Chemical Engineering Science 101 (2013) 565–576

Contents lists available at ScienceDirect

Chemical Engineering Science

journal homepage: www.elsevier.com/locate/ces

Segregation dynamics of dense polydisperse fluidized suspensions modeled using a novel formulation of the direct quadrature method of moments

Luca Mazzei *

Department of Chemical Engineering, University College London, Torrington Place, London WC1E 7JE, UK

AUTHOR-HIGHLIGHTS

- We test a novel version of DQMOM for dense multiphase flows.
- We estimate the order or magnitude of the diffusivity used in the model.
- We conduct a sensitivity analysis on the diffusivity.
- We model the segregation dynamics of polydisperse fluidized powders using the model.

ARTICLE INFO

Article history:
Received 7 December 2012
Received in revised form
20 May 2013
Accepted 5 July 2013
Available online 12 July 2013

Keywords:
Multiphase flows
Fluidization
Population balance
QMOM
DQMOM
Diffusion

ABSTRACT

Computational fluid dynamics (CFD) may be a useful design tool, provided that the mathematical models that we solve with it capture and describe well the most important features of the systems of interest. For fluidized beds, one of these features is the polydispersity of the powders: particles differ in size and alter their size distribution in time and space continuously. To model this key phenomenon, one needs to solve a population balance equation, that is, an equation that governs the evolution of the size distribution. The direct quadrature method of moments (DQMOM) allows doing so in commercial CFD codes at relatively low computational cost. This technique, successfully employed for describing dilute multiphase flows of particles that share the same velocity, still needs testing in the context of dense multiphase flows. Dense polydisperse fluidized powders can segregate or mix, depending on the process operating conditions, and to describe these phenomena one needs to let particles move with different velocities. In this work we use a recent version of DQMOM that has this feature: each quadrature class is advected with its own velocity. The transport equations of this model feature a diffusive-like contribution that allows the powders to mix at the particle length scale. We discuss how to assign a value to the diffusion coefficient and we carry out a sensitivity analysis on the latter; to do so, we simulate the mixing of powders initially segregated using different values for the diffusivity. Successively, after having estimated a suitable value for the latter, we simulate the system dynamics under conditions that should promote segregation, validating the results of the simulations experimentally.

1. Introduction

Polydisperse multiphase systems are composed of a continuous phase (a gas or a liquid) within which other discontinuous phases are dispersed (particles, droplets or bubbles); each discontinuous phase is composed of elements continuously distributed over velocity and size, and possibly other properties, such as density. Even if virtually every industrial plant contains units that treat these systems (*e.g.*, fluidized beds, bubble columns and crystallizers), designing them is still subject to great uncertainties. This is because such systems undergo numerous physical and chemical phenomena that occur concurrently: chemical reactions take place, usually implicating all the phases and affecting their properties; also, elements of the discontinuous phases can break into subelements or aggregate, while new elements may nucleate. The behavior of the units and the quality of the product strongly depend on these competing phenomena, in turn influenced by phase interactions, system fluid dynamics and, indirectly, unit geometry and size.

To describe the behavior of polydisperse multiphase systems and design process units for treating them, researchers and engineers have resorted for several years to experimental correlations and pilot plants. These correlations, however, have limited applicability as they are valid only for the specific cases investigated; so, they cannot help

* Tel.: +44 20 7679 4328; fax: +44 20 7383 2348.
E-mail addresses: l.mazzei@ucl.ac.uk, luca.mazzei.a@gmail.com

0009-2509/$ - see front matter © 2013 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.ces.2013.07.006

improve design and performance, or predict the effect of changing the size or geometry of a unit. Pilot plants, on the other hand, are costly and time-consuming and do not always lead to adequate scale up. In consequence, thanks to the high-speed computers and advanced numerical methods now available, the modeling and simulation of multiphase flows have rapidly gained importance. Due to the complexity of such flows, a relatively large number of modeling approaches have been developed in the literature (Fox, 2012). At the most fundamental level the particles are treated individually, so that the discrete structure of the dispersed material is entirely retained; here one models the behavior of each particle, accounting for its interaction with the surrounding fluid and other particles. This strategy is powerful but computationally extremely expensive. The information that these simulations provide is normally not of direct use to engineers and greatly exceeds their normal requirements. An alternative strategy is to model also the discontinuous phases as continua. Several Eulerian models of this kind have been developed (Drew and Passman, 1998); these, however, often present severe limitations.

One of the most important limitations, present also in many advanced models, is that these do not account for polydispersity, neglecting in particular that the discontinuous phases are made of elements with changing size distribution. They instead assume that the latter consist of classes of particles with equal and constant sizes. The constant-particle-size assumption significantly limits the model flexibility: classes may segregate or mix, and particles may change class, but variations in the diameters attached to each class are not allowed. Real systems are instead characterized by broad particle size distributions (PSDs) which evolve continuously owing to fluid–particle and particle–particle interactions. Particles can shrink, aggregate, break and nucleate; hence, their size distribution varies continuously in time and space. Predicting this evolution, which depends on the local conditions wherein a system operates, is key to accurately describing its behavior.

To account for polydispersity and be able to predict how PSDs evolve, one needs to solve a population balance equation (PBE), possibly along with customary multifluid balance equations for mass and/or linear momentum. Doing so is quite difficult, since the PBE dimensionality differs in general from that of classical fluid dynamic equations. In the last years numerous attempts to solve this equation, in particular within CFD codes, have been reported in the literature (in Fox, 2012 and Marchisio and Fox, 2007, for instance, one may find several references); nevertheless, *dense fluid–solid systems*, in which the phases strongly interact and do not share the same velocity field, have not been investigated extensively (Fan et al., 2004; Fan and Fox, 2008; Fox and Vedula, 2010; Mazzei, 2011; Mazzei et al., 2012), few works considering size-changing phenomena such as chemical reaction, aggregation or breakage.

Often engineers are only interested in few integral properties of the distribution function that describes the particle population. Such properties, called moments, may be important because they control the product quality or because they are easy to measure and monitor. The idea behind the method of moments is to derive transport equations for the moments of interest (Randolph and Larson, 1971). This method is attractive, for the number of equations to be solved is small; however, the moment transport equations are unclosed, because for any given set of moments that the modeler wishes to track, the equations normally involve higher-order moments external to the set (Marchisio and Fox, 2007). The quadrature method of moments (QMOM) and its direct version (DQMOM) overcome this issue by approximating the distribution function with a quadrature formula; assuming the functional form of the distribution allows to calculate, with a given approximation, the values of any higher-order moment external to the set tracked by the methods. QMOM tracks the moments of this set by integrating their evolution equations; then, once these moments are known, it calculates the nodes and weights of the quadrature formula. DQMOM, conversely, directly tracks the latter, solving the evolution equations that govern them. The models are theoretically equivalent, as we shall discuss later on, but present different issues when one solves them numerically (Shohat and Tamarkin, 1943; Akhiezer, 1965; Wright, 2007; Mazzei et al., 2010a; Petitti et al., 2010; Mazzei, 2011; Mazzei et al., 2012).

In most versions of the quadrature-based moment methods reported in the literature, the PBE solved does not feature convection in physical space; written for well-mixed systems, for which the distribution function is uniform in such a space, these models account solely for particle growth, their PBEs featuring convection just in size space (*e.g.*, Dorao and Jakobsen, 2006; Grosch et al., 2007; Aamir et al., 2009; Gimbun et al., 2009; Qamar et al., 2011). Some other models, written for nonuniform systems, account for convection in physical space, but often assume that all quadrature classes are advected with the same velocity field, so that particles share the same velocity (*e.g.*, Petitti et al., 2010). This assumption prevents solids from segregating. Dense polydisperse fluidized powders may segregate or mix, depending on the process operating conditions, and in order to describe these phenomena one needs to let particles move with different velocities.

Few models catering for nonuniform dense polydisperse fluid–solid systems have this feature, and hence have the capability to describe segregation. Among the first to be developed are those of Fan et al. (2004) and Fan and Fox (2008); these let each quadrature class be advected with its own velocity field, whose evolution is governed by a coarse-grained dynamical equation. These DQMOM models, as Mazzei et al. (2010a) reported, have a significant limitation: they do not permit powders to micromix, that is to say, to mix at the length scale of the particles. Solely macromixing, that is to say, convection-induced mixing, is possible in such models. We shall address this aspect in detail later.

To overcome this problem, Mazzei (2011) recently developed a revised version of DQMOM in which the evolution equations for the quadrature weighted nodes and weights feature a diffusive flux that compensates for the error that one makes when calculating the convective flux of a property adopting the quadrature-based approximation of the distribution function that describes the particle population. In Mazzei (2011), we did not specify which value to assign to the diffusion coefficient. This, indeed, is an open issue, which we shall address in the present work.

This work aims to simulate the segregation dynamics of inert dense polydisperse fluidized powders. The paper is organized as follows. We introduce the problem that we intend to investigate. Next, we describe the mathematical model, in particular the DQMOM model recently developed in Mazzei (2011). Since we shall need them subsequently, we also report the evolution equations of the QMOM model, showing that the two are theoretically equivalent. Their equations feature a diffusive term which allows the powders to mix at the particle length scale (micromixing). We discuss how to assign a value to the diffusion coefficient appearing in the evolution equations of the models, conducting a sensitivity analysis on the latter; to do so, we simulate the mixing of nonuniform powders using different values for the diffusivity. Finally, after having estimated a suitable value for this coefficient, we simulate the system dynamics under conditions which should promote segregation, validating the predictions of the numerical simulations experimentally.

2. Problem description

We aim to simulate the dynamics of inert dense polydisperse fluidized powders under conditions that should promote segregation.

By inert we mean that the particles do not react, grow, wear, shrink, break, aggregate or nucleate; in these conditions the local particle size distribution varies only because of mixing or segregation. A powder initially uniform, for instance, might segregate becoming nonuniform, whereas a powder initially nonuniform might homogenize becoming well-mixed.

Initially, at time $t=0$, the system of our interest is a packed bed constituted of two superposed layers of same height of polydisperse ballotini particles of equal density (2500 kg/m^3). The lower and upper powders, referred to as A and B, respectively, differ only in particle size distribution, the upper one having larger mean particle size. The minimum fluidization velocities u_A and u_B are equal to 0.01 m/s and 0.06 m/s, respectively; other experimental data are given in Mazzei et al. (2010a); in particular, the particle size distributions of the two powders, which we obtained experimentally by sieving. To fluidize the system, we fed fluid at a superficial velocity which is low enough to make the larger particles sink and the smaller particles rise; we considered two fluid velocities: 0.10 m/s and 0.05 m/s. The resulting powders, referred to as powders C and D, presented at pseudosteady-state conditions a PSD that changed continuously along the bed axis. To find the distributions, we cut off the fluid supply, letting the bed settle down, and we then divided the fixed bed into five layers, measuring the PSD of each layer by sieving. To save space, we do not report the experimental PSDs in a figure (for we would have to show ten distributions); however, we shall present and use experimental results later on in Section 7. Details about the equipment and the experimental procedure are also given in Mazzei et al. (2010a). Resorting to the direct quadrature method of moments, we intend to predict the pseudosteady-state particle size distributions and verify whether they agree with those found experimentally.

3. Multiphase population balance model

We propose to describe the evolution of the particle size distribution for an inert dense polydisperse fluidized powder. This distribution can be represented mathematically by a volume density function (VDF) that yields the volume of particles in any given differential size interval per unit volume of physical space. In particular, if $f(\boldsymbol{x},t;s)$ denotes the VDF, then $f(\boldsymbol{x},t;s)ds\,d\boldsymbol{x}$ yields the expected volume of particles present at time t in the physical volume $d\boldsymbol{x}$ about $\boldsymbol{x}$ with size in the range ds about s. If particle size varies neither continuously nor discontinuously, that is, if growth, aggregation, breakage and alike size-changing phenomena are absent, and if particles neither nucleate nor dissolve, the population balance equation, that is, the evolution equation of the VDF, reads as follows:

$$\partial_t f = -\partial_{\boldsymbol{x}}\cdot f\langle \boldsymbol{v}|s\rangle \tag{3.1}$$

where $\langle \boldsymbol{v}|s\rangle(\boldsymbol{x},t;s)$ denotes the size-conditioned particle velocity. Details about the derivation of the above equation are found in Mazzei (2011). As we see, in the conditions considered, convection in physical space is the sole cause that makes the VDF evolve.

About Eq. (3.1), we would like to point out two aspects, which are discussed in detail in Mazzei (2011). First, Eq. (3.1) accounts for the effects of particle collisions, its validity not being restricted to dilute flows in free transport regime; as discussed in Mazzei (2011), collisions affect the size-conditioned velocity field $\langle \boldsymbol{v}|s\rangle(\boldsymbol{x},t;s)$. Second, we should note that, because the advective term in Eq. (3.1) features a size-dependent velocity field, the equation presents no diffusive flux in physical space; this is because particles with different sizes are advected with different velocities. Spatial diffusion would arise if we replaced $\langle \boldsymbol{v}|s\rangle(\boldsymbol{x},t;s)$ with the mean velocity of the whole particle population, which would be averaged over s.

Quadrature-based moment methods do not solve Eq. (3.1) directly: they *assume* the functional form of the VDF, and then resort to Eq. (3.1) to determine how the quantities that this functional form leaves unspecified evolve. The form assumed for the density function is

$$f_\nu(\boldsymbol{x},t;s)\equiv \sum_{r=1}^{\nu} \phi_r(\boldsymbol{x},t)\delta[s-s_r(\boldsymbol{x},t)] \tag{3.2}$$

This quadrature formula involves 2ν functions: the weights $\phi_r(\boldsymbol{x},t)$ and the nodes $s_r(\boldsymbol{x},t)$. We do not assign them explicitly, but instead require that these conditions be met:

$$\mathcal{M}_a(\boldsymbol{x},t)\equiv \int_0^\infty s^a f(\boldsymbol{x},t;s)\,ds = \int_0^\infty s^a f_\nu(\boldsymbol{x},t;s)\,ds = \sum_{r=1}^{\nu} \phi_r(\boldsymbol{x},t)[s_r(\boldsymbol{x},t)]^a \quad \text{for} \quad 0\le a\le 2\nu-1 \tag{3.3}$$

These 2ν conditions allow to obtain the 2ν functions ϕ_r and s_r. The quantities $\mathcal{M}_a(\boldsymbol{x},t)$ are called moments of the VDF; they are integer moments, because a is integer. So, to obtain the functions ϕ_r and s_r, we require that the first 2ν integer moments of f_ν be identical to those of the VDF. Note that we do not have to preserve necessarily this set of moments; we could choose, for instance, other integer moments or even real moments of the distribution. But the first integer moments represent important physical quantities, such as the overall volume fraction of particles, the mean size of the latter and the variance of the distribution; also, preserving these moments turns f_ν into a Gaussian quadrature formula (Marchisio and Fox, 2007), which yields a higher mathematical accuracy when one uses it to calculate the approximate values of moments of the VDF external to the set considered in Eqs. (3.3).

Eqs. (3.3) allow to compute *the values* of the quadrature nodes and weights in any spatial point $\boldsymbol{x}$ and at any time t provided we know the values of the first 2ν integer moments of the VDF. At time $t=0$ we suppose that the VDF and therefore its moments are known; thus, we know the initial values of the quadrature nodes and weights in any spatial point $\boldsymbol{x}$. To obtain the values of the latter at any subsequent time, we need to know *the evolution equations* of the moments, or equivalently of the quadrature nodes and weights. QMOM deals with the moments of the density function, solving evolution equations for the latter and employing Eqs. (3.3) to determine the values of the quadrature nodes and weights. DQMOM, conversely, deals with the quadrature nodes and weights, solving evolution equations for them and employing Eqs. (3.3) to determine the values of the moments of the density function. Dealing with quadrature nodes and weights instead of moments is more convenient from a numerical standpoint, because calculating the latter from the former is much simpler than the other way around; this is particularly true for multivariate density functions, which here, nevertheless, we do not consider (Marchisio and Fox, 2007; Mazzei et al., 2012).

3.1. DQMOM evolution equations

To derive the DQMOM evolution equations, we might think to introduce Eq. (3.2) into Eq. (3.1); a few passages then yield the desired equations (Mazzei et al., 2010a). This simple procedure, nevertheless, leads to incorrect equations, insofar as, since the quadrature-based approximation of the VDF differs from the actual VDF, the evolution equation of the former differs from that of the latter. Following Mazzei (2011), we assume that the equation reported below:

$$\partial_t f_\nu = -\partial_{\boldsymbol{x}}\cdot f_\nu\langle \boldsymbol{v}|s\rangle + \partial_{\boldsymbol{x}}\cdot\mathcal{D}_x\partial_{\boldsymbol{x}} f_\nu \tag{3.4}$$

governs the quadrature-based approximation of the VDF, in which $\mathcal{D}_x(\boldsymbol{x})$ represents a diffusion coefficient. Details are given in the cited article; here we just point out that the diffusive flux introduced compensates for the error that we make when computing the convective flux of the real VDF, that is, $\partial_{\boldsymbol{x}}\cdot f\langle \boldsymbol{v}|s\rangle$, by adopting its quadrature-based approximation.

To derive the DQMOM evolution equations, we now introduce Eq. (3.2) in the revised population balance equation reported above; this, after few manipulations, gives the following:

$$\begin{aligned}\partial_t\phi_r &= -\partial_{\boldsymbol{x}}\cdot\phi_r\boldsymbol{v}_r + \partial_{\boldsymbol{x}}\cdot\mathcal{D}_x\partial_{\boldsymbol{x}}\phi_r + c_r^{\phi} \quad ;\\ \partial_t\sigma_r &= -\partial_{\boldsymbol{x}}\cdot\sigma_r\boldsymbol{v}_r + \partial_{\boldsymbol{x}}\cdot\mathcal{D}_x\partial_{\boldsymbol{x}}\sigma_r + c_r^{\sigma}\end{aligned} \tag{3.5}$$

where it is

$$\sigma_r(\boldsymbol{x},t)\equiv\phi_r(\boldsymbol{x},t)s_r(\boldsymbol{x},t),\ \boldsymbol{v}_r(\boldsymbol{x},t)\equiv\langle\boldsymbol{v}|s\rangle[\boldsymbol{x},t;s_r(\boldsymbol{x},t)] \tag{3.6}$$

The quantity σ_r denotes the rth quadrature weighted node; following Marchisio and Fox (2007), we prefer to operate in terms of these variables instead of the quadrature nodes. We must note that weights and weighted nodes are not conservative; the source terms $c_r^{\phi}(\boldsymbol{x},t)$ and $c_r^{\sigma}(\boldsymbol{x},t)$ that appear in their evolution equations are given by this set of linear equations:

$$\begin{aligned}&(1-a)\sum_{r=1}^{\nu} s_r^a c_r^{\phi} + a\sum_{r=1}^{\nu} s_r^{a-1}c_r^{\sigma}\\ &\quad = a(a-1)\mathcal{D}_x\sum_{r=1}^{\nu}\phi_r s_r^{a-2}\partial_{\boldsymbol{x}}s_r\cdot\partial_{\boldsymbol{x}}s_r \quad \text{for} \quad 0\le a\le 2\nu-1\end{aligned} \tag{3.7}$$

One can obtain the evolution equations of the quadrature nodes by combining the evolution equations above; this gives the following:

$$\phi_r D_t s_r \equiv \phi_r(\partial_t s_r + \boldsymbol{v}_r\cdot\partial_{\boldsymbol{x}}s_r) = \partial_{\boldsymbol{x}}\cdot\phi_r\mathcal{D}_x\partial_{\boldsymbol{x}}s_r + c_r^{s} \tag{3.8}$$

where it is

$$c_r^{s} = c_r^{\sigma} - s_r c_r^{\phi} + \mathcal{D}_x\partial_{\boldsymbol{x}}\phi_r\cdot\partial_{\boldsymbol{x}}s_r \tag{3.9}$$

Notice that if the diffusive term in Eq. (3.4) is neglected, that is, if we set $\mathcal{D}_x = 0$, assuming that the evolution equations of the actual and quadrature-based VDFs coincide, the diffusive and source terms in all equations vanish, Eq. (3.8) predicting (incorrectly) that quadrature nodes do not change along pathlines; in this case, only macromixing can occur, but powders cannot micromix (*i.e.*, mixing at the particle length scale is impossible). This clearly shows that the diffusive contribution is essential; the issue that arises is assigning a value to the diffusivity. We shall address this in Section 4.

Following Fan et al. (2004), to determine the velocity field $\boldsymbol{v}_r(\boldsymbol{x},t)$ we use a coarse-grained dynamical equation of this form:

$$\begin{aligned}\partial_t(\rho_s\phi_r\boldsymbol{v}_r) = &-\partial_{\boldsymbol{x}}\cdot(\rho_s\phi_r\boldsymbol{v}_r\boldsymbol{v}_r) + \partial_{\boldsymbol{x}}\cdot\boldsymbol{S}_r + \boldsymbol{f}_r\\ &+\sum_{u=1}^{\nu}(\boldsymbol{f}_{ru}+\boldsymbol{\zeta}_{ru}) + \rho_s\phi_r\boldsymbol{g}\end{aligned} \tag{3.10}$$

where ρ_s is the solid density (which is the same for all particles), $\boldsymbol{S}_r(\boldsymbol{x},t)$ is the effective solid stress tensor, accounting for kinetic and collisional stress, $\boldsymbol{f}_r(\boldsymbol{x},t)$ is the fluid–particle interaction force, whilst $\boldsymbol{f}_{ru}(\boldsymbol{x},t)$ and $\boldsymbol{\zeta}_{ru}(\boldsymbol{x},t)$ are the interaction force and the linear momentum exchanged (owing to mass transfer) between the quadrature classes r and u, respectively.

In Eq. (3.10), the effective solid stress tensor and the particle–particle interaction force are functions of the granular temperatures of the quadrature classes (Lu and Gidaspow, 2003). To find the granular temperature for the rth quadrature class, we solve this pseudointernal energy balance equation:

$$\begin{aligned}\partial_t(\rho_s\phi_r U_r) = &-\partial_{\boldsymbol{x}}\cdot\rho_s\phi_r U_r\boldsymbol{v}_r - \partial_{\boldsymbol{x}}\cdot\boldsymbol{q}_r + \boldsymbol{S}_r:\partial_{\boldsymbol{x}}\boldsymbol{v}_r\\ &+G_r - S_r + \sum_{u=1}^{\nu}(G_{ru}-S_{ru})\end{aligned} \tag{3.11}$$

Here $U_r(\boldsymbol{x},t)\equiv 3\Theta_r(\boldsymbol{x},t)/2$ is the pseudointernal energy, $\Theta_r(\boldsymbol{x},t)$ is the granular temperature and $\boldsymbol{q}_r(\boldsymbol{x},t)$ is the pseudothermal heat flux; moreover, $G_r(\boldsymbol{x},t)$ is a source term owing to fluctuating fluid–particle forces, $S_r(\boldsymbol{x},t)$ is a sink term owing to the viscous resistance to particle motion, whilst $G_{ru}(\boldsymbol{x},t)$ and $S_{ru}(\boldsymbol{x},t)$ are a source term and a sink term, respectively, the first related to the pseudointernal energy exchanged (owing to mass transfer) between the rth and uth quadrature classes and the second to the pseudointernal energy dissipated by means of inelastic particle collisions (Fan and Zhu, 1998; Jackson, 2000). For the constitutive equations employed to express the unclosed terms in the equations above, we refer to Mazzei et al. (2010a); further details can also be found in Fan and Zhu (1998) and Lu and Gidaspow (2003).

To determine the volume fraction of fluid, denoted as $\varepsilon(\boldsymbol{x},t)$, we do not have to use a transport equation; we can find this field using the relation

$$\varepsilon = 1-\sum_{r=1}^{\nu}\phi_r = 1-\mathcal{M}_0 \tag{3.12}$$

where $\mathcal{M}_0(\boldsymbol{x},t)$, the zeroth-order moment of the distribution, represents the overall volume fraction of solid. The fluid velocity field, conversely, is given by a course-grained dynamical equation similar to Eq. (3.10). For further details we refer to Mazzei and Lettieri (2008) or Lettieri et al. (2003).

3.2. QMOM evolution equations

In this work we intend to simulate the system behavior using only DQMOM. However, because subsequently we shall need the QMOM evolution equations for some considerations, we now derive them here. This will also allow us to show that the two quadrature-based methods are equivalent on theoretical grounds, and that consequently what holds for DQMOM is also true for the other method.

To derive the QMOM evolution equations we could operate on Eq. (3.4), multiplying all its terms by s^a and then integrating out the size coordinate (Mazzei et al., 2012). Here we follow an alternative route: we use the DQMOM evolution equations as starting point. To this end, let us multiply the evolution equations of the quadrature weights by $(1-a)s_r^a$ and those of the quadrature weighted nodes by as_r^{a-1}. With a few passages, omitted for brevity, we obtain the following:

$$\begin{aligned}&(1-a)s_r^a\partial_t\phi_r + as_r^{a-1}\partial_t\sigma_r = \partial_t(\phi_r s_r^a), \quad (1-a)s_r^a\partial_{\boldsymbol{x}}\cdot\phi_r\boldsymbol{v}_r\\ &\quad + as_r^{a-1}\partial_{\boldsymbol{x}}\cdot\sigma_r\boldsymbol{v}_r = \partial_{\boldsymbol{x}}\cdot\phi_r s_r^a\boldsymbol{v}_r\end{aligned} \tag{3.13}$$

Moreover, it is

$$\begin{aligned}&(1-a)s_r^a\partial_{\boldsymbol{x}}\cdot\mathcal{D}_x\partial_{\boldsymbol{x}}\phi_r + as_r^{a-1}\partial_{\boldsymbol{x}}\cdot\mathcal{D}_x\partial_{\boldsymbol{x}}\sigma_r = \partial_{\boldsymbol{x}}\cdot\mathcal{D}_x\partial_{\boldsymbol{x}}(\phi_r s_r^a)\\ &\quad -a(a-1)\mathcal{D}_x\phi_r s_r^{a-2}\partial_{\boldsymbol{x}}s_r\cdot\partial_{\boldsymbol{x}}s_r\end{aligned} \tag{3.14}$$

Consequently, from Eqs. (3.5), we obtain the following:

$$\begin{aligned}\partial_t(\phi_r s_r^a) = &-\partial_{\boldsymbol{x}}\cdot\phi_r s_r^a\boldsymbol{v}_r + \partial_{\boldsymbol{x}}\cdot\mathcal{D}_x\partial_{\boldsymbol{x}}(\phi_r s_r^a)\\ &-a(a-1)\mathcal{D}_x\phi_r s_r^{a-2}\partial_{\boldsymbol{x}}s_r\cdot\partial_{\boldsymbol{x}}s_r + (1-a)s_r^a c_r^{\phi} + as_r^{a-1}c_r^{\sigma}\end{aligned} \tag{3.15}$$

Finally, summing over the index r for $1\le r\le\nu$ and resorting to Eqs. (3.3) and (3.7), we obtain the following evolution equations:

$$\partial_t\mathcal{M}_a = -\partial_{\boldsymbol{x}}\cdot\sum_{r=1}^{\nu}\phi_r s_r^a\boldsymbol{v}_r + \partial_{\boldsymbol{x}}\cdot\mathcal{D}_x\partial_{\boldsymbol{x}}\mathcal{M}_a \quad \text{for} \quad 0\le a\le 2\nu-1 \tag{3.16}$$

This is the same result that one would obtain by operating, as described above in Eq. (3.4). This confirms, as expected, that QMOM and DQMOM are theoretically equivalent. We should not be surprised, because both methods are based on the same presumed functional form of the volume density function. So, on theoretical grounds no method is superior, both methods presenting the same limitations. In particular, in both methods neglecting the diffusive contribution renders particle micromixing impossible; this is plainly revealed solely by the DQMOM evolution equations, but we know that the same issue is present in QMOM, for its evolution equations can be derived from those of DQMOM.

The equations above, which hold for the first 2ν integer moments of the VDF, show that the moments, as opposed to the quadrature weighted nodes and weights, are conservative in the case under study. Reminding that, for instance, $\mathcal{M}_0$ represents the overall volume fraction of particles, its evolution equation states that the volume (and therefore the mass) of particles is a conservative quantity, as one should expect in this case. This condition is met because the source terms in the moment evolution equations vanish.

We should note, however, that the same is not true for the diffusive terms, which instead survive; this may appear incorrect, but we can explain it as follows. If to derive the evolution equation of $\mathcal{M}_a$ we adopt Eq. (3.1), that is, the PBE written in terms of the real VDF, we obtain the following:

$$\partial_t \mathcal{M}_a = -\partial_{\boldsymbol{x}} \cdot \int_0^\infty \langle \boldsymbol{v}|s\rangle s^a f \, ds \tag{3.17}$$

The last two equations give the following:

$$-\mathcal{D}_x \partial_{\boldsymbol{x}} \mathcal{M}_a = \int_0^\infty \langle \boldsymbol{v}|s\rangle s^a f \, ds - \sum_{r=1}^{\nu} \phi_r s_r^a \boldsymbol{v}_r \quad \text{for} \quad 0 \le a \le 2\nu - 1 \tag{3.18}$$

This shows why it is indeed necessary that the equations which govern the evolution of the moments feature diffusion; if they did not, we would be claiming that the fluxes on the right-hand side of Eqs. (3.18) are equal, this being in general untrue. We model the deviation flux as a diffusive flux. The issue, as said, is assigning a suitable value to the coefficient of diffusion.

4. Discussion on the diffusivity order of magnitude

Let t_c and x_c be the time and length scales, respectively, in the bulk of the domain. By definition, the scales of the independent variables have to render the derivatives of the scaled nondimensional dependent variables of unit order of magnitude. As for the dependent variables, ϕ_r is already dimensionless and of order unity; thus, we only need to introduce scales for the velocity fields and for the quadrature nodes. Let these be v_c and s_c, respectively. Then, if we introduce the dimensionless variables

$$\bar{t} \equiv t/t_c, \quad \bar{\boldsymbol{x}} \equiv \boldsymbol{x}/x_c, \quad \bar{\boldsymbol{v}}_r \equiv \boldsymbol{v}_r/v_c, \quad \bar{s}_r \equiv s_r/s_c \tag{4.1}$$

the first of Eqs. (3.5) gives the following:

$$\partial_{\bar{t}} \phi_r = -\frac{v_c t_c}{x_c} \{\partial_{\bar{\boldsymbol{x}}} \cdot \phi_r \bar{\boldsymbol{v}}_r\} + \frac{\mathcal{D}_x t_c}{x_c^2} \left\{\partial_{\bar{\boldsymbol{x}}} \cdot \partial_{\bar{\boldsymbol{x}}} \phi_r + \bar{c}_r^{\phi}\right\} \tag{4.2}$$

As done in the equation above, in what follows we shall assume that $\mathcal{D}_x$ is constant. Two characteristic times arise, x_c/v_c and $x_c^2/\mathcal{D}_x$, which relate to convection and diffusion, respectively. Their ratio is the inverse of the Peclet number, $\varphi \equiv 1/\mathrm{Pe}$, where $\mathrm{Pe} \equiv v_c x_c/\mathcal{D}_x$. Since the equation is scaled, the term on the left-hand side and the bracketed terms on the right-hand side have unit order of magnitude; consequently, if $\varphi \ll 1$ the characteristic time must be x_c/v_c, while if $\varphi \gg 1$ it must be $x_c^2/\mathcal{D}_x$. Thus, the characteristic time depends on the value of Pe and, as one would expect, the time scale is dictated by the term that dominates the quadrature weight rate of change.

As we shall presently explain, we expect that in the problem that we are studying φ is a small parameter. Consequently, we choose $t_c \equiv x_c/v_c$. This yields

$$\partial_{\bar{t}} \phi_r = -\partial_{\bar{\boldsymbol{x}}} \cdot \phi_r \bar{\boldsymbol{v}}_r + \varphi \{\partial_{\bar{\boldsymbol{x}}} \cdot \partial_{\bar{\boldsymbol{x}}} \phi_r + \bar{c}_r^{\phi}\} \tag{4.3}$$

Eq. (3.8) instead gives the following:

$$\phi_r D_{\bar{t}} \bar{s}_r \equiv \phi_r (\partial_{\bar{t}} \bar{s}_r + \bar{\boldsymbol{v}}_r \cdot \partial_{\bar{\boldsymbol{x}}} \bar{s}_r) = \varphi \{\partial_{\bar{\boldsymbol{x}}} \cdot \phi_r \partial_{\bar{\boldsymbol{x}}} \bar{s}_r + \bar{c}_r^{s}\} \tag{4.4}$$

Hence, over short dimensionless times of order φ convection dictates the evolution of the quadrature weights and nodes, causing macromixing. But diffusion plays a crucial role over longer times, allowing micromixing to occur and quadrature nodes to change along pathlines; the effects of diffusion are negligible compared to those of convection over short times, but these small effects build up, becoming important over long times. This is an example of singularity in the unbounded time domain (Simmonds and Mann, 1998). So, to correctly predict the pseudosteady state of the system – which by definition refers to long times – and its particle size distribution in particular, we cannot neglect the diffusive contribution.

It is worth emphasizing that the considerations above, and in particular the reason why we cannot neglect the deviation flux between actual and approximated convective fluxes, has nothing whatsoever to do with the numerical code that we employ to solve the equations of the model and with the errors that such a code might generate. The reason has only to do with the form that the quadrature-based methods assume for the volume density function that describes the particle population.

To explain why we find it reasonable to assume that φ is a small parameter, we note that $-\mathcal{D}_x \partial_{\boldsymbol{x}} \mathcal{M}_a$ is the error that we make when approximating the convective flux of $\mathcal{M}_a$, that is, the integral on the right-hand side of Eq. (3.17), with the quadrature formula. The greater the number of classes used in the latter, the smaller the error; however, the error is never zero. There must be, therefore, a minimum quadrature order n (that is, a minimum number of classes) which ensures that the error is small enough to render $\varphi < 1$. In what follows, we shall refer to quadrature formulae of order ν that satisfy this condition, assuming accordingly that $\nu \ge n$. This ensures that

$$\mathcal{D}_x < \mathcal{D}_{x,2} \equiv v_c x_c \tag{4.5}$$

Here $\mathcal{D}_{x,2}$ denotes an upper bound value for the diffusion coefficient. As we know that the higher the order of the quadrature formula, the smaller the error that this makes in approximating moment convective fluxes, we know that there must exist a more accurate upper bound value dependent on the order of the formula, and that this must decrease as the order ν increases. However, being unable to determine it, we must refer to the value identified above, which ensures that convection dominates over diffusion.

In the problem at hand, reasonable orders of magnitude for the length and velocity scales are given by the characteristic dimension of the vessel containing the fluidized bed and the fluidization velocity, respectively (as one can verify numerically). The diameter of the vessel used in this work is 0.35 m, while the fluidization velocities are 0.10 and 0.05 m/s. To be conservative, we take $x_c = 10^{-2}$ m and $v_c = 10^{-1}$ m/s, which give $\mathcal{D}_{x,2} = 10^{-3}$ m^2/s as an upper bound value for $\mathcal{D}_x$.

The upper bound value just estimated is not sufficient for deciding which value to assign to $\mathcal{D}_x$. We need also a lower bound value, which we denote as $\mathcal{D}_{x,1}$. We know that it must be greater than zero, and we expect it to be some orders of magnitude lower than the upper bound value $\mathcal{D}_{x,2}$. Nevertheless, we have no criterion based on physical grounds for estimating it. This, in principle, poses an issue, insofar as we do not know how small $\mathcal{D}_x$ should be. The numerical diffusion generated by the CFD code, however, renders the problem less critical, as we shall presently see.

The numerical scheme that the CFD code uses to discretize the spatial derivatives is diffusive; so, the real nondimensionalized evolution equation solved for the rth quadrature weight (similar considerations hold for the quadrature nodes and weighted nodes) reads as follows:

$$\partial_{\bar{t}} \phi_r = -\partial_{\bar{\boldsymbol{x}}} \cdot \phi_r \bar{\boldsymbol{v}}_r + \varphi \{[1 + (\mathcal{D}_n/\mathcal{D}_x)] \partial_{\bar{\boldsymbol{x}}} \cdot \partial_{\bar{\boldsymbol{x}}} \phi_r + \bar{c}_r^{\phi}\} \tag{4.6}$$

Here $\mathcal{D}_n$ is the coefficient of numerical diffusion, which depends on the numerical discretization scheme and on the grid size. The smaller $\mathcal{D}_n/\mathcal{D}_x$, the less numerical diffusion affects the solution; thus, one should favor discretization schemes that are little

diffusive – preferring, for instance, a second-order discretization scheme to a first-order one. The drawback is that these schemes are often less stable. We did try to solve the model using a second-order discretization scheme, but all the simulations invariably crashed almost immediately. Only the first-order discretization scheme permitted us to run the simulations. Probably, this was because to implement the DQMOM evolution equations we had to modify significantly the default model of the code. So, as for the discretization scheme, we had no real choice.

The coefficient of numerical diffusion for a first-order discretization scheme can be estimated, following Ferziger and Peric (2002), using the relation $2\mathcal{D}_n = uL$, where L is the length of the computational cell side and u is the velocity of advection, which has the same order of magnitude as v_c. Therefore, the sole degree of freedom available to reduce $\mathcal{D}_n$ is the computational cell side; in particular, to reduce the order of magnitude of $\mathcal{D}_n$ one needs to reduce the *order of magnitude* of L. Doing so is often unfeasible, insofar as it renders the computational times unacceptably long. In setting a value for L, one needs to compromise between accuracy of the solution and length of the computational times. In this work we adopted a uniform computational grid made of square cells of side $L=5$ mm. Employing a time step of 10^{-4} seconds, to simulate the systems for ten real-time seconds, which was sufficient for them to reach pseudosteady-state conditions, we had to accept computational times as long as four and a half months. Reducing the cell size to 1 mm – *which does not alter the order of magnitude of* $\mathcal{D}_n$ – results already in unacceptable computational times (running the simulations in parallel would not alleviate the problem significantly, especially because we had to run several simulations and in consequence we needed several processors). With such a cell size the order of magnitude of $\mathcal{D}_n$ results to be 10^{-4} m^2/s, just one order of magnitude lower than that of $\mathcal{D}_{x,2}$. Since we cannot set $\mathcal{D}_x$ lower than $\mathcal{D}_n$, for otherwise numerical diffusion would dominate over real diffusion, even if we had been able to estimate a value for $\mathcal{D}_{x,1}$, this would be overridden by that of $\mathcal{D}_n$. So, at least in this work, the numerics imposes a lower bound value for the coefficient of diffusion.

As seen, the order of magnitude of $\mathcal{D}_n$ is lower than that of $\mathcal{D}_{x,2}$. This is important, since, if this condition was not met, being $\mathcal{D}_n/\mathcal{D}_{x,2} > 1$, as it must be $\mathcal{D}_x/\mathcal{D}_{x,2} < 1$, it would be $\mathcal{D}_n/\mathcal{D}_x \gg 1$. This would have two consequences. The first is that numerical diffusion would prevail over real diffusion – the diffusion rate being dictated by the numerics; the second is that *the diffusive and source terms present in* Eq. (4.6) *would no longer be balanced*, the former prevailing over the latter. These effects, particularly the second, would lead to wrong predictions. Note that the strongest imbalance between diffusion and generation is not found for large values of the numerical diffusion coefficient, but for $\mathcal{D}_x = 0$, insofar as, owing to numerical diffusion, the diffusive term survives, whilst the source term vanishes. This shows another reason, this time numerical, for which we must not neglect the diffusive terms in the evolution equations.

In light of this, we conclude that $\mathcal{D}_x$ must lie between $\mathcal{D}_n$ and $\mathcal{D}_{x,2}$. This ensures that convection prevails over diffusion (a necessary condition for quadrature formulae that well approximate the moments external to the set tracked by the methods, and in particular the convective fluxes) and that the balance between diffusion and generation in the evolution equations is preserved. Any value external to this range would lead to entirely wrong results. Larger values would allow diffusion to erase the gradients generated by convection, flattening the spatial profiles of the quadrature weights and weighted nodes, of the moments and of the volume fraction of fluid; in bubbling beds, for instance, this would make bubbles disappear. Smaller values, as stated, would alter the balance between diffusion and generation, letting the former dominate; the effect is that weights and weighted nodes would essentially be modeled – incorrectly – as conservative quantities. We shall confirm all this numerically in Section 6, in which we conduct a sensitivity analysis on the diffusivity.

To conclude, let us summarize concisely how to estimate the diffusion coefficient $\mathcal{D}_x$. First, one needs to estimate the upper bound value $\mathcal{D}_{x,2}$ employing Eq. (4.5). To this end, one needs to estimate the macroscopic length scale x_c characterizing the system (dictated by the dimensions of the latter) and the velocity scale v_c (assumed to be of the same order of magnitude as the fluidization velocity and accordingly dependent on the properties of the powder being fluidized). Then, one has to estimate the numerical diffusion coefficient $\mathcal{D}_n$ (dependent on the discretization scheme and on the grid size used, and so related only to the numerics). The value for $\mathcal{D}_x$ has to lie between the lower and upper bound values just identified.

5. Implementation of the DQMOM model in CFD

To run the simulations we employed the commercial CFD code Fluent 12.1, implementing the evolution and constitutive equations in the multifluid model of the package, which is based on a Eulerian flow description, and adopting user-defined functions and subroutines. We used a quadrature approximation of order two (that is, $\nu = 2$), tracking the evolution of two quadrature weights and weighted nodes and defining three phases in the multifluid model: one gas and two particle phases. As pointed out, we solved balance equations for linear momentum and pseudointernal energy for each phase.

The first equations that we had to implement were those governing the quadrature weights. These differ from customary continuity equations, because they feature diffusion (we should notice that the quantities ϕ_r are not volume fractions of real particle phases, but weights of a quadrature formula). The simplest strategy to implement such equations would have been introducing diffusion and generation in the default continuity equations of the code that govern the evolutions of the volume fractions of the particle phases with which the quadrature classes are associated. This, however, was impossible, because Fluent does not allow to introduce diffusion in such equations.

Accordingly, we had to resort to another strategy, which does not rely on the continuity equations for the particle phases available in the code. First of all, we had to disable these equations, which therefore the code no longer solved. Then, we had to treat quadrature weights as user-defined scalars, for which the code allows to define evolution equations involving both convection and diffusion. Fluent allows to associate scalars with the mixture or with a phase. In the first case, the equation reads as follows:

$$\partial_t(\rho_m\psi) + \partial_{\boldsymbol{x}}\cdot\rho_m\psi\boldsymbol{v}_m - \partial_{\boldsymbol{x}}\cdot\mathcal{D}_x\partial_{\boldsymbol{x}}\psi = S_\psi \tag{5.1}$$

ψ being a generic scalar, ρ_m and $\boldsymbol{v}_m$ the mixture density and velocity, respectively, while S_ψ a source term. In the second case, it reads as follows:

$$\partial_t(\rho_r\alpha_r\psi) + \partial_{\boldsymbol{x}}\cdot\rho_r\alpha_r\psi\boldsymbol{v}_r - \partial_{\boldsymbol{x}}\cdot\alpha_r\mathcal{D}_x\partial_{\boldsymbol{x}}\psi = S_\psi \tag{5.2}$$

where α_r is the volume fraction of the solid phase with which the scalar ψ is associated. Both equations are unsuitable; the first because our scalars are not advected at the velocity of the mixture, but at the velocities of their respective quadrature classes, while the second because it multiplies the scalar and the diffusivity by the phase volume fraction (refer to Eqs. (3.5)).

The simplest way to overcome the issue is associating the scalars with the mixture, so that the diffusion term is correctly modeled, and changing the accumulation and advection terms via user-defined functions; in particular, we implemented the changes $\rho_m \to 1$ and $\rho_m\boldsymbol{v}_m \to \boldsymbol{v}_r$. The first ensures that the density does not

appear in the accumulation term, whereas the second replaces the mixture mass flux with the phase volume flux. To implement these modifications, we resorted to the user-defined functions DEFINE_UDS_UNSTEADY and DEFINE_UDS_FLUX. Finally, we set the following:

$$S_\psi = c_r^\phi = \frac{6(\phi_s \mathcal{D}_x \partial_{\boldsymbol{x}} s_s \cdot \partial_{\boldsymbol{x}} s_s - \phi_r \mathcal{D}_x \partial_{\boldsymbol{x}} s_r \cdot \partial_{\boldsymbol{x}} s_r)}{(s_r - s_s)^2} \tag{5.3}$$

where the indexes r and s denote the two quadrature classes. This is the expression for the source term c_r^ϕ that one obtains from Eqs. (3.7) when setting $\nu = 2$. We derived the gradients of the quadrature nodes from those of the quadrature weights and weighted nodes, as it is

$$\partial_{\boldsymbol{x}} s_r = (\partial_{\boldsymbol{x}} \sigma_r - s_r \partial_{\boldsymbol{x}} \phi_r)/\phi_r \tag{5.4}$$

Notice that Eq. (5.3) shows that the sources for the two quadrature classes are equal in magnitude and opposite in sign, so that the overall volume of particles (and consequently also their mass) is conserved. We checked this numerically, verifying that the volume integral of the zeroth-order moment over the entire computational domain is constant. We should also point out that, in each cell, at each time step and in each iteration of the computation, the values of the particle phase volume fractions (for which the CFD code no longer solved any evolution equations) were set equal to those of the weights of the quadrature formula.

To implement the evolution equations of the quadrature weighted nodes we proceeded similarly, treating them as user-defined scalars associated with the mixture. Also in this case we used the user-defined functions DEFINE_UDS_UNSTEADY and DEFINE_UDS_FLUX to set the expressions for accumulation and convection, respectively; furthermore, we used this expression for the source terms

$$S_\psi = c_r^\sigma = \frac{2\phi_s(s_s + 2s_r)\mathcal{D}_x \partial_{\boldsymbol{x}} s_s \cdot \partial_{\boldsymbol{x}} s_s - 2\phi_r(s_r + 2s_s)\mathcal{D}_x \partial_{\boldsymbol{x}} s_r \cdot \partial_{\boldsymbol{x}} s_r}{(s_r - s_s)^2} \tag{5.5}$$

where the indexes r and s denote the two quadrature classes. This is the expression for the source term c_r^σ that one obtains from Eqs. (3.7) for $\nu = 2$.

We do not report in this paper the boundary conditions and the numerical techniques used to solve the model, because these are the same as those presented in Mazzei et al. (2010a). As for the initial conditions, from the initial particle size distributions of the powders, one can calculate the initial values of the first four integer moments of the VDF and of the nodes and weights of the quadrature formula. We explained how to do this in Mazzei et al. (2010a), and thus we do not repeat it here. Table 1 reports the results for $\varepsilon = 0.400$. These values, which provide the initial conditions for the DQMOM evolution equations, are a function of the voidage ε because, whereas the PSD refers to solid mass fractions on a void-free basis, the VDF accounts for voids and provides volumes of solid per unit volume of physical space.

Table 1
Initial values of the moments of the volume density function and of the quadrature nodes and weights obtained from the experimental PSDs referred to $\varepsilon = 0.400$.

Powder	Moments of the volume density function			
	$\mathcal{M}_0$ (−)	$\mathcal{M}_1$ (μm)	$\mathcal{M}_2$ (μm^2)	$\mathcal{M}_3$ (μm^2)
A	0.600	5.45×10^1	5.06×10^3	4.82×10^5
B	0.600	1.70×10^2	4.98×10^4	1.52×10^7
	Quadrature nodes and weights			
Powder	s_1(μm)	ϕ_1(−)	s_2(μm)	ϕ_2(−)
A	75	0.262	103	0.338
B	240	0.380	355	0.220

6. Sensitivity analysis on the diffusion coefficient

In this section we test the considerations presented in Section 4, reporting how sensitive the numerical results are on the value selected for the coefficient of diffusion $\mathcal{D}_x$. To this end, we simulated several times the same system under identical operating conditions, each time using a different value for $\mathcal{D}_x$. We then compared the numerical results. The system investigated was described in Section 2. Initially, at time $t = 0$, the bed is packed and constituted of two superposed layers; these are 15 mm high and together occupy half of the vessel. The lower and upper powders differ solely in particle size distribution, the upper one having larger mean particle size. We suddenly start fluidizing the system, setting the value of the superficial velocity of the fluid equal to 0.15 m/s, which is sufficient to attain very good mixing. Each simulation ran for ten real-time seconds, a time long enough to reach pseudosteady-state conditions. The operating conditions used, as said, are those promoting nearly perfect mixing; the reason is simple: in this case we can calculate *analytically* the pseudosteady-state values of the quadrature nodes and weights, and in consequence we can easily assess the accuracy of the numerical results.

The procedure required for determining analytically the values of the quadrature nodes and weights after perfect mixing is reported in Mazzei (2011); thus, we do not repeat it here. The values of these quantities are reported in Table 2 and are referred to as *correct values*. The table reports also another set of values, referred to as *incorrect values*, whose meaning we now explain. We said that the diffusive terms cannot be neglected in the evolution equations of the quadrature weighted nodes and weights, for otherwise micromixing cannot take place. If we set $\mathcal{D}_x = 0$ and solve Eqs. (3.5) numerically using a CFD code, in these equations the source terms vanish, insofar as Eqs. (3.7) become homogeneous, while the diffusive terms survive owing to numerical diffusion. Micromixing therefore takes place, but the results are incorrect for the diffusive and source terms are imbalanced. These are the incorrect results reported in Table 2. To obtain them analytically, one needs to regard the quadrature weighted nodes and weights as conservative quantities (this is incorrect, but consistent with the evolution equations solved for $\mathcal{D}_x = 0$ and $\mathcal{D}_n \neq 0$, that is, in the presence of numerical diffusion) and simply require that they be conserved throughout the mixing process.

One expects that these incorrect results are obtained also for finite diffusivities when $\mathcal{D}_x \ll \mathcal{D}_n$, since in this case the diffusive and source terms remain strongly imbalanced. Conversely, one expects that the correct results reported in Table 2 are obtained for any $\mathcal{D}_x \gg \mathcal{D}_n$, regardless of the specific value employed for $\mathcal{D}_x$. The correct and incorrect values for the quadrature weighted nodes and weights just identified and reported in Table 2 are hence limiting values, and one might expect that the predicted values yielded by the numerical simulations should fall between them.

Fig. 1 shows the pseudosteady-state values, predicted numerically, of the quadrature nodes and weights after good mixing has taken place as a function of $\mathcal{D}_x$. The vertical dashed line identifies the value of $\mathcal{D}_n$. We also remind that the upper bound value $\mathcal{D}_{x,2}$

Table 2
Correct and incorrect values (refer to Section 6) of the quadrature nodes and weights obtained after mixing powders A and B for $\varepsilon = 0.400$. The values are computed analytically, not numerically.

	Quadrature nodes and weights			
	s_1 (μm)	ϕ_1 (−)	s_2 (μm)	ϕ_2 (−)
Correct values	104	0.366	316	0.234
Incorrect values	173	0.321	202	0.279

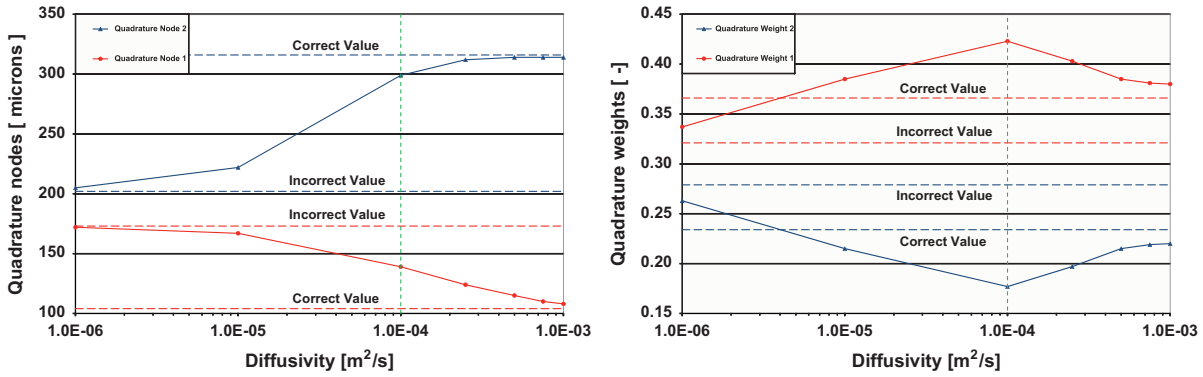

Fig. 1. Values of the VDF quadrature nodes and weights at pseudosteady-state conditions in the mixed fluidized bed as a function of the diffusion coefficient $\mathcal{D}_x$. The (modified) quadrature weights refer to a spatially uniform void fraction equal to 0.400.

formerly estimated for $\mathcal{D}_x$ is 10^{-3} m^2/s. The horizontal dashed lines identify the correct and incorrect values of the quadrature nodes and weights reported in Table 2. As we see, for $\mathcal{D}_x \ll \mathcal{D}_n$ the numerical values tend towards the incorrect values predicted analytically. In this case the order of magnitude of the diffusive terms in the evolution equations is much larger than that of the source terms, the quadrature weighted nodes and weights are essentially modeled as conservative quantities, and so the results are entirely incorrect. As the value of $\mathcal{D}_x$ increases, in particular for $\mathcal{D}_x > 0.1\mathcal{D}_n$, the quadrature nodes change monotonically, their values tending toward the correct values reported in Table 2. We note, in particular, that for $\mathcal{D}_x \sim 10\mathcal{D}_n$ the agreement is nearly perfect, this being true also for larger values of the diffusivity. Consequently, *from this standpoint*, the larger the value the better. The quadrature weights do not vary monotonically; in particular, we see that their worst values are obtained for $\mathcal{D}_x \sim \mathcal{D}_n$. Nevertheless, for larger values of $\mathcal{D}_x$ the quadrature weights change quite rapidly, their values tending towards the correct ones calculated analytically. For $\mathcal{D}_x \sim 10\mathcal{D}_n$ the agreement is very good, this being true also for larger values of the diffusivity. Apparently, once again, the larger the value of $\mathcal{D}_x$ the better.

We should highlight an important aspect. The values of the quadrature weights reported in the tables and in Fig. 1 *refer to a reference value of the void fraction*, which we chose to be equal to 0.400. This value is close to that observed in packed beds, into which the fluidized bed turns when we cut off the fluid supply. We could have chosen ε equal to zero, in which case the weights would have referred to a void-free powder. We have to choose a reference value for ε because we want to *eliminate the apparent nonuniformities* in the particle size distribution of the powder induced by the presence of bubbles, or more in general by the spatial variations of the void fraction. Since the powder is nearly perfectly mixed, we would expect the functions $\phi_r(\boldsymbol{x}, t)$ to have nearly flat spatial profiles, their values being virtually independent of $\boldsymbol{x}$. This does not happen, however, due to the variations in $\varepsilon(\boldsymbol{x}, t)$, which are induced by the fluid dynamics and have nothing to do with the powder particle size distribution. To eliminate these effects, so that the functions $\phi_r(\boldsymbol{x}, t)$ reflect solely properties of the PSD, we need to refer these quantities to a spatially uniform reference value of the void fraction. As said, we chose a reference value close to that observed in loosely packed powders. For what we are going to report now, we need to clearly distinguish between the original functions $\phi_r(\boldsymbol{x}, t)$ yielded by the CFD simulations (which in a bubbling bed, even for a well-mixed powder, are not uniform in space, owing to the fluid dynamic effects just mentioned) and those referring to a spatially uniform reference value of the void fraction (which in a bubbling bed, for a well-mixed powder, are indeed uniform in space). So, in the considerations reported below we shall denote as $\phi_r^\star(\boldsymbol{x}, t)$ the latter modified functions.

Let us return to our former considerations. We have seen that for $\mathcal{D}_x \sim 10\mathcal{D}_n$ the values of the functions $s_r(\boldsymbol{x}, t)$ and $\phi_r^\star(\boldsymbol{x}, t)$, and accordingly the particle size distribution of the well-mixed powder, are accurately predicted. The diagrams shown in Fig. 1, in particular, seem to indicate that we can choose the value of $\mathcal{D}_x$ as large as we like, that is, of order even larger than $10\mathcal{D}_n$. But this is not true for the functions $\phi_r(\boldsymbol{x}, t)$, and in turn for the function $\varepsilon(\boldsymbol{x}, t)$, whose values strongly depend on the system fluid dynamics. As said, another condition that has to be met is $\mathcal{D}_x < \mathcal{D}_{x,2}$, for otherwise diffusion would erase the spatial nonuniformities that convection generates; in this instance, also the functions $\phi_r(\boldsymbol{x}, t)$ would become uniform in space and the bubbles, or more generally the void fraction gradients within the fluidized bed, would be partly or entirely lost. This is what we observed numerically.

Fig. 2 reports the spatial profiles, after mixing has occurred, of the void fraction and of the unmodified and modified quadrature weights for various values of the diffusivity. As we can see, for $\mathcal{D}_x \sim \mathcal{D}_n$ and lower orders of magnitude, the powder is well mixed, the functions $\phi_r^\star(\boldsymbol{x}, t)$ being nearly uniform in space. But the same is not true for the functions $\phi_r(\boldsymbol{x}, t)$ and $\varepsilon(\boldsymbol{x}, t)$, whose spatial gradients, caused by the fluid dynamics, are clearly visible. This is the picture that one would expect. For $\mathcal{D}_x \sim \mathcal{D}_{x,2}$ and larger orders of magnitude, conversely, these gradients are lost, for diffusion partly or entirely erases them. The powder tends to become homogeneous, and the bed height is accordingly overpredicted.

These findings confirm that the value of $\mathcal{D}_x$ must lie between $\mathcal{D}_n$ and $\mathcal{D}_{x,2}$. If these two values differed by several orders of magnitude, one could choose a value for the diffusivity such that $\mathcal{D}_n \ll \mathcal{D}_x \ll \mathcal{D}_{x,2}$. Such a condition would be ideal, because it would ensure that the values of the quadrature nodes and weights and the spatial profile of the voidage be very accurately predicted. But this condition cannot always be achieved. As observed, $\mathcal{D}_{x,2}$ is dictated by the system and the operating conditions, whilst the order of magnitude of $\mathcal{D}_n$ is dictated by the computational times that one may accept. In the problem investigated here, the two values differ only by one order of magnitude, and therefore this ideal condition is not found. The results, however, show that the value $\mathcal{D}_x = 5.0 \times 10^{-3}$ m^2/s is a good compromise, because it does not corrupt the void fraction profiles while predicting reasonably well the values of the quadrature nodes and weights.

7. Simulations of segregating fluidized powders

After having estimated a reasonable value for $\mathcal{D}_x$, we went on to simulate the dynamics of the system under operating conditions

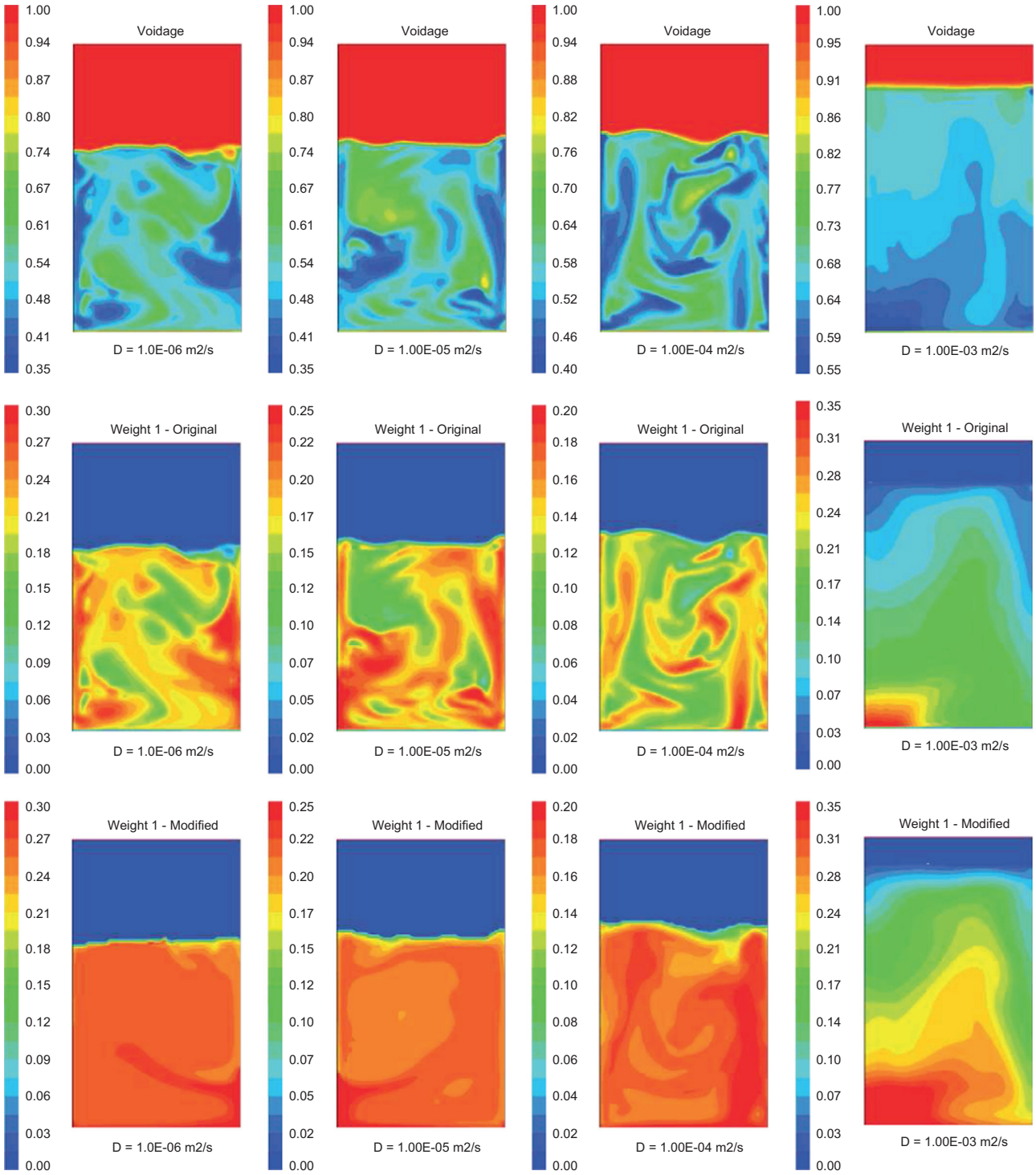

Fig. 2. Profiles of the void fraction and of the unmodified and modified quadrature weights at pseudosteady-state conditions in the mixed fluidized bed for different values of the diffusion coefficient $\mathcal{D}_x$. The values of the modified quadrature weights refer to a spatially uniform void fraction equal to 0.400.

that should promote segregation. The system modeled is the same as before (refer again to Section 2), but this time we considered superficial fluid velocities of 0.10 and 0.05 m/s, values that should be sufficiently low to induce segregation.

Before presenting the experimental and numerical results, let us briefly describe how we derived them. As formerly mentioned, initially the system is made up of two superposed uniform powders, the powder with particles of lower mean size being placed underneath the other; this ensured that smaller particles were free to migrate towards the bed top, while larger particles toward the bed bottom. In the experiments, after loading the powders, we fed nitrogen at a constant superficial velocity for about 10 min, a time sufficient to attain pseudostationary conditions. After, we carried out the so-called bed freeze test: we abruptly cut off the gas supply to the bed and vented the gas present in the windbox of the vessel to the atmosphere. We then split the fixed bed resting on the distributor plate into five layers of equal height, collected each layer using a sampling probe, and finally sieved them to obtain their PSDs. From these we calculated first the VDF integer moments of interest and then the quadrature

nodes and weights; details about the experimental apparatus and procedure can be found in Mazzei et al. (2010a). The simulations did not exactly mirror the experiments, for we did not simulate the bed collapse. Doing so is unnecessary and might even be detrimental. Numerically, we can easily determine the VDFs when the bed is still fluidized; to this end, we just have to divide the bed in layers and from the numerical profiles of the VDF moments determine their average values in each layer and then the corresponding average values of the quadrature nodes and weights. There is also another reason for which it is preferable to calculate the VDFs while the bed is still fluidized. As Mazzei et al. (2010b) reported, freezing the bed is detrimental, since while the experimental collapse is instantaneous, and the bed conserves its particle size distribution, the simulated collapse is not and so allows the bigger particles to sink toward the bottom of the vessel, altering the original segregation profile. For these reasons, we simulated solely the fluidization phase, determining the VDFs in pseudostationary conditions. The experimental and numerical values of the quadrature nodes and weights are reported in Figs. 3 and 4 for the superficial fluid velocities of 0.10 and 0.05 m/s, respectively. Good match between the values implies good prediction of the particle size distributions. This is how we tested the quadrature-based model.

Let us first describe the numerical results obtained for the superficial fluid velocity of 0.10 m/s. Fig. 3 shows the pseudosteady-state values, predicted numerically and measured experimentally, of the quadrature nodes and weights. As we can see, the numerical values of the quadrature nodes are spatially uniform, that is, they are the same in each bed layer, while the experimental values are nearly the same in all layers except the bottom one, where they are slightly larger. Overall, the agreement is reasonably good; the mean percent error is about 20% for s_1 and 10% for s_2 (from this point of view the figure might mislead, insofar as the node s_1 appears to be better predicted; but the smaller absolute error visible in the figure turns into the larger percent error reported above because s_1 is the smaller node). We should note that setting $\mathcal{D}_x = 0$ would have resulted (in the presence of numerical diffusion, which allows micromixing to take place and nodes to homogenize) in mean percent errors of about 70% for s_1 and 35% for s_2.

As said, the numerical profiles of the nodes are spatially uniform. This always occurs for very long times, provided size-changing phenomena such as growth, aggregation and breakage are absent. Initially the nodes are not uniform and therefore the diffusive and source terms present in Eq. (3.8) are not zero, making the nodes vary along the pathlines and the material derivatives of the nodes differ from zero. For long times, however, the node gradients tend to vanish, thus driving the material derivatives to zero. This occurs regardless of the superficial fluid velocity employed. Consequently, if segregation takes place, this is reflected by nonuniform spatial profiles of the quadrature weights – in particular of the functions $\phi_r^\star(\boldsymbol{x}, t)$ introduced in the previous section and reported in Fig. 3, whose values for the smaller nodes we expect to be smaller in the lower bed region. This is what we do observe in the figure: as we move downwards through the bed the weight $\phi_1^\star(\boldsymbol{x}, t)$, which refers to the smaller node, progressively decreases, revealing that segregation has taken place. Also in this case the overall agreement is reasonably good, the mean percent error being less than 10%. Segregation, as we can see, is only slightly overpredicted in the top two layers of the bed.

We now go on to discuss the results obtained for the superficial fluid velocity of 0.05 m/s. Fig. 4 shows the pseudosteady-state values, predicted numerically and measured experimentally, of the quadrature nodes and weights. As we expected, the numerical values of the quadrature nodes are spatially uniform and equal to those found for the larger superficial velocity; the experimental values, conversely, vary from layer to layer, but not substantially. Overall, the agreement is again reasonably good; the mean percent error is also in this case about 20% for s_1 and 10% for s_2.

The spatial profiles of the quadrature weights predicted numerically are not uniform, which confirms that segregation has taken place; however, the degree of segregation is far less than that observed experimentally. The experimental profile shows that in the three lower bed layers the quadrature weights are nearly uniform, the larger quadrature node being dominant; in the two upper bed layers the quadrature weights are also nearly uniform, but here the smaller node is dominant. Thus, the bed is almost split into two regions. The segregation profile predicted numerically is smoother, $\phi_1^\star(\boldsymbol{x}, t)$ changing gradually along the bed axis; the mean percent error in this case is about 35%. This result should be improved.

The question that we asked ourselves is whether the inaccuracy of the quadrature weight profile is related to the DQMOM model. We believe that it is not. As mentioned, if size-changing phenomena such as growth, aggregation and breakage are all absent, the quadrature nodes eventually become uniform in space; then, the evolution equations of the quadrature weights become as follows:

$$\partial_t \phi_r = -\partial_{\boldsymbol{x}} \cdot \phi_r \boldsymbol{v}_r + \partial_{\boldsymbol{x}} \cdot \mathcal{D}_x \partial_{\boldsymbol{x}} \phi_r \tag{7.1}$$

because, as Eqs. (3.7) indicate, the source terms vanish. This reveals that, when pseudosteady-state conditions have been achieved, the quadrature weights become conservative quantities. In Section 3 we explained why the diffusive term above survives; this, however, is comparatively smaller than the convective term, and we may regard it as a correction to the latter. In these conditions, that is, *once the source terms have vanished*, neglecting the diffusion term should not alter significantly the numerical predictions of the spatial profiles of the functions $\phi_r(\boldsymbol{x}, t)$. If we

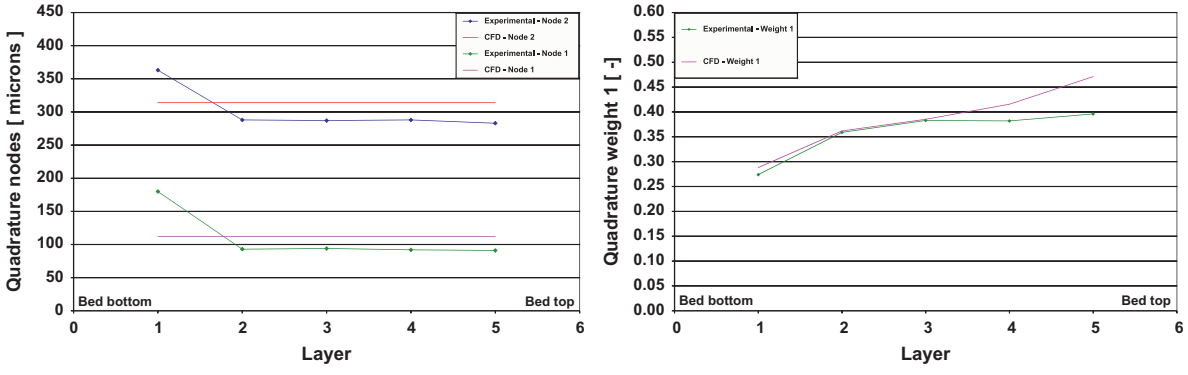

Fig. 3. Profiles of the VDF quadrature nodes and of the weight relative to the smaller node at pseudosteady-state conditions in the segregated fluidized bed for $u = 0.10$ m/s and $\mathcal{D}_x = 0.005$ m^2/s. The values of the (modified) quadrature weights refer to a spatially uniform void fraction equal to 0.400.

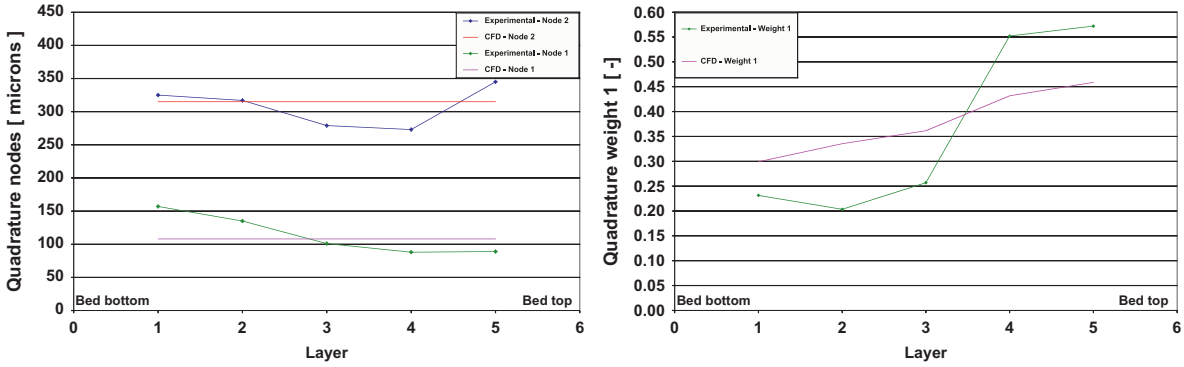

Fig. 4. Profiles of the VDF quadrature nodes and of the weight relative to the smaller node at pseudosteady-state conditions in the segregated fluidized bed for $u = 0.05$ m/s and $\mathcal{D}_x = 0.005$ m^2/s. The values of the (modified) quadrature weights refer to a spatially uniform void fraction equal to 0.400.

neglect this contribution, Eq. (7.1) reduces to the customary continuity equation solved for solid phases by Eulerian CFD codes, and in particular by Fluent. Of course, because of numerical diffusion, the equation really solved is as follows:

$$\partial_t \phi_r = -\partial_{\boldsymbol{x}} \cdot \phi_r \boldsymbol{v}_r + \partial_{\boldsymbol{x}} \cdot \mathcal{D}_n \partial_{\boldsymbol{x}} \phi_r \tag{7.2}$$

where the coefficient $\mathcal{D}_x$ has been replaced by $\mathcal{D}_n$. But the diffusive term introduced by numerical diffusion is also comparatively smaller than the convective term, provided the computational grid is sufficiently fine. So, if the pseudosteady-state spatial profiles of the quadrature weights are inaccurately predicted, there can only be two reasons: either we employed a value for $\mathcal{D}_x$ that does not render the diffusion term comparatively smaller than the convective term, or the dynamical equations used in the code, based on the kinetic theory of granular gases, lack sufficient accuracy to well match the profiles experimentally measured. The equations that we adopted are a default option in Fluent, are based on the work of Lun et al. (1984), Syamlal (1987) and Gidaspow (1994), and are customary choices in multiphase modeling of granular flows.

To investigate this aspect further, we simulated the behavior of a binary powder made up of particles with sizes equal to the pseudosteady-state values of the quadrature nodes predicted by the DQMOM equations and fluidized at a superficial velocity of 0.05 m/s. In doing so, we used the same dynamical equations previously used, replacing the evolution equations for the quadrature weights with the customary continuity equations obtained from Eqs. (7.1) by setting $\mathcal{D}_x = 0$. The computational grid adopted was the same as before, but this time to discretize in space we used a second-order numerical scheme; doing so was possible because now we were using the default equations of the code, and thus we did not encounter the same stability problems that the DQMOM equations posed. The reason for adopting a second-order numerical scheme was minimizing as much as possible the numerical diffusion. Fig. 5 reports the new and old profiles of the function $\phi_1^\star(\boldsymbol{x}, t)$, that is, of the modified quadrature weight. As we see, the two profiles do not differ significantly, both predicting a gradual change of the weight along the bed axis. Segregation is slightly more pronounced in the new profile, but the qualitative difference between the numerical and experimental results remains: the numerical results do not predict that the bed essentially separates into two regions. For the new profile, the mean percent error is nearly the same as before, being about 35%.

This is what we conclude from this investigation. The DQMOM model employed correctly predicts the evolution of the quadrature nodes, as long as the diffusion coefficient $\mathcal{D}_x$ is correctly estimated (we reported how to do this at the end of Section 4). Setting this coefficient to zero, or to a low value compared to that of the numerical diffusion coefficient, leads to wrong results. The

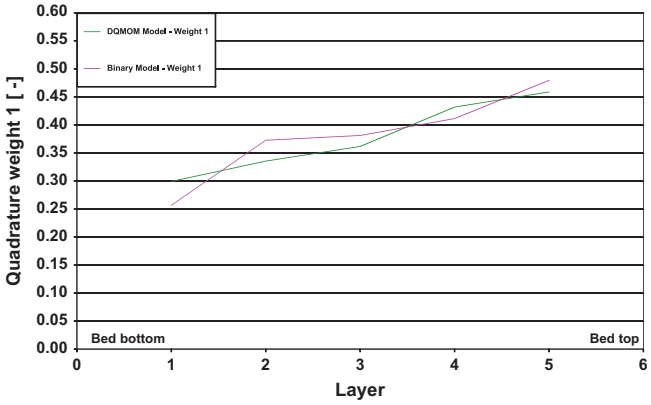

Fig. 5. Profiles of the VDF quadrature weight relative to the smaller node at pseudosteadystate conditions in the segregated fluidized bed for $u = 0.05$ m/s and $\mathcal{D}_x = 0.005$ m^2/s calculated using DQMOM and a normal binary model. The values of the (modified) quadrature weights refer to a spatially uniform void fraction equal to 0.400.

quadrature weights, conversely, are not very well predicted, because in the simulation the segregation profile is more gradual than what observed experimentally. This limitation, however, is not related to the DQMOM evolution equations, but to those adopted to model the velocity and granular temperature fields associated with each quadrature class. The constitutive relations used in such equations give results that are qualitatively correct, but quantitatively inaccurate. It is necessary, therefore, that these relations be improved.

8. Conclusions

In this work we tested the new DQMOM model recently developed by Mazzei (2011), using it to model the segregation dynamics of dense polydisperse fluidized powders of inert particles. The novelty of the model is that it allows mixing at the particle length scale, thereby allowing the nodes of the quadrature formula to vary along the pathlines. Previous models reported in the literature predicted, for similar systems, that nodes are constant along the pathlines, which implies that nonuniform powders cannot homogenize in space. To permit micromixing, the model introduces in the evolution equations of the quadrature weighted nodes and weights a diffusive term; which value to assign to the diffusion coefficient present in this term had not been specified in Mazzei (2011). In this work we first of all discussed this problem, and then we tested the ideas advanced numerically, simulating the mixing of powders initially segregated

using different values for the diffusivity. This sensitivity analysis gave a positive outcome, confirming that the diffusive term is essential and that the diffusivity $\mathcal{D}_x$ must be larger than the coefficient $\mathcal{D}_n$ representing the numerical diffusion generated by the computational code. Specifically, the analysis showed that $\mathcal{D}_x \sim 10\mathcal{D}_n$ suffices for good accuracy. Smaller values render the source and diffusion terms present in the evolution equations of the quadrature weighted nodes and weights imbalanced, leading to grossly wrong predictions, a condition found, in particular, when one sets $\mathcal{D}_x = 0$, that is, when one neglects the diffusion term altogether. We also identified an upper bound value for the diffusion coefficient, denoted as $\mathcal{D}_{x,2}$. Values of $\mathcal{D}_x$ smaller than this value permit convection to dominate over diffusion, a necessary condition for quadrature formulae that well approximate the moments external to the set tracked by the DQMOM model, in particular the convective fluxes. Larger values of $\mathcal{D}_x$ would allow diffusion to erase the gradients generated by convection, thereby flattening the spatial profiles of the quadrature weighted nodes and weights, of the moments and of the fluid voidage. The numerical results also confirmed such aspects. After having identified, in light of this analysis, a suitable value for $\mathcal{D}_x$, we went on to simulate the dynamics of the system under conditions which promote segregation, validating the results of the simulations experimentally. We considered two superficial fluid velocities, 0.10 and 0.05 m/s. For the first we found a good match between simulations and experiments, whilst for the second only the quadrature nodes were reasonably well predicted; in particular, the simulations underpredicted the degree of segregation in the bed. Qualitatively the results are correct, but quantitatively the accuracy needs improvement. A test that we conducted, however, suggests that the issue is not related to the DQMOM model but has more to do with the constitutive relations used to model the velocity and granular temperature fields associated with each quadrature class. These relations need to be improved.

References

Aamir, E., Nagy, Z.K., Rielly, C.D., Kleinert, T., Judat, B., 2009. Combined quadrature method of moments and method of characteristics approach for efficient solution of population balance models for dynamic modeling and crystal size distribution control of crystallization processes. Ind. Eng. Chem. Res. 48, 8575–8584.

Akhiezer, N.I., 1965. The Classical Moment Problem and Some Related Questions in Analysis (N. Kemmer, Trans.). Hafner Publishing Co.

Dorao, C.A., Jakobsen, H.A., 2006. Numerical calculation of the moments of the population balance equation. J. Comput. Appl. Math. 196, 619–633.

Drew, D.A., Passman, S.L., 1998. Theory of Multicomponent Fluids. Applied Mathematical Sciences, Springer.

Fan, R., Fox, R.O., 2008. Segregation in polydisperse fluidized beds: validation of a multi-fluid model. Chem. Eng. Sci. 63, 272–285.

Fan, L.S., Zhu, C., 1998. Principles of Gas-Solid Flows. Cambridge University Press.

Fan, R., Marchisio, D.L., Fox, R.O., 2004. Application of the direct quadrature method of moments to polydisperse gas-solid fluidized beds. Powder Technol. 139, 7–20.

Ferziger, J.H., Peric, M., 2002. Computational Methods for Fluid Dynamics. Springer.

Fox, R.O., 2012. Large-eddy-simulation tools for multiphase flows. Annu. Rev. Fluid Mech. 44, 47–76.

Fox, R.O., Vedula, P., 2010. Quadrature-based moment model for moderately dense polydisperse gas-particle flows. Ind. Eng. Chem. Res. 49, 5174–5187.

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

Gimbun, J., Nagy, Z.K., Rielly, C.D., 2009. Simultaneous quadrature method of moments for the solution of population balance equations using a differential algebraic equation framework. Ind. Eng. Chem. Res. 48, 7798–7812.

Grosch, R., Briesen, H., Marquardt, W., Wulkow, M., 2007. Generalization and numerical investigation of QMOM. AIChE J. 53, 207–227.

Jackson, R., 2000. The Dynamics of Fluidized Particles. Cambridge Monographs on Mechanics. Cambridge University Press.

Lettieri, P., Cammarata, L., Michale, G., Yates, J.G., 2003. CFD simulations of gas-fluidized beds using alternative Eulerian–Eulerian modelling approaches. Int. J. Chem. React. Eng. 1, 1–21.

Lu, H., Gidaspow, D., 2003. Hydrodynamics of binary fluidization in a riser: CFD simulation using two granular temperatures. Chem. Eng. Sci. 58, 3777–3792.

Lun, C.K.K., Savage, S.B., Jeffrey, D.J., Chepurniy, N., 1984. Kinetic theories for granular flow: inelastic particles in Couette flow and slightly inelastic particles in a general flow field. J. Fluid Mech. 140, 223.

Marchisio, D.L., Fox, R.O., 2007. Multiphase Reacting Flows: Modelling and Simulation CISM Courses and Lectures No. 492. International Centre for Mechanical Sciences. Springer.

Mazzei, L., 2011. Limitations of quadrature-based moment methods for modeling inhomogeneous polydisperse fluidized powders. Chem. Eng. Sci. 66, 3628–3640.

Mazzei, L., Lettieri, P., 2008. CFD simulations of expanding/contracting homogeneous fluidized beds and their transition to bubbling. Chem. Eng. Sci. 63, 5831–5847.

Mazzei, L., Marchisio, D.L., Lettieri, P., 2010a. Direct quadrature method of moments for the mixing of inert polydisperse fluidized powders and the role of numerical diffusion. Ind. Eng. Chem. Res. 49, 5141–5152.

Mazzei, L., Casillo, A., Lettieri, P., Salatino, P., 2010b. CFD simulations of segregating fluidized bidisperse mixtures of particles differing in size. Chem. Eng. J. 156, 432–445.

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

Petitti, M., Nasuti, A., Marchisio, D.L., Vanni, M., Baldi, G., Mancini, N., Podenzani, F., 2010. Bubble size distribution modeling in stirred gas-liquid reactors with QMOM augmented by a new correction algorithm. AIChE J. 56, 36–53.

Qamar, S., Mukhtar, S., Ali, Q., Seidel-Morgenstern, A., 2011. A Gaussian quadrature method for solving batch crystallization models. AIChE J. 57, 149–159.

Randolph, A.D., Larson, M.A., 1971. Theory of Particulate Processes. Academic Press.

Simmonds, J.G., Mann, J.E., 1998. A First Look at Perturbation Theory. Dover Publications Inc.

Shohat, J.A., Tamarkin, J.D., 1943. The Problem of Moments. American Mathematical Society.

Syamlal, M., 1987. The Particle-Particle Drag Term in a Multiparticle Model of Fluidization. National Technical Information Service. DOE/MC/21353-2373, NTIS/DE87006500.

Wright, D.L., 2007. Numerical advection of moments of the particle size distribution in Eulerian models. J. Aerosol Sci. 38, 352–369.