; readers are referred to [1] for further details.

When $m_{j,k,i}$ are being solved, the $m_{j+1,k,i}$ and $m_{j,k+1,i}$ need to be known, which implies

that Eq. (7) is not closed. In this work, it is closed using the quadrature method of moments (QMOM) [39], in which any higher-order mixed moment can be calculated as

$$m_{j,k,i} = \sum_{\beta=1}^{N} w_\beta u(d_\beta)^j v(d_\beta)^k d_\beta^i, \tag{8}$$

where w_β and d_β are the weights and abscissas, and N is the number of the nodes. The weights and abscissas can be calculated using the moments inversion algorithm (e.g., the Wheeler algorithm [40]) from the initial pure moments $m_{0,0,i}$. In Eq. (8) it is moreover assumed that the velocity of the elements of the disperse phase depends on size only via the algebraic functions $u(d)$ and $v(d)$. A suitable approximation [1] is to use a second-order polynomial:

$$u\left(d_\beta\right) = u_0 + u_1 d_\beta + u_2 d_\beta^2, v\left(d_\beta\right) = v_0 + v_1 d_\beta + v_2 d_\beta^2. \tag{9}$$

Eq. (9) is the particle velocity, conditioned on the particle size, and the right-hand side is a local second-order velocity polynomial approximation (VPA) of this conditional velocity. The coefficients u_i and v_i are the velocity polynomial coefficients (VPCs). Once the VPCs are determined, the conditional particle velocities can be calculated, then the mixed-moments transport equations reported in Eq. (7) are closed and ready to be solved. As explained above this assumption is valid only for moderate Stokes numbers. This is indeed the case for gas bubbles moving in a liquid, but this assumption does not have to be valid for general multiphase flows.

The VPCs can be obtained by rewriting the joint NDF as [1, 36]

$$n(u, v, d) = n(d)\delta\left(u - u(d)\right)\delta(v - v(d)). \tag{10}$$

Substituting Eq. (9) into Eq. (10) leads to:

$$n(u, v, d) = n(d)\delta(u - u_0 - u_1 d_\beta - u_2 d_\beta^2)\delta(v - v_0 - v_1 d_\beta - v_2 d_\beta^2), \tag{11}$$

and the mixed moments can be calculated by

$$m_{j,k,i} = \sum_{\beta=1}^{N} w_\beta \left(u_0 + u_1 d_\beta + u_2 d_\beta^2\right)^j \left(v_0 + v_1 d_\beta + v_2 d_\beta^2\right)^k d_\beta^i. \tag{12}$$

For example, in the expanded form of Eq. (12) the velocity component u can be written as the following equations:

$$\begin{aligned} m_{1,0,0} &= u_0 m_{0,0,0} + u_1 m_{0,0,1} + u_2 m_{0,0,2}, \\ m_{1,0,1} &= u_0 m_{0,0,1} + u_1 m_{0,0,2} + u_2 m_{0,0,3}, \\ m_{1,0,2} &= u_0 m_{0,0,2} + u_1 m_{0,0,3} + u_2 m_{0,0,4}. \end{aligned} \tag{13}$$

The velocity component u_i in Eq. (13) can be obtained by solving the following linear system:

$$\begin{bmatrix} u_0 \\ u_1 \\ u_2 \end{bmatrix} = \begin{bmatrix} m_{0,0,0} & m_{0,0,1} & m_{0,0,2} \\ m_{0,0,1} & m_{0,0,2} & m_{0,0,3} \\ m_{0,0,2} & m_{0,0,3} & m_{0,0,4} \end{bmatrix}^{-1} \begin{bmatrix} m_{1,0,0} \\ m_{1,0,1} \\ m_{1,0,2} \end{bmatrix}. \tag{14}$$

As long as the initial pure moments and the initial particle velocity are given, the VPCs for u component can be calculated using Eq. (14). The VPCs for v component can be calculated in a similar way. After the VPCs are calculated, the remaining spatial moments can be calculated by the QMOM, and the pure convection mixed-moments transport equations can be solved. It should be stressed here that when Eq. (7) is solved using the finite-volume method, the flux of the mixed moments should be able to ensure that the mixed moments can be realized for the next time step. In this work, the higher-order realizable flux split scheme developed by Vikas et al. [41] was employed. It is called the Vikas scheme for simplicity.

The original Vikas scheme was developed for the kinetic equation, in which the velocity is the internal coordinate. However, it is straightforward to extend the Vikas scheme for the GPBE, in which a joint NDF distribution is employed in this work. The spirit of the

Vikas scheme is to employ the monotonic upwind scheme for conservation laws (MUSCL) type of reconstruction to the moments flux at the grid interface, which is calculated by the piecewise linear (or polynomial) reconstruction of the weights and the piecewise constant reconstruction of other variables. It can be divided into two main steps:

1. Discretize the 1-D version of Eq. (7) using an explicit time-marching scheme:

$$M_{j,k,i}^{I,t+\Delta t} = M_{j,k,i}^{I} - \frac{\Delta t}{\Delta x}\left(F^{I+1/2} - F^{I-1/2}\right), \tag{15}$$

   where $M_{j,k,i}^{I,t+\Delta t}$ is the average value of $m_{j,k,i}$ over the Ith grid at $t+\Delta t$, $F^{I+1/2}$ and $F^{I-1/2}$ are the moments flux functions defined at the interface $I+1/2$ and $I-1/2$ for Ith grid, respectively.

2. Instead of interpolating the flux from the moments, the Vikas scheme calculates from the weights and abscissas as follows:

$$\begin{aligned} F^{I+1/2} = &\sum_{\beta=1}^{N} \max(u_\beta^I, 0)\Delta S\left(w_\beta^I + \sigma_\beta^I \frac{\Delta x}{2}\right)\left(d_\beta^I\right)^i\left(u_\beta^I\right)^j\left(v_\beta^I\right)^k \\ &+ \sum_{\beta=1}^{N} \min(u_\beta^{I+1}, 0)\Delta S\left(w_\beta^{I+1} - \sigma_\beta^{I+1} \frac{\Delta x}{2}\right)\left(d_\beta^{I+1}\right)^i\left(u_\beta^{I+1}\right)^j\left(v_\beta^{I+1}\right)^k, \end{aligned} \tag{16}$$

   where $\Delta S = \Delta y \Delta z$, which is the grid interface surface normal to the x plane. The flux calculation corresponds to the flux-vector splitting procedure: the first term on the R.H.S of Eq. (16) denotes the bubbles moving from the Ith grid crossing the $I+1/2$ interface, whereas the second denotes the bubbles moving from the $I+1$th grid crossing the $I+1/2$ interface. The flux at $I-1/2$ can be calculated in a similar way.

Readers may find the Vikas scheme only achieves 1st-order accuracy over time. Following the spirit of total variation diminishing (TVD) time stepping, multi-stage explicit time-integration schemes can be used in practice to obtain a higher-order time scheme (the

subscripts are omitted for brevity):

$$\begin{aligned} m^* &= m + F(m), \\ m^{**} &= m^* + F(m^*), \\ m^{t+\Delta t} &= \frac{1}{2}\left(m^* + m^{**}\right), \end{aligned} \tag{17}$$

where F is the flux function. It can easily be proven that the two-stage second-order Runge-Kutta method combined with the flux calculation based on the Vikas scheme, as in the equations above, can ensure that the moments are realizable.

The solution procedure (operator splitting step 1) described above can be used to simulate pure convection problems, i.e. where no other acceleration forces are considered. If acceleration is included, the operator splitting step 2 cannot be omitted, as discussed below. After the convection step, the pure moments $m_{0,0,i}$ and the mixed moments $m_{1,0,i}$ can be used to compute the so-called sample VPCs u_1^* and v_1^* based on Eq. (14). Further, the sample size weights w_β^* and abscissas d_β^* need to be calculated. Then, the sample velocities corresponding to each d_β^* are calculated by

$$u^*\left(d_\beta^*\right) = u_0^* + u_1^* d_\beta^* + u_2^* {d_\beta^*}^2, v^*\left(d_\beta^*\right) = v_0^* + v_1^* d_\beta^* + v_2^* {d_\beta^*}^2. \tag{18}$$

The velocities calculated in Eq. (18) correspond to the new particle velocity after the moments convections. Afterwards, the sample velocities need to be updated due to the acceleration term $\mathbf{A}$. In the case of a single particle with the particle size abscissas d_β^*, the sample velocity can be updated by the following ordinary differential equation (ODE):

$$m_{d_\beta^*}\frac{\mathrm{d}\mathbf{U}_{d_\beta^*}}{\mathrm{d}t} = m_{d_\beta^*}\mathbf{g} - \frac{m_{d_\beta^*}}{\rho_{d_\beta^*}}\nabla p_\mathrm{c} + \frac{m_{d_\beta^*}}{\rho_{d_\beta^*}}\frac{1}{\tau}\left(\mathbf{U}_\mathrm{c} - \mathbf{U}_{d_\beta^*}\right), \tag{19}$$

where $\mathbf{U}_{d_\beta^*} = \left(u^*\left(d_\beta^*\right), v^*\left(d_\beta^*\right), 0\right)$ for 2-D cases. The first term and the second term on the R.H.S. of Eq. (22) represent the gravity/buoyancy force and the drag force, respectively.

Substituting the Bernoulli equation

$$\nabla p_{\mathrm{c}} = \rho_{\mathrm{c}} \mathbf{g} \tag{20}$$

into Eq. (22) leads to

$$m_{d^*_\beta} \frac{\mathrm{d}\mathbf{U}_{d^*_\beta}}{\mathrm{d}t} = m_{d^*_\beta} \mathbf{g} \left(1 - \frac{\rho_{\mathrm{c}}}{\rho_{d^*_\beta}} \right) + \frac{m_{d^*_\beta}}{\rho_{d^*_\beta}} \frac{1}{\tau} \left(\mathbf{U}_{\mathrm{c}} - \mathbf{U}_{d^*_\beta} \right). \tag{21}$$

The analytical solution of Eq. (21) is

$$\mathbf{U}_{d^*_\beta}^{t+\Delta t} = \mathbf{U}_{d^*_\beta} e^{-\tau^{-1} \cdot \Delta t / \rho_{d^*_\beta}} + \left(\mathbf{U}_{\mathrm{c}} + \frac{\mathbf{g} \left(1 - \rho_{\mathrm{c}} / \rho_{d^*_\beta} \right)}{\tau} \right) \left(1 - e^{-\tau \cdot \Delta t / \rho_{d^*_\beta}} \right). \tag{22}$$

It should be noted here that other temporal integration schemes can be applied to solve Eq. (22), such as the Euler implicit method and the Euler explicit method. After the velocities have been updated by the ODE, the mixed moments for the next time step can be calculated with Eq. (8).

In summary, the GPBE for the disperse phase can be transformed into the moment transport equations, which can be solved using the following procedure:

1. From the given particle diameters and the velocities, calculate the initial pure moments and mixed moments with Eq. (8).
2. Calculate the higher-order mixed moments with Eq. (12) using QMOM.
3. Update the mixed moments as reported in Eq. (7) for the next time step. Points 1 to 3 constitute the first step of the operator splitting procedure.
4. Re-calculate the weights and abscissas with the moments inversion algorithm. Re-calculate the sample VPCs with the linear system as reported in Eq. (14) from the updated pure moments and mixed moments.
5. Calculate the sample velocities as reported in Eq. (22).
6. Re-calculate the mixed moments with Eq. (8); they are ready to enter the next time step. Meanwhile, update the sample VPCs and the sample velocities for the next time

step. Points 4 to 6 constitute the second step of the operator splitting procedure.

On theoretical grounds, to solve the multivariate NDF transported equation, the CQ-MOM can be used as an alternative method. Therefore, it is interesting to compare the VPA with the CQMOM. For simplicity, considering the bivariate NDF $n(x,t,u,d)$, the mixed moments in the CQMOM can be computed with

$$m_{j,k} = \sum_{\beta=1}^{N_\beta} w_\beta d_\beta^i \left(\sum_{\gamma=1}^{N_\gamma} w_{\beta,\gamma} u_{\beta,\gamma}^j \right), \tag{23}$$

where $w_{\beta,\gamma}$ and $u_{\beta,\gamma}$ are the conditional weights and velocity abscissas for d_β, respectively, while N_β and N_γ are the node numbers for the diameter and velocity, respectively. If one node is employed for the velocity abscissas, Eq. (23) degenerates into the following

$$m_{j,k} = \sum_{\beta=1}^{N_\beta} w_\beta d_\beta^k u_\beta^j, \tag{24}$$

since $w_{\beta,\gamma} = 1$. In order to calculate the unknown u_β, Eq. (24) can be expanded in the following form:

$$\begin{aligned} m_{1,0} &= w_1 u_1 + w_2 u_2 + w_3 u_3, \\ m_{1,1} &= w_1 d_1 u_1 + w_2 d_2 u_2 + w_3 d_3 u_3, \\ m_{1,2} &= w_1 d_1^2 u_1 + w_2 d_2^2 u_2 + w_3 d_d^2 u_3. \end{aligned} \tag{25}$$

The velocity abscissas can be solved in the following matrix form:

$$\begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix} = \begin{bmatrix} w_1 & w_2 & w_3 \\ w_1 d_1 & w_2 d_2 & w_3 d_3 \\ w_1 d_1^2 & w_2 d_2^2 & w_3 d_3^2 \end{bmatrix}^{-1} \begin{bmatrix} m_{1,0} \\ m_{1,1} \\ m_{1,2} \end{bmatrix}. \tag{26}$$

The unknown u_β ($\beta = 1, 2, 3$) are actually the velocity conditional moments. Readers may

find that if a first-rank VPA is employed (the second and third VPCs equal to 0), Eq. (24) is identical with Eq. (12) for bivariate NDF and the velocities which are conditional on the bubble diameter are exactly the same. This implies the one-node CQMOM is equivalent with the first-rank VPA. However, the VPA approach offers some advantages compared to the CQMOM in particular if the EQMOM is employed to reconstruct the NDF, in which both the NDF and the velocity conditioned on size has a continuous shape, and the moments flux based on any bubble diameter can be calculated. Further, if more velocity nodes are employed in the CQMOM, additional moments inversion step to find the secondary velocity weights and abscissas from the velocity conditional moments u_β cannot be avoided which will increase the computational costs. The dimensional splitting procedure should also be used to update the moments transport equations. Moreover, different CQMOM permutation should be considered for 3D velocity phase. These drawbacks comes with an advantage that the multi-nodes CQMOM can be employed to predict the PTC, which will never be predicted by the VPA approach. However, the implementation of the VPA, no matter first-rank or higher-rank, is identical with the one-node CQMOM, and the secondary moments inversion step in the multi-nodes CQMOM can be avoided due to monokinetic assumption.

2.3. Coupling procedure of the Eulerian QBMM

The coupling procedure of the Eulerian QBMM resembles the procedure of the E-L method. In the one-way coupling, the disperse phase can be seen as a passive scalar. Therefore, the phase fraction α_c in Eq. (1) and Eq. (2) is assumed to be 1.0, and the momentum interface exchange terms for the continuous phase are assumed to be null. Due to the simplicity of the one-way coupling, in the following we focus on the two-way coupling.

The first step is to obtain the pressure Poisson equation from the continuity equation of the continuous phase. The semi-discretised form of Eq. (2) can be written as:

$$a_P \mathbf{U}_P - \frac{\mathbf{U}_P^t}{\Delta t} + \sum a_N \mathbf{U}_N = -\alpha_\mathrm{c} \nabla \frac{p_\mathrm{c}}{\rho_\mathrm{c}} - \frac{1}{\rho_\mathrm{c}} \mathbf{M}_\mathrm{d}, \tag{27}$$

where a_P is the matrix diagonal coefficients, a_N is the matrix non-diagonal coefficients, $\mathbf{U}_P$ is the unknown velocity defined at the cell center, $\mathbf{U}_N$ is the known velocity for the cell neighbours. Here, we assume that the continuous phase is incompressible, and the gravity body force is neglected since buoyancy is only relevant for the disperse phase. It is straightforward to extend the incompressible code into a compressible one by retaining the density within these terms instead of dividing it out. The predicted velocity is defined as:

$$\mathbf{HbyA} = \frac{1}{a_P}\left(\frac{\mathbf{U}_P^t}{\Delta t} - \sum a_N \mathbf{U}_N\right). \tag{28}$$

Substituting Eq. (28) in Eq. (27), we have:

$$\mathbf{U}_P = \mathbf{HbyA} - \frac{\alpha_\mathrm{c}}{a_P}\nabla\frac{p_\mathrm{c}}{\rho_\mathrm{c}} - \frac{1}{a_P\rho_\mathrm{c}}\mathbf{M}_\mathrm{d}. \tag{29}$$

It should be stressed here that the momentum interface exchange term $\mathbf{M}_\mathrm{d}$ in the Eulerian QBMM is different with the modeling in the TFM. This is indeed the essence of the Eulerian QBMM. In the Eulerian QBMM, particles of different sizes make their own contributions. Therefore, $\mathbf{M}_\mathrm{d}$ can be decomposed into their separate contributions:

$$\mathbf{M}_\mathrm{d} = \sum_{\beta=0}^{N}\left(\alpha_\beta\frac{3}{4}\frac{C_{\mathrm{D},d_\beta}\rho_\mathrm{c}}{d_\beta}|\mathbf{U}_\mathrm{c} - \mathbf{U}_{d_\beta}|\left(\mathbf{U}_\mathrm{c} - \mathbf{U}_{d_\beta}\right)\right), \tag{30}$$

where α_β, C_{D,d_β} and $\mathbf{U}_{d_\beta}$ are the phase fraction, drag coefficients and the velocity of the particles with a diameter d_β, respectively. Obviously, the momentum interface exchange term in the Eulerian QBMM as reported in Eq. (30) is different from that calculated with the TFM method. Moreover, if the abscissas are identical, the $\mathbf{M}_\mathrm{d}$ in the Eulerian QBMM will degenerate into the $\mathbf{M}_\mathrm{d}$ in the TFM method.

As some of the drag contribution can boost the diagonal dominance of the continuous phase momentum equation, Eq. (30) is decomposed into two parts and Eq. (29) can be

re-written as

$$\mathbf{U}_P = \mathbf{HbyA} - \frac{\alpha_\mathrm{c}}{a_P}\nabla\left(\frac{p_\mathrm{c}}{\rho_\mathrm{c}}\right) - \frac{1}{a_P\rho_\mathrm{c}}\sum_{\beta=0}^{N}\left(\alpha_\beta\frac{3}{4}\frac{C_{\mathrm{D},d_\beta}\rho_\mathrm{c}}{d_\beta}|\mathbf{U}_\mathrm{c}-\mathbf{U}_{d_\beta}|\mathbf{U}_\mathrm{c}\right). \tag{31}$$

This mathematical manipulation is usually called the momentum predictor step, and Eq. (31) provides the predicted velocity **HbyA**, which is not divergence-free since the pressure has not been updated yet. Substituting Eq. (31) into Eq. (1) leads to

$$\frac{\partial\alpha_\mathrm{c}}{\partial t} + \nabla\cdot\alpha_\mathrm{c}\left(\mathbf{HbyA} - \frac{\alpha_\mathrm{c}}{a_P}\nabla\left(\frac{p_\mathrm{c}}{\rho_\mathrm{c}}\right) - \frac{1}{a_P}\sum_{\beta=0}^{N}\frac{3\alpha_\beta C_{\mathrm{D},d_\beta}}{4d_\beta}|\mathbf{U}_\mathrm{c}-\mathbf{U}_{d_\beta}|\mathbf{U}_\mathrm{c}\right) = 0. \tag{32}$$

Re-arranging Eq. (32), the pressure Poisson equation can be written as follows:

$$\nabla\cdot\left(\frac{\alpha_\mathrm{c}^2}{a_P}\nabla\left(\frac{p_\mathrm{c}}{\rho_\mathrm{c}}\right)\right) = \frac{\partial\alpha_\mathrm{c}}{\partial t} + \nabla\cdot\alpha_\mathrm{c}\left(\mathbf{HbyA} - \frac{1}{a_P}\sum_{\beta=0}^{N}\frac{3\alpha_\beta C_{\mathrm{D},\beta}}{4d_\beta}|\mathbf{U}_\mathrm{c}-\mathbf{U}_{d_\beta}|\mathbf{U}_\mathrm{c}\right). \tag{33}$$

The pressure predicted in Eq. (33) can be used to update the new velocity as reported in Eq. (31) by adding back the contribution of the drag force in the disperse phase. In the spirit of the PISO algorithm [42], 2 or 3 inner iterative steps will ensure that the continuous phase velocity field is divergence-free.

Different sparse linear system solvers can be employed to solve the discretized equations. For the moments equations, a non-iterative diagonal solver can be used due to explicit time marching in the Vikas scheme, and the diagonal matrix is constructed. The preconditioned bi-conjugate gradient (PBICG) solver is usually chosen to solve for the continuous phase velocity and turbulence-related equations (e.g., k and ε equations) since a non-symmetric sparse linear system is constructed due to the existence of the convection term. For the Poisson pressure equation, the preconditioned conjugate gradient (PCG) solver can be used due to the symmetric structure of the equation. Moreover, the geometric algebraic multi-grid (GAMG) solver can be also used in certain cases to speed up the simulations.

3. Solver structure

In this section, we briefly introduce the code `twoWayGPBEFoam`. The main feature of `twoWayGPBEFoam` is its strict adherence to OpenFOAM coding practices, meaning that its design is entirely in the form of C++ classes. The strong power of the C++ object-oriented programming (OOP) technique implies that the solvers can be extended easily, as seen in other works [43, 44, 45]. For example, in *force/drag*, we implemented different drag models: the `SchillerNaumann`, `Grace` and `constantDrag` models. Other drag models can easily be implemented by copying the existing drag model, then implementing the specific `CeRe()` function in the `.C` file.

In the code, readers will find three different folders: *src*, *solvers* and *tutorials*. Most of the GPBE/QBMM solving algorithms are encapsulated in several classes in *src*. The coupling between the disperse phase and the continuous phase was implemented in *solvers*, in which `oneWayGPBEFoam` and `twoWayGPBEFoam` are included. The *tutorials* folder includes several test cases investigated in this work. Let us examine `twoWayGPBEFoam` as an example of the structure of the implementation. In `twoWayGPBEFoam.C`, the GPBE class can be provoked by simply including the header file, and created with an auto pointer:

```
#include "MOM.H"

...

autoPtr<MOM> GPBE(MOM::New(mesh, GPBEProperties));

...
```

The 3D moment transport equations were solved by the `solve()` pointer function:

```
GPBE->solve();
```

This function actually invokes the `solve()` function defined in `MOM.C` and located at *src*, in which the moment transport equations were solved using the finite volume method:

```
/* * * * * * * * * * * * Moments advection step * * * * * * * * * * * */


    forAll(M_[0], cellI)
    {
        //- Update weighs and abscissas
        calculateAbs(cellI);
        //- Update VPCs
        updateVPC(cellI);
        //- Update bubble velocities
        updateVelocity(cellI);
    }


    //- Correct boundaries and update mean diameter
    correctBoundaryAndCells();


    //- Update flux by realizable high-order scheme
    realizableMom_.correctFlux
    (
        weis_,
        abs_,
        M_,
        mFluxX_,
        mFluxY_,
        mFluxZ_,
        particleU_,
        pureMomentsNumber_,
        nodes_
```

```
    );


    Info<< "Solving for moments advection";
    forAll(M_, kth)
    {
        scalarField& psiIf = M_[kth];
        const scalarField& psi0 = M_[kth].oldTime();
        const scalar deltaT = mesh_.time().deltaT().value();
        psiIf = 0.0;
        surfaceScalarField phiPsi =
            mFluxX_[kth] + mFluxY_[kth] + mFluxZ_[kth];

        //- Update moments explicitly
        fvc::surfaceIntegrate(psiIf, phiPsi);
        psiIf = (psi0/deltaT - psiIf)*deltaT;

        M_[kth].correctBoundaryConditions();
    }
    Ud_.correctBoundaryConditions();

    Info<< ", End moments advection " << nl
        << "Solving for moments force updating";
```

The `mFluxX_[kth]` and other similar fluxes (e.g., `mFluxY_[kth]` and `mFluxZ_[kth]`) are calculated from Eq. (16) in `Vikas.C`. For the procedure of determing the weights and abscissas, the mathematical eigenValue and eigenVector calculation codes were taken from another open-source code, `OpenQBMM`, written by Passalacqua et al. [46]. After the moments were convected, the operator split step 2 was launched:

```
/* * * * * * * * * * * * Force computing step * * * * * * * * * * * * */


    if (forceTerm_)
    {
        updateMomentsForce();
    }
    Info<< ", End moments updating\n";
```

`updateMomentsForce()` function was defined in `MOM.C`, in which the operator split step 2 was implemented:

```
    forAll(M_[0], cellI)
    {
        ...
        //- (1)
        calculateAbs(cellI);
        //- (2)
        updateVPC(cellI);
        //- (3)
        updateVelocity(cellI);
        //- (4)
        velocityODE(cellI, Uc[cellI], muc[cellI], rhoc[cellI], dT, g,
            curlUc[cellI], gradPhaseFrac, bPGrad, k[cellI], nut[cellI]);
        //- (5)
        mixedMomentsUpdate(cellI);
        ...
    }
```

Afterwards, the disperse phase fraction was updated and fed into the continuous phase momentum equation:

```
alphac = GPBE->alphac();
...
//- Construct the momentum equation
fvVectorMatrix UcEqn
(
    fvm::ddt(GPBE->alphac(), Uc) + fvm::div(alphaPhic, Uc)
  - fvm::Sp(fvc::ddt(GPBE->alphac()) + fvc::div(alphaPhic), Uc)
  + continuousPhaseTurbulence->divDevRhoReff(Uc)
 ==
  - fvm::Sp(pre, Uc)
```

Finally, the pressure Poisson equation was constructed and solved using the transient PISO algorithm [42]:

```
fvScalarMatrix pEqn
(
    fvm::laplacian(sqr(alphacf)*rAUcf, p)
  ==
    fvc::ddt(GPBE->alphac()) + fvc::div(alphacf*phiHbyA)
);
...
pEqn.solve(mesh.solver(p.select(pimple.finalInnerIter())));
...
//- Update the continuous phase velocity
Uc = HbyA + rAUc*fvc::reconstruct
   ((phicForces - pEqn.flux()/alphacf)/rAUcf);
}
```

To compile the libraries and the solver, the reader should make sure that OpenFOAM-5.x is successfully installed, then run the `Allwmake` script located at *GPBEFoam*. If the reader

wants to run some test cases, all of them can be launched by the `Allrun` script in each tutorial test case.

It should be noted here that another open-source multi-phase solvers were provided in the OpenQBMM suite by Passalacqua et al. [46]. However, the closure and coupling procedure of our code is different from theirs. On the basis of the solver structure, the codes implemented in this work are based on the existing OpenFOAM solver `DPMFoam`, in which an E-L method is implemented. The typical multi-phase solver in the OpenQBMM, namely the so-called `polydisperseBubbleFoam`, is based on the existing OpenFOAM solver `twoPhaseEulerFoam`, in which an TFM is implemented. Therefore, in our code, the momentum equation of the disperse phase is discarded, in which the disperse phase volume fraction is transported. Strictly speaking, the momentum equation of the disperse phase can be seen as the 3rd moment transport equation if the internal coordinate is the bubble diameter. However, in our code, all the moments transport equations are solved in the form of Eq. (7). Moreover, the velocity in our code is calculated by the VPA. It can be seen as a one-node CQMOM algorithm, in which the Vandermonde matrix algorithm can be discarded due to the monokinetic limit.

4. Test cases

4.1. One-way coupled test case: particle size segregation

Let us first investigate the one-way coupling. We simulate the movement of particles along the x direction as reported in Fig. 2. Only the drag force is included in this example. Initially, all the particles are at a certain location, with an equal velocity of 1 m/s. The continuous phase velocity was set at 0 m/s. As coupling is only one-way, the state of the continuous phase will not change. This implies that the continuous phase velocity will constantly be 0 m/s. When the particles move, they should stop due to the drag force at a certain time. The key point is that, since the drag force depends on the particle size, particles of different sizes decelerate differently; this is called size segregation. It becomes

obvious that this kind of segregation can only be predicted using the GPBE with the self-closure of the size-conditioned velocity. Otherwise the bubble transport velocity is unknown and no solution can be obtained. Therefore, the classical CFD-PBE cannot be used for such problems. By the use of the VPA approach, the size-conditioned velocity can be calculated from the moments and the equation system can be calculated.

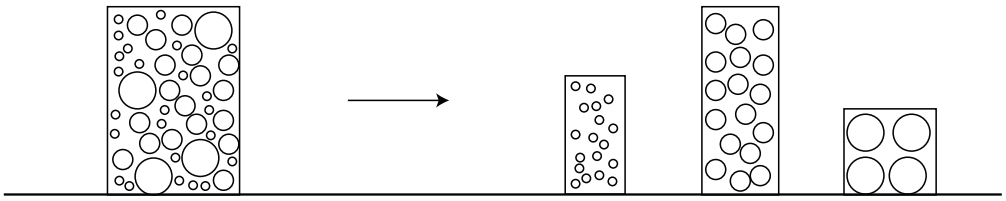


Figure 2: Particle size segregation phenomenon due to the different drag forces exerted on particles of different sizes.

In this specific case, an analytical solution can be obtained starting out from the Lagrangian equation:

$$\frac{\mathrm{d}u(d)}{\mathrm{d}t} = -\frac{u(d)}{\tau(d)}, \tag{34}$$

$$\mathrm{d}x = u(d)\mathrm{d}t, \tag{35}$$

in which x denotes the travel distance. The analytical solutions of Eq. (34) and Eq. (35) are

$$u(d) = \exp\left(-\frac{t}{\tau(d)}\right) u^0(d), \tag{36}$$

and

$$x(d) = \int_0^t \exp\left(-\frac{t}{\tau(d)}\right) \mathrm{d}t, \tag{37}$$

in which the $u^0(d)$ represents the initial particle velocity (1 m/s).

The initial moments are reported in Table 1. It should be noted that highly accurate values (e.g., with 16-digit numbers by binary format) should be provided, otherwise moments corruption occurs and the moment inversion algorithm fails. In order to generate the moments from a mean Sauter diameter, one way is to calculate the moments with 16

digital numbers from the given abscissas and weights by Eq. (8). Another way is to assume a log-normal NDF distribution, and the moments with 16-digit numbers can be computed using the following equation:

$$m_{j,k,l,i} = u^j v^k w^l \exp\left(i\mu_{\log} + i^2\sigma_{\log}^2/2\right), \tag{38}$$

where $\mu_{\log}$ and $\sigma_{\log}$ are the mean and variance, respectively. They are the two parameters defining the log-normal distribution:

$$\mu_{\log} = \log\left(\frac{{d_{\text{mean}}}^2}{\sqrt{v_{\log} + d_{\text{mean}}^2}}\right), \tag{39}$$

$$\sigma_{\log} = \sqrt{\log\left(\frac{v_{\log}}{d_{\text{mean}}^2} + 1\right)}, \tag{40}$$

where d_{mean} is equal to the mean diameter, $\sqrt{v_{\log}}$ is the user-defined standard deviation (e.g., 15%). In this work, we employed the second method.

For the first test case, three-node QMOM was employed and moments were generated from $d_{\text{mean}} = 0.5$ mm. The corresponding values of the abscissas are 0.40305 mm, 0.52275 mm and 0.67801 mm, respectively. The relaxation time τ was simply set to be $10d^{2/3}$, as implemented in the `constantDrag` model. From these known parameters, the distance reported in Eq. (37) can be obtained. All the simulations were run for 30 seconds. From the analytical solution, the particles of the largest size ($d = 0.67801$ mm) should move 7.55 mm. The particles of the smallest size ($d = 0.40305$ mm) should move 5.43 mm. Other particles ($d = 0.52275$ mm) should move 6.43 mm. The comparison of the predicted weights with an analytical solution using the quasi-second-order Vikas scheme and the first-order upwind scheme were reported in Figs. 3 and 4. It can be seen that the unavoidable numerical dissipation due to the upwind scheme is decreased by the quasi-second-order Vikas scheme. Therefore, the following simulations are performed with the Vikas scheme to reduce the numerical diffusion. Meanwhile, both the predicted results agree well with the analytical

Moments order	One-way coupled flow	Sudden enlargement	Bubble plume
m_0	1.0000×10^{-1} mm^{-3}	$\approx 1.1165 \times 10^{7}$ m^{-3}	$\approx 6.9359 \times 10^{6}$ m^{-3}
m_1	5×10^{-2} mm^{-2}	$\approx 2.2331 \times 10^{4}$ m^{-2}	$\approx 3.5026 \times 10^{4}$ m^{-2}
m_2	$\approx 2.5562 \times 10^{-2}$ mm^{-1}	$\approx 4.5668 \times 10^{1}$ m^{-1}	$\approx 1.8086 \times 10^{2}$ m^{-1}
m_3	$\approx 1.3362 \times 10^{-2}$ mm^{0}	$\approx 9.5492 \times 10^{-2}$ m^{0}	$\approx 9.5492 \times 10^{-1}$ m^{0}
m_4	$\approx 7.1426 \times 10^{-3}$ mm^{1}	$\approx 2.0416 \times 10^{-4}$ m^{1}	$\approx 5.1552 \times 10^{-3}$ m^{1}
m_5	$\approx 3.9037 \times 10^{-3}$ mm^{2}	$\approx 4.4634 \times 10^{-7}$ m^{2}	$\approx 2.8457 \times 10^{-5}$ m^{2}

Table 1: The initial moments for all test cases (test case 1: one-way coupled flow, test case 2: sudden enlargement, test case 3: bubble plume).

solution, which confirms the suitability of the algorithm and verifies its implementation.

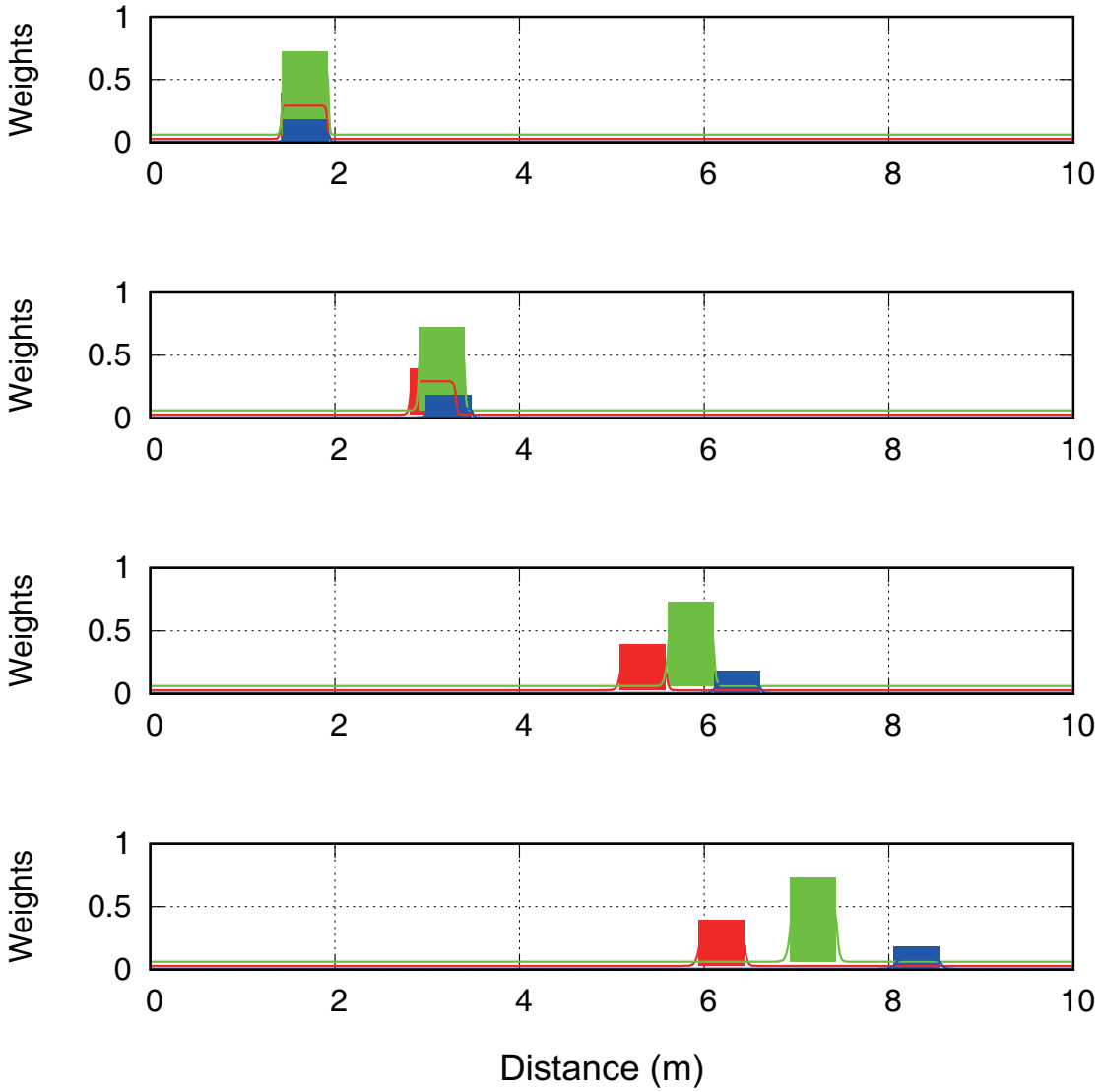


Figure 3: Comparison of the weights' location (coloured lines) predicted by the quasi-second-order Vikas scheme with the analytical solution (coloured blocks) at different times (from top to bottom).

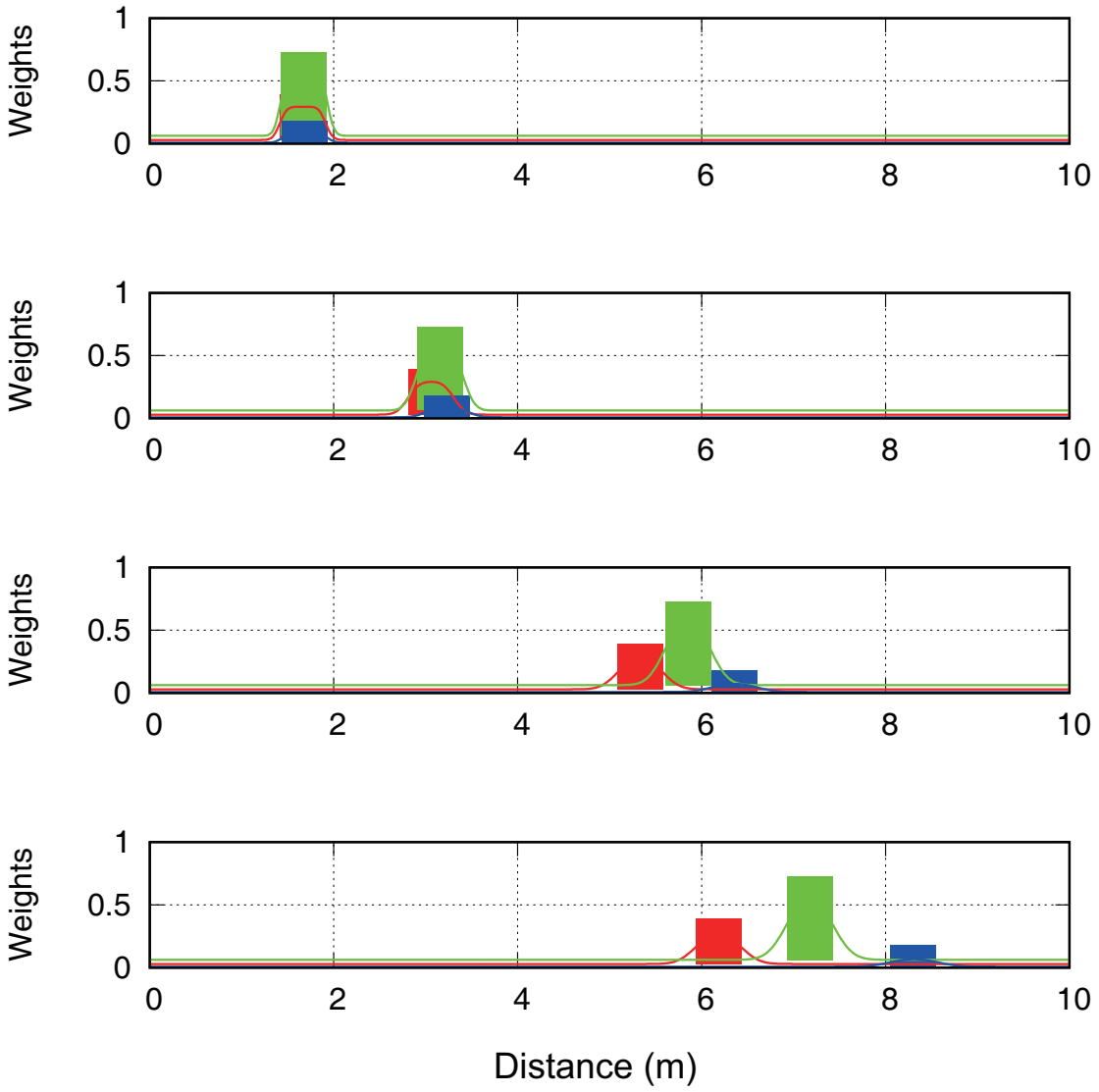


Figure 4: Comparison of the weights' location (coloured lines) predicted by the first-order upwind scheme with the analytical solution (coloured blocks) at different times (from top to bottom).

4.2. Two-way coupled test case: pipe with sudden enlargement

Next, we investigate a test case with two-way coupling. The test case studied here is that of a bubbly air/water upward flow through a pipe with a sudden enlargement, for which experiments have been carried out by Bel F'dhila [47], and simulations with the TFM have been reported by Rusche [48]; Oliveira and Issa [49]; Behzadi et al. [50]; Ullrich [51]. The feature of this test case is that there is a large separation zone at the bottom of the pipe. This case has been employed extensively to verify the two-fluid model algorithms and

implementations [50, 49, 52, 51]. Therefore, it is a good test case to compare the TFM and Eulerian QBMM algorithms. The geometry is depicted in Fig. 5. The diameters of the two pipe sections are 50 mm and 100 mm, respectively. An axial-symmetric uniform mesh ($\Delta z = 5$ mm, $\Delta x = 2.5$ mm), which provides a mesh-independent solution compared with the results predicted by higher-resolution mesh, was generated by `blockMesh`. At the inlet, the mean liquid and gas velocities are 1.57 and 1.87 m/s. The inlet phase fraction is characterised by a wall peak in experiments. However, as such experimental data is not available, the average value of the phase fraction at the inlet was given as 5% and the mean Sauter diameter was assumed to be 2 mm, as suggested in other works [49, 48].

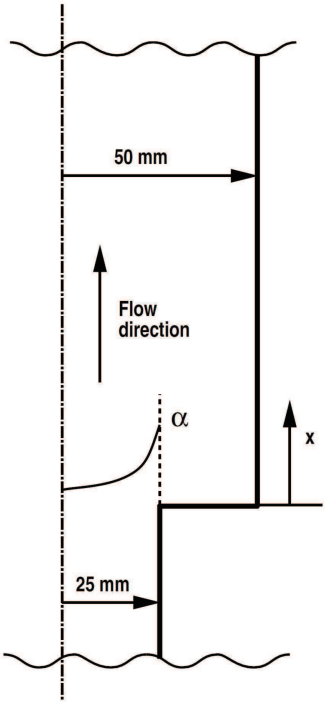


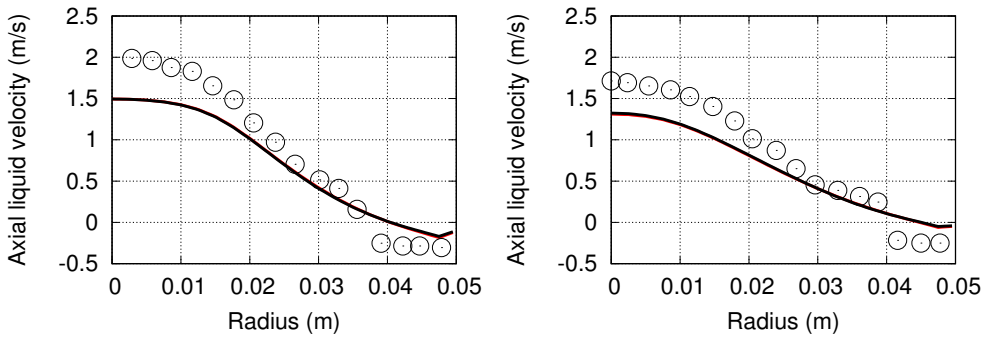
Figure 5: Schematics of the setup for the sudden enlargement test case.

The corresponding moments are calculated by assuming an initial log-normal bubble size distribution, following the calculation reported in our previous work [17]. The approximate values are given in Table 1. It should be noted that the reader should provide highly accurate values (e.g., with 16-digits numbers by binary format), otherwise moments corruption occurs

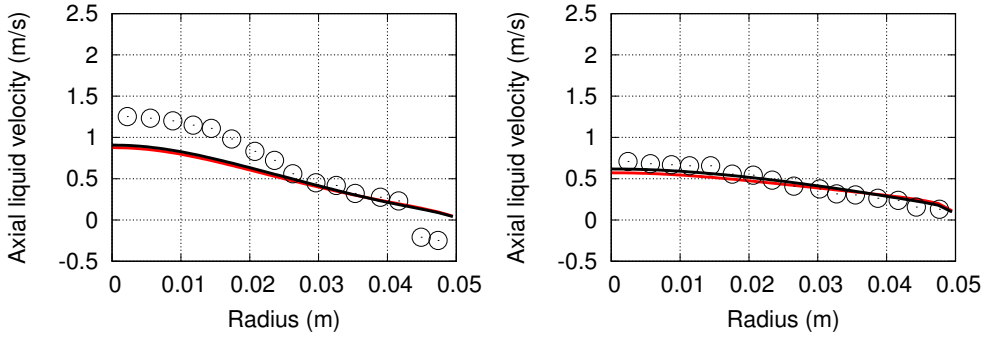
and the moment inversion algorithm fails. The gas phase is assumed to be balanced by the buoyancy and drag force. The drag coefficient was calculated following the work of Schiller and Naumann [53]. The $k - \varepsilon$ turbulence model with standard wall function correction was used. The turbulence kinetic energy of the liquid phase is calculated from the inlet velocity fluctuations (≈ 0.05) assuming isotropy. The velocity and pressure were updated using the PISO algorithm [42]. Unless otherwise stated, all the two-way coupled simulations are performed with the calculation/configuration described above.

Fig. 6 shows the profiles of the axial liquid velocities at five different cross-sections at $z = 0.13, 0.18, 0.25, 0.32$ m downstream of the sudden enlargement predicted by `twoPhaseEulerFoam` (TFM) and `twoWayGPBEFoam` (Eulerian QBMM). It can be seen that the liquid velocity is quickly diffused and, further downstream at $z = 0.32$ m, is almost uniform over the pipe radius. This is because of the increased turbulence viscosity along the streamline. Both the results predicted by `twoPhaseEulerFoam` and `twoWayGPBEFoam` agree well with the experimental data. The predicted axial liquid velocity at the lower cross-section near the pipe center is slightly underestimated. This underestimation can be improved by adjusting the mean inlet liquid velocity and the inlet phase fraction to a given "wall peak" value which is more consistent with the experimental data. The predicted profiles of the axial bubble velocities and the phase fractions are shown in Fig. 7. It can be seen that the bubbles are accelerated by the buoyancy along the pipe's axial direction, and the large bubbles move faster than the small bubbles due to the size-dependent drag coefficients. This phenomenon can only be predicted by the size-conditional velocities implemented in `twoWayGPBEFoam`. However, since all the bubbles are theoretically assumed to be equal in size, they move with the same velocity as predicted by the TFM solver `twoPhaseEulerFoam`. The contour plots and the profile plot at $z = 0.18$m of the turbulent kinetic energy and the turbulent viscosity predicted by `twoPhaseEulerFoam` and `twoWayGPBEFoam` are reported in Fig. 8 and 9. It is interesting to note that most of the turbulent diffusion is produced far from the enlargement, and the results predicted by `twoWayGPBEFoam` is consistent with `twoPhaseEulerFoam`, which

confirms that the QBMM is successfully coupled with the N-S equations of the continuous phase.



(a) $z = 13$ cm (b) $z = 18$ cm



(c) $z = 25$ cm (d) $z = 32$ cm

Figure 6: Comparison of the mean axial liquid velocity predicted by the TFM (red line) and the Eulerian QBMM (black line) with experimental data (circles) at different positions.

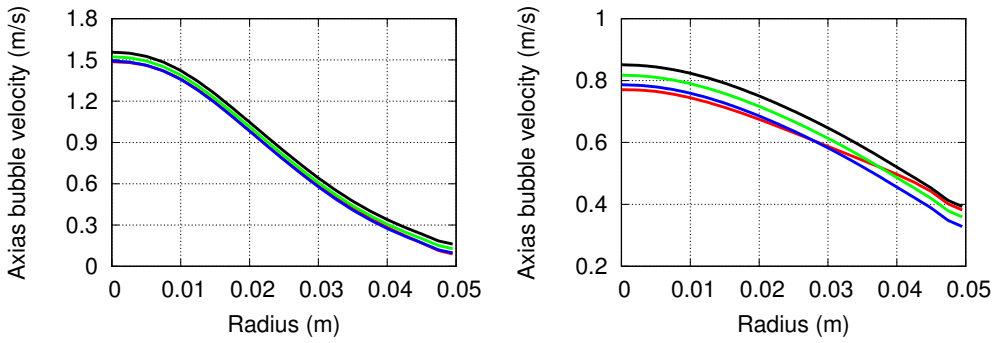


Figure 7: Comparison of the bubble velocities predicted by the TFM (red line) and the Eulerian QBMM (blue line for smallest bubbles, green line for medium bubbles, black line for largest bubbles) at $z = 0.18$ m (left) and $z = 0.32$ m (right).

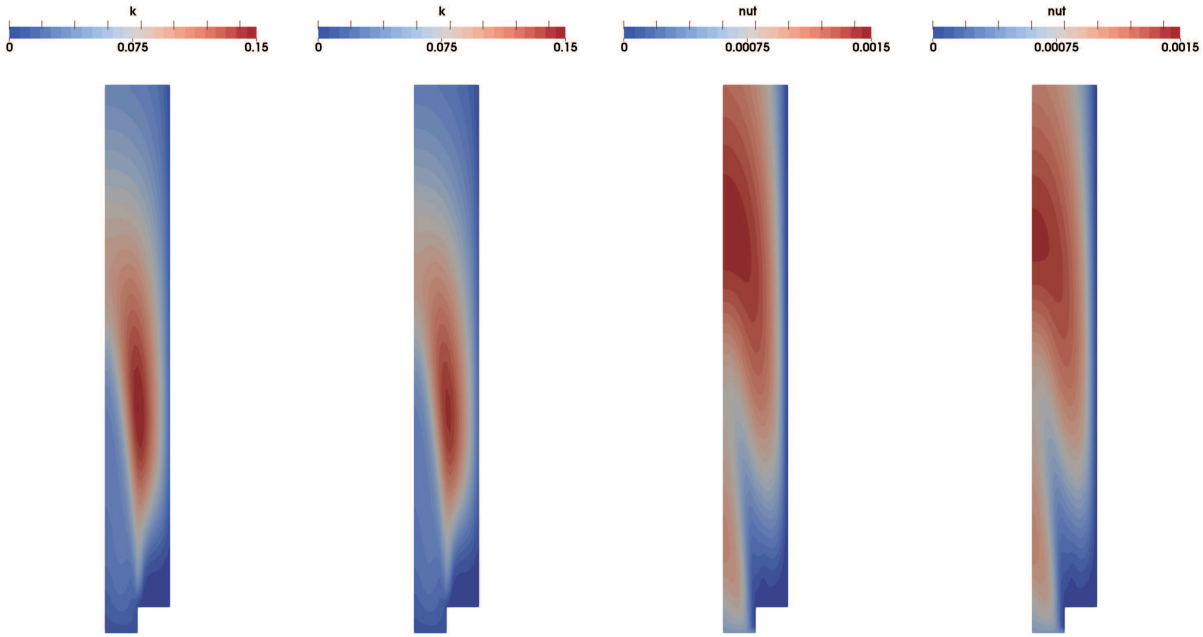


Figure 8: The contour plots of the turbulent kinetic energy and the turbulent viscosity predicted by the Eulerian QBMM (left) and the TFM (right), respectively.

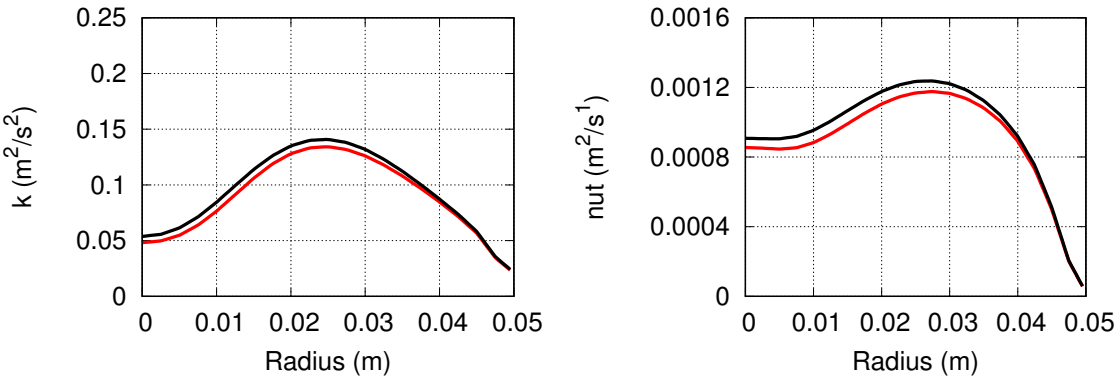


Figure 9: Comparison of the turbulent kinetic energy (left) and the turbulent viscosity (right) predicted by the Eulerian QBMM (black line) and the TFM (red line).

The plots of the predicted phase fraction produced by both solvers in Fig. 10 are not in good agreement with the experimental data. One reason is that the inlet phase fraction was given as a uniform value ($\alpha_{\mathrm{d}} = 0.05$) instead of the "wall peak" profile visualized in the experiments, which is not available. Another possible reason is that the wall peak

phenomenon is a coupled effect of shear, the wake phenomenon and deformation of the lift force and the turbulence of the liquid phase. These necessary momentum interface exchange terms (such as the lift force and the turbulent dispersion force) need to be implemented and investigated in the future to predict the phase fraction accurately. It was also shown in other works that the predicted phase fraction is highly dependent on the turbulence model [50]. As this work focuses on the coupling procedure of the Eulerian QBMM, we show that such complex phenomenon cannot be predicted by only including the drag force.

It is also interesting to compare the computational cost between the `twoPhaseEulerFoam` and `twoWayGPBEFoam`. We run the sudden enlargement test case on a i7-5820k CPU in serial. `twoPhaseEulerFoam` and `twoWayGPBEFoam` took 401s and 1452s, respectively. It can be seen that the E-QBMM requires more computational resources than the TFM. It can be explained by that the higher-order Vikas scheme needs to evaluate the fluxes for all moments by explicit interpolation at the cell interface. For example, as is seen in Eq. (7), overall 15 moments transport equations need to be updated, which implies that $15 \times 2 \times 3 \times 3 = 270$ moments flux $m_{j,k,l,i}$ need to be evaluated at the cell interface. Here, the second 2 denotes the upwind and downwind flux, the third 3 denotes three dimensions and the last 3 denotes the moments flux corresponding to each diameter if 3 nodes are employed. Moreover, we compared the computational efficiency between the preconditioned conjugate gradient (PCG) solver and the geometric algebraic multi-grid (GAMG) for the pressure Poisson equation. In each time step, around 30 times and 7 iterations are typically needed for these two methods. Thus, the simulation time difference between these different pressure Poisson solvers is negligible.

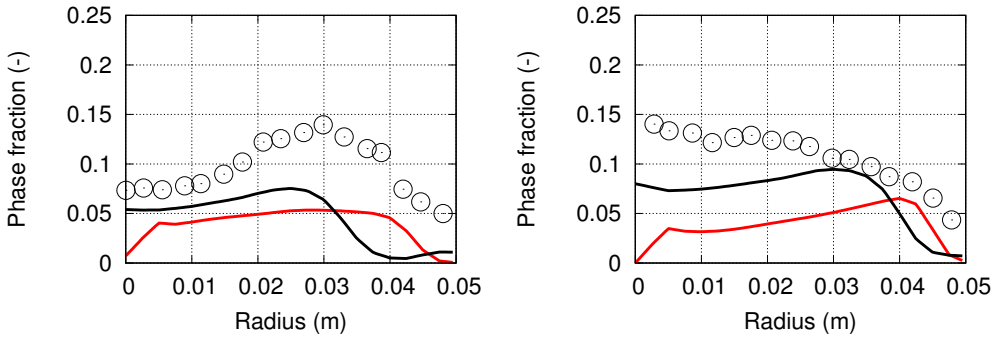


Figure 10: Comparison of the phase fraction predicted by the TFM (red line) and the Eulerian QBMM (black line) with the experimental data (points) at $z = 0.18$ m (left) and $z = 0.32$ m (right).

4.3. Two-way coupled test cases: bubble plume

In this section, let us investigate another important gas-liquid mixing phenomenon: the periodically oscillating bubble plume. It is often used for industrial multiphase mixing [54]. Moreover, it is a suitable test case to verify the two-way coupling implementation in `twoWayGPBEFoam`, since the liquid velocity is stagnant initially, and liquid vortices are generated by the coupling between the phases. Last but not least, it was proven that the periodically oscillating bubble plumes can be predicted simply by adding the drag forces [55, 56] without breakage and coalescence [57].

The first bubble column experiments were carried out by Díaz et al. [55]. The experimental setup included a 0.2-m-wide, 1.8-m-high, and 0.04-m-deep rectangular bubble column, which was filled with water (H=0.45 m) at room temperature and atmospheric pressure, while air was fed at different velocities from the gas chamber through a sparger (eight centered holes of 0.001 m in diameter and 0.006 m in pitch). A fully orthogonal hexahedral grid measuring 40 (height) × 17 (width) × 7 (depth) cells was employed, as we confirmed that the non-uniform hexahedral grids measuring 17 cells in width, 40 cells in height and 7 cells in depth are enough to capture the plume oscillations in our previous investigation [57]. The area of the sparger through which gas enters the domain was modeled as a rectangle with an area equal to 0.000108 m^2. This simplification has been proven to be efficient for meshing [55, 4, 58]. The mean Sauter diameter of the particles entering the computational

domain was set to 0.00505 m for the three-dimensional simulations, which is the same as in the literature [55]. The phase fraction was set to be 0.5, and the inlet bubble velocity calculated by the given superficial velocity (0.0024 m/s and 0.0071 m/s). The corresponding moments were given in Table. 1. Breakage and coalescence are not considered. Other necessary aspects of the setup are essentially the same as outlined in the sudden enlargement test case in the previous section and are not repeated here for simplicity.

The instantaneous phase fraction, continuous phase velocity vectors and streamlines are reported in Fig. 11. It can be seen that the oscillating bubble plumes were successfully captured and the vortices were generated as a result of the two-way coupling. The predicted horizontal liquid velocities at two different points and different superficial velocities are reported in Figs. 12 and 13. All the horizontal liquid velocities predicted by the Eulerian QBMM show periodic behavior. Not only the period but also the peak horizontal liquid velocity predicted by `twoWayGPBEFoam` agree well with the experimental data [55] and the simulated results predicted by the TFM [55, 57]. The global hold-up and the plume oscillation periods (POPs) predicted by `twoWayGPBEFoam` are reported in Table 2. All of them agree well with the experimental data. Moreover, it can be seen in Figs. 12 and 13 that the predicted vertical velocities for small ($d \approx 4$ mm), medium ($d \approx 5.2$ mm) and large bubbles ($d \approx 6.7$ mm) are different, and can be only predicted by the size-conditional velocity algorithm implemented in `twoWayGPBEFoam`, since different drag forces are exerted on the bubbles of different sizes.

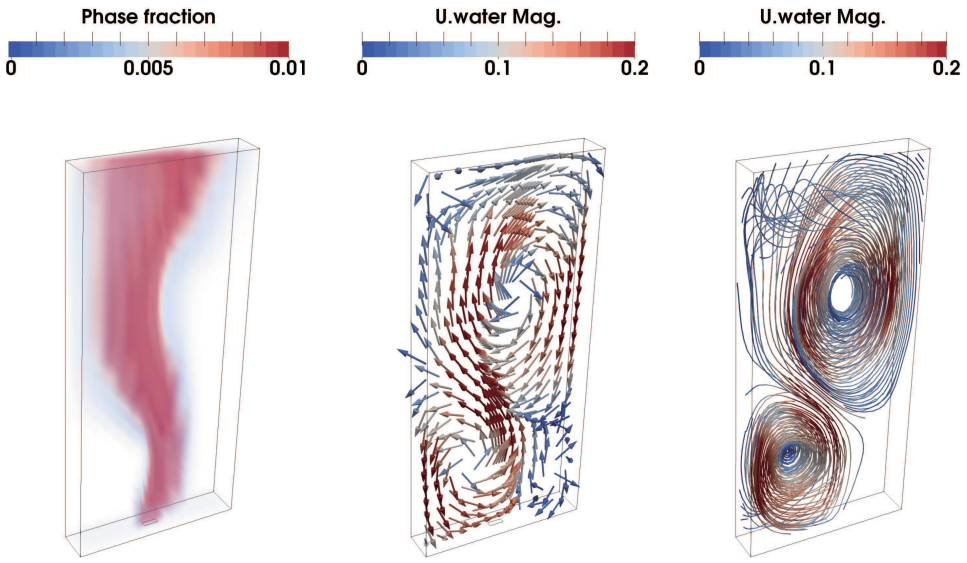


Figure 11: Contour plots from left to right of the instantaneous phase fraction, liquid velocity vectors and liquid velocity streamlines predicted by `twoWayGPBEFoam`.

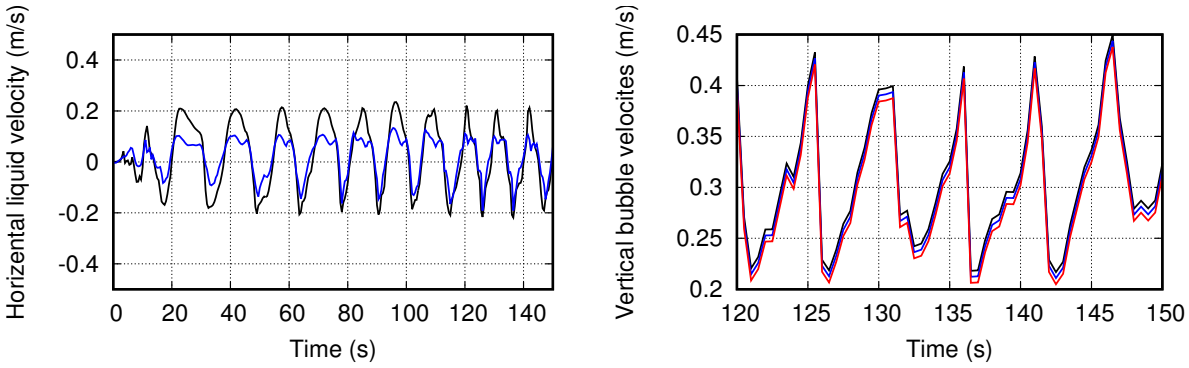


Figure 12: Left: The prediction of the horizontal liquid velocity by `twoWayGPBEFoam` at two different points (superficial velocity is 0.0024 m/s). Black line: $x = 0.1$ m, $y = 0.225$ m, $z = 0.02$ m. Blue line: $x = 0.05$ m, $y = 0.225$ m, $z = 0.02$ m. Right: the prediction of the vertical buble velocities by `twoWayGPBEFoam` for small bubbles (black line), medium bubbles (blue line) and large bubbles (red line) at $x = 0.1$ m, $y = 0.225$ m, $z = 0.02$ m.

$\mathbf{U}_g$	Sim. hold-up	Exp. hold-up	Sim. POP	Exp. POP
0.0024 m/s	0.006	0.0064	12.9	11.38
0.0071 m/s	0.012	0.018	6.31	5.69

Table 2: Comparison of the predicted global hold-up and POPs with the experimental data at different superficial velocities.

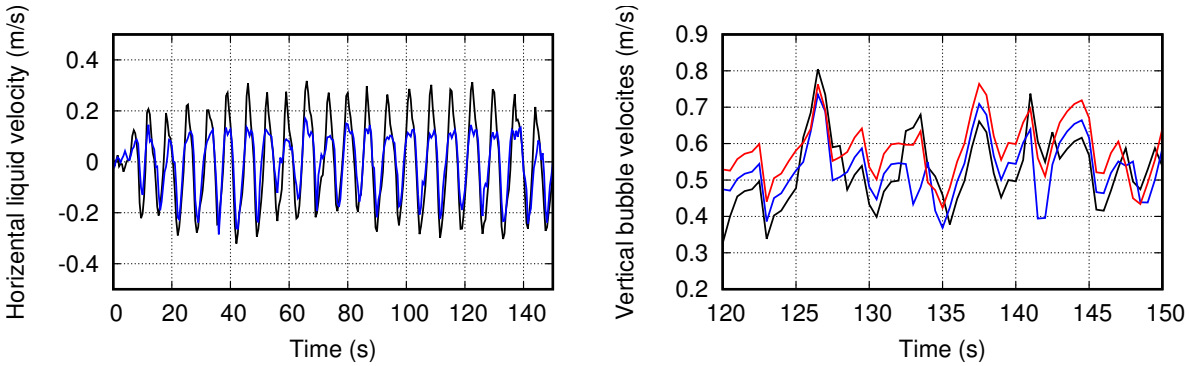


Figure 13: Left: The prediction of the horizontal liquid velocity by `twoWayGPBEFoam` at two different points (superficial velocity is 0.0071 m/s). Black line: $x = 0.1$ m, $y = 0.225$ m, $z = 0.02$ m. Blue line: $x = 0.05$ m, $y = 0.225$ m, $z = 0.02$ m. Right: the prediction of the vertical bubble velocities by `twoWayGPBEFoam` for small bubbles (black line), medium bubbles (blue line) and large bubbles (red line) at $x = 0.1$ m, $y = 0.225$ m, $z = 0.02$ m.

To further validate the two-way coupling algorithm, the experiments carried out by Pfleger et al. [59] were considered, to compare the results. This test case has been proven suitable to investigate bubbly flows and used in a number of studies to validate numerical models [60, 61, 62]. Moreover, the vertical liquid velocity was monitored using Laser Doppler Anemometry (LDA) and these results can be used for comparison with the simulations. The calculations were performed with an air flow rate of 48 l/h, and the mean Sauter diameter is 2 mm. All other numerical settings are identical to the previous test case. The predicted averaged vertical liquid velocities are reported in Fig. 14. It can be observed that the resulting time-averaged flow pattern consists in a non-uniform velocity distribution presenting an upward flow in the column center and a downward flow along the column walls: a liquid circulation mode, commonly referred to as the "Gulf-stream" or "Cooling tower" flow regime [63, 55]. This type of flow differs considerably from the instantaneous flow pattern previously described. It is therefore essential to capture the dynamic nature of the flow to accurately

describe the hydrodynamics as well as mixing and transfer processes in bubble columns.

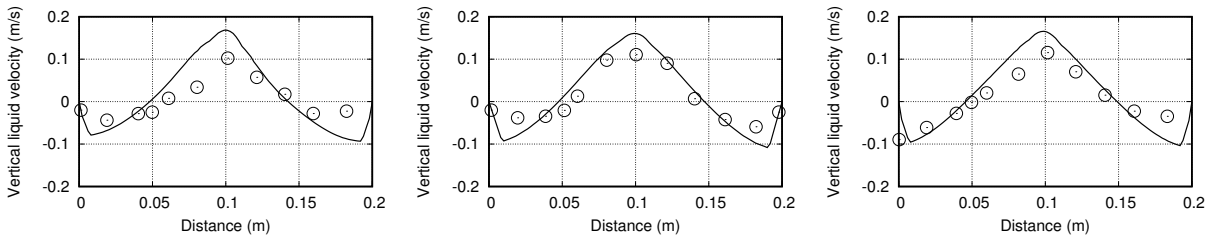


Figure 14: Comparison of the averaged vertical liquid velocity predicted by `twoWayGPBEFoam` (black line) with the experimental data (circle point) at different cross-sections in the work of Pfleger et al. [59]. From left to right: $y = 13$ cm, $y = 25$ cm, $y = 37$ cm.

5. Conclusion

In this work, a two-way coupled solver `twoWayGPBEFoam` has been developed, together with a one-way coupled testing solver oneWayGPBEFoam, in which the Eulerian QBMM was implemented. The latter is designed entirely within the OpenFOAM software framework and is a highly extensible and fully object-oriented C++ based approach. The code is released under the same license as the OpenFOAM base, at a publicly available software repository that includes documentation and example tutorials. It is intended to be a research platform for those interested in the Eulerian QBMM.

`oneWayGPBEFoam` has been used in this work to verify the particle size segregation phenomenon, and good agreement with the analytical solutions was obtained, implying that the QBMM algorithm which was implemented is correct. A turbulent bubbly air/water upward flow through a pipe with a sudden enlargement was employed to verify the results predicted by the two-way coupled Eulerian QBMM and the TFM. Both of them predict good results compared with the experimental data. To complete the two-way verifications, two classic bubble column experiments with different superficial velocities were employed, as used by Díaz et al. [55] and Pfleger et al. [59]. It was shown that the results predicted by `twoWayGPBEFoam` (e.g., the bubble plume period, the global phase fraction and the averaged vertical liquid velocity) agree well with the experimental data.

The development of `twoWayGPBEFoam` is still ongoing. Other momentum interface exchange models was also implemented and investigated in other works [64]. Optimisations and bug fixes are often applied to the repository. It is also worth mentioning that, besides `twoWayGPBEFoam`, readers are also referred to the open-source code (`OpenQBMM`) developed by Passalacqua et al. [46], which includes different QBMM algorithms and well-organized test cases, although the two-way coupling procedure used in their work is different to ours. It is our hope that these open-source QBMM-based CFD codes, as strong supplements of the existing TFM and E-L method, will be used and be extended by different research and industrial groups from different disciplines.

Acknowledgements

This work was partially supported by Helmholtz-Zentrum Dresden-Rossendorf. Dongyue Li would like to acknowledge the small, but active, CFD community in China for all their continued encouragement. He had a baby last week (2019-08-14)!

Appendix. Particle size segregation problem by continuous size distribution

In the previous test case, three-node QMOM was used to approximate the size NDF. Considering a univariate NDF, it can be represented as

$$n(d) = \sum_{\beta=1}^{N} w_\beta \delta(d - d_\beta). \tag{41}$$

If no other source terms (e.g., breakage or coalescence) were considered, the particle sizes were fixed as three distinct values and the particles stopped at three different locations. Even this is a rigid assumption due to the QMOM algorithm, we further proved that using a three-node QMOM is enough for our bubbly flow test cases and the computing procedure is relatively fast. In order to remove this rigid assumption, the extended QMOM (EQMOM) can be used to cover the full spectrum of particle sizes. In the EQMOM, the NDF is continuous and is

spread parameter	m_0	m_1	m_2
0.02	1	0.5	0.0251
0.1	1	0.5	0.02525

Table 3: Spread parameter and moments for the test cases.

more consistent with the VPA method investigated in this work. Specifically, we employ log-normal distribution as the EQMOM kernel function since the particle size is strictly larger than 0. Other EQMOM kernel functions may introduce problems (e.g., the Gaussian kernel function will predict negative particle sizes). In log-normal EQMOM, the univariate NDF can be represented as

$$n(d) = \sum_{\beta=1}^{N} w_\beta \delta_\sigma(d, d_\beta) = \sum_{\beta=1}^{N} w_\beta \left(\sum_{\theta=1}^{N_\theta} w_\theta \delta(d - d_{\beta\theta}) \right) \tag{42}$$

where $\delta_\sigma(d, d_\beta)$ is the standard log-normal distribution, σ is the spread parameter, which controls the shape of the NDF. The smaller σ is, the narrower the shape of the NDF is. w_θ is the secondary weights and $d_{\beta\theta}$ is the secondary abscissas. In the field of kinetic equation, the primary nodes can be seen as large-scale velocities and the secondary nodes can be seen as small-scale velocity dispersions [33, 65]. In this work, the primary nodes and secondary nodes can be seen as main particle size and particle size dispersions. The dispersion nodes can be used to cover the possible particle sizes. The more secondary nodes were employed, the moments calculation is more accurate. Readers are referred to other works for more information of the EQMOM [66, 1, 17].

A similar one-way coupled test case as reported in section 4.1 was investigated. Physically, the small particles trails behind the leading large particles. The initial moments were calculated from Eq. (38) with $\sqrt{v_{\log}} = 0.02$ and $\sqrt{v_{\log}} = 0.1$, which corresponds to a small spread parameter (narrow NDF shape) and a large spread parameter (wide NDF shape). One primary node and four secondary nodes were employed since it predicts identical results when more secondary nodes were employed. The calculated kth moments are reported in Table 3. Since the EQMOM covers the continuous NDF, analytical solution can be hardly

obtained. Fig. 15 shows the comparison of the predicted m_0 for small spread parameter and large spread parameter test cases. The solution calculated by the QMOM was not shown for brevity as it was investigated in section 4.1. It can be seen that the predicted m_0 is more spreading for the large spread parameter test case. It was consistent with the theory since a large spread parameter corresponds to a wide NDF. Last but not least, the shape of the predicted m_0 is continuous instead of three blocks predicted by the QMOM as was shown in Fig .3.

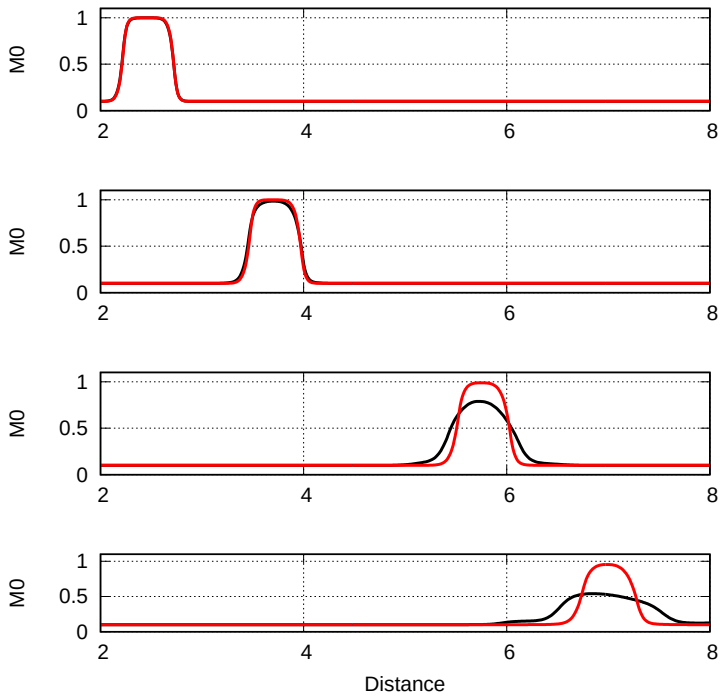


Figure 15: Comparison of the location of m_0 predicted by the EQMOM for small spread parameter (red line) and large spread parameter (black line).

References

References

[1] D. Marchisio, R. Fox, Computational models for polydisperse particulate and multiphase systems, Cambridge University Press, 2013.

[2] D. Drew, Mathematical modeling of two–phase flow, Annual Review of Fluid Mechanics 15 (1982) 261–291.

[3] Y. Tsuji, T. Kawaguchi, T. Tanaka, Discrete particle simulation of two-dimensional fluidized bed, Powder technology 77 (1993) 79–87.

[4] V. Buwa, D. Deo, V. Ranade, Eulerian–Lagrangian simulations of unsteady gas–liquid flows in bubble columns, International Journal of Multiphase Flow 32 (2006) 864–885.

[5] D. Lucas, A. Tomiyama, On the role of the lateral lift force in poly-dispersed bubbly flows, International Journal of Multiphase Flow 37 (2011) 1178–1190.

[6] Y. Liao, D. Lucas, E. Krepper, Application of new closure models for bubble coalescence and breakup to steam–water vertical pipe flow, Nuclear Engineering and Design 279 (2014) 126–136.

[7] A. Vaidheeswaran, W. Fullmer, M. Lopez De Bertodano, Effect of collision force on well-posedness and stability of the two-fluid model for vertical bubbly flows, Nuclear Science and Engineering 184 (2016) 353–362.

[8] R. Patel, O. Desjardins, B. Kong, J. Capecelatro, R. Fox, Verification of Eulerian–Eulerian and Eulerian–Lagrangian simulations for turbulent fluid–particle flows, AIChE Journal 63 (2017) 5396–5412.

[9] M. Banaei, N. Deen, M. van Sint Annaland, J. Kuipers, Particle mixing rates using the two-fluid model, Particuology 36 (2018) 13–26.

[10] A. Vaidheeswaran, M. Lopez De Bertodano, Stability and convergence of computational Eulerian two-fluid model for a bubble plume, Chemical Engineering Science 160 (2017) 210–226.

[11] H. Struchtrup, Macroscopic transport equations for rarefied gas flows, in: Macroscopic Transport Equations for Rarefied Gas Flows, Springer, 2005, pp. 145–160.

[12] A. Kaufmann, M. Moreau, O. Simonin, J. Helie, Comparison between lagrangian and mesoscopic Eulerian modelling approaches for inertial particles suspended in decaying isotropic turbulence, Journal of Computational Physics 227 (2008) 6448–6472.

[13] D. Ramkrishna, Population balances: Theory and applications to particulate systems in engineering, Academic Press, 2000.

[14] A. Buffo, M. Vanni, D. Marchisio, On the implementation of moment transport equations in OpenFOAM: Boundedness and realizability, International Journal of Multiphase Flow 85 (2016) 223–235.

[15] D. Li, A. Buffo, W. Podgórska, D. Marchisio, Z. Gao, Investigation of droplet breakup in liquid-liquid dispersions by CFD-PBM simulations: The influence of the surfactant type, Chinese Journal of Chemical Engineering 25 (2017) 1369–1380.

[16] Z. Gao, D. Li, A. Buffo, W. Podgórska, D. Marchisio, Simulation of droplet breakage in turbulent liquid–liquid dispersions with CFD-PBM: Comparison of breakage kernels, Chemical Engineering Science 142 (2016) 277–288.

[17] D. Li, A. Buffo, W. Podgórska, Z. Gao, D. Marchisio, Droplet breakage and coalescence in liquid-liquid dispersions: comparison of different kernels with EQMOM and QMOM, AIChE Journal 63 (2017) 2293–2311.

[18] A. Buffo, M. Vanni, D. Marchisio, Simulation of a reacting gas–liquid bubbly flow with CFD and PBM: Validation with experiments, Applied Mathematical Modelling 44 (2017) 43–60.

[19] C. Vik, J. Solsvik, M. Hillestad, H. Jakobsen, A multifluid-PBE model for simulation of mass transfer limited processes operated in bubble columns, Computers & Chemical Engineering 110 (2018) 115–139.

[20] L. Xie, Q. Liu, Z. Luo, A multiscale CFD-PBM coupled model for the kinetics and liquid–liquid dispersion behavior in a suspension polymerization stirred tank, Chemical Engineering Research and Design 130 (2018) 1–17.

[21] D. Li, Z. Li, Z. Gao, Compressibility induced bubble size variation in bubble column reactors: Simulations by the CFD–PBE, Chinese Journal of Chemical Engineering 26 (2018) 2009–2013.

[22] D. Li, Z. Li, Z. Gao, Quadrature-based moment methods for the population balance equation: An algorithm review, Chinese Journal of Chemical Engineering 27 (2019) 483–500.

[23] E. Krepper, D. Lucas, T. Frank, H. Prasser, P. Zwart, The inhomogeneous MUSIG model for the simulation of polydispersed flows, Nuclear Engineering and Design 238 (2008) 1690–1702.

[24] D. Lucas, T. Frank, C. Lifante, P. Zwart, A. Burns, Extension of the inhomogeneous MUSIG model for bubble condensation, Nuclear Engineering and Design 241 (2011) 4359–4367.

[25] E. Krepper, M. Beyer, D. Lucas, M. Schmidtke, A population balance approach considering heat and mass transfer–Experiments and CFD simulations, Nuclear Engineering and Design 241 (2011) 2889–2897.

[26] A. Vié, F. Laurent, M. Massot, Size-velocity correlations in hybrid high order moment/multi-fluid methods for polydisperse evaporating sprays: Modeling and numerical issues, Journal of Computational Physics 237 (2013) 177–210.

[27] P. Dems, J. Carneiro, W. Polifke, Large eddy simulation of particle-laden swirling flow with a presumed function method of moments, Progress in Computational Fluid Dynamics 12 (2012) 92–102.

[28] J. Carneiro, P. Dems, V. Kaufmann, W. Polifke, Eulerian simulations of polydisperse flows using a moments model with a relaxation approach for the moment transport velocities, in: 7th Int. Conf. on Multiphase Flow, ICMF 2010, 2010.

[29] D. Marchisio, R. Fox, Solution of population balance equations using the direct quadrature method of moments, Journal of Aerosol Science 36 (2005) 43–73.

[30] R. Fox, F. Laurent, M. Massot, Numerical simulation of spray coalescence in an Eulerian framework: direct quadrature method of moments and multi-fluid method, Journal of Computational Physics 227 (2008) 3058–3088.

[31] L. Mazzei, D. Marchisio, P. Lettieri, New quadrature-based moment method for the mixing of inert polydisperse fluidized powders in commercial CFD codes, AIChE Journal 58 (2012) 3054–3069.

[32] C. Yuan, R. Fox, Conditional quadrature method of moments for kinetic equations, Journal of Computational Physics 230 (2011) 8216–8246.

[33] C. Chalons, R. Fox, M. Massot, A multi-Gaussian quadrature method of moments for gas-particle flows in a LES framework, in: Studying Turbulence Using Numerical Simulation Databases, Center for Turbulence Research, Summer Program 2010, Stanford University, 2010.

[34] A. Passalacqua, R. Fox, R. Garg, S. Subramaniam, A fully coupled quadrature-based moment method for dilute to moderately dilute fluid–particle flows, Chemical Engineering Science 65 (2010) 2267–2283.

[35] B. Kong, R. Fox, A solution algorithm for fluid–particle flows across all flow regimes, Journal of Computational Physics 344 (2017) 575–594.

[36] C. Yuan, B. Kong, A. Passalacqua, R. Fox, An extended quadrature-based mass-velocity

moment model for polydisperse bubbly flows, The Canadian Journal of Chemical Engineering 92 (2014) 2053–2066.

[37] M. Lopez De Bertodano, W. Fullmer, A. Clausse, V. Ransom, Two-Fluid Model Stability, Simulation and Chaos, Springer, 2016.

[38] M. Andrews, P. O'rourke, The multiphase particle-in-cell MP-PIC method for dense particulate flows, International Journal of Multiphase Flow 22 (1996) 379–402.

[39] R. McGraw, Description of aerosol dynamics by the quadrature method of moments, Aerosol Science and Technology 27 (1997) 255–265.

[40] J. Wheeler, Modified moments and gaussian quadratures, The Rocky Mountain Journal of Mathematics 4 (1974) 287–296.

[41] V. Vikas, Z. Wang, A. Passalacqua, R. Fox, Realizable high-order finite-volume schemes for quadrature-based moment methods, Journal of Computational Physics 230 (2011) 5328–5352.

[42] R. Issa, A. Gosman, A. Watkins, The computation of compressible and incompressible recirculating flows by a non-iterative implicit scheme, Journal of Computational Physics 62 (1986) 66–82.

[43] S. Longshaw, M. Borg, S. Ramisetti, J. Zhang, D. Lockerby, D. Emerson, J. Reese, mdfoam+: Advanced molecular dynamics in OpenFOAM, Computer Physics Communications 224 (2018) 1–21.

[44] L. Zhu, S. Chen, Z. Guo, dugksfoam: An open source OpenFOAM solver for the Boltzmann model equation, Computer Physics Communications 213 (2017) 155–164.

[45] M. Karimi, H. Droghetti, D. Marchisio, PUFoam: A novel open-source CFD solver for the simulation of polyurethane foams, Computer Physics Communications 217 (2017) 138–148.

[46] A. Passalacqua, F. Laurent, E. Madadi-Kandjani, J. Heylmun, R. Fox, An open-source quadrature-based population balance solver for OpenFOAM, Chemical Engineering Science 176 (2018) 306–318.

[47] R. Bel F'dhila, Analyse expérimentale et modélisation d'un écoulement vertical à bulles dans un élargissement brusque, Ph.D. thesis, Toulouse, INPT (1991).

[48] H. Rusche, Computational fluid dynamics of dispersed two–phase flows at high phase fractions, Ph.D. thesis, Imperial College London (2003).

[49] P. Oliveira, R. Issa, Numerical aspects of an algorithm for the Eulerian simulation of two-phase flows, International Journal for Numerical Methods in Fluids 43 (2003) 1177–1198.

[50] A. Behzadi, R. Issa, H. Rusche, Modelling of dispersed bubble and droplet flow at high phase fractions, Chemical Engineering Science 59 (2004) 759–770.

[51] M. Ullrich, Second-moment closure modeling of turbulent bubbly flows within the two-fluid model framework, Ph.D. thesis, Technische Universität (2017).

[52] D. Cokljat, M. Slack, S. Vasquez, A. Bakker, G. Montante, Reynolds-stress model for eulerian multiphase, Progress in Computational Fluid Dynamics, An International Journal 6 (2006) 168–178.

[53] L. Schiller, A. Naumann, A drag coefficient correlation, VDI Zeitung 77 (1935) 51–86.

[54] A. Sokolichin, G. Eigenberger, A. Lapin, A. Lübert, Dynamic numerical simulation of gas-liquid two-phase flows Euler/Euler versus Euler/Lagrange, Chemical Engineering Science 52 (1997) 611–626.

[55] M. Díaz, A. Iranzo, D. Cuadra, R. Barbero, F. Montes, M. Galán, Numerical simulation of the gas–liquid flow in a laboratory scale bubble column: influence of bubble size distribution and non-drag forces, Chemical Engineering Journal 139 (2008) 363–379.

[56] A. Buffo, D. Marchisio, M. Vanni, P. Renze, Simulation of polydisperse multiphase systems using population balances and example application to bubbly flows, Chemical Engineering Research and Design 91 (2013) 1859–1875.

[57] D. Li, H. Christian, Simulation of bubbly flows with special numerical treatments of the semi-conservative and fully conservative two-fluid model, Chemical Engineering Science 174 (2017) 25–39.

[58] A. Buffo, M. Vanni, D. Marchisio, R. Fox, Multivariate quadrature-based moments methods for turbulent polydisperse gas–liquid systems, International Journal of Multiphase Flow 50 (2013) 41–57.

[59] D. Pfleger, S. Gomes, N. Gilbert, H. Wagner, Hydrodynamic simulations of laboratory scale bubble columns fundamental studies of the Eulerian–Eulerian modelling approach, Chemical Engineering Science 54 (1999) 5091–5099.

[60] R. Bannari, F. Kerdouss, B. Selma, A. Bannari, P. Proulx, Three-dimensional mathematical modeling of dispersed two-phase flow using class method of population balance in bubble columns, Computers and Chemical Engineering 32 (2008) 3224–3237.

[61] T. Ma, T. Ziegenhein, D. Lucas, J. Fröhlich, Large eddy simulations of the gas–liquid flow in a rectangular bubble column, Nuclear Engineering and Design 299 (2016) 146–153.

[62] E. Askari, P. Proulx, A. Passalacqua, Modelling of bubbly flow using CFD-PBM solver in OpenFOAM: Study of local population balance models and extended quadrature method of moments applications, ChemEngineering 2 (2018) 8.

[63] J. Chen, M. Jamialahmadi, S. Li, Effect of liquid depth on circulation in bubble columns: a visual study, Chemical Engineering Research & Design 67 (1989) 203–207.

[64] D. Li, D. Marchisio, C. Hasse, D. Lucas, Comparison of Eulerian Qbmm and classical Eulerian-Eulerian method for the simulation of poly-disperse bubbly flows, accepted by AIChE Journal.

[65] F. Laurent, A. Vie, C. Chalons, R. Fox, M. Massot, A hierarchy of eulerian models for trajectory crossing in particle-laden turbulent flows over a wide range of stokes numbers, Center for Turbulence Research Annual Research Briefs 2012 (2013) 193.

[66] C. Yuan, F. Laurent, R. Fox, An extended quadrature method of moments for population balance equations, Journal of Aerosol Science 51 (2012) 1–23.