Titre: Title:	Toward high-order CFD-DEM: development and validation
Auteurs: Authors:	Toni El Geitani, Shahab Golshan, & Bruno Blais
Date:	2023
Type:	Article de revue / Article
Référence: Citation:	El Geitani, T., Golshan, S., & Blais, B. (2023). Toward high-order CFD-DEM: development and validation. Industrial & Engineering Chemistry Research, 62(2), 1141-1159. https://doi.org/10.1021/acs.iecr.2c03546

URL de PolyPublie: PolyPublie URL:	https://publications.polymtl.ca/52602/
Version:	Version finale avant publication / Accepted version Révisé par les pairs / Refereed
Conditions d'utilisation: Terms of Use:	Tous droits réservés / All rights reserved

Titre de la revue: Journal Title:	Industrial & Engineering Chemistry Research (vol. 62, no. 2)
Maison d'édition: Publisher:	ACS
URL officiel: Official URL:	https://doi.org/10.1021/acs.iecr.2c03546
Mention légale: Legal notice:	

Towards High-Order CFD-DEM: Development and Validation

Toni El Geitani[a], Shahab Golshan[a], and Bruno Blais[a,*]

[a]*Research Unit for Industrial Flows Processes (URPEI), Department of Chemical Engineering, École Polytechique de Montréal, PO Box 6079, Stn Centre-Ville, Montréal, QC, Canada, H3C 3A7*

E-mail: bruno.blais@polymtl.ca

Abstract

CFD-DEM is used to simulate solid-fluid systems. DEM models the motion of discrete particles while CFD models the fluid phase. Coupling both necessitates the calculation of the void fraction and the solid-fluid forces resulting in a computationally expensive method. Additionally, evaluating volume-averaged quantities locally restricts particle to cell size ratios limiting the accuracy of the CFD. To mitigate these limitations, we develop a unified finite element CFD-DEM solver which integrates the CFD and DEM solvers into a single software resulting in faster and cheaper coupling between the solvers. It supports dynamically load-balanced parallelization. This allows for more efficient simulations as load balancing ensures the even distribution of workloads among processors; thus, exploiting available resources efficiently. Our solver supports high order schemes; thus, allowing the use of larger elements enhancing the validity and stability of the void fraction schemes while achieving better accuracy. We verify and validate our CFD-DEM solver with a large array of test cases: particle sedimentation, a fluidized bed, the Rayleigh Taylor instability and a spouted bed.

Introduction

Multi-phase flows are prevalent in several industries such as food processing, oil and gas, pharmaceutical industries and many more.[1,2] Understanding the underlying phenomena that control the behavior of such flows is important to develop state of the art equipment for solid-fluid contactors. Solid-fluid flows and particularly solid-gas flows, are characterized based on the number density of the particles. When this number is small, the flow is termed dilute solid-gas flow and the gas dominates the behavior of the flow. In this case, the solid particles exhibit negligible effects on the gas. When the number density is large, the flow is termed dense solid-gas flow. In this case, the movement of the solid phase becomes mainly controlled by particle-particle collisions. When this number density is bounded and thus lies in between the two previously mentioned cases, the flow is referred to as dispersed flow where the solids constitute the dispersed phase and the fluid constitute the continuous phase. In this case, the flow is affected equally by both phases.[3]

We encounter solid-gas flows in several technological applications such as pollutant control systems, combustion systems, and drying systems. Fluidized and spouted beds are among the common industrial applications which deal with solid-gas flows. Through the application of gas flow at the bottom of the bed, and when the gas flow is large enough to exert sufficient drag on the granular material allowing it to overcome gravity, particles fluidize. In the fluidized state, particles behave as a mixer allowing for a higher mass and heat transfer rates between the two phases resulting in a uniform temperature distribution.[4] This results in improved yields and efficiency in chemical and physical processes which employ such beds.

Fluidized and spouted beds differ in the way they operate. In fluidized beds, the fluid is introduced usually at the entire surface of the inlet while in spouted beds, the fluid enters through a small orifice of the base of the bed. This results in different fluidization behaviors. Contrary to a single fluidizing section in a fluidized bed, spouted beds can be divided into three sections. The spout which is the central core through which the fluid flows. The fountain, which contains the particles entrained by the spout, appears above the bed surface. The annulus is the surrounding annular region of the bed where the particles travel down from the fountain.[5]

These beds are complex in their design, building, and operation when compared to other types of beds such as packed beds and stirred tank reactors and are more difficult to scale-up.[6] However, their advantages for the chemical industries make it vital to predict their behavior in order to accomplish design, scale-up, optimization, and troubleshooting of the processes involved.[2] Even though there exist several intrusive and non intrusive experimental techniques to investigate fluidized and spouted beds, their implementation is generally expensive and often infeasible for complex applications. Furthermore, the information that can be extracted from experiments is limited especially at the particle scale. Consequently, modelling can be used to understand the various phenomena occurring in different processes, enabling sensitivity analysis on different input parameters, and testing various configurations and operational conditions at a much cheaper cost.[2]

Among the various modeling approaches, unresolved CFD-DEM is often used to study solid-fluid systems. It is an Eulerian-Lagrangian approach in which the flow field is divided into cells larger than the solid particles' size but sufficiently small to capture the flow field itself. Few commercial and open source CFD-DEM software exist to simulate solid-fluid flows using CFD-DEM. Usually, CFD and DEM software exist separately and are coupled through a coupling interface. Commercial DEM software include EDEM,[7] Rocky[8] and Aspherix[9] while open source DEM software include LIGGGHTS[10] and Yade.[11] Commercial CFD software include COMSOL,[12] Ansys Fluent,[13] StarCCM+[14] among others while open source CFD software include MFiX[15] and OpenFOAM.[16] Additionally, there exists several in-house codes used to publish studies related to solid-fluid flows.[17–19] To the best of our knowledge, MFiX, StarCCM+, and PFC[20] are the only unified CFD-DEM software meaning that both the CFD and DEM solvers are integrated into a single piece of software. However, such software do not support high order elements.Only Aspherix[9] supports load balancing and can be integrated to CFD solvers using the CFDEM interface[21] .

Unresolved CFD-DEM simulations are computationally expensive and possess some major limitations. The time step of integration is limiting as the DEM scheme is usually explicit and requires the particle time step to be very low (between 10^{-7} s to 10^{-4} s, depending on the size and the stiffness of the particles). Another limitation is the size of the mesh cells which should usually be at least three times the particle diameter in CFD-DEM simulations.[2] This is particularly important to ensure that certain void fraction calculation schemes are valid. In order to overcome this limitation, certain analytical schemes have been developed where the cell and particle intersections were determined analytically.[22,23] However, these methods are computationally expensive as they involve the calculation of trigonometric functions at each time step.[23,24] Other non-analytical approaches have been developed such as the divided particle volume methods (DPVM)[25,26] which represents the particle as a porous cube. These methods added spatial smoothing which diffused the void fraction and thus, the presence of particles was weakly felt by the fluid.[27] Additionally, numerical strategies have been developed in order to increase the accuracy of the CFD simulation when coarser grids have to be used to ensure adequate calculation of the void fraction. These include the use of separate grids for CFD and for the void fraction where the void fraction grid is coarser than the CFD grid. Such studies included the use of two level grids[28–30] and three level grids.[31,32] However, the mapping of information between grids has a non negligible computational cost that should be considered. The use of multiple grid is also challenging when the geometry of the simulation domain is complex.

More accuracy is required sometimes on the CFD side which is generally, at best, second-order accurate. Therefore, there is a trade off between accuracy and stability in such situations. As such, we propose high order schemes as an important factor in alleviating these challenges. They allow for better accuracy without decreasing the mesh size. Currently existing CFD-DEM software do not exploit high order schemes as they are more complicated and harder to implement than low order methods, and their computational memory requirement is much greater. The importance of parallelization becomes evident when CFD-DEM can be run on multiple processors to achieve better computational efficiency. Parallelizing CFD-DEM simulations requires paralellizing the CFD, the DEM and the coupling between both models. Adding dynamic load balancing to the recipe allows us to benefit more from parallelization where the work loads are divided evenly among processors.[33] To the best of our knowledge, there is no CFD-DEM software that implements dynamic load balancing on all three components (CFD, DEM and coupling).

In this paper, we build a high-order FEM stabilized CFD-DEM solver within the open-source software Lethe[34] available at `https://github.com/lethe-cfd/lethe` which is based on the Deal.II framework.[35] This was achieved by coupling the Volume Averaged Navier-Stokes (VANS) solver[36] with the DEM model of Lethe.[33] The coupling is built in Lethe and does not require a separate coupling interface. The coupled solver includes all the advantages of the VANS solver.[36] It is a fully implicit solver that requires no stability condition on the fluid side. It is globally mass conservative and it accounts for the time variation of the void fraction in the continuity equation. Moreover, it supports both models A and B of the VANS equations. It enables fundamental comparison between the two models in CFD-DEM studies as results obtained from both models are analysed. Furthermore, and to the best of our knowledge, it represents one of the first unresolved CFD-DEM solvers with load balancing capabilities. This enables it to leverage available computing resources by assigning more processors to tasks with greater work loads leading to more efficient utilization of resources. Finally, it is the first high order CFD-DEM solver. As such, coarser meshes can be used without sacrificing accuracy on the CFD side.

The coupling is achieved through the void fraction as well as the particle-fluid forces. In order to verify and validate our unresolved CFD-DEM solver, we consider four test cases. We simulate a particle sedimentation case to determine the particle's terminal velocity. We model a fluidized bed and study the pressure drop and the bed height before and after fluidization. We simulate a heavy fluid on top of a lighter fluid to obtain the Rayleigh Taylor instability and determine its mixing width. Finally, we study a spouted bed and determine the time averaged particle velocity. The obtained results verify correct implementation of the solid-fluid forces, demonstrate global mass conservation, and show physically correct predictions of the hydrodynamics of the systems being studied.

Governing Equations and Coupling Strategy in CFD-DEM

In the CFD-DEM approach, the incompressible Newtonian fluid phase is modeled as a continuum. It is described by the volume averaged Navier Stokes (VANS) equation which takes into account the dependency of the fluid's volume within a cell on the solid phase's volume occupying it. In the present work, these equations are solved using the finite element method in an Eulerian description of matter. On the other hand, the dispersed solid particles are modeled as a discrete phase. The latter is described using the discrete element method.[33] Therefore, the CFD-DEM model represents an Eulerian-Lagrangian approach for solving multiphase flows.

Description of Fluid Phase

We model the fluid flow using either forms A or B of the volume averaged Navier Stokes equations. The continuity equation for an incompressible fluid is:

$$\rho_f \frac{\partial \left(\epsilon_f\right)}{\partial t} + \rho_f \nabla \cdot \left(\epsilon_f \boldsymbol{u}_f\right) = m' \tag{1}$$

where ρ_f the fluid density, $\boldsymbol{u}_f$ is the fluid velocity, ϵ_f the fluid's void fraction, and m' the volumetric source of mass.[37] $m' = 0$ in the case of non reactive flows or when there is no source term added to the simulation. The momentum equation for model A of the incompressible VANS equations is:

$$\rho_f \left(\frac{\partial \left(\epsilon_f \boldsymbol{u}_f\right)}{\partial t} + \nabla \cdot \left(\epsilon_f \boldsymbol{u}_f \otimes \boldsymbol{u}_f\right) \right) = -\epsilon_f \nabla p + \epsilon_f \nabla \cdot \left(\boldsymbol{\tau}_f\right) + \boldsymbol{F}^{\boldsymbol{A}}_{\boldsymbol{pf}} + \rho_f \epsilon_f \boldsymbol{f} \tag{2}$$

while that of Model B is:

$$\rho_f \left(\frac{\partial \left(\epsilon_f \boldsymbol{u}_f \right)}{\partial t} + \nabla \cdot \left(\epsilon_f \boldsymbol{u}_f \otimes \boldsymbol{u}_f \right) \right) = -\nabla p + \nabla \cdot \left(\boldsymbol{\tau}_f \right) + \boldsymbol{F}_{\boldsymbol{pf}}^{\boldsymbol{B}} + \rho_f \epsilon_f \boldsymbol{f} \tag{3}$$

with

$$\boldsymbol{\tau}_f = \mu_f \left(\left(\nabla \boldsymbol{u}_f \right) + \left(\nabla \boldsymbol{u}_f \right)^T \right) - \frac{2}{3} \mu_f \left(\nabla \cdot \boldsymbol{u}_f \right) \boldsymbol{I} \tag{4}$$

where p is the pressure, $\boldsymbol{\tau}_f$ the deviatoric stress tensor, $\boldsymbol{f}$ the external force (ex. gravity), and $\boldsymbol{F}_{pf}$ is the momentum transfer term between the solid and fluid phases and includes forces such as drag, virtual mass, Basset force, Saffman lift, and Magnus lift.[38] $\boldsymbol{I}$ is the identity or unit tensor.

The difference between the two models lies in the formulation of the momentum equation specifically in the derivation of the pressure and the stress tensor. In model A, the shear and pressure gradient forces are calculated and added to the fluid phase according to cell properties such as cell void fraction and cell pressure drop. However, in model B these forces are calculated on individual particles and then lumped up with the drag force before being added to the fluid phase. Consequently, these solid-fluid interactions are defined explicit model B whereas, they are implicit in model A.[39]

Finite Element Discretization

The weak form of the VANS equations for both the continuity and momentum equations include the streamline-upwind/Petrove-Galerkin (SUPG) stabilization, the pressure-stabilizing/Petrov-Galerkin (PSPG) stabilization, and the grad-div stabilization. These equations are:

$$\textstyle\int_{\Omega} \left(\frac{\partial \left(\epsilon_f \right)}{\partial t} + \epsilon_f \nabla \cdot \boldsymbol{u} + \boldsymbol{u} \nabla \epsilon_f \right) q d\Omega + \underbrace{\mathcal{S}\mathcal{R} \cdot \left(\tau_u \nabla q \right) d\Omega_k}_{PSPG} = 0 \tag{5}$$

Form A:

$$\int_{\Omega} \left(\epsilon_f \frac{\partial \boldsymbol{u}}{\partial t} + \epsilon_f \boldsymbol{u} \cdot \nabla \boldsymbol{u}\right) \cdot \boldsymbol{v} d\Omega + \frac{1}{\rho_f} \Bigg(\int_{\Omega} \left(\epsilon_f \nu \left(\nabla^2 \boldsymbol{u}\right) + \nu \nabla \boldsymbol{u} \nabla \epsilon_f\right) \cdot \boldsymbol{v} d\Omega \tag{6}$$

$$\begin{aligned} &+ \int_{\Omega} m' \boldsymbol{u} \cdot \boldsymbol{v} d\Omega - \int_{\Omega} \left(\epsilon_f p \cdot \nabla \boldsymbol{v} + p \nabla \epsilon_f \cdot \boldsymbol{v}\right) d\Omega + \int_{\Omega} \frac{\boldsymbol{F_{pf}}}{V_{\Omega}} \cdot \boldsymbol{v} d\Omega \Bigg) - \int_{\Omega} \epsilon_f \boldsymbol{f} \cdot \boldsymbol{v} d\Omega \\ &+ \underbrace{\sum_K \int_{\Omega_k} \mathcal{SR} \cdot \left(\tau_u \boldsymbol{u} \cdot \nabla \boldsymbol{v}\right) d\Omega_k}_{SUPG} + \underbrace{\sum_K \int_{\Omega_k} \gamma \left(\frac{\partial \epsilon_f}{\partial t} + \nabla \cdot \left(\epsilon_f \boldsymbol{u}\right)\right) \left(\nabla \cdot \boldsymbol{v}\right) d\Omega_k}_{grad-div} = 0 \end{aligned} \tag{7}$$

Form B:

$$\int_{\Omega} \left(\epsilon_f \frac{\partial \boldsymbol{u}}{\partial t} + \epsilon_f \boldsymbol{u} \cdot \nabla \boldsymbol{u}\right) \cdot \boldsymbol{v} d\Omega + \frac{1}{\rho_f} \Bigg(\int_{\Omega} \nu \left(\nabla \boldsymbol{u}\right) \nabla \boldsymbol{v} d\Omega \tag{8}$$

$$\begin{aligned} &+ \int_{\Omega} m' \boldsymbol{u} \cdot \boldsymbol{v} d\Omega - \int_{\Omega} p \nabla \cdot \boldsymbol{v} d\Omega + \int_{\Omega} \frac{\boldsymbol{F_{pf}}}{V_{\Omega}} \cdot \boldsymbol{v} d\Omega \Bigg) - \int_{\Omega} \epsilon_f \boldsymbol{f} \cdot \boldsymbol{v} d\Omega \\ &+ \underbrace{\sum_K \int_{\Omega_k} \mathcal{SR} \cdot \left(\tau_u \boldsymbol{u} \cdot \nabla \boldsymbol{v}\right) d\Omega_k}_{SUPG} + \underbrace{\sum_K \int_{\Omega_k} \gamma \left(\frac{\partial \epsilon_f}{\partial t} + \nabla \cdot \left(\epsilon_f \boldsymbol{u}\right)\right) \left(\nabla \cdot \boldsymbol{v}\right) d\Omega_k}_{grad-div} = 0 \end{aligned} \tag{9}$$

and

$$\mathcal{SR} = \epsilon_f \frac{\partial \boldsymbol{u}}{\partial t} + \epsilon_f \boldsymbol{u} \nabla \cdot \boldsymbol{u} + \frac{1}{\rho_f} \left(m' \boldsymbol{u} + \nabla p - \nu \left(\nabla^2 \boldsymbol{u}\right) + \frac{\boldsymbol{F_{pf}}}{V_{\Omega}} \right) - \epsilon_f \boldsymbol{f} \tag{10}$$

where $\mathcal{SR}$ is the strong residual associated with the PSPG and SUPG stabilizations. $\boldsymbol{q}$ and $\boldsymbol{v}$ are the pressure and velocity interpolation functions. Both the test functions and the interpolation functions are Lagrange polynomial of arbitrary order, but we limit ourselves to linear (Q1) and quadratic (Q2) tensor elements in this work. V_{Ω} is the volume of the domain Ω and γ is a parameter related to the augmented Lagrangian formulation.[34] The stabilization parameter τ_u is defined by Tezduyar[40] and is explained in detail in the Lethe paper.[34] For transient simulations, τ_u is defined as:

$$\tau_u = \left[\left(\frac{1}{\Delta t}\right)^2 + \left(\frac{2\left|\boldsymbol{u}\right|}{h_{conv}}\right)^2 + 9\left(\frac{4\nu}{h_{diff}^2}\right)^2 \right]^{-1/2} \tag{11}$$

where Δt is the time step, h_{conv} and h_{diff} are the size of the element related to the convection transport and diffusion mechanism respectively.[34] The complete description, derivation and verification of the weak form of the VANS equations using the finite element method (FEM) along with the various stabilization techniques implemented is presented by El Geitani et al.[36]

All stabilization methods applied support both implicit stabilization where we use the velocity of the current time step in the stabilization formulation and explicit stabilization where we use the velocity at the previous time step. The last term in Eqs. (2) and (3) is not added in our solver as we don't solve for gravity in the fluid phase. The consequence of this is the addition of the buoyancy force on the solid phase and the redefinition of the pressure to include the hydrostatic pressure. It is important to note that this force is not explicitly added to the fluid phase of the VANS equations as it is implicit in the pressure.

The γ weight factor parameter for the grad-div stabilization implemented in the VANS equations[36] is calculated according to:[41]

$$\gamma = \nu + c^{*}\boldsymbol{u}_{f_{\Omega_e}} \tag{12}$$

where c^{*} depends on the pressure behavior in element Ω_e. Since this information is rarely available, Olshanskii et al.[41] set c^{*} to a global constant of order 1. In our solver, we allow c^{*} to be a user defined value. The choice of $c*$ remains arbitrary in the literature and there is no clear consensus to the optimal value that should be used. Grad-div stabilization plays an important role not only in mass conservation but in the condition number of the linear systems arising when solving the VANS equations.

Order of the Scheme

$\boldsymbol{q}$ and $\boldsymbol{v}$ are functions used for the interpolation of the velocity and pressure within the cells. These functions can take many forms, but in the present work we use Lagrange polynomials. If the polynomial is of degree 1, then the underlying scheme is second-order accurate and it is denoted by Q1 in a tensor element (quadrilateral in 2D, hexahedron in 3D). If it is of degree 2, it results in a third-order scheme and it is denoted by Q2 in a tensor element. In general, a polynomial of degree n is denoted as Q_n and results in a $n + 1$ order scheme. However, it is important to note that in tensor elements, the order of the scheme applies to the edges of the element. The diagonal along the faces and the inside of the element are interpolated by a polynomial resulting from the tensor product of the polynomials at the edges of the element.[42] In 3d, the diagonal polynomial on the faces of the elements will be of degree $2n$ and within the elements they will be of degree $3n$. Thus, although the Q1 elements are formally second-order accurate in space, they possess high-order components along the faces and within the element and this greatly increases their accuracy when the velocity gradient is not aligned with the edges of the cells.

The velocity and the pressure can be interpolated with different functions. For example, a 2D problem having a scheme that is third order accurate in velocity and second order accurate in pressure can be expressed as a Q2-Q1 scheme. The polynomials interpolating the velocity and pressure at the sides of the element will be of degree 2 and 1 respectively. In our work, the polynomial interpolating the void fraction is always taken to be of degree 1 (Q1) since this keeps the void fraction bounded within the elements.

Void Fraction Calculation

In the literature, the derivative of the void fraction with respect to time in the continuity equation is usually set to zero as it introduces major instabilities to the pressure field. In our solver, we found that this quantity can be added to the equation if the density of the fluid is low (eg. for a gas). However, if the fluid is dense (eg. a liquid), this quantity causes the solution to diverge.

We calculate the void fraction in a cell by:

$$\epsilon_f = \frac{v_{element} - v_{particles}}{v_{element}} \tag{13}$$

where $v_{element}$ is the volume of an element and $v_{particles}$ is the volume of all particles present within the element. In our solver, the latter is calculated by either the particle centroid method (PCM) due to its simplicity [43,44] or by the satellite point method (SPM)[44] which is more accurate than the PCM but more expensive. In the PCM method, if the centroid of the particle lies in a given cell, then the entire volume of the particle is considered to be in the cell. In the SPM method, each particle is divided into smaller sub-particles known as satellite particles each possessing its own volume. If the centroid of the satellite particle lies in a given cell, then the entire volume of the satellite particle will be considered to be in the cell.

In the context of the finite element method, the PCM and SPM are both calculated per element and then projected onto its nodes. The void fraction calculation using PCM and SPM is given by:

$$\epsilon_{f_{pcm}} = 1 - \frac{\sum^{N_p} V_p}{V_{\Omega_e}} \tag{14}$$

$$\epsilon_{f_{spm}} = 1 - \frac{\sum^{N_p} \sum^{N_{sp}} V_{sp}}{V_{\Omega_e}} \tag{15}$$

where V_p, V_{Ω_e}, V_{sp}, N_p and N_{sp} are respectively the particle volume, the element volume, the satellite point volume, the number of particles in an element and the number of satellite points in the particle that are also in the element. The full description of the void fraction scheme used and how it is implemented in the context of finite element is explained by El Geitani et al.[36]

Description of Solid Phase

Newton's second law describes the motion of the solid particles. There are two distinct particle motions in DEM simulations. The particles' translational movement is described by:

$$m_i \frac{\partial \boldsymbol{u}_{p,i}}{\partial t} = \sum_{j \in C_i} \boldsymbol{f}_{ij} + \sum_{w} \boldsymbol{f}_{w,i} + m_i \boldsymbol{g} + \boldsymbol{f_{fp,i}} \tag{16}$$

where m_i is the mass of particle i, $\boldsymbol{u}_{p,i}$ is the particle i's velocity, $\boldsymbol{f}_{ij}$ are the interactions between particle i and particle j and C_i includes all particles in the contact list of particle i, $\boldsymbol{f}_{w,i}$ are the particle i and walls interactions and w loops over all walls in contact with particle i, $\boldsymbol{g}$ is the gravity and $\boldsymbol{f_{fp,i}}$ are the fluid and particle i interactions which represent the coupling forces between the two phases. The particles' rotational movement is described by:

$$I_i \frac{\partial \boldsymbol{\omega}_{p,i}}{\partial t} = \sum_{j \in C_i} \left(\boldsymbol{M}_{ij}^{t} + \boldsymbol{M}_{ij}^{r} \right) + \boldsymbol{M}_{i}^{ext} \tag{17}$$

where I_i is the moment of inertia of particle i, $\boldsymbol{\omega}_{p,i}$ is the angular velocity of particle i, $\boldsymbol{M}_{ij}^{t}$ and $\boldsymbol{M}_{ij}^{r}$ are respectively the tangential and rolling friction torques due to the contact between particle i and j, and $\boldsymbol{M}_{i}^{ext}$ denotes all other external torques. In our solver, we use the Rayleigh characteristic time in order to determine the critical time step required for stability of the DEM simulation. We calculate the critical time step of all particles and take the smallest as the Rayleigh time step. It is given by:[45]

$$\Delta t_{crit} = \min_{i \in N_p} \left(\frac{\pi r_i}{0.1631 \nu_i + 0.8766} \sqrt{\frac{\rho_i}{G_i}} \right) \tag{18}$$

where N_p is the total number of particles, ρ_i is the particle's density, G_i is the particle's shear modulus, ν_i is the particle's Poisson's ratio, and r_i is the particle's radius. It is common to use a DEM time step that is 20% of the Rayleigh time step. For simulations where particles reach high velocities, a time step smaller than the 20% Rayleigh time step might be required.

Particle-Particle and Particle-Wall Interactions

The soft sphere model is used to calculate contact forces using artificial overlaps and a set of imaginary springs and dashpots and torques resulting from particle collisions.[46] During collision, this overlap as well as the collision force change. The time step for such simulation should be small enough so that the contact between particles is processed in several time intervals.[2] We use a non-linear visco-elastic model to calculate the normal and tangential spring and damping constants.[33] The respective equations are presented in Table 1 where δ_n is the normal overlap, e is the coefficient of restitution, ν is Poisson's ratio, and i and j represent the particles in contact.

Table 1: Normal and tangential spring and damping constants for the visco-elastic model.

Parameters	**Equations**
Normal spring constant	$k_n = \frac{4}{3} Y_e \sqrt{R_e \delta_n}$
Normal damping coefficient	$\eta_n = -2\sqrt{\frac{5}{6}}\beta\sqrt{S_n m_e}$
Tangential spring constant	$k_t = 8 G_e \sqrt{R_e \delta_n}$
Tangential damping coefficient	$\eta_t = -2\sqrt{\frac{5}{6}}\beta\sqrt{S_t m_e}$
Effective mass	$\frac{1}{m_e} = \frac{1}{m_i} + \frac{1}{m_j}$
Effective radius	$\frac{1}{R_e} = \frac{1}{R_i} + \frac{1}{R_j}$
Effective shear modulus	$\frac{1}{G_e} = \frac{2(2-\nu_i)(1+\nu_i)}{Y_i} + \frac{2(2-\nu_j)(1+\nu_j)}{Y_j}$
Effective Young's modulus	$\frac{1}{Y_e} = \frac{(1-\nu_i^2)}{Y_i} + \frac{(1-\nu_j^2)}{Y_j}$
Effective Young's modulus	$\frac{1}{Y_e} = \frac{(1-\nu_i^2)}{Y_i} + \frac{(1-\nu_j^2)}{Y_j}$
β	$\beta = \frac{\ln e}{\sqrt{\ln^2 e + \pi^2}}$
S_n	$S_n = 2 Y_e \sqrt{R_e \delta_n}$
S_t	$S_t = 8 G_e \sqrt{R_e \delta_n}$

The contact force between two particles is calculated as a combination of normal and tangential contributions:[33]

$$\boldsymbol{f}_{ij} = \begin{cases} \boldsymbol{f}_{ij}^{n} = -(k_n \delta_n)\boldsymbol{n}_{ij} - (\eta_n \boldsymbol{v}_{rn}) \\ \\ \boldsymbol{f}_{ij}^{t} = -(k_t \delta_t) - (\eta_t \boldsymbol{v}_{rt}) \end{cases} \tag{19}$$

where $\boldsymbol{v}_{rn}$ and $\boldsymbol{v}_{rt}$ are the normal and tangential components of the relative contact velocity. For additional details about DEM, we refer the reader to the Lethe-DEM paper.[33]

To integrate Newton's equation of motion, we implement the Velocity Verlet scheme.[47]

CFD-DEM Coupling

CFD and DEM are mainly coupled through the void fraction calculation and the fluid-particle interactions. In our code, the CFD and DEM both use the same mesh. This prevents the need to transfer solutions between meshes or localize the particle onto the CFD mesh and significantly reduces the computational cost. Additionally, our solver is fully parallelized. The parallelization of our solver is explained in detail in the Lethe[34] and Lethe-DEM[33] papers for the CFD and DEM components respecively. This parallelization supports adaptive mesh refinement, is flexible and scalable thanks to the method presented by Gassmöller et al.[48] and implemented in the Deal.II library.[35] Every fluid-particle interaction is calculated using values interpolated at the particle's location by the FEM interpolation. The summation of individual fluid-particle forces over all particles in a cell results in the force applied on the fluid due to the particles. All forces are calculated only once per time step. The detailed algorithm is shown in the following section.

Coupling Scheme

The coupling strategy is given in detail in Fig. 1. It couples the VANS solver[36] with Lethe-DEM.[33] The CFD and DEM solvers are coupled by introducing the DEM iterator within the CFD iterator. Initially, the void fraction is calculated, the VANS equations are solved for the fluid, the fluid-particle interactions are calculated and then the DEM iterator is called to solve Newton's second equation of motion. In order to ensure temporal agreement between the CFD and DEM iterator, we restrict the choice of the time step to the CFD solver. The DEM time step is chosen implicitly by specifying a DEM coupling frequency given as:

$$f_{coupling} = \frac{\Delta t_{CFD}}{\Delta t_{DEM}} \tag{20}$$

where $f_{coupling}$ is the frequency at which the coupling occurs and Δt is the time step. As shown in the VANS article,[36] our solver can achieve high order accuracy in both time and space. We can achieve second order or third-order accuracy in time using backward difference formulation (BDF2 or BDF3) and third order accuracy in space for the velocity field by using Q2-Q1 finite elements for velocity and pressure respectively. On the CFD side, our code is

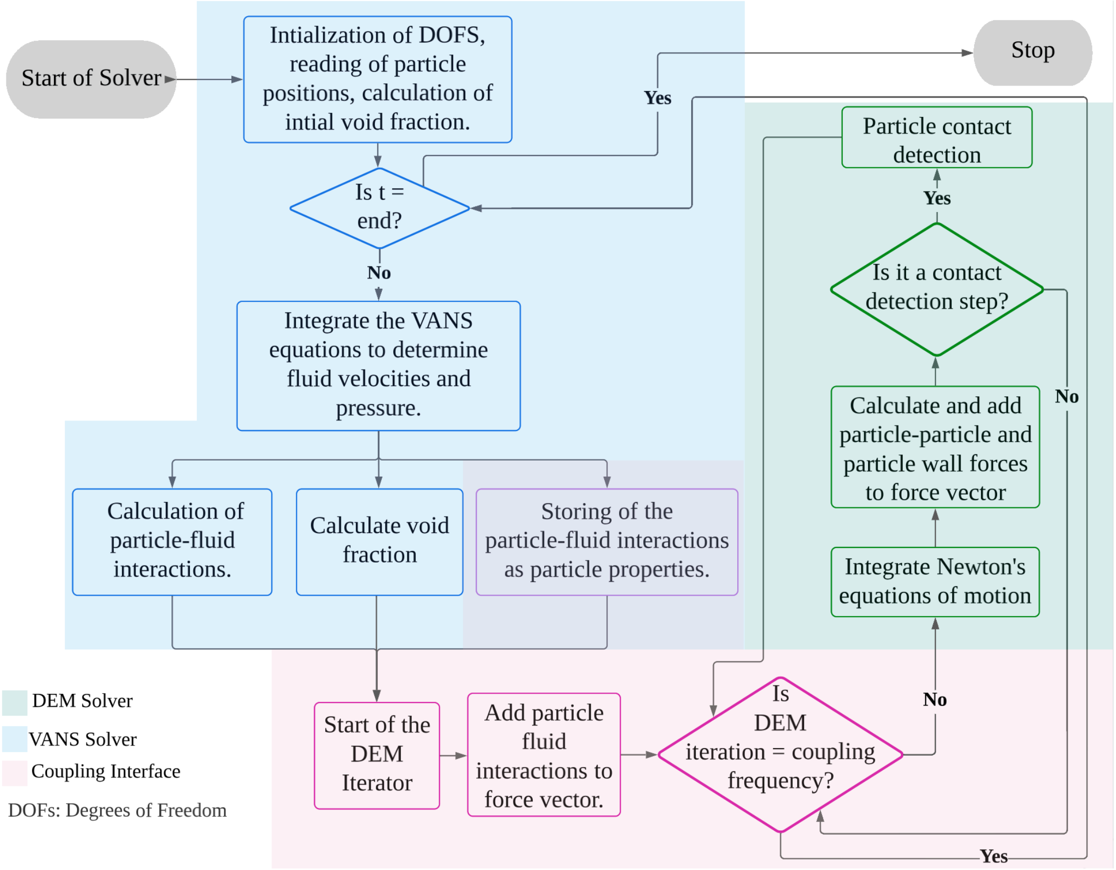


Figure 1: The CFD-DEM coupling scheme.

unconditionally stable since we use an implicit formulation in our evaluation of the VANS solution. On the DEM side, and in order to produce valid results, the DEM time step should be small enough to accurately capture particle-particle interactions. Since the fluid modeling is based on evaluating locally volume-averaged quantities, an element (Ω_e) volume should be at least one order of magnitude greater than the particle volume ($V_{\Omega_e} > 10V_p$).[49]

For the inter-phase coupling, and as we aim to solve dense solid-fluid systems, we assume 4 way coupling between the solid and fluid phases. In this case, the fluid affects the particles' motion and the particles affect the fluid flow. In addition to this, particles motion is also affected by collisions between particles and between particles and walls where these interactions play an important role in characterizing the particles' behavior.[50]

Load Balancing

Our CFD-DEM coupled solver supports load balancing. Load balancing allows the distribution of all tasks among all available processors such that the overall computational time is optimized. Load balancing was possible thanks to the p4est[51] and Deal.II[35] finite element library. P4est allows for a user-specified weight function that returns a non-negative integer weight for each finite element. Thus, instead of partitioning the work into a uniform number of elements, it distributes it evenly by weight. This is important as sometimes elements have varying mathematical unknowns to store and compute on.[51] This is true for CFD-DEM as the number of particles is not equal among elements. In the CFD-DEM solver of Lethe, each cell is assigned a weight based on its computational load. The sum of these weights is then distributed evenly among the available processors. In the absence of particles, the default cell weight is 1000. In CFD-DEM, we attribute a weight for a particle. The overall cell weight becomes the sum of the cell weight and all particles' weights in the cell. Consequently, a cell with more particles will normally have a larger weight.

Aspherix[9] also supports load balancing. However, due to the non unified nature of Aspherix, load balancing is only performed on the DEM side. On the CFD side, load-balancing is performed by OpenFOAM. Even if both the CFD and DEM are separately load balanced, the information about the particles calculated by the DEM (position, velocity, angular velocity, etc.) must be sent from multiple cores to multiple cores using MPI since the parallel decomposition of the DEM and the CFD do not coincide. In this case, the particle-fluid coupling, which includes the calculation of the solid-fluid forces and the void fraction, is not balanced. Nothing prevents all particles from ending up in a few cores of the CFD component of the code, which would be a significant bottleneck when calculating the coupling. This is where the Lethe CFD-DEM solver becomes attractive as it aims to overcome this particular limitation. In Lethe, load balancing ensures that communication between processors is performed only when particles moves between processors and not at each coupling step. This is not the case for non-unified solvers where communication is performed at each coupling step thus may result in more overhead as shown by the schematic of Fig. 2.

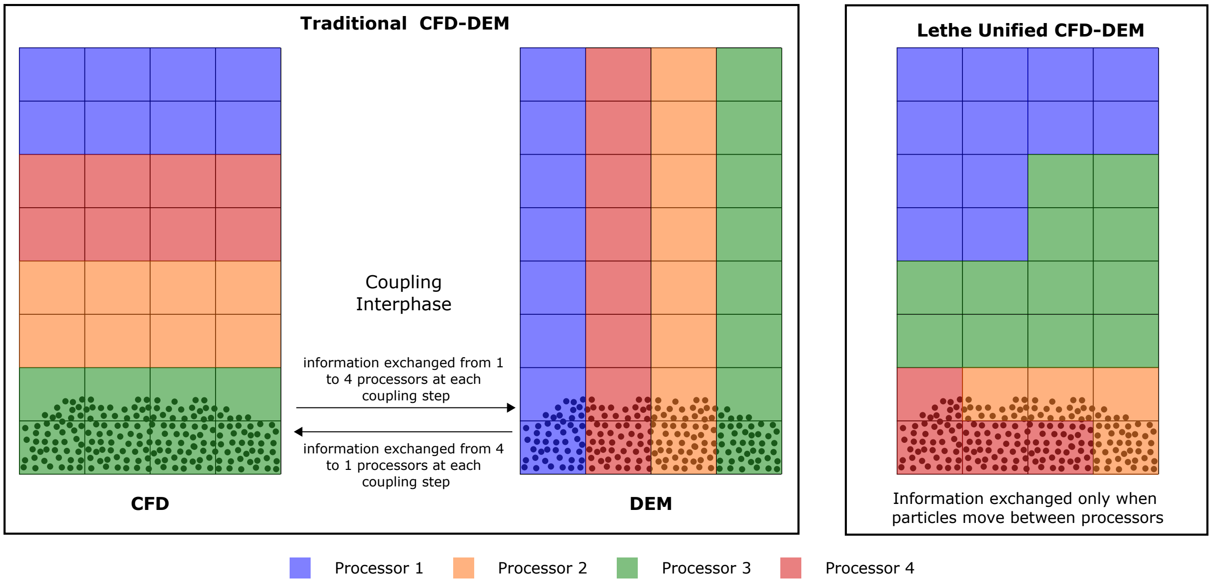


Figure 2: Representation of load balancing with 4 processors in a unified CFD-DEM solver and in a separately coupled CFD and DEM solvers

The application of load balancing in CFD-DEM software is relatively new. To our knowledge, our CFD-DEM solver demonstrates one of the first solvers with load balancing capabilities that take into account both physics in a unified software. Further studies must be performed in order to determine the optimum particle to cell weight ratios as this is affected by several factors. For example, a simulation where particle-particle contact forces are frequent usually requires a larger particle weight, whereas the same simulation with Q2-Q1 elements instead of Q1-Q1 elements will require a larger cell weight.

Particle-fluid Interactions

There exists several important forces that should be accounted for in the coupling process. These include the drag force, the buoyancy force, the pressure force, and the shear force. Since we are mainly interested in gas-solid flows in this work, we neglect virtual mass force, Basset force and lift force.[43]

Drag Force

It is the force experienced due to the relative motion of the particle and fluid around it. We determine the drag force based on:

$$\boldsymbol{f_D} = \beta(\boldsymbol{u}_f - \boldsymbol{u}_{p,i}) \tag{21}$$

where $\boldsymbol{u_{p,i}}$ is the particle i's velocity, and β is the inter-phase momentum exchange coefficient. According to Gidaspow, the interphase momentum exchange coefficient β differs between models A and B:[37]

$$\beta_B = \frac{\beta_A}{\epsilon_f} \tag{22}$$

On the other hand, Zhou et al.[39] implement the same drag force for both models. In our derivation, we use the same drag force for both models. This is because only when considering the same drag force are the models A and B mathematically equivalent. When simulating a stationary solid phase, as the case of the packed bed which we simulated,[36] we use the assumption of Eq. (22) as we only consider drag and buoyancy. Eqs. (39) and (41) demonstrate that the pressure gradient force should be applied back to the fluid phase only in model B. As such, not implementing this important force in the packed bed for model B necessitates the use of the momentum exchange coefficient as written in Eq. (22). This penalizes the pressure in model B by a factor of ϵ_f in order to compensate for the pressure gradient force resulting in a good pressure drop in the bed. This is not the case for model A as the pressure gradient force is already implicitly applied by the term $\epsilon_f \nabla p$ of the momentum equation.

We calculate the drag force on a single particle as:

$$\boldsymbol{f_D} = \frac{1}{2}\rho_f C_D A_{ref} \left|\boldsymbol{u}_{f,p} - \boldsymbol{u}_{p,i}\right| \left(\boldsymbol{u}_{f,q} - \boldsymbol{u}_{p,avg}\right) \tag{23}$$

where $\boldsymbol{u}_{f,p}$ is the interpolated fluid velocity at the particle's location calculated at the previous time step, $\boldsymbol{u}_{f,q} = \boldsymbol{u}_f$ is the fluid velocity at the quadrature point, and $\boldsymbol{u}_{p,avg}$ is the average particles' velocity in the cell both calculated at the current time step. We use a Q_n interpolation that is homogeneous with the FEM scheme in order to interpolate the velocity at the particle's location. A_{ref} is the particle's reference area which is taken as the cross-section:

$$A_{ref} = \pi r_p^2 \tag{24}$$

where r_p is the particle's radius and C_D is the drag coefficient. At the time of writing, our code supports Dallavalle,[52] DiFelice,[53] Rong,[54] Koch and Hill,[55] Beetstra,[56] and Gidaspow[37] drag models. The inter-phase momentum exchange coefficient for Dallavalle model:[52]

$$\beta = \frac{1}{2}\rho_f \left(0.63 + \frac{4.8}{\sqrt{Re_p}}\right)^2 A_{ref} \left|\boldsymbol{u}_f - \boldsymbol{u}_{p,i}\right| \tag{25}$$

and Re_p is the particle Reynolds number and is expressed as:

$$Re_p = \frac{\rho_f \epsilon_f \left| \boldsymbol{u}_f - \boldsymbol{u}_{p,i} \right| d_p}{\mu_f} \tag{26}$$

where μ_f is the fluid's dynamic viscosity and d_p is the particle's diameter. The inter-phase momentum exchange coefficient for the Di Felice model[53] is given:

$$\beta = \frac{1}{2}\rho_f \Big(0.63 + \frac{4.8}{\sqrt{Re_p}}\Big)^2 A_{ref} \left| \boldsymbol{u}_f - \boldsymbol{u}_{p,i} \right| \epsilon_f^{\,2-\left[3.7-0.65e^{\left(\frac{-(1.5-log_{10}(Re_p))^2}{2}\right)}\right]} \tag{27}$$

while that for the Rong et al. model is given as:[54]

$$\beta = \frac{1}{2}\rho_f \Big(0.63 + \frac{4.8}{\sqrt{Re_p}}\Big)^2 A_{ref} \left| \boldsymbol{u}_f - \boldsymbol{u}_{p,i} \right| \epsilon_f^{\,2-\left[2.65(\epsilon_f+1)-(5.3-3.5\epsilon_f)\epsilon_f^2 e^{\left(\frac{-(1.5-log_{10}(Re_p))^2}{2}\right)}\right]} \tag{28}$$

The Koch and Hill drag model is derived based on results obtained from lattice-Boltzmann simulations. It is calculated according to Eq. (21) where the momentum exchange coefficient β is determined as:[55]

$$\beta = \frac{18\mu_f \epsilon_f^2 \epsilon_p}{d_p^2} \Big(F_0(\epsilon_p) + \frac{1}{2}F_3(\epsilon_p) Re_p\Big) \frac{V_{p,i}}{\epsilon_p} \tag{29}$$

where $\epsilon_p = 1 - \epsilon_f$ is the particles' void fraction, $V_{p,i}$ is the particle's volume, and where

$$F_0(\epsilon_p) = \begin{cases} \frac{1+3\sqrt{\frac{\epsilon_p}{2}}+\frac{135}{64}\epsilon_p ln(\epsilon_p)+16.14\epsilon_p}{1+0.681\epsilon_p-8.48\epsilon_p^2+8.14\epsilon_p^3} & \epsilon_p < 0.4 \\ \frac{10\epsilon_p}{\epsilon_f^3} & \epsilon_p \geq 0.4 \end{cases} \tag{30}$$

and

$$F_3(\epsilon_p) = 0.0673 + 0.212\epsilon_p + \frac{0.0232}{\epsilon_f^5} \tag{31}$$

The Beetstra[56] drag model was obtained from lattice-Boltzmann simulations and the drag force for a single particle is expressed as:

$$\boldsymbol{f_D} = 3\pi\mu_f d_p \epsilon_f \boldsymbol{u}_f F_i \tag{32}$$

where F_i is the normalized drag for a mono-dispersed system and is defined as:

$$F_i = \frac{10\epsilon_p}{\epsilon_f^2} + \epsilon_f^2(1 + 1.5\epsilon_p^{0.5}) + \frac{0.413Re}{24\epsilon_f^2}\left[\frac{\epsilon_f^{-1} + 3\epsilon_p\epsilon_f + 8.4Re_p^{-0.343}}{1 + 10^{3\epsilon_p}Re_p^{-(1+4\epsilon_p)/2}}\right] \tag{33}$$

The Gidaspow[37] drag model is a combination between the Ergun equation and Wen-Yu drag model. It is defined as:

$$\beta = \begin{cases} \frac{150(1-\epsilon_f)^2\mu_f}{\epsilon_f d_p^2} + \frac{1.75(1-\epsilon_f)\rho_f(\boldsymbol{u}_f - \boldsymbol{u}_{p,i})}{d_p} & \epsilon_f < 0.8 \\ \frac{3}{4}C_D\frac{\epsilon_f\left|\boldsymbol{u}_f - \boldsymbol{u}_{p,i}\right|\rho_f(1-\epsilon_f)}{d_p}\epsilon_f^{-2.65} & \epsilon_f \geq 0.8 \end{cases} \tag{34}$$

where for the case of $\epsilon_f \geq 0.8$, the drag coefficient C_D is determined as:

$$C_D = \begin{cases} \frac{24}{Re_p}\left(1 + 0.15(Re_p)^{0.687}\right) & Re_p < 1000 \\ 0.44 & Re_p \geq 1000 \end{cases} \tag{35}$$

For additional information about the different drag models available, we refer the reader to the article by Norouzi et al.[57] or by Bérard et al.[58]

Buoyancy Force

It is the force exerted by the fluid on the submerged particle that opposes its weight. It is given by the following equation:[37]

$$\boldsymbol{f}_B = -\rho_f V_{p,i} \boldsymbol{g} \tag{36}$$

The buoyancy force becomes important in fluidized and spouted beds when the ratio of particle to fluid densities is significant. The buoyancy force is only applied on the particles, as the pressure calculated using the VANS equations includes the hydro-static pressure.

Undisturbed Flow Forces

These forces are given by:[58]

$$\boldsymbol{f}_{\nabla p} = -V_p \left(\frac{\partial p}{\partial x} \right) = -V_{p,i} \nabla P \tag{37}$$

$$\boldsymbol{f}_{\nabla \cdot \tau} = -V_p \left(\frac{\partial \tau}{\partial x} \right) = -V_{p,i} \nabla \cdot \tau \tag{38}$$

where $\boldsymbol{f}_{\nabla p}$ and $\boldsymbol{f}_{\nabla \cdot \tau}$ are the pressure and shear forces respectively, and ∇P and $\nabla \cdot \tau$ are the gradient of the pressure and divergence of the stress tensor interpolated at the particle's location respectively. For Model A, these forces are implicitly added to the fluid since the void fraction multiplies the pressure and stress tensor gradients, and therefore should only be added explicitly to the particles. For Model B, it should be explicitly added for both solid and fluid phases.

All of the fluid-particle interactions are applied to both the solid and fluid phases to ensure that Newton's third law of motion is respected. The expression of the particle-fluid interactions for the VANS equations are $\boldsymbol{F}_{pf}$ for the force applied on the fluid phase and $\boldsymbol{f}_{fp}$

for the force applied on the solid phase. For model B, these forces become:

$$\boldsymbol{F}_{pf_B} = \frac{1}{\Delta V_{\Omega_e}} \sum_{N_p} \Big(\boldsymbol{f}_{D_B} + \boldsymbol{f}_{\nabla p} + \boldsymbol{f}_{\nabla\cdot\tau} \Big) \tag{39}$$

$$\boldsymbol{f}_{fp_B,i} = \Big(\boldsymbol{f}_{D_B,i} + \boldsymbol{f}_{B,i} + \boldsymbol{f}_{\nabla p,i} + \boldsymbol{f}_{\nabla\cdot\tau,i} \Big) \tag{40}$$

where V_{Ω_e} is the volume of the finite element Ω_e. The particle-fluid force of model A is given by:

$$\boldsymbol{F}_{pf_A} = \frac{1}{\Delta V_{\Omega_e}} \sum_{N_p} \Big(\boldsymbol{f}_{D_A} \Big) \tag{41}$$

$$\boldsymbol{f}_{fp_A,i} = \Big(\boldsymbol{f}_{D_A,i} + \boldsymbol{f}_{B,i} + \boldsymbol{f}_{\nabla p,i} + \boldsymbol{f}_{\nabla\cdot\tau,i} \Big) \tag{42}$$

Verification and Validation of the CFD-DEM Model

In order to verify and validate our model, we simulate several test cases. The series of test cases include particle sedimentation, a fluidized bed, the Rayleigh Taylor instability and a spouted bed.

Particle Sedimentation Test Case

We simulate a dense particle initially at rest falling in a stagnant liquid and we measure the instantaneous velocity of the particle. We then compare the velocity obtained from the simulation with that calculated by numerically solving the following ordinary differential equation (ODE):

$$m_p \frac{\partial u}{\partial t} = \Big(\boldsymbol{F}_D - V_p(\rho_p - \rho_f)\boldsymbol{g} \Big) \tag{43}$$

where m_p is the particle's mass and ρ_p is the particle's density. After some time, the particle should reach a constant velocity known as the settling or terminal velocity. This occurs when the drag balances the gravitational force and buoyancy.[37] The settling velocity $\boldsymbol{v}_r$ is given as:

$$\boldsymbol{v}_r = \sqrt{\frac{V_p(\rho_p - \rho_f)\boldsymbol{g}}{\frac{1}{2}\rho_f C_D A_{ref}}} \tag{44}$$

where C_D is the drag coefficient given based on the drag model used. We use the DiFelice drag model.

Simulation Setup

The cylinder in which the particle will fall should be large enough so that its walls have no effect on the sedimentation of the particle. For this, we choose a cylinder with a radius 25 times greater than the particle's diameter (2 mm). The particle was inserted in the middle at a height of 0.08 from the center of the cylinder having the following coordinates (0.08,0,0). The gravitational acceleration of 9.81 m/s^2 was applied in the x-direction. The DEM time step chosen was 3.28% of the Rayleigh time step. We chose a very coarse mesh as to avoid velocity fluctuations as the particle moves across cells. Only the drag and buoyancy forces were enabled for this simulation as to respect the ODE of Eq. (43). The physical properties chosen are typical values often used for water as a fluid and glass beads as the solids. We used Q1-Q1 second order finite elements in velocity and pressure. All simulation parameters are given in Table S1.

Results and Discussion

We show the instantaneous velocity of the particle in Fig. 3. Based on the obtained results, the velocity of the particle follows the analytical solution until it reaches the correct value of the settling velocity $\boldsymbol{v}_r = 0.2328 m/s$.

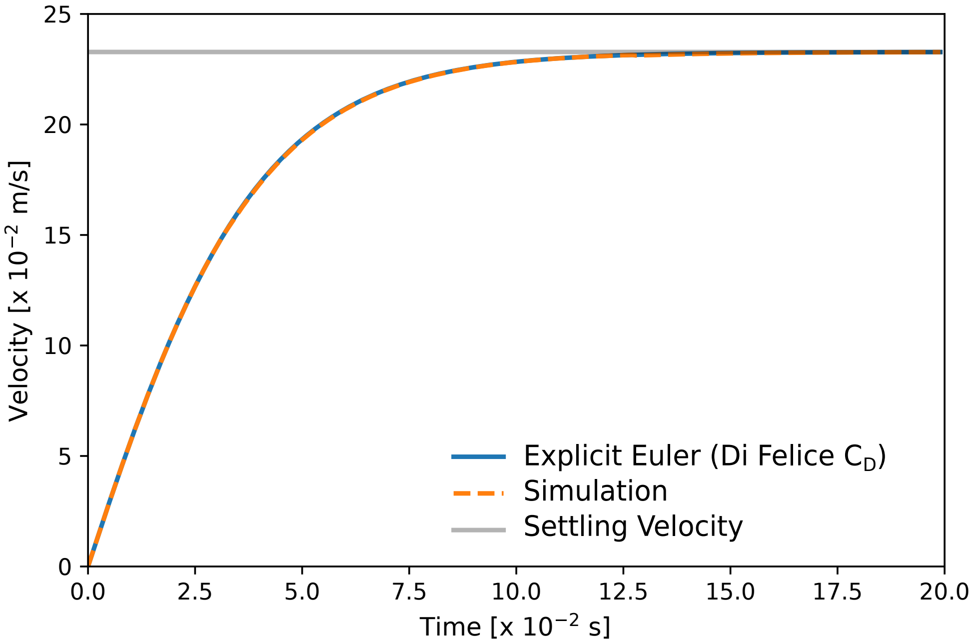


Figure 3: Instantaneous velocity of particle during sedimentation.

This simple test case allows the validation of our drag and buoyancy implementation.

Fluidized Bed Test Case

We study the pressure drop as a function of inlet velocity in a fluidized bed. We compare the measured pressure drop with that obtained from the Ergun equation. The Ergun equation is a correlation that predicts the pressure drop in a packed bed and is given by:[59]

$$\Delta p = \frac{150(1-\epsilon_f)^2 \boldsymbol{u}_f \mu_f H_b}{\epsilon_f^3 d_p^2} + \frac{1.75(1-\epsilon_f)\rho_f \boldsymbol{u}_f^2 H_b}{\epsilon_f^3 d_p} \tag{45}$$

where H_b is the bed's height. After fluidization, the pressure drop becomes constant and proportional to particle weight held by the fluid and is given as:

$$\Delta p = \frac{N_p V_p (\rho_p - \rho_f) g}{A_b} \tag{46}$$

where N_p is the total number of particles in the bed and A_b is the bed's cross sectional area. In a fluidized bed, the pressure drop is only the contribution of the drag and buoyancy forces. We simulate the fluidized bed test case using both Q1-Q1 and Q2-Q1 elements to demonstrate the high order capabilities of our CFD-DEM solver. For the discretization in time, we use the first order backward difference scheme (BDF1).

Simulation Setup

We first use the DEM solver to fill a cylindrical bed with 200,000 spherical particles with a diameter of 500 μm belonging to the Geldart B group. The gravitational acceleration of 9.81 m/s^2 was applied in the x-direction. The DEM time step chosen was 10.38% of the Rayleigh time step. The particles are packed at a distance of 0.04 m above the inlet to allow the fluid flow to develop before entering the packing. This also eliminates any effects the void fraction might have at the inlet. The DEM simulation had a final time of 1 s to allow the particles to settle and for their kinetic energy to dissipate. Air flow is then introduced at the bottom of the bed at varying inlet velocities from 0.02 m/s to 0.28 m/s with increments of 0.02 m/s. An additional velocity of 0.15 m/s was added as it represents the value close to the minimum fluidization, thus it allows us to better understand the behavior at minimum fluidization. The simulation was run once using the PCM and once using the SPM with 7 satellite particles for void fraction calculation. We choose a smoothing length for the void fraction equivalent to 4 d_p^2. We study the fluidization curve of the bed obtained using the Di Felice[53] drag model. The simulation parameters are given in Table S2.

There exists various correlations to predict the minimum fluidization velocity. Based on the particles' diameter and density, we compare the simulation results with different correlations. We used Ergun,[59] Wen-Yu,[60] and Noda et al.[61] correlations for the minimum fluidization velocity. For a more detailed review on the different available correlations, we refer the reader to the article by Anantharaman et al.[62] The Ergun minimum fluidization Re_{mf} is defined as:

$$Re_{mf} = \left(\left(\frac{42.86(1-\epsilon_{mf})}{\phi} \right)^2 + 0.571\epsilon_{mf}^3 \phi Ar \right)^{0.5} - \left(\frac{42.86(1-\epsilon_{mf})}{\phi} \right) \tag{47}$$

where ϵ_{mf} is the void fraction at minimum fluidization, ϕ is the sphericity of the particles ($\phi = 1$ for spherical particles), and Ar is the Archimedes number and is defined as:

$$Ar = \frac{g\rho_f(\rho_p - \rho_f)d_p^3}{\mu_f^2} \tag{48}$$

The Wen-Yu minimum fluidization Re_{mf} is defined as:

$$Re_{mf} = \left(33.7^2 + 0.0408Ar \right)^{0.5} - 33.7 \tag{49}$$

The Noda et al. minimum fluidization Re_{mf} is defined as:

$$Re_{mf} = \left(19.29^2 + 0.0276Ar \right)^{0.5} - 19.29 \tag{50}$$

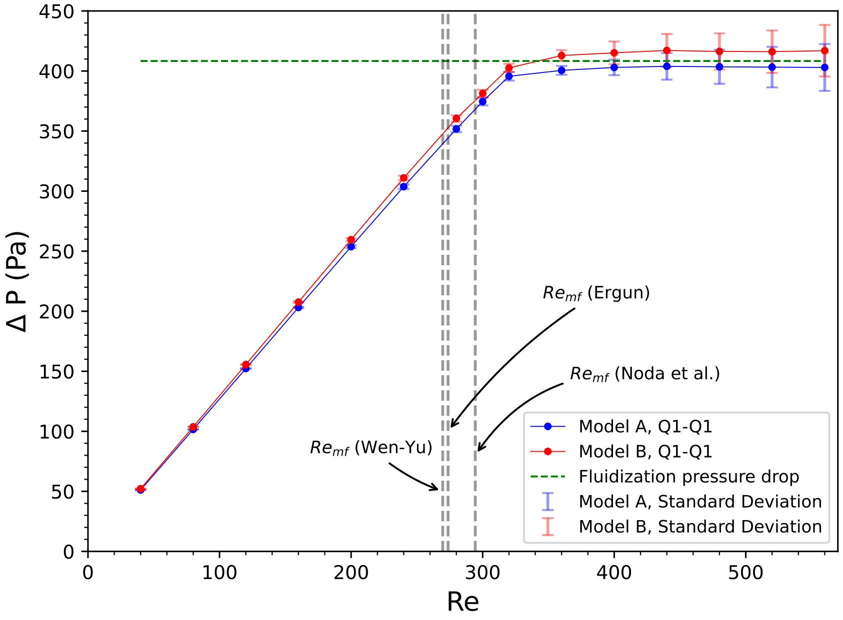
The minimum fluidization velocity (U_{mf}) is obtained from Re_{mf} by:

$$Re_{mf} = \frac{\rho_f U_{mf} d_p}{\mu_f} \tag{51}$$

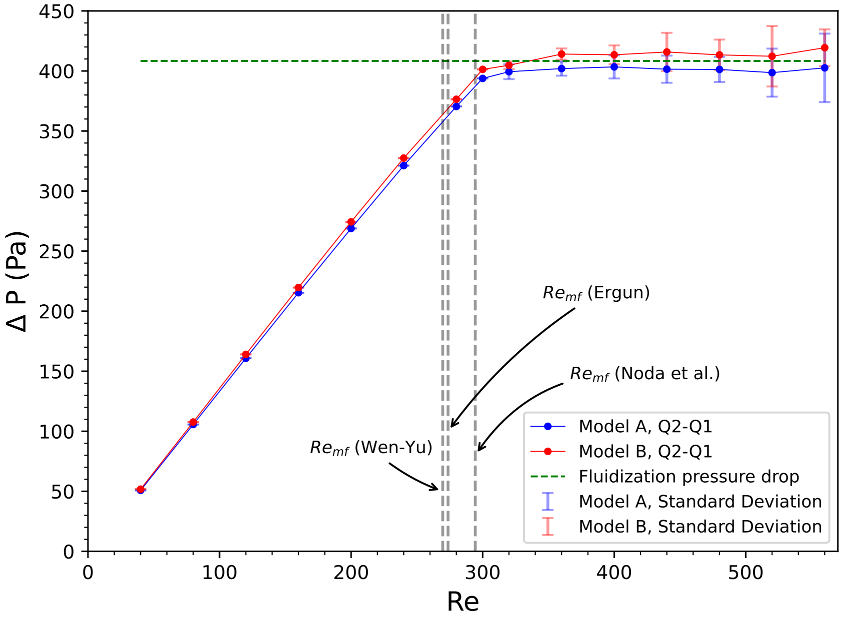
For this case, we found the minimum fluidization velocity using Eq. (47) to be $U_{mf} = 0.137\ m/s$, Eq. (49) to be $U_{mf} = 0.135\ m/s$, and Eq. (50) to be $U_{mf} = 0.147\ m/s$. The fluidization pressure drop is calculated using Eq. (46) to be $\Delta p_{fluidization} = 408.34\ Pa$. Grad-div stabilization was particularly interesting in this case when solving model B of the VANS equations where stability was dependant on the choice of c^*. Model A showed an enhanced stability with an optimal choice of c^* taken to be the size of the element for all cases. This was not the case for model B, where the solver was unstable for some cases. Hence, the choice of $c^* = 1$.

Results and Discussion

We calculate the pressure drop instantaneously, and we average it over 0.5 s after the pressure stabilizes. Fig. 4 shows the pressure drop as a function of the different inlet velocities for both models A and B of the VANS equations for different finite element orders using the SPM void fraction scheme. As the results for pressure drop did not differ between using the PCM and SPM void fraction schemes, only results obtained uing SPM are presented. The convergence of model B with Q2-Q1 elements was difficult as the system of model B is already stiff and using high order elements exacerbates the stiffness of the system. The use of SPM instead of PCM for void fraction calculation did not help in the convergence of Q2-Q1 model B simulations. The stiffness of model B comes from explicitly adding the pressure and shear forces to the fluid. For model B with Q2-Q1 elements, few simulations with inlet velocities above minimum fluidization were not converging to the tolerance of 10^{-9} for longer simulation times. Model A is considerably more stable than model B in CFD-DEM simulations as convergence was obtained easily for the different elements orders. The bars shown represent the standard deviation over a period of 0.5 s.



(a) pressure drop using Q1-Q1 elements

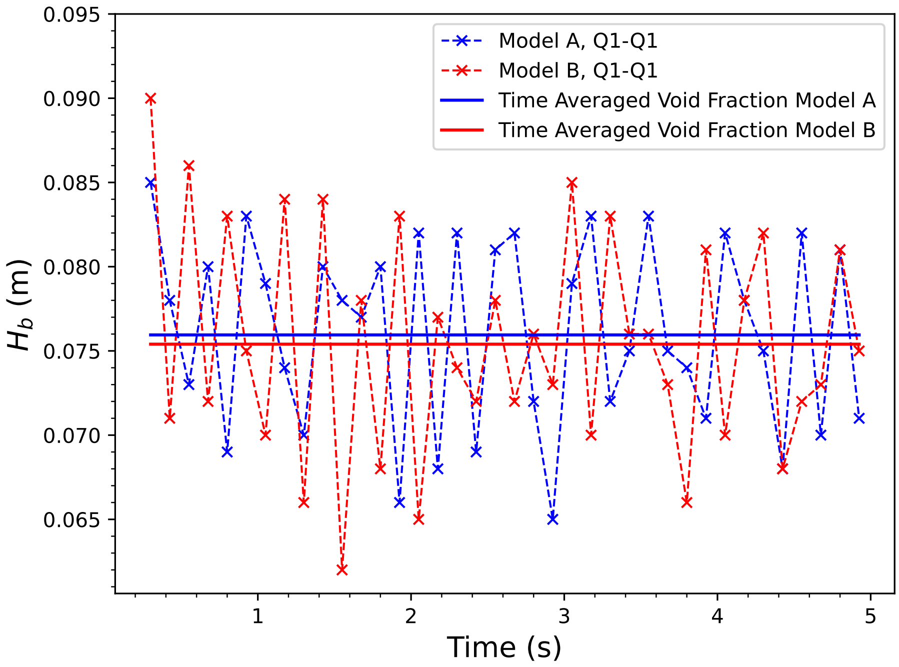


(b) pressure drop using Q2-Q1 elements

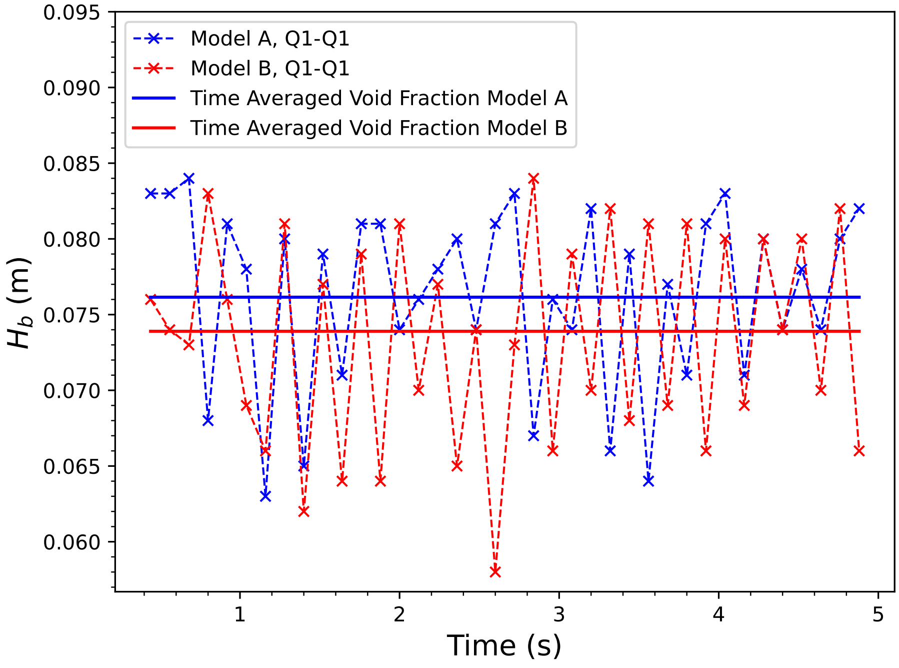
Figure 4: Pressure drop in fluidized bed as a function of inlet Re number using SPM.

For both finite element orders, it can be seen that models A and B result in a relatively close pressure drop in the bed. Model B results in a slightly larger standard deviation and that can be attributed to its instantaneous pressure drop fluctuating to larger magnitudes than that of model A. The fluctuations in pressure drop for both models is slightly higher for second order elements. Since higher order elements capture better the turbulence in the flow, we expect to observe higher fluctuations in the pressure drop.

Fig. 5 presents the instantaneous bed height for both models using Q1-Q1 elements and an inlet velocity of 0.28 m/s. The height of both models is highly oscillatory explained by the slug flow behavior of the fluidized bed where large bubbles form at the inlet of the bed and break at the outlet of the packing region. Models A and B do not lead to the same instantaneous physical behavior. This is in part due to the chaotic behavior of a fluidized bed. The time averaged height of both models is relatively close and as such, the beds behavior is statistically similar. We notice that the two void fraction schemes lead to different bed fluctuations at the same time-step as shown in Fig. 5, although the average height barely changes between SPM and PCM schemes. The change was more noticeable in model B where the average height is around 0.075 for PCM and 0.0725 for SPM compared to model A where the average height was 0.075 for PCM and 0.076 for SPM.



(a) Height fluctuation using PCM



(b) Height fluctuation using SPM

Figure 5: Instantaneous bed height fluctuation for inlet velocity of 0.28 m/s.

For Q1-Q1 elements, we compare the time averaged height in the bed for both models as shown in Fig. 6. The bed height was averaged over a period of 1 s. Differences between average heights using PCM and SPM are minor so only SPM results are presented.

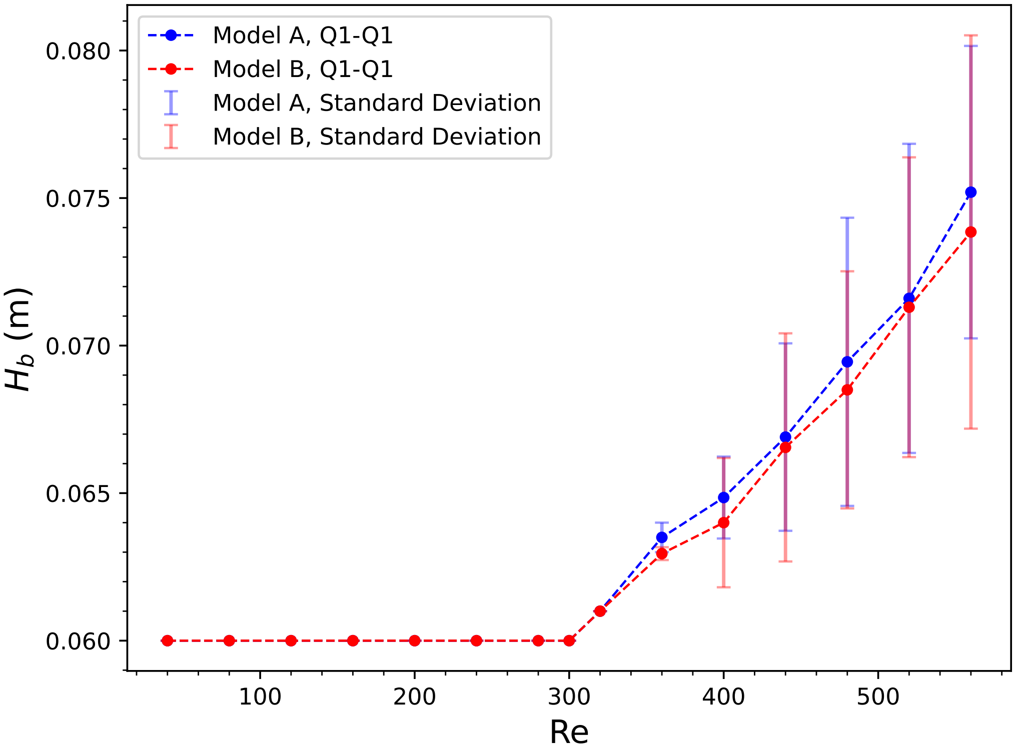


Figure 6: Average bed height as a function of inlet Re number using Q1-Q1 elements using SPM.

The time averaged height is relatively close in both models for the different inlet velocities. The variation at each inlet velocity over 3 s is shown as the standard deviation which represents the magnitude of fluctuation of the bed height over this period. Model B results in fluctuations that are more pronounced when compared to those of model A as demonstrated by the standard deviation in Fig. 6. Therefore, model A and B, though mathematically equivalent are physically different when considering the instantaneous behavior of the fluidized bed. This is caused by the different treatment of the pressure force on the fluid.

The Rayleigh-Taylor Instability Test Case

We study the evolution of the Rayleigh-Taylor instability as a function of different fluid and particle densities. This is a well known case in which a heavy fluid is located above a light fluid. The difference in densities makes this case an inherently unstable problem. The fluids start mixing together under the effect of gravity. Initially, an exponential instability growth period occurs after which the mixing layer is described according to the following:[63]

$$h = \alpha A g t^2 \tag{52}$$

where h is the width of the mixing layer, g is the gravitational acceleration, t is the time and A is the Atwood number and is defined as:

$$A = \frac{\rho_{hf} - \rho_{lf}}{\rho_{hf} + \rho_{lf}} \tag{53}$$

where $\rho_{hf} = \epsilon_f \rho_f + (1 - \epsilon_f)\rho_p$ is the density of the heavy fluid since the heavy fluid is a mixture of fluid and particles and $\rho_{lf} = \rho_f$ is the density of the light fluid. The constant α varies slightly with the Atwood number. It is believed to be insensitive to initial conditions.[64] Therefore, we start all our simulations with the same initial conditions. According to Snider, it lies between 0.05 and 0.07.[63] A historical survey by Dimonte et al.[64] for α obtained from experiments and numerical simulations found that α can have values between 0.03 and 0.08.[64] It is important to note that Eq. (52) is only valid after the initial exponential growth period. We expect to obtain slightly larger values of α in our simulations as we are working in 3D where single modes grow faster than in 2D.[64]

Simulation Setup

Similar to the fluidized bed test case setup, we insert the particles using the DEM solver before starting the CFD-DEM simulation. Particles of the same diameter and density were non-uniformly and randomly inserted in the upper section (initial height of particles is 0.05 m) of a 3 dimensional rectangular bed. The gravitational acceleration of 9.81 m/s^2 was applied in the y-direction. No initial perturbation is placed on the interface of the two fluids; however, small perturbations occur as a result of the random distribution of particles. We use the Dallavalle drag model[52] as we want a drag force independent of the void fraction. We solve model A of the VANS equations. We apply slip boundary conditions on the side walls of the beds as to eliminate boundary effects. We apply no slip boundary conditions on the top and bottom walls to prevent fluid re-circulation. Three test cases were simulated with varying Atwood numbers. The ratio of the DEM time step chosen to the Rayleigh time step for each case as well as all parameters are presented in Table 2.

Table 2: Parameters for the setup of the Rayleigh-Taylor Instability.

Particle radius (m)	0.0001	0.0001	0.0001
Particle density (kg/m^3)	3	5	10
Fluid density (kg/m^3)	1	1	1
Initial particle volume fraction	0.088	0.088	0.088
gravity (m/s)	0,-9.8,0	0,-9.8,0	0,-9.8,0
Number x,y,z mesh size	$18 \times 150 \times 18$	$18 \times 150 \times 18$	$18 \times 150 \times 18$
Bed x,y,z size (m)	$0.01 \times 0.15 \times 0.01$	$0.01 \times 0.15 \times 0.01$	$0.01 \times 0.15 \times 0.01$
Atwood Number	0.0817	0.151	0.286
Number of Particles	837925	837925	837925
CFD time step (s)	5×10^{-4}	5×10^{-4}	5×10^{-4}
CFD-DEM coupling frequency	500	500	500
$\Delta t_{DEM}/\Delta t_{crit}$ (%)	58.04	44.95	31.78

Our choice of densities result in a close range of Atwood numbers, therefore, we expect that the difference in the instabilities and the mixing length will be relatively small.

Results and Discussion

We measure the mixing width as the height of the bed starting from the first position where we have the diluted region of particles and ending with the last denser layer having less particles than the initial particle packing. This ensure we capture the complete mixing width. Fig. 7 shows the plot of the growth rate of the mixing layer as a function of time.

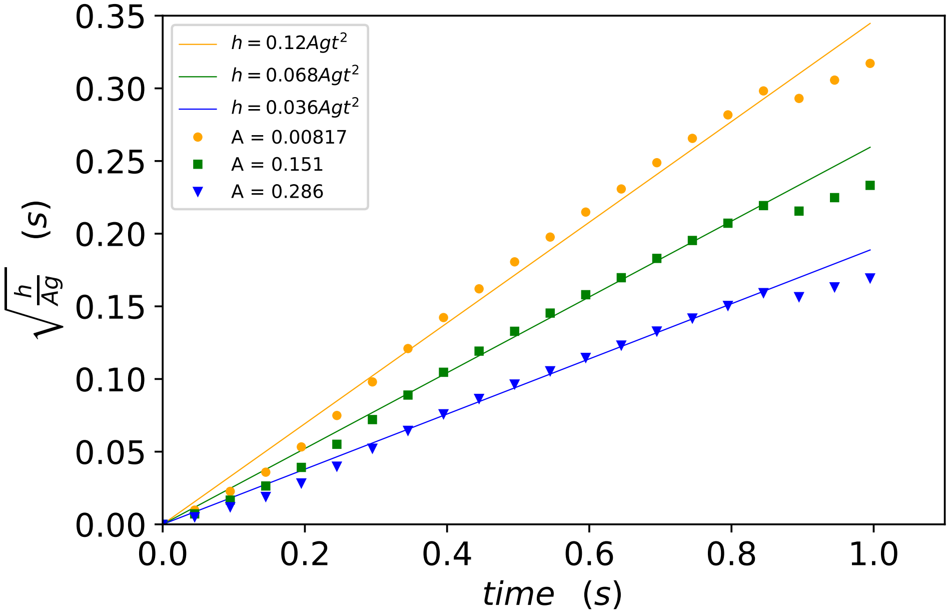


Figure 7: Growth rate of the mixing layer as a function of time.

The measurement of the mixing layer agrees well with the correlation of the mixing layer. The initial 0.2 s do not agree with the correlation as they represent the initial exponential instability growth where correlation (52) is not valid. Additionally, the sudden drop observed at around time 0.9 s is attributed to a big bubble of the light fluid reaching the top of the bed. As such, the packing at the top of the bed is disturbed, and not enough particles remain to feed the instability which eventually leads to the breaking of the mixing layer, thus rendering correlation (52) invalid after 0.9 s. We observe this behavior in Fig. 8 which shows snapshots depicting the growth of the Rayleigh-Taylor instability at different times in the simulation for $A = 0.286$. In Fig. 8, the fluid is not shown and only the particles' are presented along with their velocities.

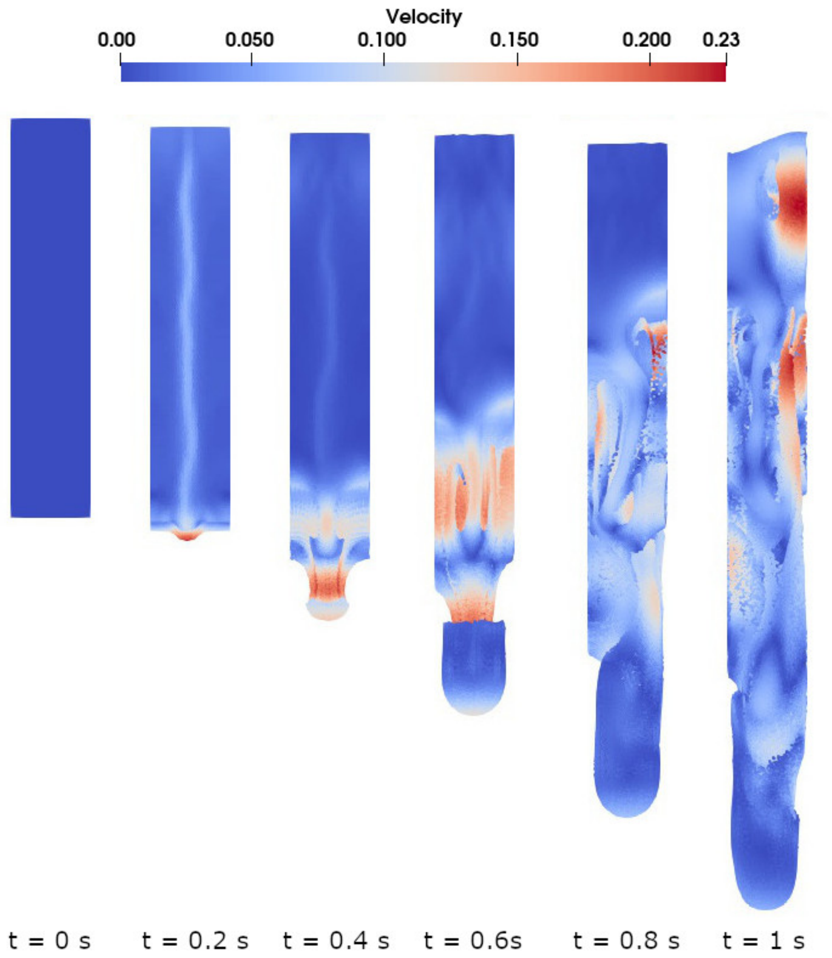


Figure 8: Side view of the 3D growing Rayleigh-Taylor instability plumes showing the particles' positions and velocity at different simulation times.

The bubble at $t = 1\ s$ is responsible for the sudden turn over in the curves observed at the end of the mixing layer growth of Fig. 7. From an angle, we show the 3D plumes for $A = 0.286$ at $t = 0.45\ s$ in Fig. 9 .

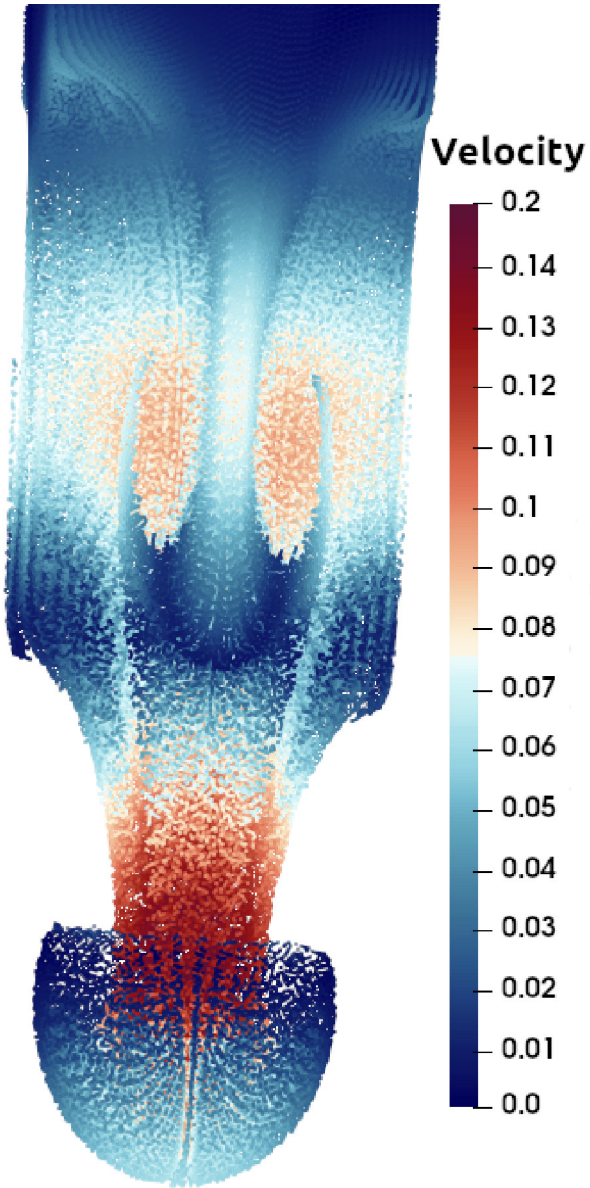


Figure 9: Angle view of the 3D growing Rayleigh-Taylor instability plumes for A=0.286 and t=0.45 s.

The instability is formed as a major single central mushroom-shaped plume surrounded by minor plumes. The instability remains symmetrical until $t = 0.5\ s$ after which it breaks down. The symmetrical plumes created demonstrate the mass conservation property of our solver. This is clear from the formation of the plumes where the light fluid moves upward to allow space for the heavy fluid moving downwards. Additionally, the circular shape of the mushroom inwards at its extremity is only achieved due to the highly accurate Q_n

interpolation of the particles' properties and position within a cell.

Spouted Bed Test Case

We simulate a rectangular spouted bed. We calculate the time averaged particles' velocities at different positions in the bed and compare the data obtained with experimental results obtained using particle image velocimetry.[65]

Simulation Setup

The bed size, geometry, solid and fluid properties are extracted from the experiment of Yue et al.[65] Figure 10 shows the schematic geometry of the simulation.

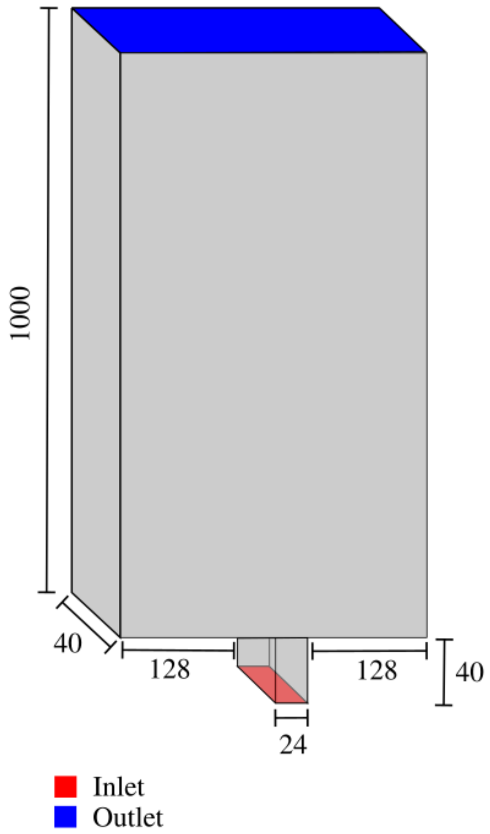


Figure 10: Scheme of the spouted bed geometry (unit: mm).

The authors did not specify the number of particles used but they specified the static bed height to be 200 *mm*. Based on the given geometry, and in order to obtain a similar static bed height, we use 175800 particles with the same properties as those used by Yue et al.[65] The particles used in this test case belong to the Geldart D group. In order to obtain better distribution of the flow and stability of the solver, we add to the geometry a small channel at the bottom of the bed through which we introduce the gas. The geometry and mesh were generated using the open source software GMSH.[66] For this case, we solve model A of the VANS equations. The gravitational acceleration of 9.81 m/s^2 was applied in the y-direction. The DEM time step used was 4.02% of the Rayleigh time step. The study is performed using Q1-Q1 for both PCM and SPM with 7 satellite particles and Q2-Q1 elements using only PCM to calculate the void fraction, a smoothing length of $L^2 = 5d_p^2$ for the void fraction, and a grad-div stabilization parameter (c^*) equal to the element size. For the Q2-Q1 simulation, convergence was difficult for (c^*) the size of the element, so the value of (c^*) 10 times the element size was chosen as it made the Q2-Q1 simulations robust. We increment the spout inlet velocity gradually from 0 m/s at $t = 0$ s until it reached its maximum value of 20.8 m/s at $t = 0.05$ s. This ensure a smooth transition of the bed from rest to the fluidized regime and prevents an initial shock from occurring due to the sudden introduction of a high velocity at $t = 0$ s. Table S3 presents the physical and numerical parameters for this test case.

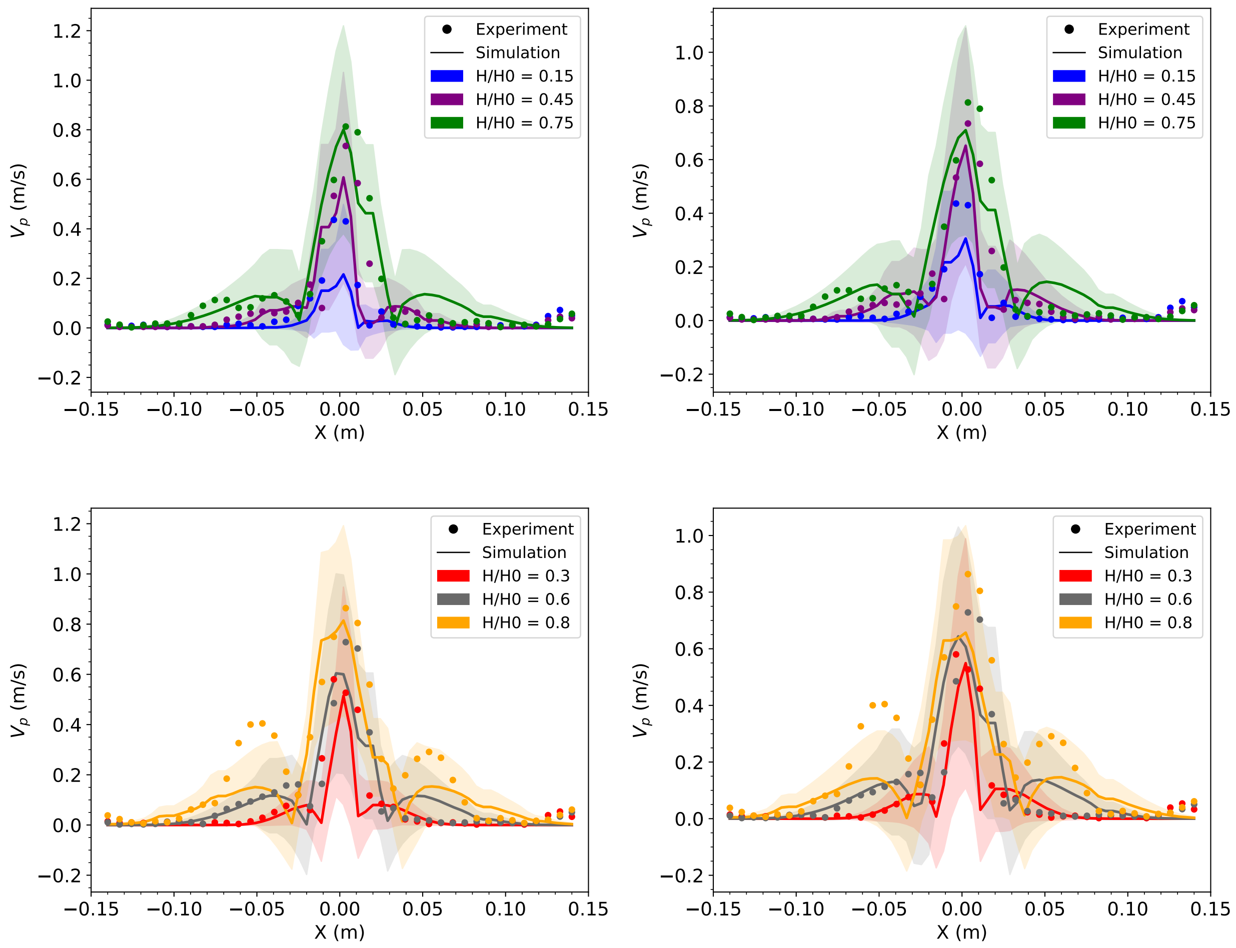
We perform the simulations for four particle-particle and particle-wall friction coefficients (0.3, 0.2, 0.1, and 0.05) and determine that a value of 0.3 for both allowed for the most realistic behavior of the spouted bed that is comparable to the behavior obtained from the experiments. The coefficient of rolling friction had some effect on the behavior of the bed and using a value of 0 led to the closest results with the experiments, thus it was neglected by assuming no rolling friction of particles. We apply slip boundary condition for all walls of the bed and the channel except the channel's base (inlet) and the bed's top wall (outlet). At the inlet, we apply a Dirichlet boundary condition with the value of the inlet velocity in the y-direction. For the outlet, we apply a zero traction boundary condition when the fluid is leaving the domain and we penalize the flow when the fluid is inbound. This prevents flow re-entry when there is turbulent structures or vortices leaving the domain. The outlet boundary condition imposed is:

$$\int_{\Gamma_o} (\nu \nabla \boldsymbol{u} \cdot \boldsymbol{n} - p\ I \cdot \boldsymbol{n} - \beta (\boldsymbol{u} \cdot \boldsymbol{n})\_ \boldsymbol{u}) \cdot \boldsymbol{v}\, d\Gamma = 0 \tag{54}$$

where β is a constant, p is the pressure, I is the identity matrix, $(\boldsymbol{u} \cdot \boldsymbol{n})\_$ is $min(0, \boldsymbol{u} \cdot \boldsymbol{n})$ $\boldsymbol{v}$ is the velocity test function and Γ_o is the outlet boundary. For further details, we refer the reader to the work of Arndt et al.[67] In order to prevent the particles from falling into the channel, we define a floating wall which is an imaginary wall only felt by the particles at the top of the channel at the intersection between the channel and the bed base.

Results and Discussion

For both Q1-Q1 and Q2-Q1 simulations, we measure the time averaged particle velocity profile in the direction of the flow at different heights (y-axis) of the bed and we compare the values with the experimental results of Yue et al.[65] The results obtained using both the PCM and SPM void fraction schemes are presented. The heights in Figures 11 and 12 are normalized by the static bed height.



Results obtained using PCM

Results obtained using SPM

Figure 11: Distribution of particle velocity magnitude in vectors along the X direction at different height levels for Q1-Q1 simulation.

Figure 11 shows the average particles' velocity magnitude at different heights in the spouted bed along the bed's width for the Q1-Q1 simulation using the PCM and SPM void fraction schemes. The spout is centralized at $x = 0\ m$ and the velocities obtained from the simulation agrees well with the experimental velocities. It is important to note that the experimental velocities were measured using Particle Image Velocimetry (PIV) which already involves errors in measurement. Around the central spout, we observe re-circulation zones at $x \in [0.05, 0.1]\ m$ & $x \in [-0.1, -0.05]\ m$. The calculated re-circulation velocity magnitude is smaller than that measured in the experiments. This could be due to errors in the PIV measurements as the zones are not symmetric in the experiments contrary to what they should be. Additionally, we calculate the particles' average velocity magnitude by averaging over the finite elements cells which is different than the averaging procedure performed in the experiments. However, the general trend of the results follows well that of the experiments. We average over the last 15 s out of the 20 s of the simulation to ensure that the averaging is performed only when the pseudo steady state has been reached. We show the standard deviation of our curves as a zone of the same color. This standard deviation represents the fluctuation of particles' velocity at this location with respect to time. All experimental velocities lie in the respective standard deviation zones except for the re-circulation velocities at $H/H_0 = 0.8$. This demonstrates that despite all possible measurement and averaging errors, the simulation gives good results comparable to the experimental results of the bed.

The difference between results obtained using PCM and QCM are minimal. The main difference lies in the spout region where the PCM scheme leads to higher average particle velocities compared to the SPM scheme. This might be explained by the strong temporal discontinuity of the PCM scheme which can lead to high gradients in the void fraction profile resulting in high velocity gradients as well. Even though the SPM is also discontinuous in time, the effect of this discontinuity is reduced due to the refinement of the particles.

To investigate and better understand the effect of high order methods, we simulate the same case using Q2-Q1 elements. This is done only using the PCM void fraction scheme, as the effect of the void fraction scheme was described using Q1-Q1 elements and the aim is to determine the effect of high order schemes on the behavior of the spouted bed. The results are shown in Fig. 12

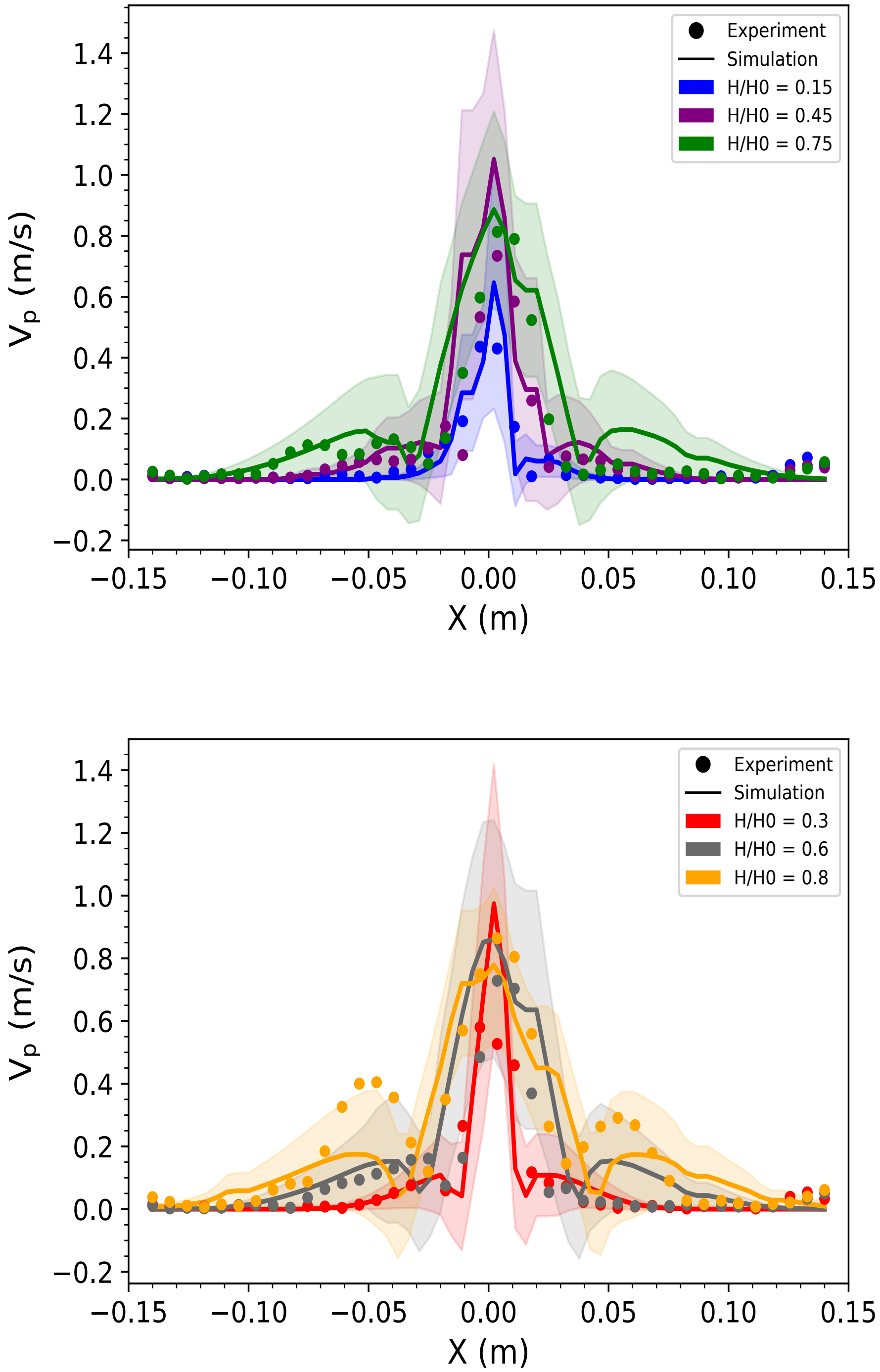


Figure 12: Distribution of particle velocity magnitude in vectors along the X direction at different height levels for Q2-Q1 simulation using PCM.

The differences between Q1-Q1 and Q2-Q1 simulations' results are mainly localized in the spout region at low values of the reference height. The four smallest heights from 0.15 till 0.6 have a much higher average particle velocity in Q2-Q1 simulations compared to the Q1-Q1 simulation while the two largest heights of 0.75 and 0.8 have a relatively similar velocity profile in both simulations. This can be attributed to the fact that the Q2 interpolation captures more adequately the fluid jet in the spout region because it induces less numerical dissipation. The Q2-Q1 scheme also has a richer interpolation space and can capture a velocity profile that abruptly changes in space. The Q2-Q1 results are thus in much better agreement with the experimental results in this region. In general, the oscillation of the curves in time are more prominent in the Q1-Q1 than in the Q2-Q1 results. This explains the overall thinner standard deviation zones in Fig. 12 compared to Fig. 11. However, around $x = 0\ m$ and for the heights closer to the inlet, the standard deviation is high. This is caused by the high velocity and turbulent structures in this region leading to strong oscillations with time. Finally, the experimental particles' velocities in the re-circulation zones for height 0.8 lies within the standard deviation zone calculated from the simulation. Thus, this zone is also better captured in the Q2-Q1 simulation.

In general, certain studies which included CFD-DEM simulations of spouted beds produced results which were similar to the results obtained by our Q1-Q1 simulation while others did not. Jajcevic et al.[55] obtained results which agreed well with their experiments for a single spout where the higher bed height possessed the larger solid velocities while Gryczka et al.[68] obtained results which did not agree with their experiments where the simulated solid velocities were much smaller than the experimentally measured velocities. Zhao et al.[69] and Duarte et al.[70] both obtained results which were tending to an inverted trend than our Q1-Q1 results. They obtained the largest particle velocities at the smallest heights and the lowest particle velocities with the largest heights. The results of Duarte et al.[70] did not agree well with experimental results of He et al.[71] against which they were comparing.

Effect of load balancing on computational time

Our CFD-DEM solver supports load balancing; however, there are many parameters that should be accounted for to ensure enhanced computational efficiency such as the ratio of solid particle to fluid cell weight and the frequency of load balancing. Among these parameters, we focus on studying the ratio of solid particle to fluid cell weight. For the purpose of this study and to reduce the computational cost of the simulation, we simulated the spouted bed test case with a coarser mesh of 18-70-4 for the bed and 2-3-4 for the channel in x-y-z and with half the number of particles (87,900 particles) using Q1-Q1 elements for different particle to fluid cell weight ratios at six different numbers of processors. The simulation was run for 1 s of real time and a time step of 0.001 s using the dynamic load balancing approach which automatically detects the load balancing steps from the distribution of particles and cells among the processors.[33] All other parameters were kept the same as given in Table S. The result is shown in Fig. 13.

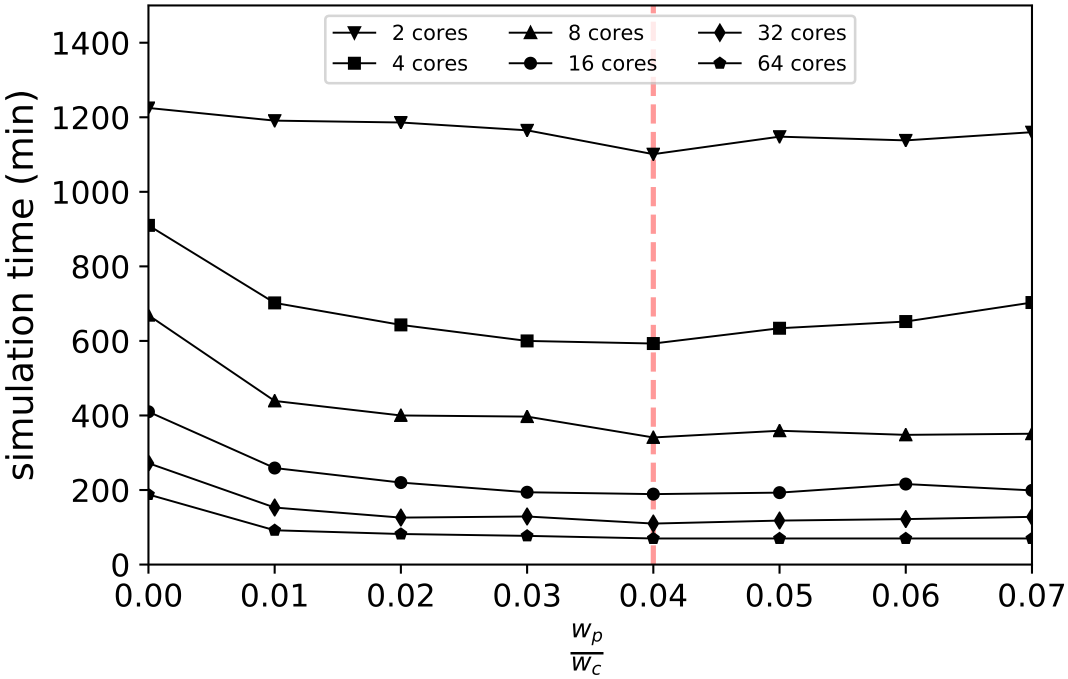


Figure 13: Overall simulation time as a function of the the particle to cell weight ratio for different number of processors.

the difference in speed between two neighboring ratios of 0.04 is negligible as it only adds few minutes to the simulation. As such, knowing the neighborhood of the optimal weight ratio to use is sufficient as it can affect drastically the speed especially for simulations with a lower number of processors. Moreover, looking at the trend of simulation time as a function of weight ratios, we observe that the bigger the number of processors used, the less effect the change in weight ratios has on the overall simulation time.

We plot the speedup of the simulation as a function of the number of processors in a logarithmic scale as shown in Fig. 14.

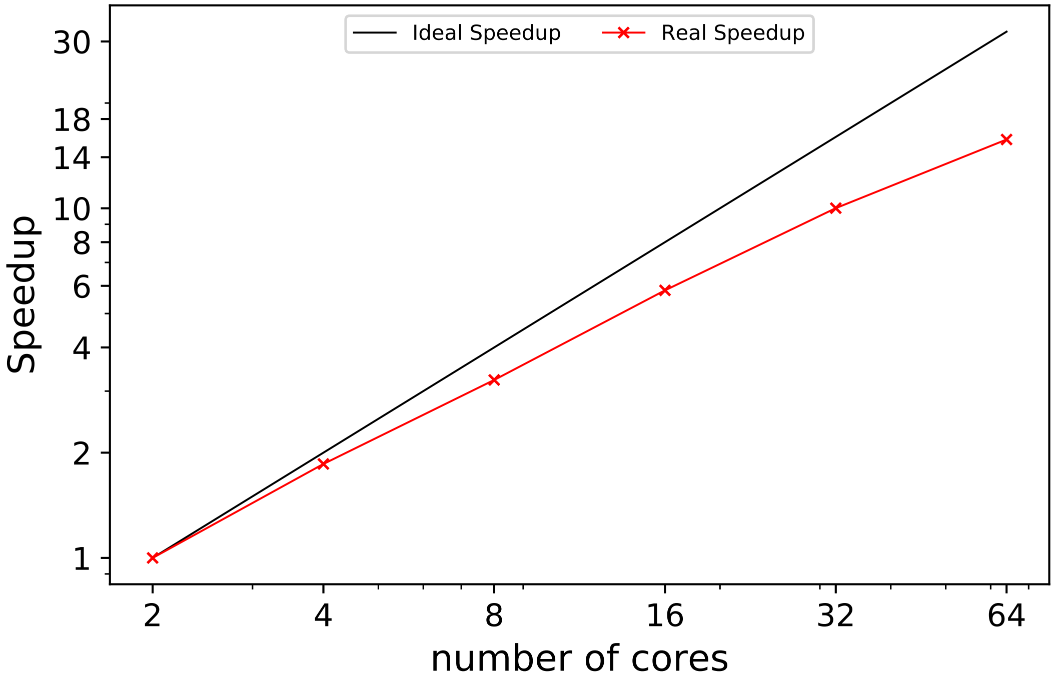


Figure 14: Simulation speedup as a function of processors in logarithmic scale

Overall, the CFD-DEM solver scales well. For a lower number of processors, our solver scales better than for an increased number of processors. This is because at lower number of processors, the matrix assembly of the CFD solver is the most consuming part and can take up to 70 % of a time step. The higher the number of processors, the faster the matrix assembly gets as the matrices on each processor become smaller. However, increasing the number of processors beyond a certain value, in this case 16, leads to a decrease in the scaling. This is due to the DEM solver that becomes slower. Our DEM solver uses the notion of ghost particles which are particles seen by the current processor but owned by another neighboring processor. As the number of processors increase, the number of ghost particles increase as well as a single particle is owned by one processor but can be seen as a ghost particle by more and more processors. This leads to an increase of the computational cost as the DEM solver has to perform more calculations related to the ghost particles. For 2 processors, an average of 43950 particles exists in a single processor. This value approximately decreases by a factor of 2 everytime we double the number of processors. At 16 processors, this value is around 5493 particles per processor. Additional decrease in this value leads to a decrease in the scalability of the DEM solver for which quasi-ideal scaling could be obtained at 20,000 particles per core.[33]

Conclusion

This work presents a verified and validated stabilized finite element approach for the modeling of multi-phase flows using unresolved CFD-DEM. Our unified coupled solver is robust, parallel and supports both models A and B of the VANS equations. It supports high order finite elements which allows for more accurate results using bigger mesh sizes thus respecting the requirement for large enough mesh sizes without compromising accuracy. Also, it is among the first CFD-DEM software to support load balancing. This helps improve the computational efficiency of our solver as available resources can be exploited more effectively by balancing work loads among processors. Moreover, the DEM and CFD portions of the code run on the same mesh which results in cheaper communication and coupling between them. Furthermore, our formulation is locally and globally conservative.

Our solver can simulate solid-fluid flows for different applications. We validate the code with different test cases. In the fluidized bed test case, we show that even though model A and B are mathematically equivalent, they result in different instantaneous physical behavior of fluidized beds. Model A is more stable and results in less stiffness of the system being solved. From the Rayleigh Taylor instability, we prove mass conservation of our solver as well as accurate interpolation of the particle-fluid forces. The spouted bed test case allows us to compare the particles' velocity at different bed heights with experimental data. In this case, the use of high-order scheme leads to time-averaged particle velocity profiles which are in better agreement with experimental results, especially in the spout region. We can thus conclude that our results are realistic and our formulation is validated. We think that the use of higher-order scheme is a promising way to simulate complex solid-fluid flows using CFD-DEM since it allows for a richer and more accurate description of the flow field while maintaining an adequate particle to cell size ratio.

Supporting Information

The tables including the details of the physical properties of fluids and particles and the numerical parameters for the setup of the simulations for the particle sedimentation, fluidized bed, and spouted bed test cases are included separately.

This information is available free of charge via the Internet at http://pubs.acs.org/.

Acknowledgements

The authors would like to thank the deal.II community for their support as well as Professor Yansong Shen for his clarification concerning the spouted bed experimental results. Bruno Blais would like to acknowledge the financial support from the Natural Sciences and Engineering Research Council of Canada (NSERC) through the RGPIN-2020-04510 Discovery Grant. The authors would also like to acknowledge technical support and computing time provided by Compute Canada and Calcul Québec.

References

(1) Yeoh, G. H.; Tu, J. In *Computational Techniques for Multiphase Flows*, second edition ed.; Yeoh, G. H., Tu, J., Eds.; Butterworth-Heinemann: Amsterdam, 2019; pp 227–319.

(2) Norouzi., H. R. *Coupled CFD-DEM Modeling : Formulation, Implementation and Application to Multiphase Flows.*, 1st ed.; Wiley: Newark, 2016.

(3) Guan Heng Yeoh,; Tu, J. In *Computational Techniques for Multiphase Flows*, second edition ed.; Yeoh, G. H., Tu, J., Eds.; Butterworth-Heinemann: Amsterdam, 2019; pp 1 – 18.

(4) van der Hoef, M.; Ye, M.; van Sint Annaland, M.; Andrews, A.; Sundaresan, S.; Kuipers, J. Multiscale Modeling of Gas-Fluidized Beds. *Advances in Chemical Engineering* **2006**, *31*, 65–149.

(5) Sahoo, P.; Sahoo, A. Fluidization and Spouting of Fine Particles: A Comparison. *Advances in materials science and engineering* **2013**, *2013*, 1–7.

(6) Cocco, R.; Karri, S. B. R.; Knowlton, T. Introduction to Fluidization. *Chemical engineering progress* **2014**, *110*, 21–29.

(7) Discrete Element Modeling - DEM Software: Altair Edem. `https://www.altair.com/edem/`.

(8) Rocky DEM. 2022; `https://rocky.esss.co/software/`.

(9) DCS computing. 2022; `https://www.aspherix-dem.com/`.

(10) Kloss, C.; Goniva, C.; Hager, A.; Amberger, S.; Pirker, S. Models, algorithms and validation for opensource DEM and CFD–DEM. *Progress in Computational Fluid Dynamics, an International Journal* **2012**, *12*, 140–152.

(11) Vaclav Smilauer,; Angelidakis, V.; Catalano, E.; Caulk, R.; Chareyre, B.; Chèvremont, W.; Dorofeenko, S.; Duriez, J.; Dyck, N.; Elias, J.; Er, B.; Eulitz, A.; Gladky, A.; Guo, N.; Jakob, C.; Kneib, F.; Kozicki, J.; Marzougui, D.; Maurin, R.; Modenese, C.; Pekmezi, G.; Scholtès, L.; Sibille, L.; Stransky, J.; Sweijen, T.; Thoeni, K.; Yuan, C. *Yade Documentation*, 3rd ed.; The Yade Project, 2021.

(12) Comsol: a multiphysics software for optimizing designs. `https://www.comsol.com/`.

(13) Ansys FLUENT. `https://www.ansys.com/products/fluids/ansys-fluent`.

(14) Multiphysics Computational Fluid Dynamics (CFD) Simulation Software: Siemens Software. `https://www.plm.automation.siemens.com/global/en/products/simcenter/STAR-CCM.html`.

(15) Syamlal, M.; Rogers, W.; OBrien, T. J. *MFIX documentation theory guide*; 1993.

(16) OpenFOAM. `https://www.openfoam.com/`.

(17) Peng, Z.; Doroodchi, E.; Luo, C.; Moghtaderi, B. Influence of void fraction calculation on fidelity of CFD-DEM simulation of gas-solid bubbling fluidized beds. *AIChE journal* **2014**, *60*, 2000–2018.

(18) ZHOU, Z. Y.; KUANG, S. B.; CHU, K. W.; YU, A. B. Discrete particle simulation of particle–fluid flow: model formulations and their applicability. *Journal of Fluid Mechanics* **2010**, *661*, 482–510 .

(19) Rui Sun,; Xiao, H. Diffusion-based coarse graining in hybrid continuum–discrete solvers: Theoretical formulation and a priori tests. *International Journal of Multiphase Flow* **2015**, *77*, 142–157.

(20) PFC. `http://www.itascacg.com/software/pfc`.

(21) CFDEM project. 2022; `https://www.cfdem.com/`.

(22) Freireich, B.; Kodam, M.; Wassgren, C. An exact method for determining local solid fractions in discrete element method simulations. *AIChE journal* **2010**, *56*, 3036–3048.

(23) C.L. Wu,; Zhan, J.; Li, Y.; Lam, K.; Berrouk, A. Accurate void fraction calculation for three-dimensional discrete particle model on unstructured mesh. *Chemical Engineering Science* **2009**, *64*, 1260–1266.

(24) Boyce, C. M.; Holland, D. J.; Scott, S. A.; Dennis, J. S. Novel fluid grid and voidage calculation techniques for a discrete element model of a 3D cylindrical fluidized bed. *Computers & chemical engineering* **2014**, *65*, 18–27.

(25) Link, J.; Cuypers, L.; Deen, N.; Kuipers, J. Flow regimes in a spout–fluid bed: A combined experimental and simulation study. *Chemical Engineering Science* **2005**, *60*, 3425–3442.

(26) Khawaja, H. A.; Scott, S. A.; Virk, M. S.; Moatamedi, M. Quantitative analysis of accuracy of voidage computations in CFD-DEM simulations. *The Journal of Computational Multiphase Flows* **2012**, *4*, 183–192.

(27) Zhengbiao Peng,; Moghtaderi, B.; Doroodchi, E. A modified direct method for void fraction calculation in CFD–DEM simulations. *Advanced Powder Technology* **2016**, *27*, 19–32.

(28) Falah Alobaid,; Ströhle, J.; Epple, B. Extended CFD/DEM model for the simulation of circulating fluidized bed. *Advanced Powder Technology* **2013**, *24*, 403–415.

(29) Falah Alobaid, A particle–grid method for Euler–Lagrange approach. *Powder Technology* **2015**, *286*, 342–360.

(30) Surya Deb,; Tafti, D. K. A novel two-grid formulation for fluid–particle systems using the discrete element method. *Powder Technology* **2013**, *246*, 601–616.

(31) Daniel Hirche,; Hinrichsen, O. Implementation and evaluation of a three-level grid method for CFD-DEM simulations of dense gas–solid flows. *Chemical Engineering Journal Advances* **2020**, *4*, 100048.

(32) Meisam Farzaneh,; Sasic, S.; Almstedt, A.-E.; Johnsson, F.; Pallarès, D. A novel multigrid technique for Lagrangian modeling of fuel mixing in fluidized beds. *Chemical Engineering Science* **2011**, *66*, 5628–5637.

(33) Golshan, S.; Munch, P.; Gassmöller, R.; Kronbichler, M.; Blais, B. Lethe-DEM: An open-source parallel discrete element solver with load balancing. *Computational Particle Mechanics* **2022**, 1–20.

(34) Blais, B.; Barbeau, L.; Bibeau, V.; Gauvin, S.; Geitani, T. E.; Golshan, S.; Kamble, R.; Mirakhori, G.; Chaouki, J. Lethe: An open-source parallel high-order adaptative CFD solver for incompressible flows. *SoftwareX* **2020**, *12*, 100579–.

(35) Daniel Arndt,; Bangerth, W.; Blais, B.; Fehling, M.; Gassmöller, R.; Heister, T.; Heltai, L.; Köcher, U.; Kronbichler, M.; Maier, M.; Munch, P.; Pelteret, J.-P.; Proell, S.; Simon, K.; Turcksin, B.; Wells, D.; Zhang, J. The `deal.II` Library, Version 9.3. *Journal of Numerical Mathematics* **2021**, *29*, 171–186.

(36) Geitani, T. E.; Golshan, S.; Blais, B. A High Order Stabilized Solver for the Volume Averaged Navier-Stokes Equations. Arxiv math.NA, Submitted on 2022-06-06, doi: 10.48550/ARXIV.2206.02842.

(37) DIMITRI GIDASPOW, In *Multiphase Flow and Fluidization: Continuum and Kinetic Theory Descriptions with Applications*; GIDASPOW, D., Ed.; Academic Press: San Diego, 1994; pp 1–29.

(38) Blais, B.; Bertrand, F. On the use of the method of manufactured solutions for the verification of CFD codes for the volume-averaged Navier–Stokes equations. *Computers & fluids* **2015**, *114*, 121–129.

(39) ZHOU, Z. Y.; KUANG, S. B.; CHU, K. W.; YU, A. B. Discrete particle simulation of particle–fluid flow: model formulations and their applicability. *Journal of fluid mechanics* **2010**, *661*, 482–510.

(40) Tezduyar, T.; Sathe, S. Stabilization Parameters in Supg and Pspg Formulations. *Journal of Computational and Applied Mechanics* **2003**, *4*, 71–88.

(41) Olshanskii, M.; Lube, G.; Heister, T.; Löwe, J. Grad–div stabilization and subgrid pressure models for the incompressible Navier–Stokes equations. *Computer Methods in Applied Mechanics and Engineering* **2009**, *198*, 3975–3988.

(42) Brenner, S.; Scott, L. *The Mathematical Theory of Finite Element Methods*; Texts in Applied Mathematics; Springer: New York, 2002.

(43) Golshan, S.; Sotudeh-Gharebagh, R.; Zarghami, R.; Mostoufi, N.; Blais, B.; Kuipers, J. Review and implementation of CFD-DEM applied to chemical process systems. *Chemical engineering science* **2020**, *221*, 115646–.

(44) Clarke, D. A.; Sederman, A. J.; Gladden, L. F.; Holland, D. J. Investigation of void fraction schemes for use with CFD-DEM simulations of fluidized beds. *Industrial & Engineering Chemistry Research* **2018**, *57*, 3002–3013 .

(45) Yanjie Li,; Xu, Y.; Thornton, C. A comparison of discrete element simulations and experiments for 'sandpiles' composed of spherical particles. *Powder Technology* **2005**, *160*, 219–228 .

(46) Blais, B.; Vidal, D.; Bertrand, F.; Patience, G. S.; Chaouki, J. Experimental Methods in Chemical Engineering: Discrete Element Method—DEM. *Canadian journal of chemical engineering* **2019**, *97*, 1964–1973.

(47) Delacroix, B.; Bouarab, A.; Fradette, L.; Bertrand, F.; Blais, B. Simulation of gran-

ular flow in a rotating frame of reference using the discrete element method. *Powder Technology* **2020**, *369*, 146–161.

(48) Gassmöller, R.; Lokavarapu, H.; Heien, E.; Puckett, E. G.; Bangerth, W. Flexible and Scalable Particle-in-Cell Methods With Adaptive Mesh Refinement for Geodynamic Computations. *Geochemistry, geophysics, geosystems : G3* **2018**, *19*, 3596–3604.

(49) Pepiot, P.; Desjardins, O. Numerical analysis of the dynamics of two- and three-dimensional fluidized bed reactors using an Euler–Lagrange approach. *Powder Technology* **2012**, *220*, 104–121, Selected Papers from the 2010 NETL Multiphase Flow Workshop.

(50) Elghobashi, S. Particle-laden turbulent flows: Direct simulation and closure models. *Applied Scientific Research* **1991**, *48*, 301–314.

(51) Burstedde, C.; Wilcox, L. C.; Ghattas, O. `p4est`: Scalable Algorithms for Parallel Adaptive Mesh Refinement on Forests of Octrees. *SIAM Journal on Scientific Computing* **2011**, *33*, 1103–1133.

(52) Dallavalle, J. M. *Micromeritics : the technology of fine particles*, 2nd ed.; Pitman: New York, 1948.

(53) Di Felice, R. The voidage function for fluid-particle interaction systems. *International journal of multiphase flow* **1994**, *20*, 153–159.

(54) Rong, L.; Dong, K.; Yu, A. Lattice-Boltzmann simulation of fluid flow through packed beds of uniform spheres: Effect of porosity. *Chemical engineering science* **2013**, *99*, 44–58.

(55) Jajcevic, D.; Siegmann, E.; Radeke, C.; Khinast, J. G. Large-scale CFD–DEM simulations of fluidized granular systems. *Chemical Engineering Science* **2013**, *98*, 298–310.

(56) Beetstra, R.; van der Hoef, M. A.; Kuipers, J. A. M. Drag force of intermediate Reynolds number flow past mono- and bidisperse arrays of spheres. *AIChE Journal* **2007**, *53*, 489–501.

(57) Hamid Reza Norouzi,; Golshan, S.; Zarghami, R. On the drag force closures for multiphase flow modeling. *Chemical Product and Process Modeling* **2022**, *17*, 531–582.

(58) Bérard, A.; Patience, G. S.; Blais, B. Experimental methods in chemical engineering: Unresolved CFD-DEM. *Canadian journal of chemical engineering* **2020**, *98*, 424–440.

(59) ERGUN, S. Fluid flow through packed columns. *Chem. Eng. Prog.* **1952**, *48*, 89–94.

(60) Wen, C.; Yu, Y. A generalized method for predicting the minimum fluidization velocity. *AIChE Journal* **1966**, *12*, 610–612.

(61) Noda, K.; Uchida, S.; Makino, T.; Kamo, H. Minimum fluidization velocity of binary mixture of particles with large size ratio. *Powder Technology* **1986**, *46*, 149–154.

(62) Anantharaman, A.; Cocco, R. A.; Chew, J. W. Evaluation of correlations for minimum fluidization velocity (Umf) in gas-solid fluidization. *Powder Technology* **2018**, *323*, 454–485.

(63) Snider, D. An Incompressible Three-Dimensional Multiphase Particle-in-Cell Model for Dense Particle Flows. *Journal of computational physics* **2001**, *170*, 523–549.

(64) Dimonte, G.; Youngs, D. L.; Dimits, A.; Weber, S.; Marinak, M.; Wunsch, S.; Garasi, C.; Robinson, A.; Andrews, M. J.; Ramaprabhu, P.; Calder, A. C.; Fryxell, B.; Biello, J.; Dursi, L.; MacNeice, P.; Olson, K.; Ricker, P.; Rosner, R.; Timmes, F.; Tufo, H.; Young, Y.-N.; Zingale, M. A comparative study of the turbulent Rayleigh–Taylor instability using high-resolution three-dimensional numerical simulations: The Alpha-Group collaboration. *Physics of Fluids* **2004**, *16*, 1668–1693.

(65) Yue, Y.; Wang, S.; Bahl, P.; de Silva, C.; Shen, Y. Experimental Investigation of Spout Deflection in a Rectangular Spouted Bed by the PIV Method. *Industrial & engineering chemistry research* **2020**, *59*, 13810–13819.

(66) Geuzaine, C.; Remacle, J.-F. Gmsh: A 3-D finite element mesh generator with built-in pre- and post-processing facilities. *International Journal for Numerical Methods in Engineering* **2009**, *79*, 1309–1331.

(67) Arndt, D.; Braack, M.; Lube, G. Finite Elements for the Navier-Stokes Problem with Outflow Condition. Numerical Mathematics and Advanced Applications ENUMATH 2015. Cham, 2016; pp 95–103.

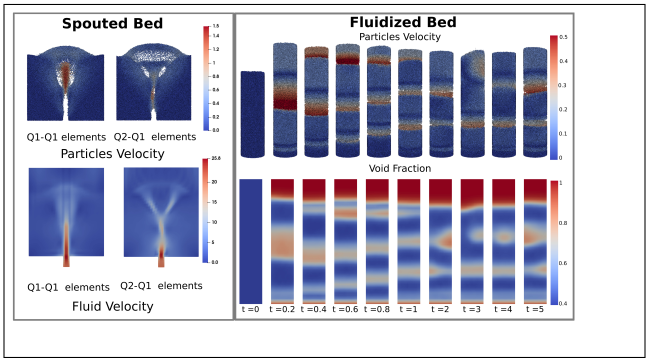
(68) O. Gryczka,; Heinrich, S.; Deen, N.; van Sint Annaland, M.; Kuipers, J.; Jacob, M.; Mörl, L. Characterization and CFD-modeling of the hydrodynamics of a prismatic spouted bed apparatus. *Chemical Engineering Science* **2009**, *64*, 3352–3375.

(69) Zhao, X.-L.; Yao, Q.; Li, S.-Q. Effects of Draft Tubes on Particle Velocity Profiles in Spouted Beds. *Chemical Engineering & Technology* **2006**, *29*, 875–881.

(70) Duarte, C.; Olazar, M.; Murata, V.; Barrozo, M. Numerical simulation and experimental study of fluid–particle flows in a spouted bed. *Powder Technology* **2009**, *188*, 195–205.

(71) He, Y.-L.; Lim, C. J.; Grace, J. R.; Zhu, J.-X.; Qzn, S.-Z. Measurements of voidage profiles in spouted beds. *The Canadian Journal of Chemical Engineering* **1994**, *72*, 229–234.

TOC Graphic


Spouted Bed
1.5
1.4
1.2
1
0.8
0.6
0.4
0.2
0.0
Q1-Q1 elements
Q2-Q1 elements
Particles Velocity
25.8
20
15
10
5
0.0
Q1-Q1 elements
Q2-Q1 elements
Fluid Velocity
Fluidized Bed
Particles Velocity
0.5
0.4
0.3
0.2
0.1
0
Void Fraction
1
0.8
0.6
0.4
t =0
t =0.2
t =0.4
t =0.6
t =0.8
t =1
t =2
t =3
t =4
t =5