

An Euler–Lagrange strategy for simulating particle-laden flows



Jesse Capecelatro*, Olivier Desjardins

Sibley School of Mechanical and Aerospace Engineering, Cornell University, Ithaca, NY 14853, USA

ARTICLE INFO

Article history:

Received 6 September 2012

Received in revised form 16 November 2012

Accepted 7 December 2012

Available online 29 December 2012

Keywords:

Euler–Lagrange

Particle tracking

Fluidized bed

Spout fluidization

Particle collisions

Discrete element model

ABSTRACT

In this work, a strategy capable of simulating polydisperse flows in complex geometries is employed where the fluid transport equations are solved in an Eulerian framework and the dispersed phase is represented as Lagrangian particles. Volume filtered equations for the carrier phase are derived in detail for variable density flows, and all unclosed terms are discussed. Special care is given to the interphase coupling terms that arise, in order to ensure that they are implemented consistently and that they converge under mesh refinement. This provides the flexibility of using cell sizes that are smaller than the particle diameter if necessary. Particle collisions are handled using a soft-sphere model that has been modified for parallel efficiency. Simulations are carried out for a number of laboratory-scale configurations, showing excellent agreement with experiments.

© 2012 Elsevier Inc. All rights reserved.

1. Introduction

Suspensions of solid particles in a carrier fluid are common in many engineering applications, including sediment transport, gas and liquid fluidized bed reactors, and turbulent risers. Such flows are ubiquitous during biomass thermochemical conversion, and with recent interests in the production of renewable fuels, great effort has been put forward in understanding and predicting such processes. The presence of the dispersed phase leads to a wide range of length and time scales that need to be considered. Consequently, a relatively large number of modeling approaches have been developed (e.g., [1–6]). Whenever possible, such models should be based on well-defined physical parameters from the underlying microscale flow physics, and not derived from interpretations of a specific flow regime [1]. With the rapidly growing computational resources, new opportunities arise to develop more reliable and predictive computational fluid dynamics (CFD) models for exploring the complex behavior of dispersed multiphase flows.

In this work, we consider a dispersed phase that consists of smooth, spherical particles. A particle p is characterized by its position $\mathbf{x}_p(t)$, its volume $\mathcal{V}_p(t)$, and its velocity $\mathbf{u}_p(t)$ at time t . Although the methods described in this paper assume solid particles, most of the formulation could be extended to droplets and bubbles as well. For an ensemble of n_p particles, it is convenient to define a phase space $\mathbf{x} = (\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_{n_p}, \mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_{n_p}, \mathcal{V}_1, \mathcal{V}_2, \dots, \mathcal{V}_{n_p})$. The phase space contains all possible values for each degree of freedom, and each particle is represented by a point in phase space [7]. The stochastic nature of the particle phase is described by a Liouville equation for the multiparticle density function, $f(t, \mathbf{x}, \mathbf{x})$, given by

$$\frac{\partial f}{\partial t} + \mathbf{u}_p \cdot \frac{\partial f}{\partial \mathbf{x}_p} + \frac{\partial}{\partial \mathbf{u}_p} \left(f \frac{\mathbf{F}_p}{m_p} \right) = \mathbb{C}, \quad (1)$$

where \mathbf{F}_p represents external forces acting on each particle (e.g., gravity, drag, added mass, etc.), m_p is the mass of each particle, and \mathbb{C} is the collision operator. Note that Eq. (1) implies a summation over all particles. Due to the large number of

* Corresponding author. Tel.: +1 8456612336.

E-mail address: jsc359@cornell.edu (J. Capecelatro).

independent variables associated with f in a three-dimensional flow, a direct solution is usually intractable, hence various mathematical approximations are made to arrive at models capable of simulating systems of interest. In a recent review paper, Fox [1] summarizes such modeling strategies, in addition to providing a systematic approach for developing large-eddy simulation (LES) tools starting from microscale physics. Another noteworthy review on that topic is due to Balachandrar and Eaton [2], where the current state-of-the-art experimental and computational techniques for turbulent dilute dispersed flows are discussed.

Two common approaches for modeling dispersed phase flows include Euler–Euler (EE) and Euler–Lagrange (EL) methods. Note that Pai and Subramaniam [8] established consistency relationships between the two representations in the framework of a probability density function formalism, presented advantages and limitations of each approach, and identified unclosed terms that arise. EE representations reduce Eq. (1) to the one-particle density function appearing in the kinetic equation [9–11]. In general, the kinetic description is unclosed and requires mesoscale models for the collision operator. EE methods approximate the kinetic equation by considering a set of moments of f , and solve both the fluid and particle-phase on a common Eulerian grid. In the small Knudsen number (highly collisional) limit with an underlying assumption that the flow is nearly at equilibrium, the particle density function is close to Maxwellian and a Chapman–Enskog expansion can be used to derive a two-fluid model (TFM) using ensemble or volume averaging [12–15]. This approach leads to particle phase transport equations that closely resemble the Navier–Stokes equations using moment closures obtained from kinetic theory. TFM greatly reduces the cost of simulating large-scale systems as it does not require interactions at the particle scale to be resolved. As a result, TFM has been used in a large number of studies, for example in dense fluidized bed reactors [16,17] and riser flows in vertical channels [18–22]. However, in the dilute limit, the constitutive models fail to correctly describe important features of the flow. Desjardins et al. [23] showed the challenges EE formulations face for finite Knudsen number and non-equilibrium flows, for which particle trajectory crossings play an important role and higher moments of the particle number density must be considered to yield an accurate result. Several studies proposed solutions to overcome the limitations of standard TFM. Simonin [24] used a moment method based on the Grad [25] approach to study non-equilibrium dilute gas-particle flow behavior. Desjardins et al. [23] proposed to approximate the kinetic equation using a two-node quadrature approach based on the quadrature method of moments [26], leading to a method capable of handling highly non-equilibrium, finite-Stokes number flows, including impinging jets, jet crossing, and particle rebound off walls, which previously could not be treated accurately with EE formulations. Following this work, Passalacqua et al. [27] implemented a third-order quadrature-based moment method coupled with a fluid solver to deal with dense flows. They demonstrated the importance of considering the effect of the local particle Knudsen number, which they found to be large if solid packing is small or the particle Mach number is large, which is the case in many dispersed two-phase flows. Yuan and Fox [28] developed a conditional quadrature method of moments (CQMOM) based on 1-D adaptive quadrature of conditional velocity moments to improve the accuracy of the quadrature-based moment method. They applied CQMOM to a range of problems involving particle trajectory crossing and collisions. While EE methods have been widely successful, in particular due to their computational efficiency when solving systems containing a large number of particles, statistical and physical closures continue to pose a challenge.

Euler–Lagrange (EL) strategies provide an alternative framework with the potential of simpler closures. EL methods represent the dispersed phase in a Lagrangian manner by discretizing the multiparticle density function introduced earlier into a number of individually tracked stochastic particles. As expected in Discrete Simulation Monte Carlo (DSMC) approaches, EL methods can yield very accurate solutions, though they can require a large number of stochastic particles to control statistical errors [29]. A few examples of applications of EL methods within the literature include simulations of impinging stream reactors [30], cluster formation of gas–solid flows in vertical channels [31,32], and dense fluidized beds [32,33]. It is typical in EL simulations to consider as many particles as there are physical particles, or a smaller number of computational particles in order to reduce simulation cost. Garg et al. [34] demonstrated that if the spatial distribution of particles becomes highly non-uniform, regions with fewer particles have higher statistical error, which might prevent numerical convergence of the interphase momentum transfer term. They proposed to dynamically ensure that the number of stochastic particles per cell remains nearly constant and showed much improved convergence. However, this strategy remains to be extended to colliding particles. Similarly, Salman and Soteriou [35] demonstrated limitations of the EL approach in their study of evaporating sprays. They reviewed typical interpolation schemes used to couple the two phases and showed that such procedures do not converge under mesh refinement. In fact, most existing EL coupling schemes require the Eulerian mesh to be an order of magnitude larger than the diameter of the largest particle, potentially preventing the capture of dynamically important mesoscale processes. This can be problematic when dealing with non-uniform meshes or a dispersed phase with a wide size distribution.

Despite these limitations, a single realization of DSMC accounts discretely for all individual particle processes such as drag and collisions, and can therefore provide much needed insights in the physics of particle-laden flows, and even help guide the development of improved EE closures. This level of modeling is referred in the literature as discrete element models (DEM) [36]. The coupling of DEM with a finite-volume description of the gas phase based on the Navier–Stokes equations was first reported by Tsuji et al. [37] and Hoomans [38] using a soft-sphere [36] and hard-sphere [39] collision models, respectively. Deen [4] provided a detailed review on DEM for the study of gas–solid flows in fluidized bed reactors.

Coupling DEM with the spatially-filtered equations of motion for the gas phase is the strategy that we follow in this paper. In the next section, we start from the point wise variable density Navier–Stokes equations, coupled to Newton's second law for the particles, and derive volume filtered continuity, momentum, and scalar transport equations appropriate for the EL

simulations. All unclosed terms and assumptions are given explicitly and discussed. Interphase coupling is discussed in detail in Section 3, and a dual filter approach is introduced to guarantee convergence under mesh refinement. Coupling with a conservative immersed boundary method [40] to account for realistic geometries is described in this section as well. A soft sphere model [36] accounts for particle collisions. This model has been modified compared to other versions in the literature in order to recover the correct close-packing limit, as well as to account for inter-particle friction and rotation in a scalable manner. The collision model and its parallel implementation are presented in Section 4. Section 5 summarizes the implementation of the numerical approach and presents parallel performance. Finally, in Section 6, several numerical tests are conducted of lab-scale particle-laden flows for validation purposes. In this work, all simulations are performed using NGA, an arbitrarily high order multi-physics CFD code [41]. It is shown that the overall strategy is capable of producing simulations of realistic, three-dimensional polydisperse flows on high core counts.

2. Mathematical description

This section presents the equations employed to describe the flow of solid particles in a low Mach number, variable density carrier fluid. We begin by presenting the point wise equations of motion for each phase. These equations require resolving the flow around each individual particle, which is excessively expensive for engineering systems of interest. Consequently, we derive a set of volume filtered equations for a variable density fluid, which can be solved on a scale larger than the particle diameter. The particles are solved in a Lagrangian framework described by Newton's laws of motion. The two phases are fully coupled through momentum exchange terms. Unclosed terms that arise from the filtering approach are presented and discussed.

2.1. Point wise description

The carrier fluid is described by the Navier–Stokes equations for a low Mach number, variable density flow. Continuity is given by

$$\frac{\partial \rho_f}{\partial t} + \nabla \cdot (\rho_f \mathbf{u}_f) = 0, \quad (2)$$

where ρ_f and \mathbf{u}_f are the point wise fluid density and velocity, respectively. Conservation of momentum is expressed as

$$\frac{\partial}{\partial t} (\rho_f \mathbf{u}_f) + \nabla \cdot (\rho_f \mathbf{u}_f \otimes \mathbf{u}_f) = \nabla \cdot \boldsymbol{\tau} + \rho_f \mathbf{g}, \quad (3)$$

where \mathbf{g} is the acceleration due to gravity, and $\boldsymbol{\tau}$ is the point wise value of the fluid stress tensor, given by

$$\boldsymbol{\tau} = -p\mathbf{I} + \mu \left[\nabla \mathbf{u}_f + \nabla \mathbf{u}_f^T - \frac{2}{3} (\nabla \cdot \mathbf{u}_f) \mathbf{I} \right]. \quad (4)$$

The hydrodynamic pressure and dynamic viscosity coefficient are given by p and μ , respectively. \mathbf{I} is the identity tensor. Interactions with the solid phase are incorporated by imposing no-slip and no-penetration boundary conditions at the surface of each particle.

The displacement of an individual particle p is calculated using Newton's second law of motion,

$$m_p \frac{d\mathbf{u}_p}{dt} = \mathbf{f}_p^{\text{inter}} + \mathbf{F}_p^{\text{col}} + m_p \mathbf{g}. \quad (5)$$

The particle mass is defined by $m_p = \pi \rho_p d_p^3 / 6$, where ρ_p and d_p are the particle density and diameter, respectively. $\mathbf{F}_p^{\text{col}}$ is the particle collision force, which will be described in detail in Section 4. The force $\mathbf{f}_p^{\text{inter}}$ exerted on a single particle p by the surrounding fluid is given by the integral of the stress tensor over the surface of the particle,

$$\mathbf{f}_p^{\text{inter}} = \int_{S_p} \boldsymbol{\tau} \cdot \mathbf{n} dS, \quad (6)$$

where \mathbf{n} is the outward unit normal vector of the particle surface S_p . This term is discussed further in Section 2.5.

The angular momentum of the particle, ω_p , is computed by taking the cross product of the position vector outward from the particle with Eq. (5), giving

$$I_p \frac{d\omega_p}{dt} = \int_{S_p} \frac{d_p}{2} \mathbf{n} \times (\boldsymbol{\tau} \cdot \mathbf{n}) d\mathbf{y} + \sum_j \frac{d_p}{2} \mathbf{n} \times \mathbf{f}_{t,j \rightarrow p}^{\text{col}}, \quad (7)$$

where $\mathbf{f}_{t,j \rightarrow p}^{\text{col}}$ is the tangential component of the collision force of particle j acting on particle p . This term will be made explicit later. I_p is the particle's moment of inertia, given for a sphere by

$$I_p = \frac{m_p d_p^2}{10}. \quad (8)$$

2.2. Volume filtering operators

In order to account for the effect of particles without requiring to resolve the fluid-phase equations on the scale of the particle surface, the point wise fluid equations can be split into microscale processes, i.e., processes that take place on the scale of a particle and below, and meso/macro-scale processes, i.e., processes that take place on a scale much larger than the particle size. Anderson and Jackson [42] detail such an approach for particle-laden flows by applying a local volume filtering operator to the Navier–Stokes equations, thereby replacing the point variables (fluid velocity, pressure, etc.) by smoother, locally filtered fields. This local volume filtering strategy is employed in this work. Another strategy based on ensemble averaging is proposed by Zhang and Prosperetti [43], which leads to comparatively similar equations.

We begin by defining a filtering kernel g with a characteristic length δ_f , such that $g(r) > 0$ decreases monotonically with increasing r , and is normalized such that its integral over the entire physical space is unity. The local voidage at a point \mathbf{x} and time t is defined as

$$\varepsilon_f(\mathbf{x}, t) = \int_{\mathcal{V}_f} g(|\mathbf{x} - \mathbf{y}|) d\mathbf{y}, \quad (9)$$

where \mathcal{V}_f indicates that the integral is taken over all points \mathbf{y} occupied by the fluid. Although this quantity is a function of both space and time, we will simply use ε_f throughout to represent the local fluid volume fraction. Similarly, the particle volume fraction can be computed by taking the volume integral of the filtering kernel over all space occupied by the solid phase. Since g varies little within a single particle, it can be taken outside of the integral and the particle volume fraction can be expressed as

$$\varepsilon_p(\mathbf{x}, t) = 1 - \varepsilon_f \approx \sum_{i=1}^{n_p} g(|\mathbf{x} - \mathbf{x}_i|) \mathcal{V}_i, \quad (10)$$

where \mathcal{V}_i , as previously mentioned, is the volume of particle i , and n_p is the number of particles.

Let $\mathbf{a}(\mathbf{x}, t)$ be any point property of the fluid of arbitrary tensorial order, and $\mathbf{a}'(\mathbf{x}, t)$ denote its residual field, such that

$$\mathbf{a}(\mathbf{x}, t) = \bar{\mathbf{a}}(\mathbf{x}, t) + \mathbf{a}'(\mathbf{x}, t), \quad (11)$$

where the volume filtered field $\bar{\mathbf{a}}(\mathbf{x}, t)$ is computed by taking the convolution product with the filtering kernel g , giving

$$\varepsilon_f \bar{\mathbf{a}}(\mathbf{x}, t) = \int_{\mathcal{V}_f} \mathbf{a}(\mathbf{y}, t) g(|\mathbf{x} - \mathbf{y}|) d\mathbf{y}. \quad (12)$$

Due to surface contributions of the solid particles, there is no commutation between filtering and differentiation. Consequently, filtering derivatives of point variables will lead to additional terms in the form of integrals of the point wise quantity about the surface of the particle. Volume filtering the gradient, divergence, and time derivative of a point property of the fluid are respectively given as

$$\int_{\mathcal{V}_f} \nabla \mathbf{a}(\mathbf{y}, t) g(|\mathbf{x} - \mathbf{y}|) d\mathbf{y} = \nabla (\varepsilon_f \bar{\mathbf{a}}(\mathbf{x}, t)) - \sum_{i=1}^{n_p} \int_{S_i} \mathbf{n} \otimes \mathbf{a}(\mathbf{y}, t) g(|\mathbf{x} - \mathbf{y}|) d\mathbf{y}, \quad (13)$$

$$\int_{\mathcal{V}_f} \nabla \cdot \mathbf{a}(\mathbf{y}, t) g(|\mathbf{x} - \mathbf{y}|) d\mathbf{y} = \nabla \cdot (\varepsilon_f \bar{\mathbf{a}}(\mathbf{x}, t)) - \sum_{i=1}^{n_p} \int_{S_i} \mathbf{n} \cdot \mathbf{a}(\mathbf{y}, t) g(|\mathbf{x} - \mathbf{y}|) d\mathbf{y}, \quad (14)$$

and

$$\int_{\mathcal{V}_f} \frac{\partial \mathbf{a}(\mathbf{y}, t)}{\partial t} g(|\mathbf{x} - \mathbf{y}|) d\mathbf{y} = \frac{\partial}{\partial t} (\varepsilon_f \bar{\mathbf{a}}(\mathbf{x}, t)) + \sum_{i=1}^{n_p} \int_{S_i} \mathbf{n} \cdot \mathbf{u}_i \mathbf{a}(\mathbf{y}, t) g(|\mathbf{x} - \mathbf{y}|) d\mathbf{y}, \quad (15)$$

where S_i represents the surface of particle i . A detailed derivation can be found in the work of Anderson and Jackson [42]. In order to reach these expressions it is assumed that the shortest distance from the location of the point property to the surface bounding the fluid is significantly larger than the radius of g . Treatment for solving filtered quantities near walls will be discussed in Section 3.2. For simplicity, all fluid properties in the remainder of the paper will be assumed to be a function of space and time and the parentheses will be left out.

For flows with density changes, such as those found in chemically reacting systems, it is often convenient to introduce a Favre filtering operation (see for example [44]). The density-weighted volume filtered quantity $\tilde{\mathbf{a}}$ is given by

$$\tilde{\mathbf{a}} = \frac{\varepsilon_f \bar{\rho_f \mathbf{a}}}{\varepsilon_f \bar{\rho_f}}, \quad (16)$$

where $\bar{\rho_f}$ is the volume filtered fluid density. Any fluid property \mathbf{a} may be split into its density-weighted and residual components $\mathbf{a} = \tilde{\mathbf{a}} + \mathbf{a}''$.

The definitions above are most useful when constructing the equations of motion if the filtered variables are invariant with the chosen filtering kernel g and the filter length δ_f . Spatial variations in point properties manifest from scales on the order of the particle diameter and interparticle spacing (from boundary layer effects and collisions) to scales comparable to the dimensions of the complete system. Choosing δ_f such that the local filtered variables are unaffected by the filtering process is not trivial. Instead, we allow the filter length to be small enough for the mesoscale flow features to be unaffected, while maintaining a filter length large enough for microscale processes such as drag to be appropriately modeled by classical models. In the context of particle-laden flows, δ_f proportional to several particle diameters meets this criterion.

2.3. Volume filtered equations of motion

Filtering Eq. (2) by multiplying by $g(|\mathbf{x} - \mathbf{y}|)$ and integrating over all fluid points yields the continuity equation in terms of the volume filtered, density weighted velocity,

$$\frac{\partial}{\partial t} (\varepsilon_f \bar{\rho}_f) + \nabla \cdot (\varepsilon_f \bar{\rho}_f \bar{\mathbf{u}}_f) = -S_\rho. \quad (17)$$

Using the fact that $\mathbf{u}_f|_{S_i} = \mathbf{u}_i + \frac{dr_i}{dt} \mathbf{n}$, where $\frac{dr_i}{dt}$ is the rate of change of the radius of particle i , S_ρ is given by

$$S_\rho = \sum_{i=1}^{n_p} \dot{m}_i g(|\mathbf{x} - \mathbf{x}_i|), \quad (18)$$

where

$$\dot{m}_i = -\rho_f|_{S_i} S_i \frac{dr_i}{dt}. \quad (19)$$

When considering phase change, a particle may change size due to evaporation or devolatilization processes. However, this term vanishes when considering inert particles.

With respect to the momentum balance, volume filtering the non-linear convective term requires special care. Applying Eq. (14)–(16) to the left-hand-side of Eq. (3) gives

$$\int_{V_f} g(|\mathbf{x} - \mathbf{y}|) \left[\frac{\partial}{\partial t} (\rho_f \mathbf{u}_f) + \nabla \cdot (\rho_f \mathbf{u}_f \otimes \mathbf{u}_f) \right] d\mathbf{y} = \frac{\partial}{\partial t} (\varepsilon_f \bar{\rho}_f \bar{\mathbf{u}}_f) + \nabla \cdot (\varepsilon_f \bar{\rho}_f \bar{\mathbf{u}}_f \otimes \bar{\mathbf{u}}_f) + \nabla \cdot \mathbf{R}_u - \mathbf{S}_{\rho u}, \quad (20)$$

where

$$\mathbf{R}_u = \varepsilon_f \bar{\rho}_f \bar{\mathbf{u}}_f'' \otimes \bar{\mathbf{u}}_f'' \quad (21)$$

is a residual stress akin to a Reynolds stress resulting from volume filtering. The momentum source term $\mathbf{S}_{\rho u}$ accounts for the momentum flow rate from the particle surface due to mass transfer, which simplifies to

$$\mathbf{S}_{\rho u} = \sum_{i=1}^{n_p} \dot{m}_i \mathbf{u}_i g(|\mathbf{x} - \mathbf{x}_i|). \quad (22)$$

Substituting $\mathbf{a} = \boldsymbol{\tau}$ in Eq. (14) and decomposing the point wise stress tensor into its filtered and residual components gives

$$\int_{V_f} g(|\mathbf{x} - \mathbf{y}|) \nabla \cdot \boldsymbol{\tau} d\mathbf{y} = \nabla \cdot (\varepsilon_f \bar{\boldsymbol{\tau}}) - \sum_{i=1}^{n_p} \int_{S_i} g(|\mathbf{x} - \mathbf{y}|) \mathbf{n} \cdot \bar{\boldsymbol{\tau}} d\mathbf{y} - \sum_{i=1}^{n_p} \int_{S_i} g(|\mathbf{x} - \mathbf{y}|) \mathbf{n} \cdot \boldsymbol{\tau}' d\mathbf{y}. \quad (23)$$

Applying Gauss' theorem to the second term on the right-hand-side of Eq. (23) and assuming both the filtered stress tensor and its derivatives vary little over distances comparable with the radius of g , the surface integral of the volume filtered stress tensor becomes

$$\sum_{i=1}^{n_p} \int_{S_i} g(|\mathbf{x} - \mathbf{y}|) \mathbf{n} \cdot \bar{\boldsymbol{\tau}} d\mathbf{y} \approx \bar{\boldsymbol{\tau}} \cdot \nabla \varepsilon_f. \quad (24)$$

Now considering the last term in the right-hand-side of Eq. (23), the filtering kernel $g(|\mathbf{x} - \mathbf{y}|)$ may be replaced by $g(|\mathbf{x} - \mathbf{x}_i|)$ and taken outside of the integral since the filtered value of $\boldsymbol{\tau}'$ taken over S_i is small compared with variations about its filtered value, and these variations are smooth. Using this and putting Eq. (24) in Eq. (23) yields

$$\int_{V_f} g(|\mathbf{x} - \mathbf{y}|) \nabla \cdot \boldsymbol{\tau} d\mathbf{y} = \varepsilon_f \nabla \cdot \bar{\boldsymbol{\tau}} - \sum_{i=1}^{n_p} g(|\mathbf{x} - \mathbf{x}_i|) \int_{S_i} \boldsymbol{\tau}' \cdot \mathbf{n} d\mathbf{y}. \quad (25)$$

For simplicity, the filtered stress tensor $\bar{\boldsymbol{\tau}}$ can be written as

$$\bar{\boldsymbol{\tau}} = -\bar{p} \mathbf{I} + \mu \left[\nabla \bar{\mathbf{u}}_f + \nabla \bar{\mathbf{u}}_f^T - \frac{2}{3} (\nabla \cdot \bar{\mathbf{u}}_f) \mathbf{I} \right] + \mathbf{R}_\mu, \quad (26)$$

where \mathbf{R}_μ arises from filtering the velocity gradients in $\boldsymbol{\tau}$. Note that when dealing with reactive flows, μ may vary significantly and will contribute additional terms to \mathbf{R}_μ .

We now exploit the similarities between Eq. (6) and Eq. (25) to elucidate the nature of the surface integral term on the right-hand-side of Eq. (25). Using the divergence theorem and the fact that $\nabla \cdot \bar{\boldsymbol{\tau}}$ varies little over the scale of a particle, Eq. (6) can be expanded as

$$\mathbf{f}_p^{\text{inter}} = \int_{S_p} \boldsymbol{\tau} \cdot \mathbf{n} d\mathbf{y} = \int_{S_p} (\bar{\boldsymbol{\tau}} + \boldsymbol{\tau}') \cdot \mathbf{n} d\mathbf{y} = \int_{V_p} \nabla \cdot \bar{\boldsymbol{\tau}} d\mathbf{y} + \int_{S_p} \boldsymbol{\tau}' \cdot \mathbf{n} d\mathbf{y} \approx V_p \nabla \cdot \bar{\boldsymbol{\tau}} + \int_{S_p} \boldsymbol{\tau}' \cdot \mathbf{n} d\mathbf{y}. \quad (27)$$

We now multiply through by $g(|\mathbf{x} - \mathbf{x}_p|)$, and sum over all particles. The left-hand-side of Eq. (27) is used to define the inter-phase momentum exchange $\mathbf{F}^{\text{inter}}$,

$$\mathbf{F}^{\text{inter}} = \sum_{i=1}^{n_p} g(|\mathbf{x} - \mathbf{x}_i|) \mathbf{f}_i^{\text{inter}}. \quad (28)$$

Now considering the first term on the right-hand-side of Eq. (27), $\nabla \cdot \bar{\boldsymbol{\tau}}$ may be taken outside the summation, and using Eq. (10) this term reduces to

$$\sum_{i=1}^{n_p} g(|\mathbf{x} - \mathbf{x}_i|) V_i \nabla \cdot \bar{\boldsymbol{\tau}} = (1 - \varepsilon_f) \nabla \cdot \bar{\boldsymbol{\tau}}. \quad (29)$$

Finally, combining Eqs. 25, 27, 28, and 29 yields

$$\int_{V_f} g(|\mathbf{x} - \mathbf{y}|) \nabla \cdot \boldsymbol{\tau} d\mathbf{y} = \nabla \cdot \bar{\boldsymbol{\tau}} - \mathbf{F}^{\text{inter}}. \quad (30)$$

From the above formulation, the momentum equation is then given by

$$\frac{\partial}{\partial t} (\varepsilon_f \bar{\rho}_f \widetilde{\mathbf{u}}_f) + \nabla \cdot (\varepsilon_f \bar{\rho}_f \widetilde{\mathbf{u}}_f \otimes \widetilde{\mathbf{u}}_f) = \nabla \cdot (\bar{\boldsymbol{\tau}} - \mathbf{R}_u) + \varepsilon_f \bar{\rho}_f \mathbf{g} - \mathbf{S}_{\rho u} - \mathbf{F}^{\text{inter}}. \quad (31)$$

2.4. Filtered scalar transport equation

A similar approach can be used to describe a generalized conservative transport equation, which is relevant when dealing with reactive flows. Given a point wise scalar quantity, ϕ , its transport equation is given by

$$\frac{\partial \rho_f \phi}{\partial t} + \nabla \cdot (\rho_f \mathbf{u}_f \phi) = \nabla \cdot (D \nabla \phi), \quad (32)$$

where D is the diffusion coefficient. Multiplying through by g and integrating the left-hand-side of the above equation yields

$$\int_{V_f} g(\mathbf{x} - \mathbf{y}) \left[\frac{\partial \rho_f \phi}{\partial t} + \nabla \cdot (\rho_f \mathbf{u}_f \phi) \right] d\mathbf{y} = \frac{\partial}{\partial t} (\varepsilon_f \bar{\rho}_f \widetilde{\phi}) + \nabla \cdot (\varepsilon_f \bar{\rho}_f \widetilde{\mathbf{u}}_f \widetilde{\phi}) + \nabla \cdot \mathbf{R}_\phi + S_{\rho \phi}, \quad (33)$$

where

$$\mathbf{R}_\phi = \varepsilon_f \bar{\rho}_f \widetilde{\mathbf{u}}_f'' \phi''. \quad (34)$$

Assuming the scalar quantity at the surface of particle i , $\phi|_{S_i}$, is uniform, the source term is given by

$$S_{\rho \phi} = \sum_{i=1}^{n_p} \dot{m}_i \phi|_{S_i} g(|\mathbf{x} - \mathbf{x}_i|) \quad (35)$$

Filtering the right-hand-side of Eq. (32) gives

$$\int_{V_f} \nabla \cdot (D \nabla \phi) g(|\mathbf{x} - \mathbf{y}|) d\mathbf{y} = \nabla \cdot (\varepsilon_f \bar{D} \nabla \bar{\phi}) - S_{\nabla \phi}, \quad (36)$$

where

$$S_{\nabla \phi} = \sum_{i=1}^{n_p} \int_{S_i} \mathbf{n} \cdot \nabla \phi D g(|\mathbf{x} - \mathbf{y}|) d\mathbf{y}. \quad (37)$$

Note that the source term $S_{\nabla \phi}$ vanishes when considering a Neumann boundary condition at the surface of the particle, namely $\nabla \phi \cdot \mathbf{n}|_{S_i} = 0$. Using the same arguments described when handling the fluid stress tensor to rewrite $\bar{D} \nabla \bar{\phi}$, the filtered general scalar transport equation is given by

$$\frac{\partial}{\partial t} (\varepsilon_f \bar{\rho}_f \tilde{\phi}) + \nabla \cdot (\varepsilon_f \bar{\rho}_f \tilde{\mathbf{u}}_f \tilde{\phi}) + \nabla \cdot \mathbf{R}_\phi = \nabla \cdot (\varepsilon_f D \nabla \tilde{\phi} + \mathbf{R}_D) - S_{\rho\phi} - S_{\nabla\phi}, \quad (38)$$

where \mathbf{R}_D is an unclosed term defined by $\mathbf{R}_D = \overline{D \nabla \phi} - D \nabla \bar{\phi}$.

2.5. Closures

As a result of filtering the equations of motion for the fluid phase, sub-grid terms (Eqs. (21), (26) and (27)) arise and require closure. The general problem consists of formulating expressions for these terms based on filtered quantities, applicable for a wide range of particle Stokes and Reynolds numbers from the dilute to dense limit. First, \mathbf{R}_u and \mathbf{R}_ϕ are similar to Reynolds stresses and involve velocity fluctuations about their filtered values. For classical turbulence the Reynolds stress is closed via a turbulent viscosity model, given by

$$\mathbf{R}_u \approx \mu_t (\nabla \bar{\mathbf{u}}_f + \nabla \bar{\mathbf{u}}_f^T), \quad (39)$$

where μ_t is the turbulent viscosity. In the context of particle-laden flows, the turbulent kinetic energy cascade is more complicated than with single-phase turbulence [45]. For varying particle-phase Stokes numbers and concentrations, the presence of the disperse phase can either augment or attenuate turbulence relative to single-phase turbulence [3]. Some studies have shown that a dilute concentration of particles located inside the fluid boundary layer suppresses the generation of turbulence, while larger particles contribute to Reynolds stresses [46]. For dense flows, fluidization can be generally associated with two kinds of turbulence; velocity fluctuations that arise from granular agitation at the particle scale (via interparticle collisions and wake instabilities), as well as from mesoscale features such as particle clustering and bubbles, and it is therefore incorrect to use standard turbulence models to close this term. For the majority of the work presented in this paper, we will assume that the carrier-phase velocity fluctuations do not contribute significantly to the flow dynamics, allowing us to neglect \mathbf{R}_u . However, in certain cases, such as spouting fluidized beds where fluid-phase Reynolds numbers may be very large (above to 58,000 in some lab-scale experiments [47]), \mathbf{R}_u requires proper closure. For these cases, a dynamic Smagorinsky model [48,49] based on Lagrangian averaging [50] is employed to estimate μ_t . Although this model does not account for turbulence modulation by the particles, it will be highlighted later that the cases considered exhibit high turbulent viscosity in regions free of particles.

Filtering the point wise velocity gradients in the viscous term leads to unclosed stresses, which can be handled in several ways, such as combining them with the stress tensor [42], accounting for them through the introduction of an effective viscosity [43,5], or simply neglecting them [51]. To deal with the unclosed term \mathbf{R}_μ in the filtered fluid stress tensor, an effective viscosity μ^* is introduced, giving

$$\mathbf{R}_\mu \approx \mu^* \left[\nabla \bar{\mathbf{u}}_f + \nabla \bar{\mathbf{u}}_f^T - \frac{2}{3} (\nabla \cdot \bar{\mathbf{u}}_f) \mathbf{I} \right], \quad (40)$$

where μ^* was derived by Gibilaro [52] for fluidized beds, and is given by

$$\mu^* = \mu (\varepsilon_f^{-2.8} - 1). \quad (41)$$

The force $\mathbf{f}_p^{\text{inter}}$ that couples the individual particles and the carrier fluid comes from the momentum exchange at the particle surface, which has been expanded in Eq. (27) to give

$$\mathbf{f}_p^{\text{inter}} \approx \nu_p \nabla \cdot \bar{\boldsymbol{\tau}} + \int_{S_p} \boldsymbol{\tau}' \cdot \mathbf{n} dS. \quad (42)$$

Eq. (42) can be better understood by considering a flow past a single, isolated particle. In that case, $\boldsymbol{\tau}$ varies on the scale of the particle diameter, and therefore $\bar{\boldsymbol{\tau}}$ is approximately constant for $\delta_f \gg d_p$. Then $\nabla \cdot \bar{\boldsymbol{\tau}}$ vanishes, and it becomes clear that $\int_{S_p} \boldsymbol{\tau}' \cdot \mathbf{n} dS$ represents the drag force due to the uniform flow, which we will write

$$\int_{S_p} \boldsymbol{\tau}' \cdot \mathbf{n} dS \approx \mathbf{f}^{\text{drag}}, \quad (43)$$

where \mathbf{f}^{drag} can be obtained through classical models. Other contributions to the surface integral might include added mass, the Basset history term, lift and Faxen forces. For gas–solid flows, the added mass term becomes negligible and is therefore not included in the simulations presented herein. However, when the density difference between fluid and particles decreases, the volume of fluid the particle needs to displace as it moves through it will provide non-negligible acceleration (or deceleration). Zhang and Prosperetti [14] give an exact expression for the added mass term for an inviscid fluid at low particle concentrations. At higher values of concentration, they include a correction to account for the local volume fraction. They also derive an expression for the lift force for spherical particles in an inviscid fluid. A quite different expression is given by Saffman [53] for viscous flows at low Reynolds numbers.

The drag model of Tenneti et al. [54] is employed in this work, which was derived from particle-resolved direct numerical simulation. An ensemble-averaging approach was used to calculate the unclosed average interphase momentum transfer of flow past random fixed configurations of monodisperse spheres. In their approach, the drag correlation models the complete

stress tensor, while our equations require only the fluctuating part. However, this drag was determined for a statistically homogeneous suspension at steady-state, and does not include the effect of the filtered pressure gradient. Assuming the contribution of the averaged stress tensor is negligible in their simulations, the correlated drag coefficient is appropriate for the equations presented in Section 2.3. The expression for the drag is given as

$$\frac{\mathbf{f}_i^{\text{drag}}}{m_p} = \frac{1}{\tau_p} (\widetilde{\mathbf{u}}_f - \mathbf{u}_p) F(\varepsilon_f, \text{Re}_p), \quad (44)$$

where the particle response time τ_p derived from Stokes flow is

$$\tau_p = \frac{\rho_p d_p^2}{18\mu\varepsilon_f}. \quad (45)$$

The dimensionless drag force coefficient is valid for a wide range of Reynolds numbers and solid packing, and is given by

$$F(\varepsilon_f, \text{Re}_p) = \frac{1 + 0.15\text{Re}_p^{0.687}}{\varepsilon_f^2} + \varepsilon_f F_1(\varepsilon_f) + \varepsilon_f F_2(\varepsilon_f, \text{Re}_p), \quad (46)$$

where the particle Reynolds number is

$$\text{Re}_p = \frac{\varepsilon_f \rho_f |\widetilde{\mathbf{u}}_f - \mathbf{u}_p| d_p}{\mu}. \quad (47)$$

The remaining two terms are given by

$$F_1(\varepsilon_f) = \frac{5.81(1 - \varepsilon_f)}{\varepsilon_f^3} + \frac{0.48(1 - \varepsilon_f)^{1/3}}{\varepsilon_f^4},$$

$$F_2(\varepsilon_f, \text{Re}_p) = (1 - \varepsilon_f)^3 \text{Re}_p \left(0.95 + \frac{0.61(1 - \varepsilon_f)^3}{\varepsilon_f^2} \right).$$

In the context of wall-bounded particle-laden flows, the accuracy of the drag coefficient and particle volume fraction diminishes at regions close to walls, where key assumptions that were used in the formulation of the models are often violated. For example, the definition of the volume filtered fluid quantity, Eq. (12), assumes that $\bar{\mathbf{a}}$ is evaluated far from the nearest fluid boundary, Γ . This allows for all surface integrals of the point property to be taken about the surface of the particle while neglecting contributions due to the fluid boundary, namely

$$\int_{\Gamma} \mathbf{n} \cdot \mathbf{a} \mathbf{g}(|\mathbf{x} - \mathbf{y}|) d\mathbf{y} = 0. \quad (48)$$

For regions of the fluid close to a wall, this assumption is no longer valid. Note that when considering $\mathbf{a} = \mathbf{u}_f$, this term vanishes due to the no-penetration condition. However, when dealing with the fluid stress tensor for example, this term must be accounted for. A model for the fluctuating stress tensor is already given in the form of a drag force, so a drag model that accounts for wall interactions directly would be convenient. In addition, the effective viscosity model defined in Eq. (41) is dubious at walls. We assume that the particle volume fraction at the surface of the wall vanishes, since only a single point of the particle surface can be in contact with the wall, and therefore we use the molecular viscosity for viscous fluxes at the walls. Details on the numerical implementation of the wall interactions will be provided in Section 3.2.

With regards to particle rotation, the first term on the right-hand side of Eq. (7) is neglected, and rotation is attributed only to the tangential component of the collision force, $\mathbf{f}_{i,j \rightarrow p}^{\text{col}}$, between particle p and all other particles j . It should be noted that particle lift forces, such as Saffman lift and Magnus effect, are not accounted for and might play a significant role in dilute flows with wall interactions, such as clustering of particles in risers, but this assumption should be appropriate in dense systems as shown herein. Note that the choice of closures proposed in this section is clearly not unique, and could be improved significantly with a better understanding of the intricate microscale physics. With the advance of high performance computing, there is an opportunity for fully-resolved particle DNS to provide data necessary to guide the development of such improved models.

3. Interphase coupling

Coupling between the carrier fluid and dispersed phase appears in the form of the volume fraction ε_f , defined by Eq. (9), and interphase exchange term $\mathbf{F}^{\text{inter}}$, defined by Eq. (28), as well as the source terms that arise during particle evaporation and devolatilization, defined by Eqs. (18), (22), and (35). These terms are first computed at the location of each particle, using information from the fluid, and are then transferred to the Eulerian grid. To interpolate the fluid variables to the particle loca-

tion, a second order trilinear interpolation scheme is used. To extrapolate the particle data back to the Eulerian mesh, it is necessary to perform a volume filtering operation that is consistent with the mathematical formulation. Classically, this filtering operation takes the form of a linear extrapolation of the particle data to the nearest cells [5,55]. Salman and Soteriou [35] and Garg et al. [56] reviewed typical interpolation schemes used to couple the two phases, and showed that such procedures lead to significant error that depends heavily on the ratio of the particle diameter to the grid spacing. With such approaches, the size of the Eulerian mesh sets the filtering length scale. As discussed in Section 2.2, the characteristic filter size should be larger than the particle, hence these approaches require for the mesh size to be much larger than the particle diameter. This requirement can become problematic under certain circumstances. For example, polydisperse particle systems where some particles are very large can require excessively coarse meshes, while boundary layer flows or flows in cylindrical geometries require locally finer meshes at the wall or at the pole, respectively, for which the Lagrangian approach seemingly breaks down. However, note that mathematically there should be no intrinsic limitation on how fine a mesh can be used for solving the fluid equations of motion (Eqs. (17) and (31)).

Transferring Lagrangian data to the Eulerian frame has already been defined within the derivation of the equations of motion (see Eqs. (10), (18), (22), (28), and (35)). Let \mathbf{b}_i be any property associated with particle i , its volume filtered value $\hat{\mathbf{b}}(\mathbf{x}, t)$ can be defined following Eq. (12) by

$$\varepsilon_p \hat{\mathbf{b}}(\mathbf{x}, t) = \int_{\mathcal{V}_S} \mathbf{b}_i(\mathbf{y}, t) g(|\mathbf{x} - \mathbf{y}|) d\mathbf{y}, \quad (49)$$

where $\mathcal{V}_S = \mathcal{V}_1 \cup \mathcal{V}_2 \cup \dots \cup \mathcal{V}_{n_p}$. Assuming \mathbf{b}_i to be constant for each particle, and since g varies little over the particle volume, we can approximate $\hat{\mathbf{b}}(\mathbf{x}, t)$ as

$$\varepsilon_p \hat{\mathbf{b}}(\mathbf{x}, t) \approx \sum_{i=1}^{n_p} \mathbf{b}_i g(|\mathbf{x} - \mathbf{x}_i|) \mathcal{V}_i. \quad (50)$$

Note that this definition is compatible with our derivation so far, in that the particle volume fraction is obtained by filtering $\mathbf{b}_i = 1$ for each particle, and $\mathbf{F}^{\text{inter}}$ by filtering $\mathbf{b}_i = \mathbf{f}^{\text{inter}} / \mathcal{V}_i$. Clearly, any mesh restriction arises from the discrete implementation of our volume filter g . If the volume filter is implemented explicitly according to Eq. (50), then we expect to retain full control of the particle diameter to mesh size ratio.

3.1. Filter discretization

Transferring the particle data to the Eulerian mesh requires discretizing the filtering kernel g , in order to solve Eq. (50). The support of the filtering kernel determines the number of Eulerian cells that are influenced by the presence of each particle. For example, consider a filter that spans $10d_p$ on a grid with cell spacing $\Delta x = 2d_p$. In three dimensions, the support of g spans approximately a $5 \times 5 \times 5$ cell volume, which would require looping over the 125 neighboring cells to compute the volume filtered particle data. This is clearly overly expensive for a system with a large number of particles. To alleviate this issue entirely, it is natural to introduce a two-step filtering process. First, the particle data is transferred on the Eulerian mesh through a conservative mollification operation. This filter has a characteristic length scale that corresponds to the mesh size, Δx . Then, the Eulerian field obtained from the first operation is diffused with an operator chosen to lead to the final filtered length scale δ_f , which should be much larger than the particle diameter. This methodology allows to use finer meshes while still following the requirements associated with local volume filtering. Each step is detailed below.

3.1.1. Mollification

The particle data is first transferred on the Eulerian mesh using a mollification kernel $g_{\mathcal{M}}$, giving

$$\varepsilon_p \mathbf{b}_{\mathcal{M}}(\mathbf{x}, t) \approx \sum_{i=1}^{n_p} \mathbf{b}_i g_{\mathcal{M}}(|\mathbf{x} - \mathbf{x}_i|) \mathcal{V}_i, \quad (51)$$

where $\mathbf{b}_{\mathcal{M}}(\mathbf{x}, t)$ is the initial particle data mollified onto the Eulerian grid. The mollification kernel is a function that decreases monotonically with the distance from the particle, and is normalized such that it integrates to unity, given by

$$g_{\mathcal{M}}(r) = \frac{1}{\sigma \sqrt{2\pi}} e^{-\frac{r^2}{2\sigma^2}}, \quad (52)$$

where σ is the kernel width, taken as $\sigma = \Delta x / 2\sqrt{2 \ln 2}$ such that the full width at half the height of the kernel, $\delta_{1/2}$, is equal to the grid spacing, and therefore the effect of the particle is typically spread out over the 27 nearest cells. The normalization ensures conservation of the particle data as it is transferred to the Eulerian mesh.

3.1.2. Diffusion

To deal with finer grids in a robust manner, a diffusion operation is implemented after the mollification step by solving

$$\frac{\partial \alpha}{\partial \tau} = \mathcal{D} \nabla^2 \alpha, \quad (53)$$

where \mathcal{D} is a diffusion coefficient associated with the transfer of particle data to the underlying mesh. The initial condition corresponds to $\alpha(\tau = 0) = \mathbf{b}_M$ and the solution corresponds to $\alpha(\tau = \tau_f) = \hat{\mathbf{b}}(\mathbf{x}, t)$, where $\hat{\mathbf{b}}(\mathbf{x}, t)$ represents the particle data after the convolution of the two filters has been applied. The quantity $\sqrt{\mathcal{D}\tau_f}$ corresponds to a length scale for this diffusion operation. An appropriate choice of value for this quantity should account for the mollification operation already accomplished, such that the application of the two operators produces a result independent of the underlying mesh size. Diffusion of the Gaussian kernel g_M with width Δx should give the full volume filtering kernel g with width δ_f , which leads to

$$\mathcal{D}\tau_f = \frac{\max(\delta_f^2 - \Delta x^2, 0)}{16 \ln 2}. \quad (54)$$

A value of $\delta_f = 3d_p$ was found to yield appropriate smoothing for a monodisperse system while allowing for detailed capture of mesoscale flow features. For polydisperse systems with wide size distributions, we take $\delta_f = \max(d_p)$ instead. Discretization of the diffusion process, Eq. (53), can be handled in several ways. For cases with few particles, Eq. (53) is solved explicitly using second order central differences. For systems with a large number of particles we solve Eq. (53) implicitly in a single step to reduce computational cost. Fig. 1 shows the initial particle data transferred to the mesh with $\delta_{1/2} = \Delta x$, as well as the diffused kernel with $\delta_{1/2} = \delta_f$.

Fig. 2 considers a single particle transferring its data in one dimension. Fig. 2a shows the mollification kernel g_M as the mesh is refined, Fig. 2b measures the error associated with the mollification operation compared to the exact solution given by Eq. (50) with $\delta_f = 3d_p$. Because this operation is a function of the mesh size, convergence is not achieved during refinement. Instead, the kernel approaches a Dirac distribution, potentially leading to volume fractions greater than unity, illustrating the fact that this operation requires the mesh size to be several times larger than the particle diameter. It is shown that the mollification kernel converges towards the reference solution until $\Delta x < \delta_f$, at which case the kernel width continues to narrow and the solution diverges. The full, two-step filtering operation is considered in Figs. 2c and 2d. As the mesh is refined, Fig. 2c shows that the kernel converges towards a Gaussian of characteristic width $\delta_f = 3d_p$. Convergence is shown in Fig. 2d for a mesh refinement from $\Delta x = 16d_p$ to $\Delta x = d_p/16$. According to Eq. (54), for cell sizes larger than δ_f no diffusion is taking place and $g(r) = g_M(r)$. Therefore, convergence for $\Delta x > \delta_f$ is based on the initial mollification. For cell sizes smaller than δ_f , the mollification kernel, $g_M(r)$, continues to narrow and approach a Dirac function (as depicted in Fig. 2a), the diffusion operation becomes active and allows to obtain second-order convergence.

3.2. Wall interactions

A conservative immersed-boundary (IB) method is employed to model realistic geometries without requiring a body-fitted grid. The method is based on a cut-cell formulation that provides discrete conservation of mass and momentum [40,57]. To compute the area and volume fractions that are at the basis of this approach, it is necessary to fully characterize the location of the immersed geometry surface. This is accomplished through the use of an implicit description in the form of an iso-surface of a smooth function, referred to as a level set function, G , chosen such that it corresponds to a standard signed distance function, i.e.,

$$|G(\mathbf{x}, t)| = |\mathbf{x} - \mathbf{x}_\Gamma|, \quad (55)$$

where \mathbf{x}_Γ corresponds to the closest point on the solid–fluid interface from \mathbf{x} , and $G(\mathbf{x}, t) > 0$ on one side of the interface, and $G(\mathbf{x}, t) < 0$ on the other side. With this definition, the immersed boundary surface Γ in a domain Ω is defined by $\Gamma(t) = \{\mathbf{x} \in \Omega : G(\mathbf{x}, t) = 0\}$.

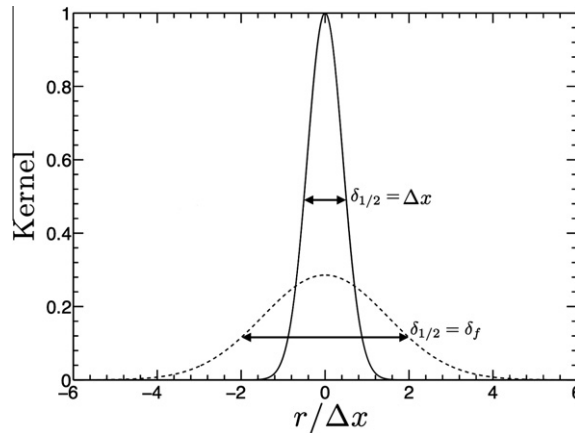


Fig. 1. Initial filtering kernel g_M with a characteristic width $\delta_{1/2} = \Delta x$ (solid line), full volume filtering kernel g with a characteristic width $\delta_{1/2} = \delta_f$ (dashed line).

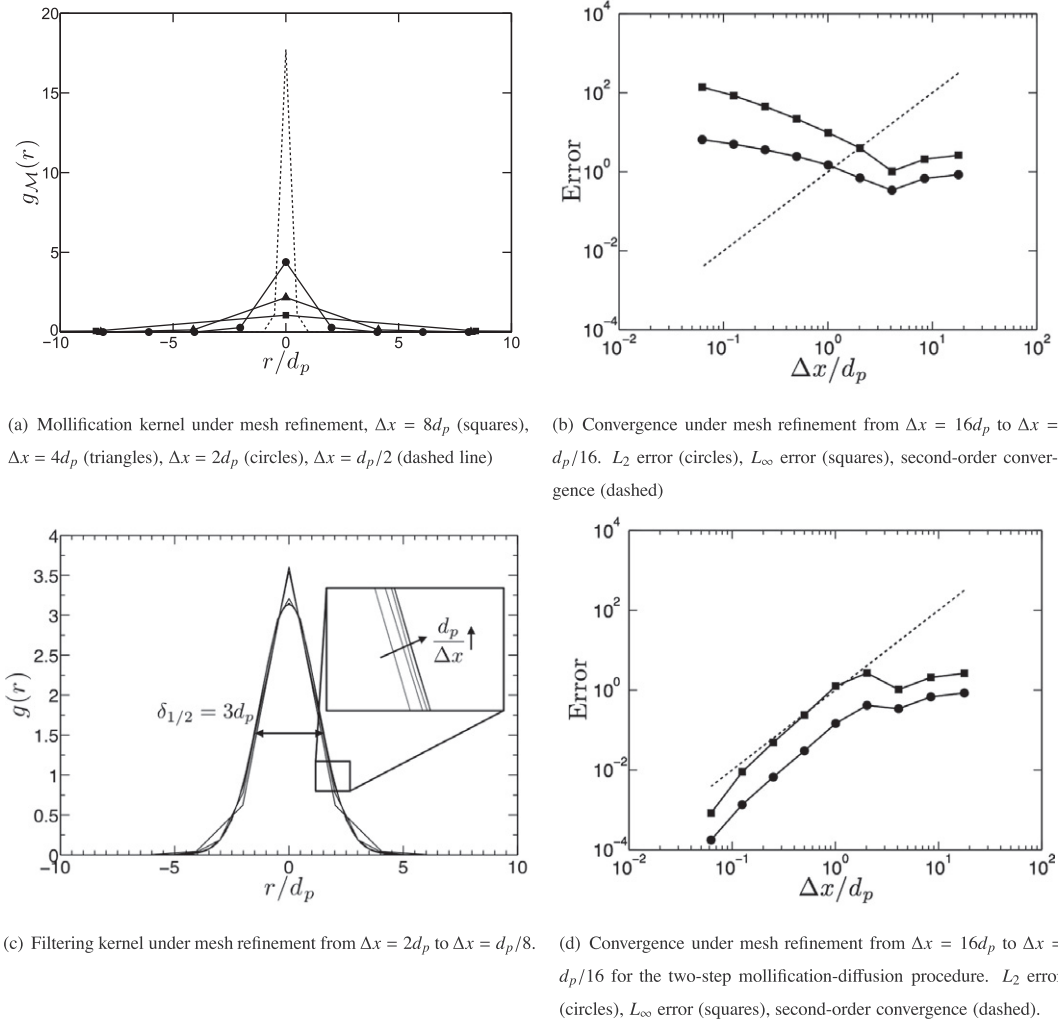


Fig. 2. Mollification and diffusion of a single particle in one-dimension under mesh refinement.

The coupling of this cut-cell strategy with our particle-laden approach must be handled carefully, such that the particle data being transferred to the underlying mesh is unaffected by the presence of small cut-cells at the walls. In particular, any over-prediction of the fluid volume fraction in near-wall cells is likely to lead to an unphysical behavior with fast fluid layers rising along the walls. During the mollification step, the Neumann boundary condition is enforced by introducing an image particle. Particles close to a wall are mirrored across the boundary and their mollification kernel is superimposed with the original kernel prior to normalization, given by

$$g_M(r) = g_M(r) + g_M(r - 2G(\mathbf{x}_p, t)), \quad (56)$$

where $G(\mathbf{x}_p, t)$ is the distance from the particle to the closest IB surface. Fig. 3 shows the analytic mollification kernel as a function of the scaled distance from the particle, along with the image kernel, and the full kernel with Neumann boundary. The solid line represents the wall and the dashed line indicates the particle position.

Eqs. (50) and (51) are implemented in a finite volume formalism. The mollification step requires computing the mollification kernel centered on particle p in cell i as

$$g_{i,p} = \frac{1}{V_i^{\text{cell}}} \int_{V_i^{\text{cell}}} g_M(\mathbf{x} - \mathbf{x}_p) d\mathbf{x}, \quad (57)$$

where V_i^{cell} is the volume of cell i . This integral is approximated by splitting the cell volume into five tetrahedra, such that

$$\int_{V_i^{\text{cell}}} g_M(\mathbf{x} - \mathbf{x}_p) d\mathbf{x} \approx \sum_{n=1}^5 g_M(\mathbf{x}_{i,n}^{\text{tet}} - \mathbf{x}_p) V_{i,n}^{\text{tet}}, \quad (58)$$

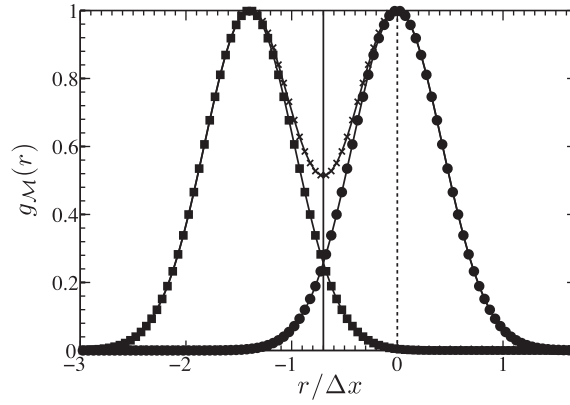


Fig. 3. Introduction of image particle to impose Neumann condition near walls. Wall (solid line), particle position (dashed line), initial kernel (circles), imaged kernel (squares), superposition (crosses).

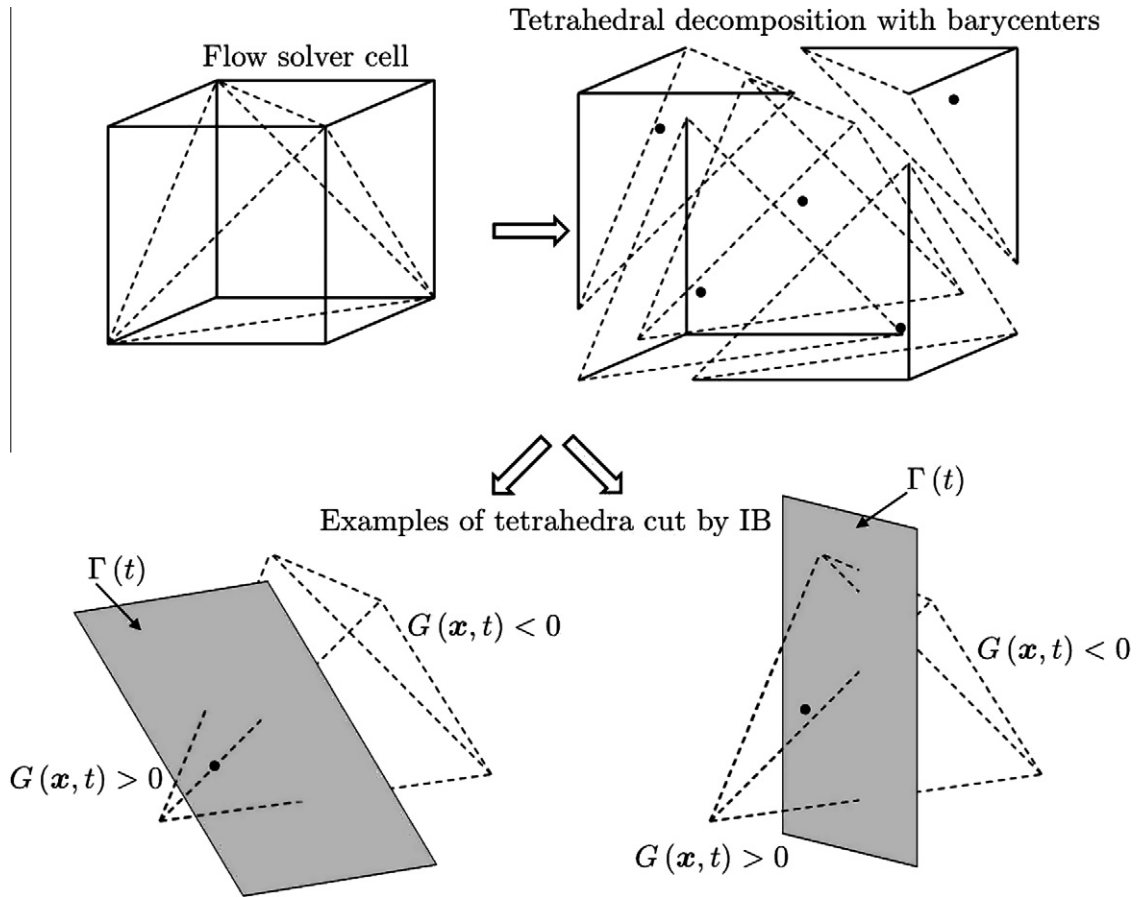


Fig. 4. Decomposition of a hexahedron cell into five tetrahedra and their corresponding barycenters (filled circles) for volume integration. Cells cut by the IB are recomputed via marching tetrahedra algorithm.

where $\mathbf{x}_{i,n}^{\text{tet}}$ and $\mathcal{V}_{i,n}^{\text{tet}}$ represent respectively the barycenter and the volume of the n th tetrahedron. Fig. 4 illustrates a flow solver cell divided into five tetrahedra with their corresponding barycenters. Tetrahedra that are cut by the IB surface are split into sub-tetrahedra, using a marching tetrahedra algorithm [58], and the barycenter and volume of the cut tetrahedra are adjusted appropriately. This approach provides an accurate sub-cell integration methodology that accounts fully for the immersed boundary.

4. Collision model

4.1. Normal collision

Particle–particle and particle–wall collisions are modeled using a soft-sphere approach originally proposed by Cundall and Strack [36]. The particles are represented as a mass-spring-dashpot system, as shown in Fig. 5. When two particles come into contact, a repulsive force $\mathbf{f}_n^{\text{col}}$ is created as

$$\mathbf{f}_{n,b \rightarrow a}^{\text{col}} = \begin{cases} -k\delta_{ab}\mathbf{n}_{ab} - \eta\mathbf{u}_{ab,n} & \text{if } d_{ab} < (r_a + r_b + \lambda), \\ 0 & \text{else.} \end{cases} \quad (59)$$

For the collision shown in Fig. 5, r_a and r_b are the radii of particles a and b , respectively, d_{ab} is the distance between the centers of the particles, δ_{ab} is the overlap between the particles, and \mathbf{n}_{ab} is the unit normal vector from particle a to particle b . The normal relative velocity between particles a and b is given by

$$\mathbf{u}_{ab,n} = ((\mathbf{u}_a - \mathbf{u}_b) \cdot \mathbf{n}_{ab})\mathbf{n}_{ab}. \quad (60)$$

λ is the force range, which will be discussed in Section 4.4, and k and η are the spring stiffness and the damping parameter, respectively. A model for the damping parameter uses a coefficient of restitution $0 < e < 1$ and an effective mass $m_{ab} = (1/m_a + 1/m_b)^{-1}$ such that

$$\eta = -2 \ln e \frac{\sqrt{m_{ab}k}}{\pi^2 + (\ln e)^2}. \quad (61)$$

Collisions with walls are handled by treating the walls as particles with infinite mass and zero radius.

4.2. Inter-particle friction

To account for friction between particles and thus particle rotation, a tangential collision model presented by van der Hoef et al. [6] is used. When particle a comes in contact with particle b , the tangential force is described as

$$\mathbf{f}_{t,b \rightarrow a}^{\text{col}} = \begin{cases} -k_t\delta_t - \eta_t\mathbf{u}_{ab,t} & \text{if } |\mathbf{f}_{t,b \rightarrow a}^{\text{col}}| \leq \mu_f |\mathbf{f}_{n,b \rightarrow a}^{\text{col}}|, \\ -\mu_f |\mathbf{f}_{n,b \rightarrow a}^{\text{col}}| \mathbf{t}_{ab} & \text{if } |\mathbf{f}_{t,b \rightarrow a}^{\text{col}}| > \mu_f |\mathbf{f}_{n,b \rightarrow a}^{\text{col}}|, \end{cases} \quad (62)$$

where k_t , δ_t , η_t , and μ_f are the tangential spring stiffness, tangential displacement, tangential damping coefficient, and friction coefficient, respectively. The relative tangential velocity, $\mathbf{u}_{ab,t}$, is defined as

$$\mathbf{u}_{ab,t} = \mathbf{u}_{ab} - \mathbf{u}_{ab,n}, \quad (63)$$

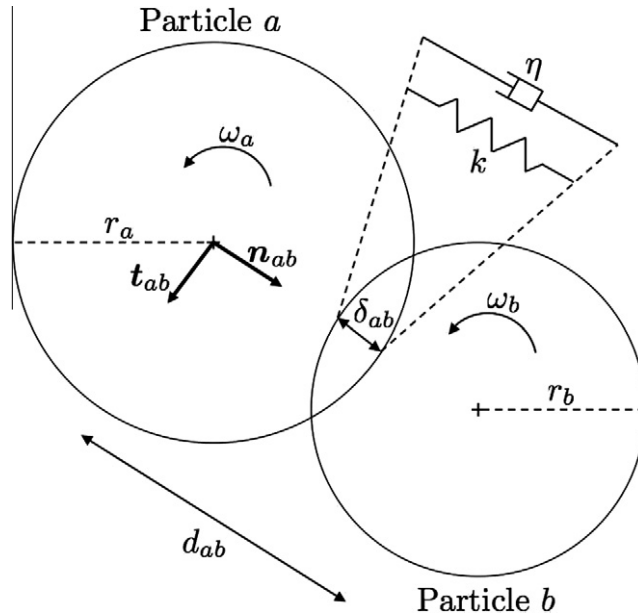


Fig. 5. Soft-sphere representation of two particles undergoing collision.

and is used to create a tangential unit vector \mathbf{t}_{ab} as

$$\mathbf{t}_{ab} = \frac{\mathbf{u}_{ab,t}}{|\mathbf{u}_{ab,t}|}. \quad (64)$$

In two dimensions, the tangential displacement is defined as

$$\delta_t = \delta_{t_0} + \int_{t_0}^t \mathbf{u}_{ab,t} dt, \quad (65)$$

where t_0 is the initial time of the collision. Computing Eq. (65) can become computationally intensive for particle flows under sustained contact. For instance, in two dimensions with a polydisperse particle system, tens of simultaneous collisions are common, and in three dimensions, hundreds of simultaneous collisions may be possible for a single particle provided the particle size distribution is wide enough. In order to deal with tangential collisions properly, the tangential displacement, δ_t , needs to be stored for every collision and every particle at each timestep. For systems with tens of thousands of particles or more, this method becomes very computationally expensive. However, the static friction model

$$\mathbf{f}_{t,b \rightarrow a}^{\text{col}} = -\mu_f |\mathbf{f}_{n,b \rightarrow a}^{\text{col}}| \mathbf{t}_{ab} \quad (66)$$

that simply multiplies the normal component of the collision force by the coefficient of friction does not require storing any additional information. Moreover, this model is valid for sustained contact, which should dominate for dense particle systems. This simplified model was tested against the full linear model, Eq. (62), as well as experiments described in Di Renzo and Di Maio's review [59]. The test configuration consists of a single particle colliding with a flat surface, as illustrated in Fig. 6. Collision parameters are given in Table 1, chosen to reflect the simulations performed in Di Renzo and Di Maio [59].

Fig. 7a shows the rebound angle α_2 , versus impact angle α_1 . The second plot, Fig. 7b, shows the final rotational velocity ω_2 , versus α_1 . The third plot, Fig. 7c, shows the tangential coefficient of restitution $e_t = |\mathbf{u}_{1,t}|/|\mathbf{u}_{2,t}|$, versus α_1 , where $\mathbf{u}_{1,t}$ and $\mathbf{u}_{2,t}$ are the tangential velocities before and after the collision, respectively. At smaller contact angles the friction model over-predicts the tangential force, though overall this approach appears to be a good compromise between accuracy and cost.

Once each individual collision force is computed, the full collision force that particle p experiences can be expressed as a sum of collisions with all other particles j undergoing collision with p , i.e.,

$$\mathbf{F}_p^{\text{col}} = \sum_j (\mathbf{f}_{n,j \rightarrow p}^{\text{col}} + \mathbf{f}_{t,j \rightarrow p}^{\text{col}}). \quad (67)$$

4.3. Stability criteria

When simulating dense particle-laden flows, particle processes often limit the simulation timestep. The particle Courant-Friedrichs-Lewy (CFL) number, given by

$$\text{CFL}_p = \frac{\Delta t_p |\mathbf{u}_p|}{d_p}, \quad (68)$$

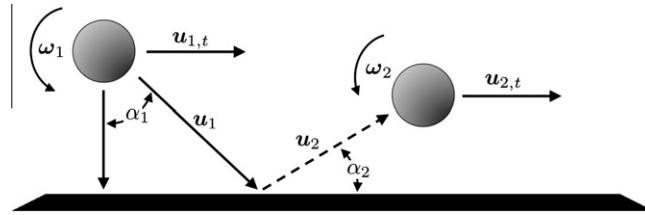


Fig. 6. Particle collision test. Angular velocity ω_p , impact angle α , impact velocity \mathbf{u}_p , normal velocity $\mathbf{u}_{ab,n}$, tangential velocity $\mathbf{u}_{ab,t}$.

Table 1

Particle collision parameters for rebound test case.

Diameter, d_p	5	mm
Density, ρ_p	4000	kg/m ³
Normal spring constant, k	1.72×10^7	N m ⁻¹
Tangential spring constant, k	1.48×10^7	N m ⁻¹
Coefficient of restitution, e	1.0	–
Friction coefficient, μ_f	9.2×10^{-2}	–
Initial velocity, \mathbf{u}_1	3.9	m s ⁻¹
Timestep, Δt	1×10^{-8}	s

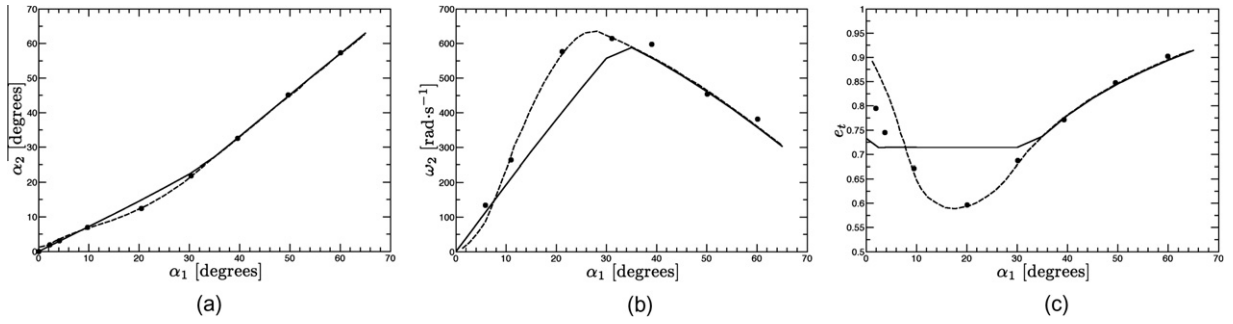


Fig. 7. Validation of the simplified tangential collision model (solid line) against full linear model [59] (dashed line) and experimental results [59] (filled circles).

characterizes the fraction of its diameter the particle moves during the particle timestep, Δt_p . If the particle response time τ_p is less than the simulation timestep Δt , the particle solver iterates with a timestep $\Delta t_p < \tau_p$ for a total time Δt . The timestep for integrating the particle motion is chosen such that $\max_{p=1..n_p} \text{CFL} \leq 0.1$, thereby limiting the inter-particle overlap during collisions.

To avoid excessive overlap which can lead to unphysical solid packing in a bed at rest, the particle stiffness should compensate for the weight of particles above it, leading to the following criterion on the spring constant

$$k \gg H_b \frac{m_p g}{d_p^2}, \quad (69)$$

where H_b is the height of the bed. Finally, the spring constant is related to the collision time τ_{col} according to

$$k = m_{ab} / \tau_{col}^2 (\pi^2 + \ln(e)^2). \quad (70)$$

To ensure proper resolution of collisions, Δt is chosen such that $\tau_{col} \geq 15\Delta t$.

4.4. Random close-packing limit

As mentioned in Section 4.1, a radius of influence λ is introduced to create a collision force when two particles are close, but not yet in contact. The collision model proposed by Patankar (2001) [5] suggests using $\lambda = 0.075d_p$ when calculating particle overlap. With a finite, non-zero radius of influence, particles at rest never come in contact, and therefore a settled bed under random close-packed conditions will exhibit an artificially large fluid volume fraction. Theory and experiments have shown that a bed of monodisperse spheres at rest will reach a random close-packed limit of 0.634 [60]. Several three-dimensional simulations were conducted to see the effect of λ on the solid packing for a static bed. The domain is periodic in the span-wise directions and consists of approximately 100,000 monodisperse spheres, shown in Fig. 8a. The particles have a diameter and density of 200 μm and 3300 kg/m^3 , respectively. A uniform mesh of $\Delta x = 2d_p$ with 24 cells in the spanwise directions and 50 cells in the height was used for this test. Fig. 8b shows that the random close-packing limit is recovered when λ approaches 0, as expected.

Following this observation, we modify the expression of the radius of influence, making it depend on the relative collision velocity. To this end, a particle collision CFL number is introduced as

$$\text{CFL}_{ab}^{\text{col}} = \frac{|\mathbf{u}_{ab,n}| \Delta t}{d_{\text{eff}}}, \quad (71)$$

where $|\mathbf{u}_{ab,n}|$ is the magnitude of the normal relative velocity between particles a and b during a collision, and d_{eff} is the effective particle diameter ($d_{\text{eff}} = d_p$ for wall collisions and $d_{\text{eff}} = r_a + r_b$ for particle–particle collisions). The timestep for integrating the particle motion is chosen such that particles do not move more than 10% of their diameter per timestep, and therefore, the largest value $\text{CFL}_{ab}^{\text{col}}$ will typically be 0.1 for particle–wall collisions, and 0.2 for particle–particle collisions. The radius of influence can then be expressed as

$$\lambda_{\text{wall}} = 0.75 \text{CFL}_{ab}^{\text{col}} d_{\text{eff}}, \quad (72)$$

$$\lambda_{\text{particle}} = 0.375 \text{CFL}_{ab}^{\text{col}} d_{\text{eff}}, \quad (73)$$

such that it decreases monotonically as the normal collisional velocity decreases, reaching zero when the particles are immobile with respect to one another. This strategy allows to recover the correct random close-packing limit in near motionless regions of the bed, while providing a robust collision model for high-speed collisions.

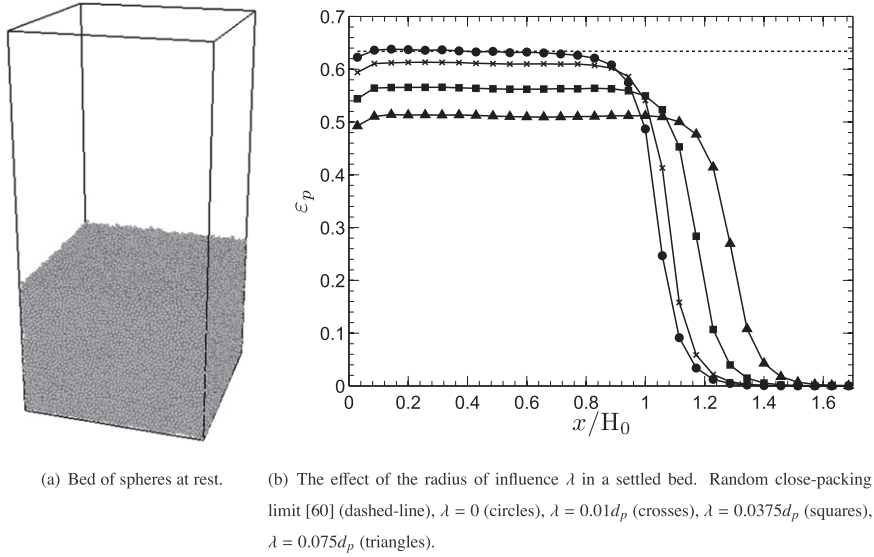


Fig. 8. Testing the effective solid packing in a bed at rest.

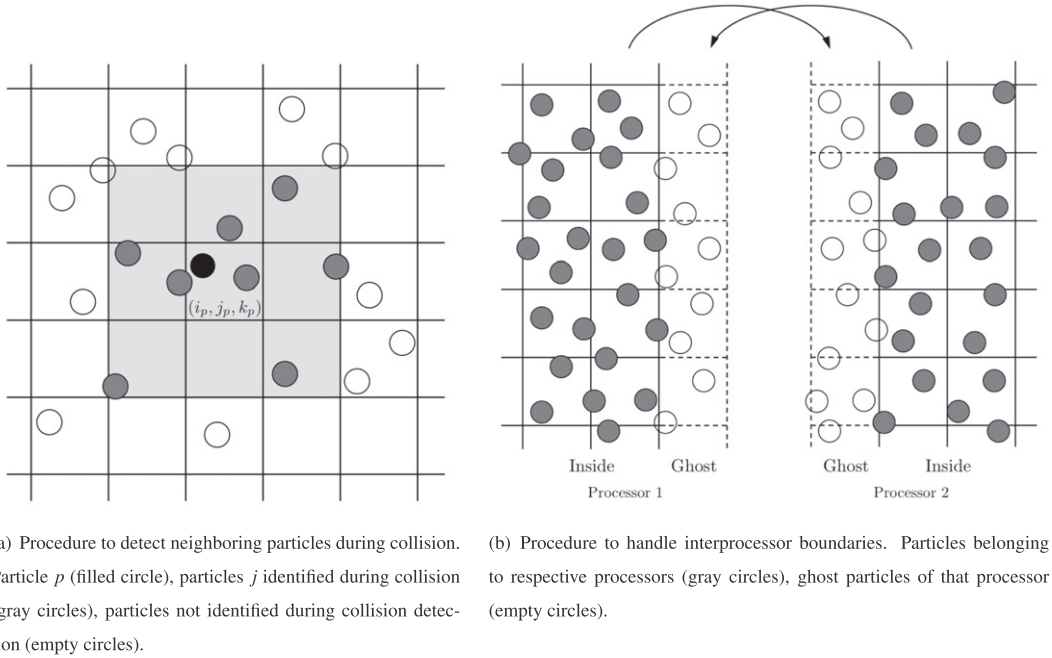


Fig. 9. Utilization of the Eulerian mesh during computation of particle collisions.

4.5. Numerical implementation

The computation of the collision force requires measuring inter-particle distances, which leads to an $\mathcal{O}(n_p^2)$ problem if implemented using a brute-force strategy. Instead, we make use of the underlying computational mesh in order to speed up the identification of likely collision partners, as illustrated in Fig. 9a. For each particle p , we identify the computational cell (i_p, j_p, k_p) to which particle p belongs. For this cell and its closest 26 neighbors, we loop over all particles j in that cell, and test whether p and j are undergoing collision. The computational cells are typically large enough to ensure all collisions are captured using this approach. Note that it is also possible to construct a separate grid proportional to the particle diameter for the computation of particle collisions, which might be useful for particles with a wide size distribution, or non-uniform meshes, leading to grid spacing smaller than the particle diameter.

The scheme also relies on ghost particles in order to facilitate the parallel implementation of the collision force calculation. These ghost particles correspond to particles located inside interprocessor ghost cells, and they are communicated between processors like any Eulerian variable, as illustrated in Fig. 9b. Each processor contains its own particles as well as the particles in the nearest layer of cells of the processor immediately neighboring it.

5. Numerical implementation and performance

The CFD code NGA [41], a high-order, fully conservative solver tailored for turbulent flow computation, solves the fluid equations on an Eulerian mesh while the particles are tracked individually in a Lagrangian framework. All simulations presented in this paper are performed using second order spatial accuracy for both convective and viscous terms in the Navier–Stokes equations. Time advancement is accomplished using the second order accurate semi-implicit Crank–Nicolson scheme of Pierce and Moin [61]. Based on a fractional step approach [62], this algorithm uses both temporal and spatial staggering between velocity, pressure, and density. The volume fraction ε_f is stored at cell centers, like the density. The details on the mass, momentum, and energy conserving finite difference scheme are available in [41].

Before each new flow solver timestep, the particle equations are solved using the fluid velocity obtained from an Adams–Bashforth prediction. The initial predictor step is second order accurate, resulting in second order temporal accuracy of the coupling between particles and gas phase [63]. The particle equations consist of a set of nine coupled ordinary differential equations per particle that are solved using a second-order Runge–Kutta scheme. Sub-stepping is used to ensure stability when the particle response time (defined in Eq. (45)) becomes smaller than the simulation timestep. For low Stokes number cases, this can become computationally expensive as many iterations might be necessary to maintain stability. An implicit solver such as DVODE [64] can be implemented to avoid this issue and reduce computational time.

Particles are distributed among processors based on the underlying domain decomposition of the fluid phase. After each timestep, the particles undergo an interprocessor communication step as they move from one processor sub-domain to another, as illustrated in Fig. 10. NGA relies on interprocessor ghost cells in order to facilitate this data exchange. Particles owned by processor 1 that are located within its ghost cells at the end of the timestep are sent to the appropriate neighboring processors before being removed from the memory of processor 1. In this work, the numerical methods have been implemented in parallel using message passing interface (MPI), and detail on the solution procedure as well as parallel performance of the scheme will be provided in the following sections.

5.1. Solution procedure

This section summarizes the solution procedure used in our code, then discusses the relative cost of individual routines.

1. Pre-timestep fluid velocity solver

- Adam–Bashforth prediction of the fluid velocity at the middle of the particle timestep

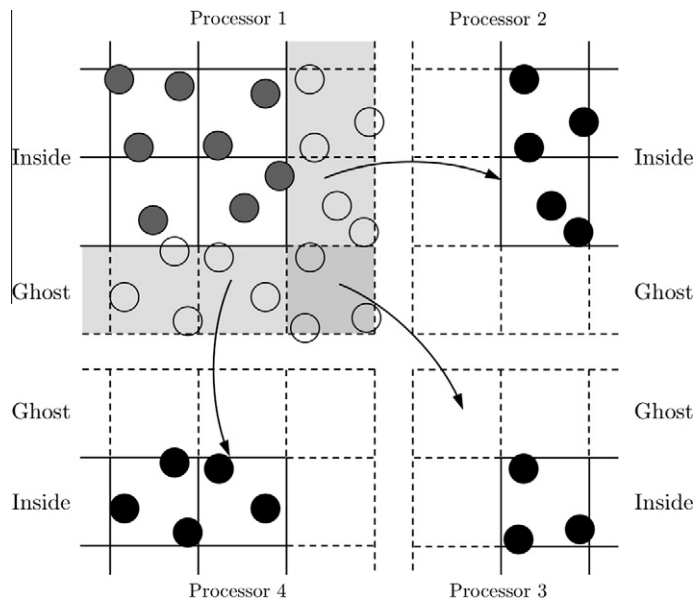


Fig. 10. Procedure to handle interprocessor particle exchange. Particles belonging to processor 1 (gray circles), particles removed from processor 1 after sending (empty circles), particles received by neighboring processors (filled circles). Dashed lines indicate ghost cell boundaries.

2. Localize particles on mesh
 - Find i, j, k of cell in which the center of the particle lies
 - Communicate particles with neighboring processors
3. Compute collision force for each particle
 - Loop through the 27 nearest cells and identify neighboring particles
 - Calculate distance d_{ab} between particle a and nearest wall (“ b ”)
 - Calculate distance d_{ab} between particle a and each neighboring particle b
 - If $d_{ab} < 0.075d_{\text{eff}}$, compute λ (Eqs. (72) and (73))
 - If $d_{ab} < \lambda$
 - * Solve normal collision force (Eq. (59))
 - * Solve tangential collision force (Eq. (66))
4. Advance particles
 - Determine particle timestep size such that $\Delta t_p < \tau_p$ (Eq. (45))
 - Subiterate for a total time Δt
 - Interpolate the fluid-phase variables ($\nabla \cdot \boldsymbol{\tau}$, ε_f , \mathbf{u}_f , ρ_f , μ) to the particle position
 - Compute forces acting on the particle (Eq. (5))
 - Update position, velocity, and angular velocity with second-order Runge–Kutta
5. Compute interphase exchange term from particles
 - Calculate momentum exchange term (Eqs. (42) and (44))
 - Exchange source term with fluid phase via mollification (Eqs. (51) and (52))
 - Filter momentum exchange term (Eq. (53))
6. Compute volume fraction from particles
 - Mollify particle volume to the Eulerian grid (Eqs. (51) and (52))
 - Divide by local cell volume
 - Filter volume fraction (Eq. (53))
7. Communicate particles with neighboring processors
8. Advance fluid velocity
 - Multiply fluid density by volume fraction
 - Compute effective viscosity (Eq. (41))
 - Update velocity using Crank–Nicolson scheme
9. Advance the timestep, $t = t + \Delta t$.
10. Repeat if $t < t_{\text{final}}$

To assess the timing of each component of the algorithm, a three-dimensional, twice periodic bed of 15.6 million particles was simulated using 20 million grid cells on 720 cores. To avoid any load imbalances, the processors were decomposed in the two span-wise directions only, resulting in approximately 22,000 particles per processor. Timing for the most expensive routines is given in Fig. 11, which shows the fraction of time spent in the LPT routine that is outlined in the algorithm above. While each timestep took approximately 6 s, it was found that the collision process (step 3) dominates the cost at approximately 72% of the total time, while the particle advancement routine (step 4) and interprocessor communication were found to take approximately 17% and 2% of the total time, respectively. It should be noted that the relative cost of interprocessor communications increases significantly when considering fewer particles per processor.

5.2. Parallel performance

A scaling analysis was performed on Red Mesa, the National Renewable Energy Laboratory’s (NREL) high performance computing system. The system consists of 15,360 cores with a peak performance of 180 TFlops. A series of simulations were run to test the parallel performance of NGA using a reference simulation of 134 million cells and 382 million particles on 4096 cores. The number of cores is progressively reduced to obtain the speed-up data, while the scale-up data is obtained by proportionally reducing the domain size and number of cores in the vertical direction while keeping the relative bed height constant at half the reactor length. Fig. 12 shows strong and weak scaling results. Scale-up is computed as

$$\text{Scale-up} = N_{\text{core}}^{\text{ref}} \left(\frac{n_x n_y n_z}{t_{\text{step}}} \right) \left(\frac{n_x^{\text{ref}} n_y^{\text{ref}} n_z^{\text{ref}}}{t_{\text{step}}^{\text{ref}}} \right)^{-1}, \quad (74)$$

where N_{core} is the number of cores, t_{step} is the time per timestep, n_i represents the number of cells in the i th direction, and the “ref” quantities corresponds to the reference simulation. Similarly, speed-up is defined as

$$\text{Speed-up} = N_{\text{core}}^{\text{ref}} \frac{t_{\text{step}}}{t_{\text{step}}^{\text{ref}}}. \quad (75)$$

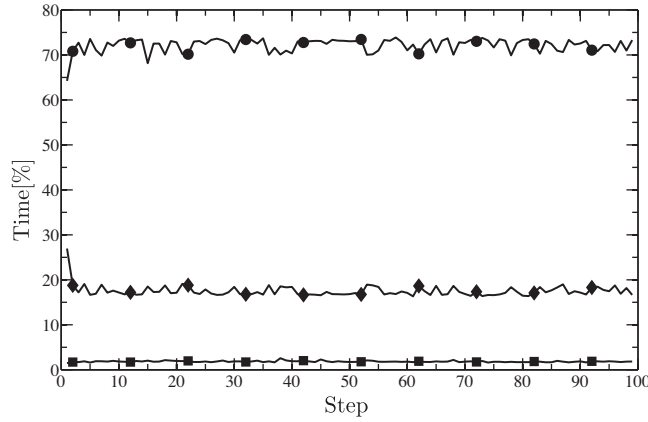


Fig. 11. Timing of specific particle tracking routines, normalized by the total time per timestep. Simulation contains 15.6 million particles, performed using 720 cores: collision scheme (circles), particle advancement (diamonds), interprocessor communication (squares).

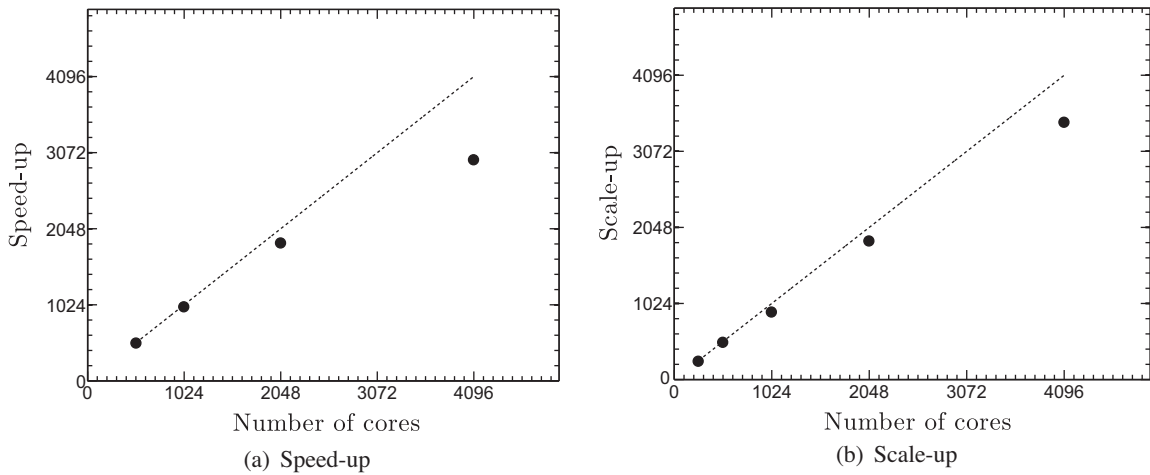


Fig. 12. Scaling analyses of NGA on Red Mesa. Dashed line indicate linear scaling.

The cost of the particle tracking scheme was found to remain roughly constant throughout the simulations. It was found that the particle tracking routines accounted for 83% of the total simulation cost, while the pressure and velocity solver accounted for 14% and 2.4%, respectively. Scaling results are very good up to 4,096 cores, the maximum number used in this test, with increased efficiency for higher loading of cells and particles per processor.

6. Numerical tests

A range of laboratory scale simulations were conducted in order to validate the methods discussed in this paper. The section begins with an analysis of the force balance at the onset of fluidization. Highly turbulent spout fluidization and bidispersed segregation are then considered in a pseudo two-dimensional reactor, and results are compared against experiments. Finally, a two-inch bubbling fluidized bed is presented, and mean statistics on the flow dynamics and bubble characteristics are discussed.

6.1. Onset of fluidization

The phenomenon of fluidization occurs when a fluid passes through a bed of granular material, causing a pressure drop that can support the weight of the bed, allowing the granular material to behave like a fluid. The critical value at which the superficial gas velocity, defined as $\varepsilon_f u_f$ if we assume the x -direction to be vertical, first begins to fluidize the material, is known as the minimum fluidization velocity U_{mf} . Superficial velocities exceeding U_{mf} may lead to rapid particle motion causing bubbling or clustering, allowing for efficient mixing to take place. A study of the onset of fluidization was conducted in order to assess the capability of the proposed strategy to reproduce theory [13]. We consider the periodic bed of particles

used in Section 4.4 and shown in Fig. 8a, and introduce an inlet velocity at the bottom which is increased until fluidization is reached. The simulation parameters for the study can be found in Table 2.

Prior to fluidization, the pressure drop across the bed balances drag, which can be shown by neglecting friction at the walls and the effect of gravity on the fluid, reducing the momentum balance given by Eq. (31) in the vertical direction to

$$-\varepsilon_f \frac{\partial p}{\partial x} = f^{\text{drag}}. \quad (76)$$

The minimum fluidization velocity is reached when the pressure drop can support the weight of the bed [13], given by

$$-\frac{\partial p}{\partial x} = (\varepsilon_f \rho_f + \varepsilon_p \rho_p) g \approx \varepsilon_p \rho_p g. \quad (77)$$

Fig. 13a shows the pressure gradient normalized by the weight of the bed per unit volume acting in the vertical direction plotted against the analytic expressions given by Eqs. (76) and (77). Eqs. (76) and (77) were solved using the volume fraction at minimum fluidization. As the superficial gas velocity increases, the pressure drop across the bed increases near-linearly until minimum fluidization has been reached, at approximately 0.03 m/s.

As the inlet velocity approaches U_{mf} , the forces acting on the particles evolve as well. The relative importance of each component in the particle equation of motion was computed by summing the individual forces acting on each particle over the total number of particles, and is given in Fig. 13b. When the bed is at rest, the collision force exactly balances the weight of the bed. The pressure drop and drag increase with the superficial gas velocity until the particles are no longer in contact and fluidization is reached.

The measured pressure drop in Fig. 13a verifies the code's ability to reach minimum fluidization for the given drag model as predicted by theory. Figs. 13a and 13b illustrate the mechanisms that control the multiphase dynamics in fluidized beds, which will be responsible for the phenomenon observed in the subsequent test cases.

6.2. Pseudo two-dimensional spout fluidization

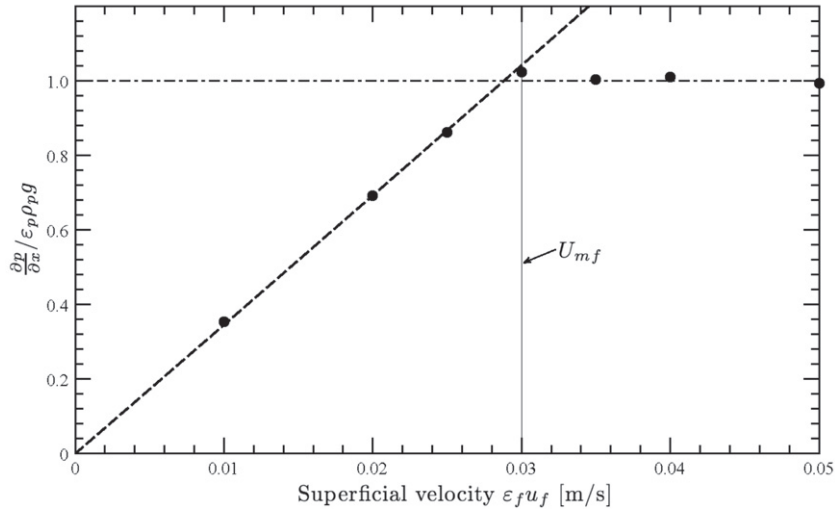
Numerical simulations of a pseudo two-dimensional spout fluidized bed under two different inflow conditions were conducted and compared to experiments by Link et al. [65,66]. The depth of the bed is large enough to prevent bridge formation between particles, though not too large such that digital-image-analysis is capable of measuring voidage. A schematic of the bed as well as its dimensions can be seen in Fig. 14. Except for the height of the reactor, which is reduced to limit the computational expense, all physical parameters used in the simulations match those of the experiments. Simulation parameters for each case can be found in Table 3. A grid size equal to the particle diameter was chosen in order to capture the unsteady nature of the gas phase. Air is injected via three separate sections at the inlet. A high-speed spout velocity is fed through the center orifice while the outside sections feed the background velocity.

For both cases, relatively high gas velocities are introduced, resulting in particle Reynolds numbers above 300, and gas Reynolds numbers through the spout above 13,000. At such large Reynolds numbers, sub-grid Reynolds stresses are likely to be significant and need to be closed. For both cases A and B, we close \mathbf{R}_u with a dynamic Smagorinsky model [48,49] based on Lagrangian averaging [50]. Note that we do not account for turbulence modulation by the particles, even though it might be important. Two viscosity models are therefore used simultaneously: an eddy viscosity model to account for turbulence, and an effective viscosity model based on experimental correlations to account for the enhancement in viscous stresses in packed regions of the bed. Interactions between these models are unclear. However, Fig. 15 reveals a distinct segregation between the models. The effective viscosity dominates in regions with greatest solid packing, while the turbulent viscosity model dominates in the particle-free turbulent regions.

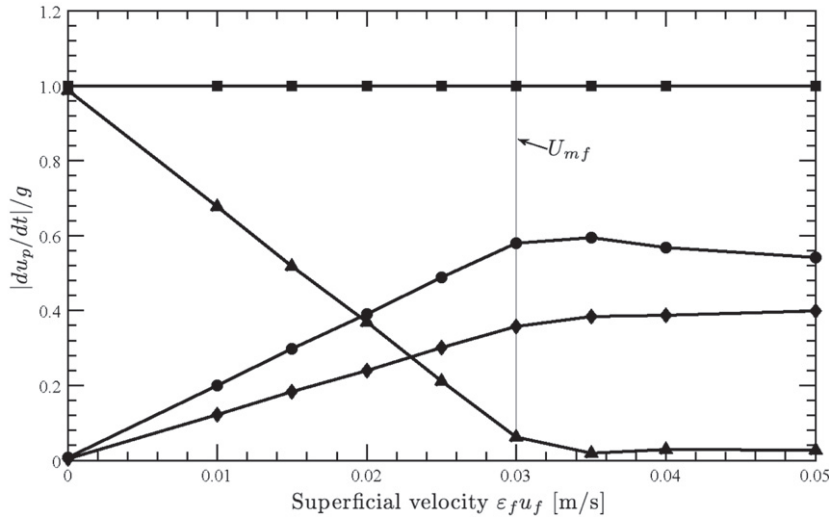
As depicted in Fig. 15, the spout velocity in case A leads to a region of fast moving particles in the middle of the bed with a recirculation region at the top. When the background fluidization velocity is increased, the spout has a greater impact on the

Table 2
Simulation parameters for the minimum fluidization test.

Height	20	mm
Width	9.6	mm
Depth	9.6	mm
Cells in x-direction, n_x	50	–
Cells in y-direction, n_y	24	–
Cells in z-direction, n_z	24	–
Timestep, Δt	5	μs
Filter width, δ_f	700	μm
Number of particles, n_p	101,568	–
Particle diameter, d_p	200	μm
Particle density, ρ_p	2600	kg/m^3
Coefficient of restitution, e	0.8	–
Coefficient of friction, μ_f	0.1	–



(a) Pressure drop normalized by the weight of the bed per unit volume. Pressure gradient measured in the simulation (circles), drag given by Eq. 76 computed using the volume fraction at minimum fluidization (dashed line), weight of the bed given by Eq. 77 (broken line).



(b) Normalized particle momentum balance. Gravity (squares), pressure drop (circles), drag (diamonds), collisions (triangles).

Fig. 13. Balance of forces during onset of minimum fluidization.

overall flow pattern. In case B the jet oscillates back and forth due to the production of gas bubbles alongside the spout, as is observed in both experiment and simulation. Fig. 16 shows qualitative agreement between the experiment and simulation for Case B.

Link et al. [65,66] used particle image velocimetry (PIV) to study the particle flow and obtained time-averaged vertical particle flux profiles, $\Phi_{p,x}$, at a height of 0.13 m for each case. A comparison of the particle flux profiles is presented in Fig. 17. Although the results fail to match the data at the region near the wall, overall very good agreement is shown in both cases. It should be noted that simulations by Link et al. [66] show discrepancies in $\Phi_{p,x}$ at the duct center and regions near the wall. It has already been discussed in Section 2.5 that several challenges exist when modeling regions near the walls. In particular, key assumptions in the mathematical derivation are violated close to solid boundaries, and a drag model that is aware of walls could alleviate this. Furthermore, a proper prediction of the volume fraction within a particle diameter away from the wall is very difficult to obtain. A Neumann boundary condition on this quantity is used for robustness, though a detailed sub-grid model might be necessary to capture the proper physics.

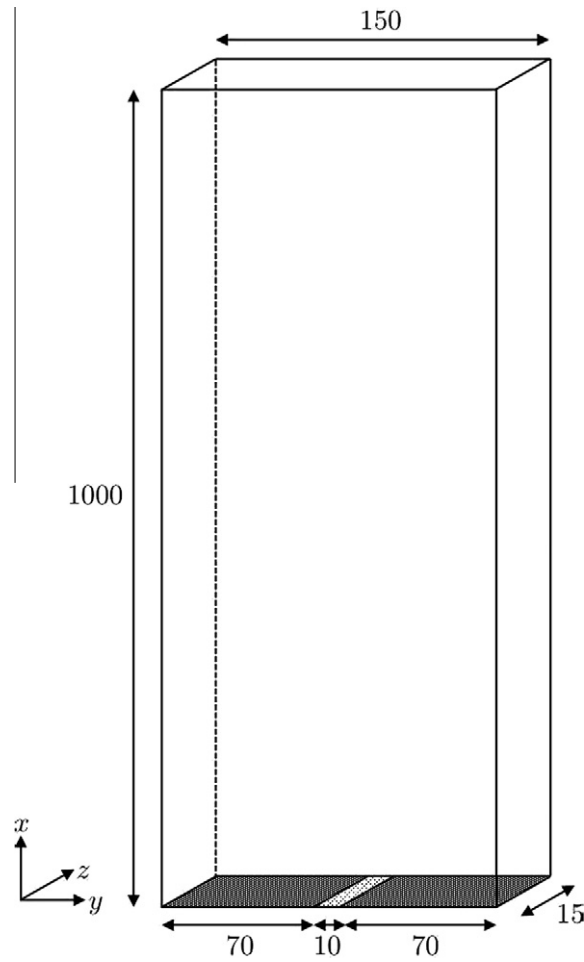


Fig. 14. Schematic of spout-fluid bed (not to scale), with dimensions in mm [65].

Table 3

Simulation parameters for the spout fluidization cases.

Height	0.75	m		
Width	0.15	m		
Depth	0.015	m		
Cells in x -direction, n_x	300	–		
Cells in y -direction, n_y	30	–		
Cells in z -direction, n_z	3	–		
Timestep, Δt	5	μs		
Filter width, δ_f	7.5	mm		
Number of particles, n_p	24,500	–		
Particle diameter, d_p	2.5	mm		
Particle density, ρ_p	2526	kg/m^3		
Spring constant, k	4,538	kg/s^2		
Coefficient of restitution, e	0.9	–		
Coefficient of friction, μ_f	0.1	–		
Case	A	B		
Background velocity	1.5	m/s	3.0	m/s
Spout velocity	30	m/s	20	m/s

6.3. Segregation dynamics

Segregation in a bidisperse fluidized bed was studied and compared to experiments performed by Goldschmidt et al. [67]. A bed of small and large glass beads was fluidized and the time evolution of both axial and lateral segregation was calculated.

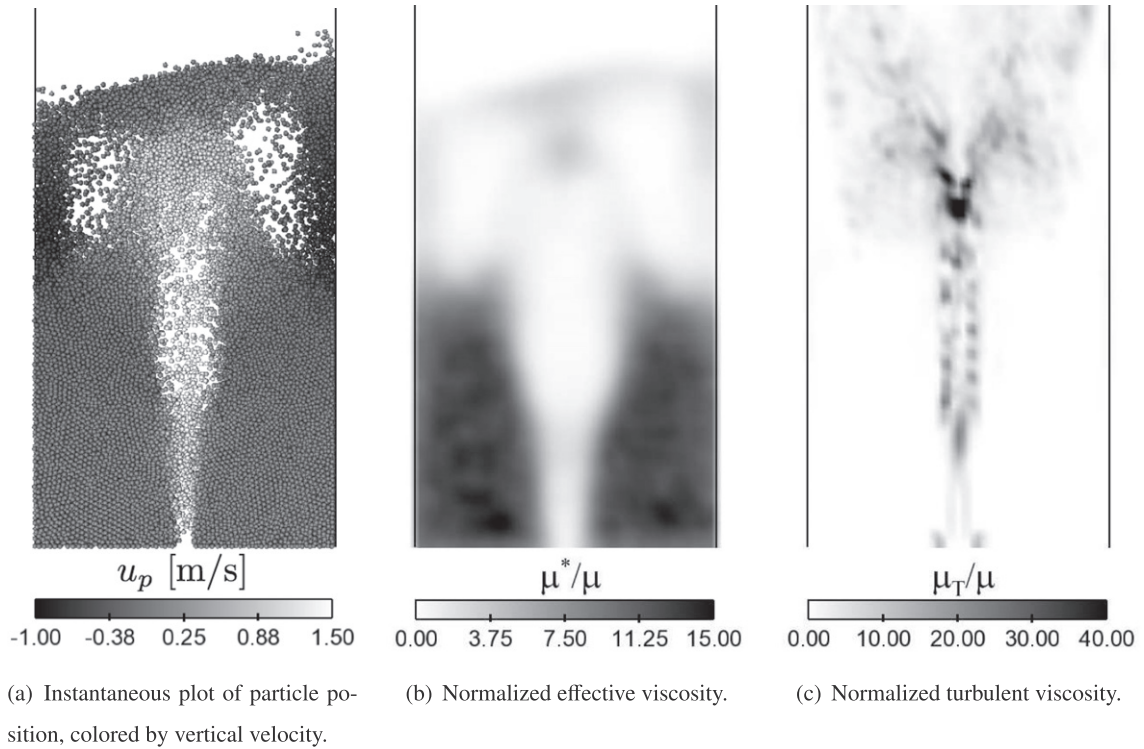


Fig. 15. Instantaneous snapshot of case A of the spout fluidization simulation and corresponding sub-grid viscosity models.

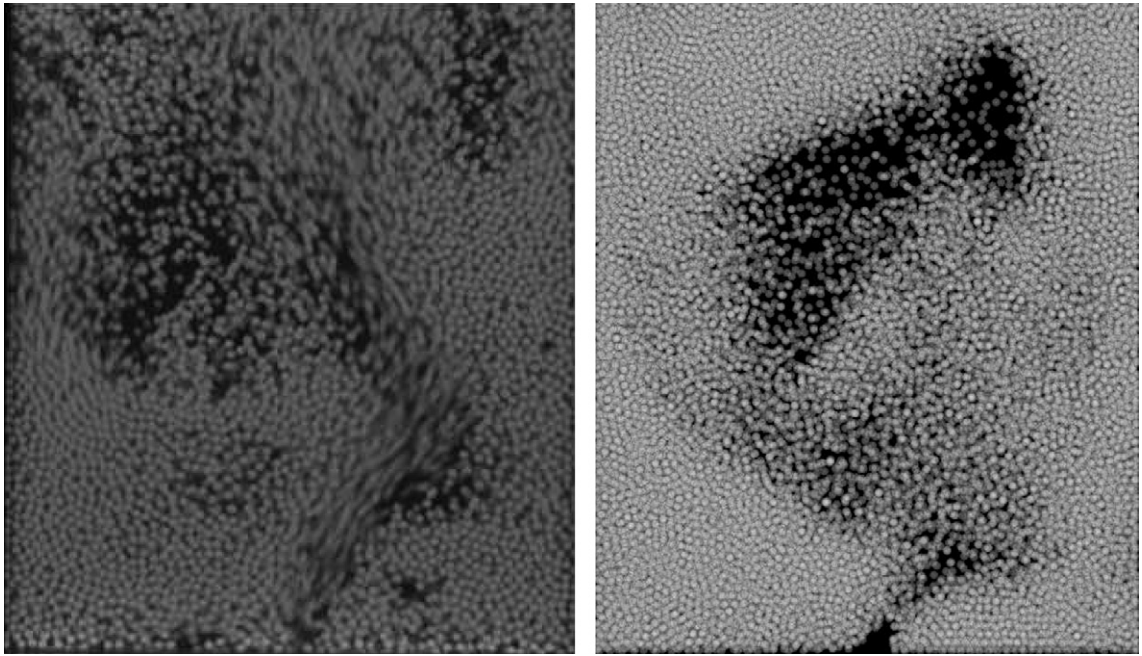


Fig. 16. Instantaneous particle location showing flapping motion from experiment [65] (left) and simulation (right) for Case B.

In gas–solid fluidized beds, segregation occurs at fluidization velocities close to minimum fluidization. Smaller particles tend to be more dynamic and migrate to the top of the bed. The experiments were conducted in a pseudo two-dimensional duct as shown in Fig. 14. Glass particles with diameters of 1.5 and 2.5 mm were fluidized with air at a fluidization velocity of

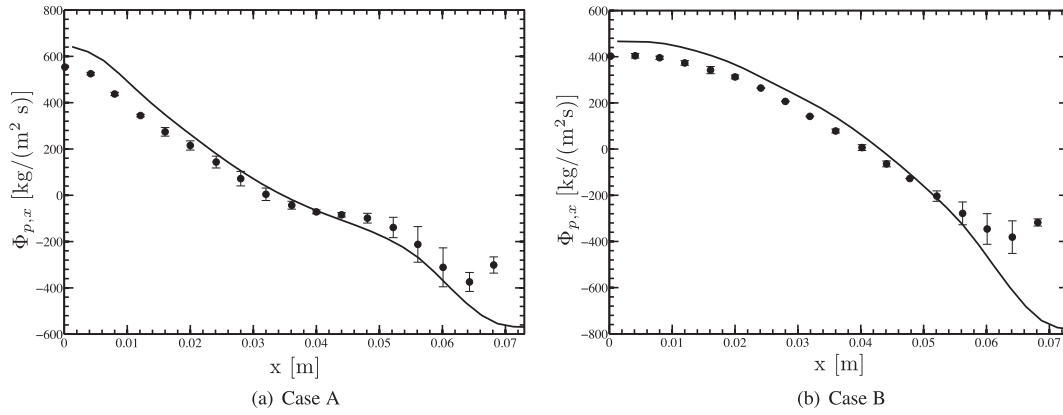


Fig. 17. Time-averaged particle flux profiles at a height of 0.13 m for two different cases. Simulations (solid lines), experiment [66] (circles).

$u_{in} = 1.3$ m/s, corresponding to $0.78U_{mf}$ and $1.25U_{mf}$, respectively. The bed consists of a mass fraction of small particles $\chi_{small} = 0.25$. Simulation parameters are given in Table 4.

The degree of segregation s is defined as

$$s = \frac{S - 1}{S_{max} - 1}, \quad (78)$$

where $S = h_{small}/h_{large}$, $S_{max} = \frac{2-\chi_{small}}{1-\chi_{small}}$. The measure s is referred to as the percentage of axial segregation, which equals 0 for a perfectly mixed bed, and 1 when the mixture is completely segregated. h_{small} and h_{large} refer to the mean height of the small and large particles, respectively. The mean height of the small particles is given by $h_{small} = \frac{\sum_{i=1}^{N_{p,small}} x_{i,small}}{N_{p,small}}$, where $x_{i,small}$ is the x -position of the i th small particle, and $N_{p,small}$ is the number of small particles. h_{large} is defined in a similar way for large particles. When calculating lateral segregation, h_{small} and h_{large} are taken to be the average y -distances from the center of the bed.

Initially, the bed is at rest and uniformly mixed. At $t = 0$ a uniform inflow is introduced at the bottom of the bed. The simulation runs for 60 s, and segregation rates are compared with experimental data. Originally, walls were present in all directions and the bed was found to remain at rest with the given inflow velocity. With three cells across the depth of the bed, a proper boundary layer could not be established with the imposed no-slip boundary condition, which led to a mismatch in effective U_{mf} , preventing fluidization to occur with the experimental parameters. This was alleviated by removing the walls in the z -direction, allowing the bed to fluidize with the given inflow. Although wall effects may be significant, no significant motion was observed with the particles normal to the z plane. Fig. 18 shows instantaneous snapshots of particle position during the simulation. The segregation rate is plotted in Fig. 19, which shows good agreement with the experiments, although the segregation rate seems to taper off in the simulation while the experiment data shows a continuous rise. The lack of walls in the z -directions might explain this discrepancy. Note that at such low fluidization velocities, the solution becomes very sensitive to the drag model, which does not account for walls appropriately.

Table 4

Simulation parameters for the bidisperse fluidized bed.

Height	0.45	m		
Width	0.15	m		
Depth	0.015	m		
Cells in x -direction, n_x	90	–		
Cells in y -direction, n_y	30	–		
Cells in z -direction, n_z	3	–		
Timestep, Δt	$1e-4$	s		
Inlet superficial velocity, u_{in}	1.3	m/s		
Filter width, δ_f	7.5	mm		
Particles	Large		Small	
Number, n_p	17940	–	27720	–
Diameter, d_p	2.5	mm	1.5	mm
Particle density ρ_p	2525	kg/m ³	2525	kg/m ³
Minimum fluidization velocity, U_{mf}	0.78	m/s ²	1.25	m/s ²
Spring constant, k	45.3	kg/s ²	9.8	kg/s ²
Coefficient of restitution, e	0.9	–	0.9	–
Coefficient of friction, μ_f	0.1	–	0.1	–

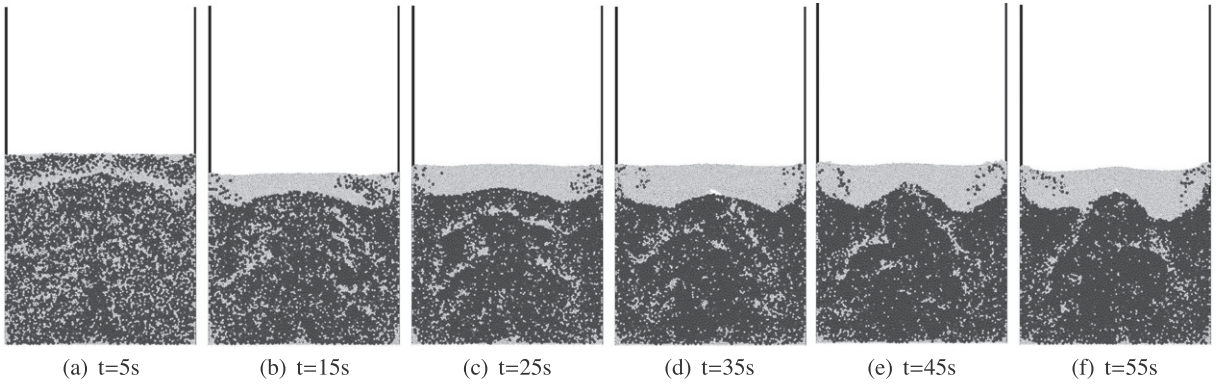


Fig. 18. Segregation of a bidisperse fluidized bed. Large and small particles are shaded dark and light, respectively.

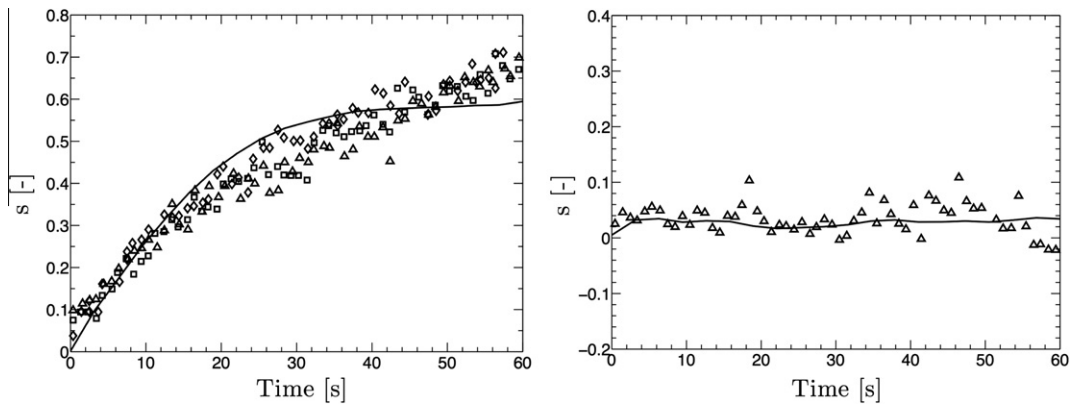


Fig. 19. Axial segregation rate (left) and lateral segregation rate (right) predicted numerically (solid line) compared to experiments [67] (symbols).

6.4. Lab-scale fluidized bed reactor

NREL is using a 4-inch fluidized bed reactor to study gasification of biomass [68]. The reactor, shown in Fig. 20a, consists of a freeboard to prevent the particles from escaping the top, a downward-angled feed tube to load the biomass particles, an exit valve to prevent pressure buildup, pressure taps at different intervals along the side of the reactor, and a distributor plate at the bottom. A simulation was conducted as if all the ports were closed off, considering only a cylindrical boundary with a freeboard, as shown in Fig. 20b. The geometry was taken to be half the scale of the physical reactor, including 15.6 million particles and approximately 20 million grid cells. Fluid properties were taken to be those of nitrogen at room temperature. The simulation was performed using 576 cores for three weeks in order to obtain flow statistics. Simulation parameters are given in Table 5.

In order to fluidize the bed material while preventing back flow, gas is forced through a distributor plate pierced with numerous small holes. Franka and Heindel [69] have shown that the superficial gas velocity affects the region directly above the distributor plate, leading to regions of high gas volume fraction extending from the distributor plate into the bed. Jets are created by a high local gas velocity passing through each hole, potentially leading to non-uniform fluidization. This non-uniform flow pattern affects the bubble characteristics within the bed [69], and has therefore been introduced in the simulation. The gas jets are modeled as Gaussian functions with a characteristic width D_{jet} , distributed along a Cartesian grid, shown in Fig. 20c. The inflow is given by

$$\sum_{j=1}^{n_{jet}} e^{-\left(\frac{(y-y_j)^2}{0.5D_{jet}^2} + \frac{(z-z_j)^2}{0.5D_{jet}^2}\right)}, \quad (79)$$

where the j th jet is centered at (x_{in}, y_j, z_j) , x_{in} is the location of the inlet, and n_{jet} is the number of jets. The inflow is then re-scaled to match the superficial gas velocity.

The particle size distribution is created from a random log-normal distribution generator. Given a mean diameter $\langle d_p \rangle$ and a standard deviation σ_d , the standard deviation and mean for a log-normal random variable are

$$\sigma_{d,log} = \sqrt{\ln(1 + (\sigma_d/\langle d_p \rangle)^2)}, \quad (80)$$

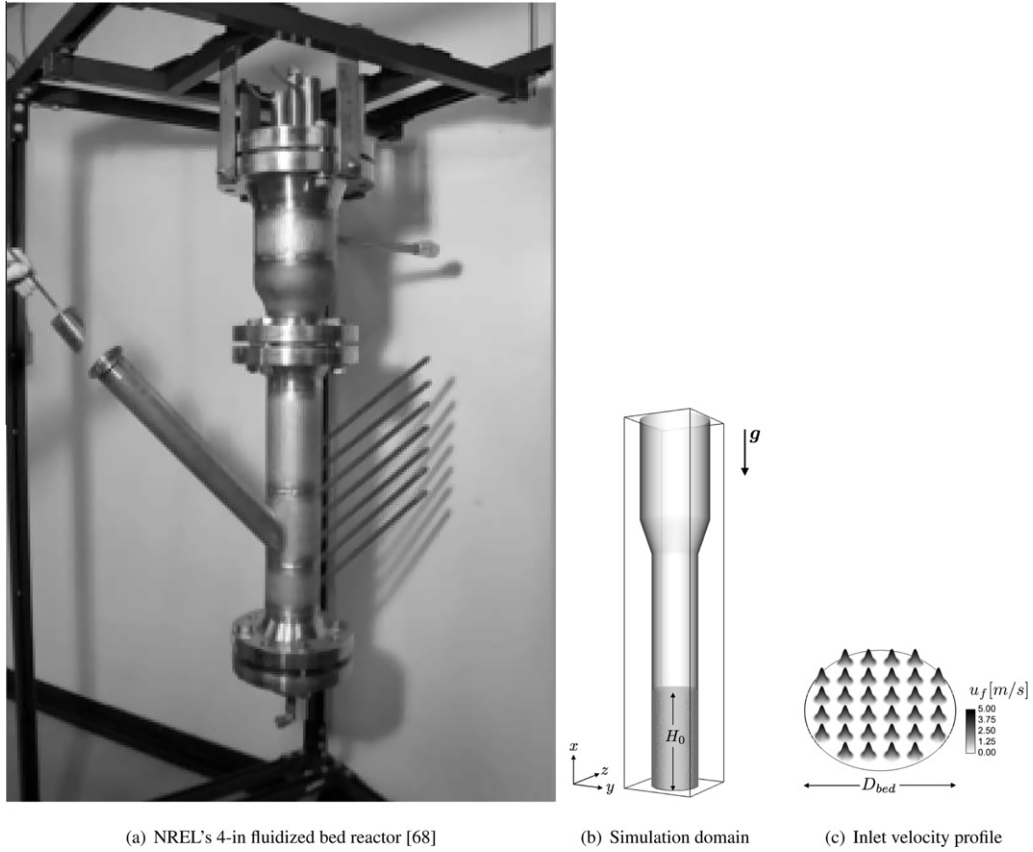


Fig. 20. Experimental and computational configuration of a lab-scale fluidized bed reactor.

Table 5
Simulation parameters for lab-scale fluidized bed reactor.

Domain length in x	m	0.4
Domain length in y	m	0.079
Domain length in z	m	0.079
Cells in x -direction, n_x	800	–
Cells in y -direction, n_y	158	–
Cells in z -direction, n_z	158	–
Bed diameter, D_{bed}	m	0.051
Initial bed height, H_0	m	0.106
Gas density, ρ_f	kg/m ³	1.13
Gas viscosity, μ	kg/m s	1.77×10^{-5}
Particle density, ρ_p	kg/m ³	3300
Inlet superficial velocity, u_{in}	m/s	0.494
Orifice diameter, D_{jet}	m	0.002
Number of holes, n_{jet}	–	32
Number of particles, n_p	–	15.6 million
Mean particle diameter, $\langle d_p \rangle$	μm	200
Min particle diameter, d_{min}	μm	102
Max particle diameter, d_{max}	μm	410
Particle standard deviation, σ_d	μm	70
Spring constant, k	kg/s ²	3.0
Coefficient of restitution, e	–	0.8
Coefficient of friction, μ_f	–	0.092
Filter width, δ_f	μm	$820 (2d_{max})$

and

$$\langle d_p \rangle_{\log} = \ln(\langle d_p \rangle) - \sigma_{d,\log}^2/2, \quad (81)$$

respectively. Using these parameters for the mean and standard deviation of a random normal distribution then exponentiating this value yields the appropriate particle diameter. A log-normal distribution was created to match the particle size data provided by NREL, as illustrated in Fig. 21.

Fig. 22 shows a volumetric rendering of the gas velocity and an iso-surface of $\varepsilon_f = 0.92$ at a simulation time of 1.5 s. The gas jets at the inlet directly lead to the formation of bubbles, which expand and accelerate as they rise until they reach the surface of the bed. The bubbles tend to form in the middle of the bed, leading to a recirculation of particles down the sides of the walls. It is apparent in the figure that the gas velocity is greatest within the bubbles. Instantaneous plots of fluid volume fraction and velocity are presented in Fig. 23a and c, further exemplifying this correlation. Above the bed, the freeboard is seen to induce turbulence in the gas phase, as illustrated in Fig. 23c.

The simulation was run until statistical stationarity was reached. Time-averaged fields of the gas phase volume fraction and velocity are given in Fig. 23b and d, respectively. The highest particle packing is observed at the walls, indicating the tendency for bubbles to form in the middle of the bed. The mean gas velocity is negative in this region as a result of entrainment from the particles. Radial profiles of particle velocities at various heights in the bed are given in Fig. 24a, which clearly identifies recirculation at the walls due to rising bubbles. The presence of the bubbles lead to periodic oscillations in the bed height, as depicted in Fig. 24b. The bed height is based on the 90th-percentile of particle height, and indicates convergence in macroscale motions of the bed.

Pepiot and Desjardins [33] showed that large variations of residence times are directly related to the local gas volume fraction, and thus bubbling has the potential to affect the chemical processes in reactive beds. In order to develop more in-

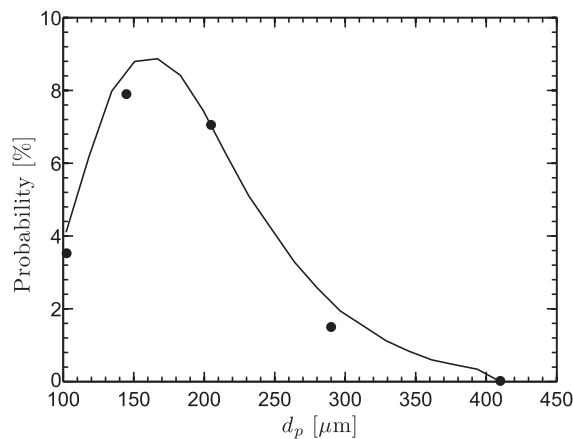


Fig. 21. Particle size distribution. Data provided by NREL (circles), simulation data (solid line).

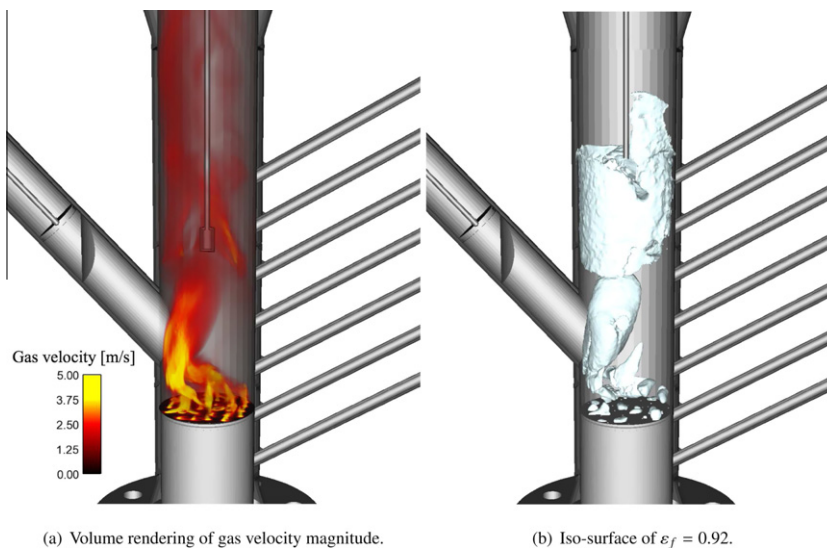


Fig. 22. Simulation of a two-inch fluidized bed reactor at $t = 1.5$ s.

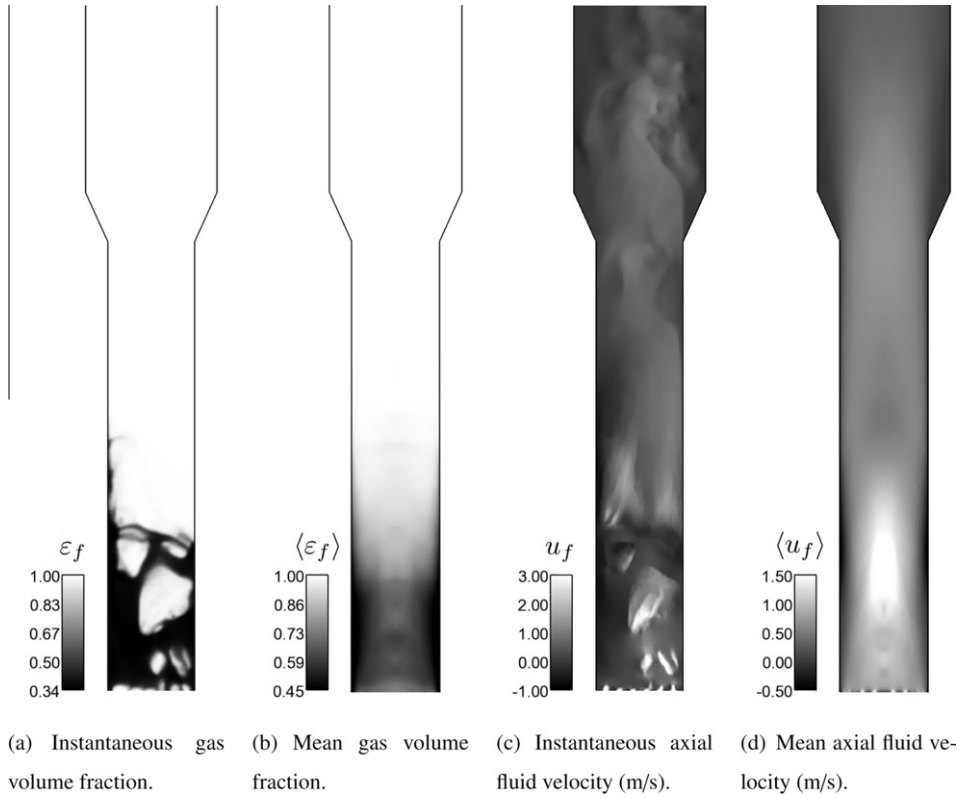


Fig. 23. Instantaneous and time-averaged statistics of the two-inch fluidized bed reactor.

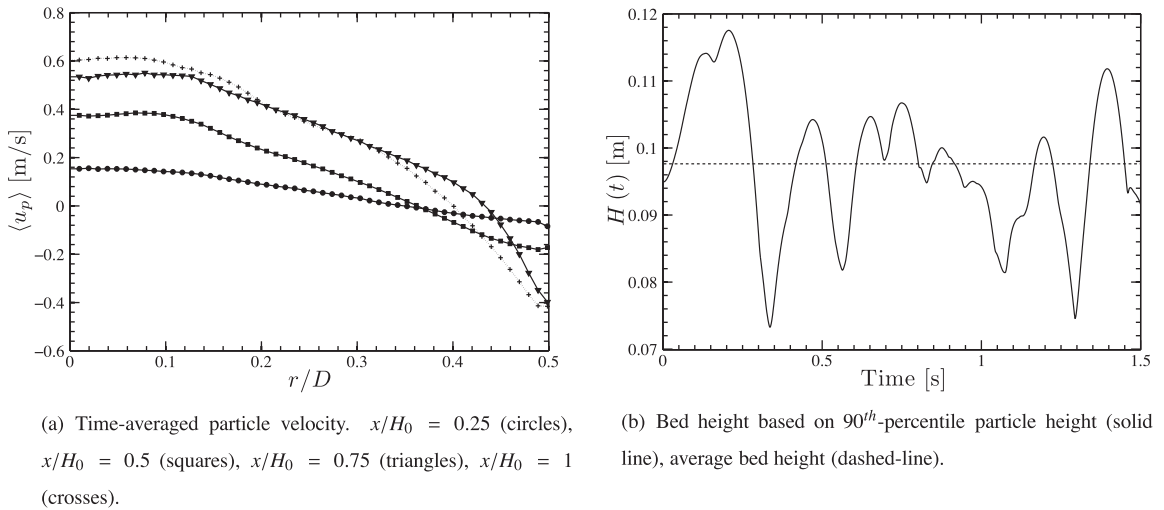


Fig. 24. Particle statistics of the two-inch fluidized bed reactor.

sight on the role of bubbles in fluidized bed dynamics, the band growth algorithm of Herrmann et al. [70] was implemented to identify and track individual bubbles. For details on the bubble tracking scheme, see Pepiot and Desjardins [33]. With this tool, the volume of each identified structure is calculated and an effective diameter is computed by assuming a perfect sphere. The mean bubble diameter was found to be 18% of the bed diameter, and on average 8 bubbles existed within the bed at any time. Bubbles tend to form at the bottom of the bed and grow in size and velocity as they rise. The bubble height was compared with Darton's correlation [71] in Fig. 25, which gives the bubble diameter as a function of height by

$$D_b = 0.54(U_{in} - U_{mf})^{0.4} \left(h + 4\sqrt{A_0} \right)^{0.8} g^{-0.2}, \quad (82)$$

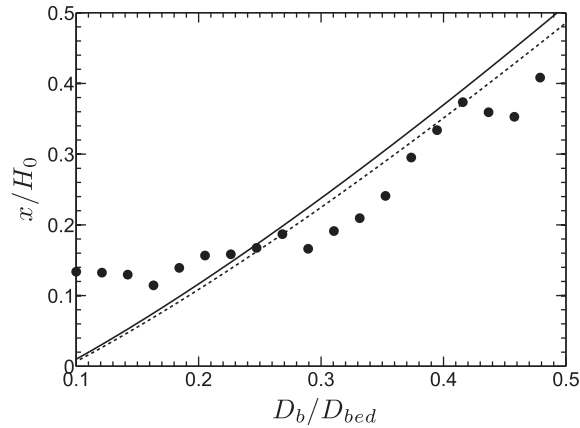


Fig. 25. Bubble height as a function of diameter compared to Darton's correlation [71]. Correlation corresponding to $u_{in} = 6U_{mf}$ (dashed-line), correlation corresponding to $u_{in} = 11U_{mf}$ (solid line).

where A_0 is the catchment area of the distributor plate (area of the plate per orifice). For this case, $A_0 = \pi D_{jet}^2/4$. Due to the non-linearity of the drag, U_{mf} cannot be determined *a priori*, as this would require knowledge of the voidage at fluidization. However, this value of the volume fraction can be postulated, allowing us to compare the correlation with the simulation results. Typically, the voidage at minimum fluidization, ε_{mf} , is taken to be that of a packed bed of uniform spheres in cubic arrangement [13], giving $\varepsilon_{mf} = 1 - \pi/6$. Using this value, u_{in} corresponds to $6U_{mf}$. However, when calculating U_{mf} for experimental runs, NREL uses a value of $\varepsilon_{mf} = 0.405$ to account for polydispersity, corresponding to $11U_{mf}$. It is shown in the figure that the value of U_{mf} does not change the correlation substantially, and it is assumed the simulation fluidization velocity falls between these two values. Due to the highly turbulent nature of the flow, bubble coalescence and breakup lead to numerous small diameter bubbles throughout the bed. With this large number of small bubbles throughout the bed, smaller bubbles are found to deviate from the correlation, although overall we see good agreement between the simulation data and correlation.

7. Conclusions

A numerical strategy for simulating gas–solid particle-laden flows in an Eulerian–Lagrangian framework was developed and tested. Starting from the point wise equations of motion that govern each phase, a volume filtered description initially introduced by Anderson and Jackson [42] was presented then extended for chemically reacting flows. All assumptions and models were stated explicitly and discussed. A two-step filtering process was derived in the context of the volume filtering approach that converges under mesh refinement and allows for cell sizes smaller than the particle diameter if necessary. A conservative immersed boundary method was used for modeling complex geometries. Due to the inherent nature of this approach, detailed information of the particle's distance to the nearest boundary was readily available. Particles near walls were mirrored across the boundary to yield appropriate boundary conditions. A soft-sphere collision model was employed, which was modified for parallel efficiency. Replacing the full linear model for tangential collisions with the static friction model was shown to be a good compromise for cost, as the simpler model does not require storing the collision history. A radius of influence proportional to the relative collisional velocity was used to allow for robust collisions while recovering random close packing for a bed at rest.

Several lab-scale simulations of experiments were conducted for validation purposes. Choosing appropriate values for the simulation timestep that account for both the particle CFL as well as particle stiffness is necessary to prevent excessive overlap leading to unphysical packing while correctly capturing particle collisions. Particle flows close to the minimum fluidization velocity as well as highly turbulent flows were simulated and provided overall good agreement with experiments. For systems with large gas phase Reynolds numbers, a dynamic Smagorinsky eddy viscosity model was used to close the residual stress tensor \mathbf{R}_u . Although interactions between the viscosity models are unclear, a distinct segregation was observed where the effective viscosity dominates in regions with greatest solid packing, while the turbulent viscosity model dominates in the particle-free turbulent regions. Finally, a two-inch fluidized bed reactor was simulated, consisting of 15.6 million particles on approximately 20 million grid cells. Mean statistics of the flow were computed and a band-growth algorithm was used to study bubbling in the bed. Gas jets form at the distributor plate that lead to bubbles that grow in size as they approach the surface of the bed, displaying good agreement with experimental correlations.

It was shown that several challenges exist for properly treating regions near walls. When deriving the volume filtered equations, underlying assumptions are violated when the filter radius is larger than the distance between particles and solid boundaries. Drag models for spherical particles that account for wall interactions are necessary for proper closure. An accurate computation of the volume fraction near walls that accounts for the distribution of distances between particle surfaces

is also needed to capture the correct physical behavior. When dealing with high Reynolds number flows, a standard eddy viscosity model does not properly account for turbulence modulation by particles, and significant physical processes might be compromised. Accounting for the exchange of angular momentum from the fluid phase, leading to lift forces that are otherwise neglected, is of interest as well. These shortcomings could be addressed by improved models developed using particle-resolved DNS data. With these considerations, the approaches described in this paper are shown to work well on a range of cases. A filter length scale was chosen such that microscale processes (e.g., collisions and drag) were handled with well-accepted models from the literature and mesoscale features were resolved explicitly, allowing to properly capture important multiphase phenomenon including particle segregation and bubbling. However, this segregation in length scales limits the simulation strategy to $\mathcal{O}(10^8)$ Lagrangian particles, and more work is required for scale-up. Furthermore, the efficiency of the proposed methodology, in terms of accuracy versus computational expense, remains to be investigated and compared to that of existing approaches.

Acknowledgements

This work has been supported in part by the Office of the Biomass Program of the US Department of Energy under Contract XCO-0-40641-01 with the National Renewable Energy Laboratory. We also gratefully acknowledge Dr. Perrine Pepiot's initial contributions towards integrating and developing the particle tracking framework with NGA, as well as all ongoing discussions.

References

- [1] R. Fox, Large-eddy-simulation tools for multiphase flows, *Annual Review of Fluid Mechanics* (2012).
- [2] S. Balachandar, J. Eaton, Turbulent dispersed multiphase flow, *Annual Review of Fluid Mechanics* 42 (2010) 111–133.
- [3] C. Crowe, J. Schwarzkopf, M. Sommerfeld, Y. Tsuji, *Multiphase flows with droplets and particles*, CRC press, 2011.
- [4] N. Deen, M. Van Sint Annaland, M. Van der Hoef, J. Kuipers, Review of discrete particle modeling of fluidized beds, *Chemical Engineering Science* 62 (2007) 28–44.
- [5] N. Patankar, D. Joseph, Modeling and numerical simulation of particulate flows by the Eulerian-Lagrangian approach, *International Journal of Multiphase Flow* 27 (2001) 1659–1684.
- [6] M. Van der Hoef, M. Ye, M. van Sint Annaland, A. Andrews, S. Sundaresan, J. Kuipers, Multiscale modeling of gas-fluidized beds, *Advances in Chemical Engineering* 31 (2006) 65–149.
- [7] R. Fox, Introduction and fundamentals of modeling approaches for polydisperse multiphase flows, *Multiphase Reacting Flows: Modelling and Simulation* (2007) 1–40.
- [8] M. Pai, S. Subramaniam, A comprehensive probability density function formalism for multiphase flows, *Journal of Fluid Mechanics* 628 (2009) 181–228.
- [9] S. Chapman, T. Cowling, *The mathematical theory of non-uniform gases: an account of the kinetic theory of viscosity, thermal conduction, and diffusion in gases*, Cambridge Univ Press, 1991.
- [10] C. Cercignani, R. Illner, M. Pulvirenti, *The mathematical theory of dilute gases*, volume 106, Springer, 1994.
- [11] H. Struchtrup, *Macroscopic transport equations for rarefied gas flows: approximation methods in kinetic theory*, Springer, Verlag, 2005.
- [12] J. Kuipers, K. Van Duin, F. Van Beckum, W. Van Swaaij, A numerical model of gas-fluidized beds, *Chemical Engineering Science* 47 (1992) 1913–1924.
- [13] D. Gidaspow, *Multiphase flow and fluidization: continuum and kinetic theory descriptions*, Academic, Pr, 1994.
- [14] D. Zhang, A. Prosperetti, Averaged equations for inviscid disperse two-phase flow, *Journal of Fluid Mechanics* 267 (1994) 185–220.
- [15] E. Peirano, B. Leckner, Fundamentals of turbulent gas-solid flows applied to circulating fluidized bed combustion, *Progress in Energy and Combustion Science* 24 (1998) 259–296.
- [16] J. Pritchett, T. Blake, S. Garg, A numerical model of gas fluidized beds, in: *AIChE Symp. Ser.*, volume 176, pp. 134–148.
- [17] J. Bouillard, R. Lyczkowski, D. Gidaspow, Porosity distributions in a fluidized bed with an immersed obstacle, *AIChE Journal* 35 (1989) 908–922.
- [18] H. Enwald, E. Peirano, A. Almstedt, Eulerian two-phase flow theory applied to fluidization, *International Journal of Multiphase Flow* 22 (1996) 21–66.
- [19] Y. Tsuo, D. Gidaspow, Computation of flow patterns in circulating fluidized beds, *AIChE Journal* 36 (1990) 885–896.
- [20] S. Benyahia, H. Arastoopour, T. Knowlton, H. Massah, Simulation of particles and gas flow behavior in the riser section of a circulating fluidized bed using the kinetic theory approach for the particulate phase, *Powder Technology* 112 (2000) 24–33.
- [21] H. Arastoopour, Numerical simulation and experimental analysis of gas/solid flow systems: 1999 Fluor-Daniel plenary lecture, *Powder Technology* 119 (2001) 59–67.
- [22] D. Gidaspow, J. Jung, R. Singh, Hydrodynamics of fluidization using kinetic theory: an emerging paradigm: 2002 Fluor-Daniel lecture, *Powder Technology* 148 (2004) 123–141.
- [23] O. Desjardins, R. Fox, P. Villedieu, A quadrature-based moment method for dilute fluid-particle flows, *Journal of Computational Physics* 227 (2008) 2514–2539.
- [24] O. Simonin, Prediction of the dispersed phase turbulence in particle-laden jets, in: *4th International Symposium On gas-solid flows*, Portland, pp. 23–26.
- [25] H. Grad, On the kinetic theory of rarefied gases, *Communications on Pure and Applied Mathematics* 2 (1949) 331–407.
- [26] D. Marchisio, R. Fox, Solution of population balance equations using the direct quadrature method of moments, *Journal of Aerosol Science* 36 (2005) 43–73.
- [27] 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.
- [28] C. Yuan, R. Fox, Conditional quadrature method of moments for kinetic equations, *Journal of Computational Physics* (2011).
- [29] G. Bird, *Molecular gas dynamics and the direct simulation of gas flows*, Oxford, United Kingdom: Clarendon Press (Oxford Engineering Science Series, 42 (1995)).
- [30] A. Kitron, T. Elperin, A. Tamir, Monte carlo simulation of gas-solids suspension flows in impinging streams reactors, *International Journal of Multiphase Flow* 16 (1990) 1–17.
- [31] T. Tanaka, S. Yonemura, K. Kiribayashi, Y. Tsuji, Cluster formation and particle-induced instability in gas-solid flows predicted by the DSMC method, *JSME International Journal-Series B – Fluids and Thermal Engineering* 39 (1996) 239–245.
- [32] Y. Tsuji, Multi-scale modeling of dense phase gas-particle flow, *Chemical Engineering Science* 62 (2007) 3410–3418.
- [33] P. Pepiot, O. Desjardins, Numerical analysis of the dynamics of two- and three-dimensional fluidized bed reactors using an Euler-Lagrange approach, *Powder Technology* (2010).

- [34] R. Garg, C. Narayanan, S. Subramaniam, A numerically convergent lagrangian-eulerian simulation method for dispersed two-phase flows, *International Journal of Multiphase Flow* 35 (2009) 376–388.
- [35] H. Salman, M. Soteriou, Lagrangian simulation of evaporating droplet sprays, *Physics of Fluids* 16 (2004) 4601.
- [36] P. Cundall, O. Strack, A discrete numerical model for granular assemblies, *Geotechnique* 29 (1979) 47–65.
- [37] Y. Tsuji, T. Kawaguchi, T. Tanaka, Discrete particle simulation of two-dimensional fluidized bed, *Powder Technology* 77 (1993) 79–87.
- [38] B. Hoomans, J. Kuipers, W. Briels, W. Van Swaaij, Discrete particle simulation of bubble and slug formation in a two-dimensional gas-fluidised bed: a hard-sphere approach, *Chemical Engineering Science* 51 (1996) 99–118.
- [39] C. Campbell, C. Brennan, Computer simulation of granular shear flows, *Journal of Fluid Mechanics* 151 (1985) 167–88.
- [40] P. Peplot, O. Desjardins, Direct numerical simulation of dense particle-laden flows using a conservative immersed boundary technique, in: *Proceedings of the Summer Program*, p. 323.
- [41] O. Desjardins, G. Blanquart, G. Balarac, H. Pitsch, High order conservative finite difference scheme for variable density low Mach number turbulent flows, *Journal of Computational Physics* 227 (2008) 7125–7159.
- [42] T. Anderson, R. Jackson, Fluid mechanical description of fluidized beds. Equations of motion, *Industrial and Engineering Chemistry Fundamentals* 6 (1967) 527–539.
- [43] D. Zhang, A. Prosperetti, Momentum and energy equations for disperse two-phase flows and their closure for dilute suspensions, *International Journal of Multiphase Flow* 23 (1997) 425–453.
- [44] N. Peters, *Turbulent combustion*, Cambridge university press, 2000.
- [45] C. Crowe, On models for turbulence modulation in fluid-particle flows, *International Journal of Multiphase Flow* 26 (2000) 719–727.
- [46] C. Rogers, J. Eaton, The effect of small particles on fluid turbulence in a flat-plate, turbulent boundary layer in air, *Physics of Fluids A: Fluid Dynamics* 3 (1991) 928.
- [47] Y. He, S. Qin, C. Lim, J. Grace, Particle velocity profiles and solid flow patterns in spouted beds, *The Canadian Journal of Chemical Engineering* 72 (1994) 561–568.
- [48] M. Germano, U. Piomelli, P. Moin, W. Cabot, A dynamic subgrid-scale eddy viscosity model, *Physics of Fluids A: Fluid Dynamics* 3 (1991) 1760.
- [49] D. Lilly, A proposed modification of the germano subgrid-scale closure method, *Physics of Fluids A: Fluid Dynamics* 4 (1992) 633.
- [50] C. Meneveau, T. Lund, W. Cabot, A lagrangian dynamic subgrid-scale model of turbulence, *Journal of Fluid Mechanics* 319 (1996) 353–385.
- [51] A. Di Renzo, F. Di Maio, Homogeneous and bubbling fluidization regimes in DEM-CFD simulations: Hydrodynamic stability of gas and liquid fluidized beds, *Chemical Engineering Science* 62 (2007) 116–130.
- [52] L. Gibilaro, K. Gallucci, R. Di Felice, P. Pagliai, On the apparent viscosity of a fluidized bed, *Chemical Engineering Science* 62 (2007) 294–300.
- [53] P. Saffman, The lift on a small sphere in a slow shear flow, *Journal of Fluid Mechanics* 22 (1965) 385–400.
- [54] S. Tenneti, R. Garg, S. Subramaniam, Drag law for monodisperse gas-solid systems using particle-resolved direct numerical simulation of flow past fixed assemblies of spheres, *International Journal of Multiphase Flow* 37 (2011) 1072–1092.
- [55] D. Snider, P. O'Rourke, M. Andrews, Sediment flow in inclined vessels calculated using a multiphase particle-in-cell model for dense particle flows, *International Journal of Multiphase Flow* 24 (1998) 1359–1382.
- [56] R. Garg, C. Narayanan, D. Lakehal, S. Subramaniam, Accurate numerical estimation of interphase momentum transfer in Lagrangian-Eulerian simulations of dispersed two-phase flows, *International Journal of Multiphase Flow* 33 (2007) 1337–1364.
- [57] M. Meyer, A. Devesa, S. Hickel, X. Hu, N. Adams, A conservative immersed interface method for Large-Eddy Simulation of incompressible flows, *Journal of Computational Physics* (2010).
- [58] G. Treece, R. Prager, A. Gee, Regularised marching tetrahedra: improved iso-surface extraction, *Computers and Graphics* 23 (1999) 583–598.
- [59] A. Di Renzo, F. Di Maio, Comparison of contact-force models for the simulation of collisions in DEM-based granular flow codes, *Chemical Engineering Science* 59 (2004) 525–541.
- [60] G. Scott, D. Kilgour, The density of random close packing of spheres, *Journal of Physics D: Applied Physics* 2 (1969) 863.
- [61] C. Pierce, Progress-variable approach for large-eddy simulation of turbulent combustion, Ph.D. thesis, Citeseer, 2001.
- [62] J. Kim, P. Moin, Application of a fractional-step method to incompressible navier-stokes equations, *Journal of Computational Physics* 59 (1985) 308–323.
- [63] P. Popov, H. Wang, S. Pope, Specific volume coupling and convergence properties in hybrid particle/finite volume algorithms for turbulent reactive flows, *Journal of Computational Physics* (2012).
- [64] P. Brown, G. Byrne, A. Hindmarsh, et al, Vode: A variable coefficient ode solver, *SIAM J. Sci. Stat. Comput* 10 (1989) 1038–1051.
- [65] J. Link, C. Zeilstra, N. Deen, H. Kuipers, Validation of a discrete particle model in a 2D spout-fluid bed using non-intrusive optical measuring techniques, *The Canadian Journal of Chemical Engineering* 82 (2004) 30–36.
- [66] J. Link, L. Cuyper, N. Deen, J. Kuipers, Flow regimes in a spout-fluid bed: A combined experimental and simulation study, *Chemical Engineering Science* 60 (2005) 3425–3442.
- [67] M. Goldschmidt, J. Link, S. Mellema, J. Kuipers, Digital image analysis measurements of bed expansion and segregation dynamics in dense gas-fluidised beds, *Powder Technology* 138 (2003) 135–159.
- [68] K. Gaston, M. Jarvis, P. Peplot, K. Smith, W. Frederick, M. Nimlos, Biomass pyrolysis and gasification of varying particle sizes in a fluidized bed reactor, *Energy and Fuels* (2011).
- [69] N. Franka, T. Heindel, Local time-averaged gas holdup in a fluidized bed with side air injection using X-ray computed tomography, *Powder Technology* 193 (2009) 69–78.
- [70] M. Herrmann, A parallel Eulerian interface tracking/Lagrangian point particle multi-scale coupling procedure, *Journal of Computational Physics* 229 (2010) 745–759.
- [71] R. Darton, R. LaNauze, J. Davidson, D. Harrison, Bubble growth due to coalescence in fluidised beds, *Chemical Engineering Research and Design* 55 (1977) 274–280.