UNIVERSIDADE FEDERAL DE SÃO CARLOS
DEPARTAMENTO DE ENGENHARIA QUÍMICA
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA QUÍMICA

Victor Oliveira Ferreira

Liquid-solid fluidized bed dynamics: experimental and computational analysis

Estudo da dinâmica de leito fluidizado sólido-líquido: uma análise experimental e computacional

São Carlos, SP

2023

Victor Oliveira Ferreira

Liquid-solid fluidized bed dynamics: experimental and computational analysis

Estudo da dinâmica de leito fluidizado sólido-líquido: uma análise experimental e computacional

Tese apresentada ao Programa de Pós-Graduação em Engenharia Química da Universidade Federal de São Carlos para obtenção do título de Doutor em Engenharia Química.

Orientadora Profa. Dra. Gabriela Cantarelli Lopes
Coorientador Prof. Dr. Bruno Blais

São Carlos, SP

2023

UNIVERSIDADE FEDERAL DE SÃO CARLOS

Centro de Ciências Exatas e de Tecnologia
Programa de Pós-Graduação em Engenharia Química

Folha de Aprovação

Defesa de Tese de Doutorado do candidato Victor Oliveira Ferreira, realizada em 06/11/2023.

Comissão Julgadora:

Profa. Dra. Gabriela Cantarelli Lopes (UFSCar)

Prof. Dr. Bruno Blais (Polytechnique Montréal)

Prof. Dr. Dyrney Araújo dos Santos (UFG)

Prof. Dr. Rodrigo Béttega (UFSCar)

Prof. Dr. Gabriel Henrique Justi (UNIPAMPA)

O Relatório de Defesa assinado pelos membros da Comissão Julgadora encontra-se arquivado junto ao Programa de Pós-Graduação em Engenharia Química.

Este trabalho é dedicado às pessoas que eu mais amo nesse mundo.
aos meus pais, Clodoaldo e Cybele;
aos meus irmãos, Tarcísio e Luiza;
à minha avó, Socorro;
à minha esposa, Izabella;
dedico este à vocês.

AGRADECIMENTOS/AKNOWLEDGEMENTS

Gostaria de agradecer aos meus pais, Clodoaldo e Cybele, por terem trilhado e pavimentado o meu caminho até aqui. Mãe e pai, muito obrigado por todo o amor que vocês sempre me deram. Mãe, muito obrigado por todo o suporte, por me compreender e acolher, por tudo que você me proporcionou e permitiu construir. Pai, muito obrigado por incentivar minha curiosidade desde meus primeiros anos de vida, por me introduzir ao mundo da ciência e da pesquisa acadêmica, e por querer sempre extrair o melhor de mim. Amo demais vocês!

Ao meu irmão, Tarcísio, pela amizade e por ter dividido comigo a incrível experiência de morar em Montréal. Muito obrigado por ter me acolhido em sua casa. Obrigado por tudo, incluindo os conselhos, as noites jogando videogame, as conversas, os abraços, os bagels. Te amo, "rimão"!

À minha irmã, Luiza, por ser uma fonte infinita de motivação e inspiração. Te amo, irmã!

À minha esposa, Izabella, que dividiu comigo intensamente cada etapa dessa aventura. Sofremos juntos com a saudade, os desafios de morar em um lugar novo, as frustrações pelo caminho. Vivemos juntos as vitórias, as alegrias, os reencontros. Muito obrigado por todo o incentivo e pela sua imensurável compreensão. Te amo demais, meu amor!

À minha orientadora, Professora Dra. Gabriela Cantarelli Lopes, por ter me aceitado como seu orientado e por toda a dedicação à minha formação enquanto pesquisador. Professora, obrigado por todo o incentivo, por cada discussão, por ter me permitido errar e aprender com os erros, por ter sido compreensiva nos momentos de dificuldade, pelos estresses, pelas risadas, e por ter me apoiado durante toda a minha jornada na Pós-Graduação. Muito obrigado por toda a confiança que você depositou em mim!

To my co-supervisor, Professor Bruno Blais, Ph.D., for receiving me in his group at Polytechnique Montréal and for his dedication to this project. Prof, I am immensely grateful that you replied to that first e-mail and for all the opportunities you have given me ever since. Your contributions to this work and to my education as a researcher are uncountable. Thank you for allowing me to be part of your team and share such a friendly and collaborative environment. Thank you for each lesson. Thank you for your deep interest in "capivaras", as well!

Aos meus companheiros de laboratório Daniel, João, Karla e Ricardo pela amizade e todo o apoio. Extendo este ao amigo Rauber, que compartilhou comigo os desafios e conquistas de se fazer intercâmbio. Muito obrigado pela amizade e pelo ambiente fraterno proporcionado por vocês durante nossa estada na Pós-Graduação.

To my friends in Montréal, thank you for receiving me and making my stay in Montréal one of the most memorable experiences I have ever had. Clara, Lucka, and Toni thank you for sharing the “bunker” with me. Amishga, Audrey, Carole-Anne, Ghazaleh, Hélène, Laura, Olivier, Paul, Saad, Shahab, Valérie, and friends from Polytechnique; from scientific discussions to “just dance” matches, thank you for everything we have shared.

À minha avó, Socorro, meu padrinho, Valdir, minha madrinha, Nicéia, e toda a minha família por todo o amor e torcida. Amo muito todos vocês!

Ao querido amigo Breno por todo o suporte, acolhimento e aprendizado que obtive de nossas conversas. Seu apoio foi fundamental para que eu chegasse aqui.

À querida amiga Iara, pela sua imensa generosiadade e por todo o carinho. Muito obrigado por tudo!

Aos amigos Paulo e Cristhian, pela dedicação com a qual me ensinaram este ofício. Extendo este aos colegas da Universidade Federal do Pará.

Aos amigos queridos que deixei em Belém e que torcem pelo meu sucesso.

À Universidade Federal de São Carlos, ao Departamento de Engenharia Química e ao Programa de Pós-Graduação em Engenharia Química.

À Polytechnique Montréal.

Aos órgãos que fomentaram esta pesquisa, incluindo a Fundação de Amparo à Pesquisa do Estado de São Paulo (FAPESP) pela Bolsa de Doutorado-Direto (processo nº 2019/19173-9) e Bolsa de Estágio de Pesquisa no Exterior - BEPE (processo nº 2020/14567-6), e o Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq, processo número 408618/2018-3) pelo financiamento da construção do aparato experimental. O presente trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior – Brasil (CAPES) – Código de Financiamento 001.

“One child, one teacher, one book, one pen can change the world.”
(Malala Yousafzai)

“O custo do cuidado é sempre menor que o custo do reparo.”
(Marina Silva)

RESUMO

A fluidização sólido-líquido é uma técnica simples de intensificação de processos. O método ainda é pouco explorado, parcialmente em razão da sua imprevisibilidade. Avaliar sua dinâmica interna sem um robusto aparato experimental não é uma tarefa trivial, enquanto a predição da sua dinâmica depende de correlações simples que não refletem as variações temporais ou espaciais. Por outro lado, com o avanço do poder de processamento, fenômenos físicos podem ser simulados utilizando modelos fenomenológicos complexos e métodos numéricos robustos implementados em computadores. Nesse contexto, o método de acoplamento não resolvido entre a Dinâmica de Fluidos Computacional (CFD, do inglês *computational fluid dynamics*) e o Método dos Elementos Discretos (DEM, do inglês *discrete elements method*) se destaca por permitir, concomitantemente, uma descrição precisa do comportamento individual das partículas (incluindo interações com outros sólidos e com o fluido) enquanto preserva um custo computacional razoável. Nesta tese, foram aplicados métodos experimentais e o acoplamento CFD-DEM não resolvido para avaliar a dinâmica do leito fluidizado líquido-sólido. Os resultados experimentais foram usados para verificar a precisão da equação de Richardson-Zaki e de um método proposto baseado em correlações de força de arraste. O método proposto demonstrou precisão superior na maioria dos casos testados. As simulações de leito fluidizado líquido-sólido foram realizadas utilizando o Lethe, um software CFD, DEM, CFD-DEM e multifísica de código aberto. As simulações dos leitos fluidizados líquido-sólido em escala piloto foram validadas para uma ampla variedade de regimes. São apresentadas discussões sobre a escolha da correlação para força de arraste, a robustez do método com diferentes topologias de malha e a importância da força de sustentação de Saffman para a reprodução do comportamento estrutural do fluido. As simulações validadas foram usadas para avaliar o comportamento de mistura de leitos fluidizados líquido-sólido aplicando-se o método dos vizinhos mais próximos (NNM, do inglês *nearest-neighbors method*) e o índice de mistura de Doucet. O componente axial foi o mais lento na maioria das simulações. O tempo de mistura em função da vazão de fluido alcança um platô para leitos mais diluídos (com frações de fluido acima de 50%). As variações nas propriedades de interação das partículas não desempenharam um papel importante na avaliação quantitativa da mistura. Em leitos de partículas com números de Stokes muito baixos ou em leitos com concentrações muito altas, o aumento do coeficiente de atrito de deslizamento das partículas demonstrou ter um impacto negativo no desempenho da mistura.

Palavras-chave: Fluidização Líquida, Leito Fluidizado Sólido-Líquido, Escoamentos Multifásicos, FEM, CFD-DEM.

ABSTRACT

The liquid-solid fluidization is a simple process intensification strategy. The method remains underexplored, in part, due to the lack of predictability. It is not trivial to assess its internal dynamics without a robust experimental setup and the predictions of its fluid dynamics behavior rely on simple correlations that do not reflect internal variances either in time or space. On the other hand, with the advance of processing power, physics-based mathematical models can now be resolved using robust numerical methods implemented on computers. The unresolved CFD-DEM (Computational Fluid Dynamics-Discrete Elements Method) coupling is especially powerful for providing a precise description of particles' individual behavior (including interactions with other solids and the fluid) while preserving a reasonable computational cost. In this thesis, experimental methods and the unresolved CFD-DEM coupling were applied to assess the liquid-solid fluidized bed dynamics. Experimental measurements of the expansion behavior of beds were obtained for a wide variety of particles. The experimental results were used to verify the precision of the Richardson-Zaki equation and a proposed alternative method based on drag correlations. The latter has shown superior accuracy for most of the tested cases. The liquid-solid fluidized bed simulations were carried out in Lethe, an open-source CFD, DEM, CFD-DEM, and multiphysics software. Lethe simulations of the pilot-scale liquid-solid fluidized beds were validated for a wide variety of regimes. Discussions on the choice of drag correlation, the robustness of the method with different mesh topologies, and the importance of the Saffman lift force for the accurate simulation of the flow structures are provided. The validated simulations were used to assess the particles' mixing by applying the nearest-neighbors method (NNM) and the mixing index introduced by Doucet. The slowest mixing component for most of the simulations was the axial. The mixing time as a function of the inlet flow rate demonstrates a plateauing behavior for looser beds (fluid fractions above 50%). Variations in the interaction properties of particles do not seem to play an important role in the quantitative mixing. Yet, for a bed of particles with very low Stokes numbers or at very high concentrated beds, increasing the sliding friction coefficient presented a negative impact on the mixing performance.

Keywords: Liquid Fluidization, Liquid-Solid Fluidized Bed, Multiphase Flows, FEM, CFD-DEM.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

Without water, there is no life. Freshwater is a limited resource. Those are well-known facts, however, humanity has systematically neglected this. With the increase in human population, the demand for more sustainable water usage is urgent, and this implies not only the creation of new technologies but also the improvement of the known ones. For either solution, the deeper the knowledge about how things work, the greater the chances of innovating, and this is what this work is about.

Water is the most manipulated material in the worldwide industry (RICHARD *et al.*, 2005). The understanding of how to redirect its flow allowed humanity to come up with solutions to sensible aspects of life. In the book entitled "The Silk Roads", Frankopan (2017) argues that the success of the Mesopotamian civilizations can be credited in great part to the proximity to the Tigris and the Euphrates rivers. According to the author, the Persian Empire (the greatest of the ancient world) created a pioneering irrigation system covering lands from the Mediterranean to the center of Asia, allowing for the growth of cities with increasingly large populations.

From 4000 B.C. until now, the success or perishing of populations is undoubtedly related to dominating liquid dynamics. In special, sanitation depends on it. According to Angelakis and Snyder (2015), water has been used for human waste disposal since the Bronze Age. Yet, Lin *et al.* (2022) report that approximately 80% of sewage generated by human activities is discharged into rivers and oceans without any treatment. According to the UNESCO 2021 World Water Development Report, almost a million human deaths per year result from diarrhea caused by unsafe drinking water, lack of sanitation, and bad hand hygiene habits (LIN *et al.*, 2022). As Lin *et al.* (2022) show, the alarming reality is that the human population is still growing and the investment in sanitation does not follow the pace.

A similar relationship can be established between the human population and the increasing demands for energy resources. Approximately 87% of the world's total energy generation comes from burning oil, coal, and natural gas (MBUNGU *et al.*, 2020). The extensive use of these highly pollutant fuels is pointed out as the main cause of global warming. Additionally, these fossil fuels are non-renewable energy resources and the estimations are that the available sources will not last until the end of the next century (MBUNGU *et al.*, 2020). Hence, the demand for clean and renewable energy resources is urgent.

Besides the challenges (and oftentimes even the lack of incentive), the scientific community is actively trying to bring solutions to these problems as demonstrated in Figure 1.1. The number of publications on wastewater treatment and renewable clean energy resources is increasing yearly, and the reflexes of these efforts are sensible. However, much more is yet to be done in terms of process intensification and cost reduction.

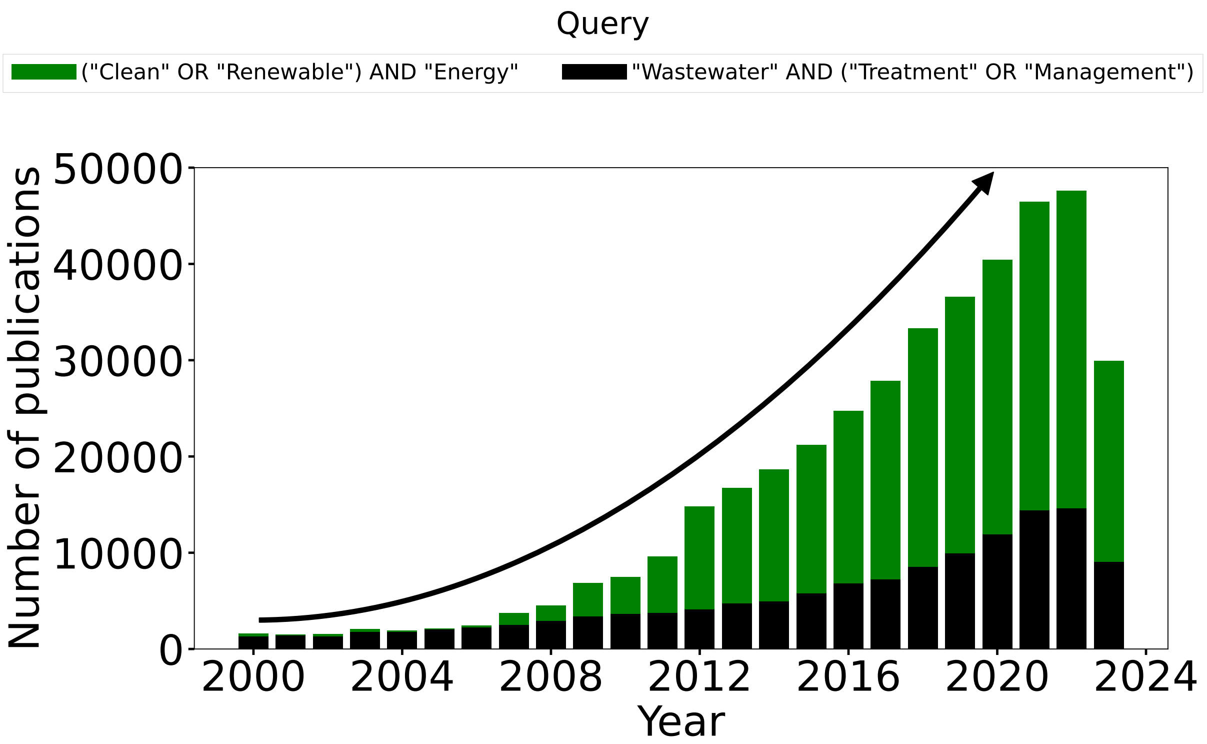


Figure 1.1 – Number of publications in wastewater treatment and clean energy resources per year. Source: www.webofknowledge.com, accessed on September 28th, 2023.

In this context, one technology arises as an alternative to fulfill both demands. Wastewater treatment using anaerobic digestion consists of exposing effluents rich in fats, oils, and greases (nutrients) to the action of anaerobic bacteria. The microorganisms feed themselves with this organic matter, breaking the long molecular chains and making it more soluble and easily degradable (NADAIS *et al.*, 2011; JAMALI *et al.*, 2017). As a side effect, the bacteria also produce biogases such as methane, that can be used as fuel.

It is a cheap, environmentally friendly, self-sustainable technology capable of remedying both demands: wastewater management and energy.

Anaerobic Digestion can be done in batches, but this process can be sensibly intensified by using liquid-solid fluidized beds (LSFBs). These equipment operate at inlet fluid velocities way higher than fixed beds. Under this condition, particles are agitated and the higher momentum exchange promotes the control of the biofilm thickness, enhancing the contact between the microorganism and the fresh nutrient. Hence, higher conversion rates and productivity are expected (SARAVANAN; SREEKRISHNAN, 2006; ALDACO *et al.*, 2007).

Although the high potential of this technology, the insufficient interest in the field restricts its use. This can be partially attributed to the lack of knowledge about the fluid and granular dynamics in liquid-solid fluidization systems. Nonetheless, with the increase of computational power in recent years, complex differential equations describing the fluid and granular dynamics can now be solved using robust numerical-computational methods. Among those, the unresolved coupling between the Computational Fluid Dynamics and the Discrete Elements method (unresolved CFD-DEM) stands out as a precise, yet computationally efficient, method to describe fluid-particle systems with high time and space resolution (ZHOU *et al.*, 2010; NOROUZI *et al.*, 2016; BéRARD *et al.*, 2020).

This work aims to fill the gap in the knowledge about liquid fluidization dynamics using experimental techniques and the unresolved CFD-DEM method. More than a hope, the author of this thesis is motivated by a deep belief that this scientific contribution brings us a step towards the development, implementation, and popularization of clean technologies such as anaerobic digestion-liquid-solid fluidized bed bioreactors. This author also believes (as ancient Persians did[1]) that the constant sharing and pursuing of knowledge will help humanity overcome the harming challenges of this century. More than an introduction, this chapter is an invitation to share. From this author's heart, this is an invitation to believe.

[1] Citing Herodotus, Frankopan (2017) mentions that this succeeded civilization had a well-known peculiar affection for different cultures and research.

CHAPTER 2

BIBLIOGRAPHIC REVIEW

The literature review introduces several ideas applied throughout the work. It starts from more general concepts related to fluid and particle dynamics and gradually tapers to more specific topics related to the present research. The sections are divided as follows:

Section 2.1 discusses fundamental topics on fluid and granular dynamics. Concepts such as Eulerian and Lagrangian frames, hypotheses underlying the modeling of particles and fluid dynamics, turbulence, and interphase momentum exchange are included in this section. In special, this section describes the foundations of the discrete elements method (DEM) formulations, the Volume-Average Navier-Stokes equations, and the interphase forces that are going to be applied in the unresolved computational fluid dynamics-discrete elements method coupling (CFD-DEM). The idea is to present the foundations of the modeling further applied to predict the behavior of particles and fluid in a liquid-solid fluidized bed.

Section 2.2 is about how the concepts in Section 2.1 can be applied to simulate the behavior of fluid-solid systems. Integration strategies on the DEM, discretization strategies in the CFD method, algorithms and solvers, and other practical aspects of solving the differential equations used to model the fluid-solid systems are provided in this section. Special attention is given to the finite elements method (FEM) and the integration of the DEM equations.

Section 2.3 is about liquid-solid fluidized beds. The section includes definitions, structural aspects, fluid-dynamics design, correlations used in its description, and applications are mentioned in this section. Trends in liquid-solid fluidized beds are also reviewed.

2.1 Fundamentals and modeling of liquid, granular, and solid-liquid flows

Flows can be modeled as a continuum or a collection of discrete entities. Those approaches are often called Eulerian and Lagrangian, respectively. In the case of gases, the number of Knudsen is often used as a criterion to decide which of the approaches is the most adequate (HOEF *et al.*, 2006):

$$K_n = \lambda / L \tag{2.1}$$

The dimensionless number is a ratio between the molecular free path length λ and the characteristic length scale of the flow L. Figure 2.1 illustrates the level of modeling per K_n.

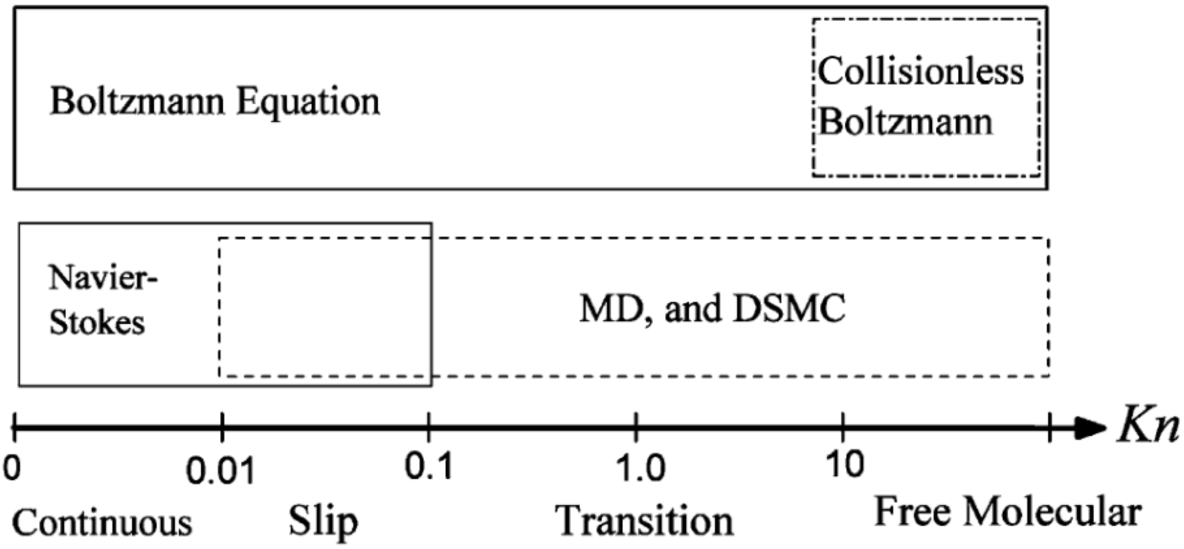


Figure 2.1 – Level of flow modeling according to K_n. Reproduced from Hoef *et al.* (2006).

Gas flows with $K_n > 10$ are almost free of molecular interaction and behave as ideal gases. Flows with $K_n > 0.1$ violate the continuum hypothesis and cannot be modeled without taking molecular dynamics into account (KARNIADAKIS *et al.*, 2005). For $K_n < 0.1$, the continuum hypothesis is valid.

Liquid and granular flows present particular characteristics that do not resemble the gas described by the classic kinetic theory of molecular gases. However, the theory constitutes the foundation of the modern understanding of momentum, heat, and mass transport. Equations such as Navier-Stokes, Fourier, and Fick laws can be derived from the Boltzmann Transport Equation. For brevity, we have chosen to start this work from one step further in history[1].

[1] We refer the reader to Kremer (2010) for a detailed description of the kinetic theory and a thorough derivation of the transport equations.

2.1.1 Granular dynamics

Granular materials are the second-most manipulated in industry (RICHARD *et al.*, 2005). They are essentially a conglomerate of individual macroscopic solid entities. Usually, a conjunct of those entities is considered granular if their characteristic length (diameter, for example) is above 100 µm (GENNES, 1998; DURAN, 2012). The size criterion is not chosen arbitrarily. Powders (< 100 µm), for example, are way more susceptible to van der Waals, capillary, and electrostatic interactions (RODUNER, 2006; TRAINA *et al.*, 2013). Granular matter with characteristic size above 3-5 mm are usually referred to as *broken solids*. Their dynamics are better approached by the Lagrangian frame due to their size. The main difference between broken and smaller granular solids are their lower mechanical properties (Young's module, restitution coefficient, sliding friction coefficient) and usual morphology (lower sphericity) (COULSON, 2019).

Granular materials can be modeled using either the Eulerian or Lagrangian frames of reference.

2.1.1.1 Eulerian approach

As a general rule, the Eulerian approach can be adopted when the behavior of the individual entities composing the flow is insignificant to the studied phenomenon and/or the system is so large compared to particles that the system behaves like a fluid. Eulerian granular flows can be represented by compressible equations, with quantities such as granular pressure and granular temperature calculated using the kinetic theory of granular flows (KTGF) (HOEF *et al.*, 2006).

The KTGF borrows concepts from the kinetic theory of molecular gases to simulate the dynamics of the bulk. The general concept is that kinetic energy at the particles' scale is responsible for macro-scale properties such as pressure, temperature, viscosity, thermal and electrical conductivity, compressibility, and others (HOEF *et al.*, 2006). Additionally, particles' velocities are described by a Maxwellian distribution function (KREMER, 2010).

The main assumptions of the kinetic theory of (molecular) gases are (CAMPBELL, 1990; RAJCHENBACH *et al.*, 1994; HOEF *et al.*, 2006; KREMER, 2010):

1. The fluid consists of particles of negligible volume compared to their container. Consequently, λ is very high and the probability of collisions between more than two particles is negligible.

2. Molecules resemble perfect hard spheres during collisions, that is, collisions do not dissipate energy (elastic).

3. External forces are negligible compared to interaction forces.

4. The amount of molecules is immeasurably high, such that the thermodynamics limit and the use of statistical mechanics is justified. Additionally, the distribution of molecules in space is uniform.

The difference in scale between granular materials and molecular gases explains the lack of resemblance between granular and molecular dynamics. Different from gases, granular material dynamics are dominated essentially by gravity and interactions with fluids (interphase momentum exchange), and other particles or surfaces (collisions). Collisions of granular materials are inelastic, consequently, granular temperature decreases rapidly to zero and a state of equilibrium (in a thermodynamics sense) is never reached. In other words, in contrast to gases, the granular temperature can only be sustained by a constant external influence (CAMPBELL, 1990).

For this reason, the main challenge in the Eulerian approach is to determine a valid expression for granular rheology and hydrodynamics incorporating interaction characteristics such as particles' roughness. The approach by Gidaspow *et al.* (1991) is among the most applied in the literature and is the default option in Fluent, a popular commercial software by Ansys®(ANSYS, 2013).

2.1.1.2 Lagrangian approach

The Lagrangian frame can be used to represent particles as individual (discrete) bodies, allowing for a description of granular dynamics at the particles' level. The family of discrete (Lagrangian) methods for granular flows includes cellular-automata methods (BAXTER; BEHRINGER, 1990; MARINACK; HIGGS, 2011), Monte-Carlo simulation (HOPKINS; SHEN, 1988; BREY; RUIZ-MONTERO, 1999), and the discrete elements method (DEM) (TSUJI *et al.*, 1992; ZHU *et al.*, 2007; BLAIS *et al.*, 2019; GOLSHAN *et al.*, 2022). Notably, the DEM is becoming very popular given its capability to accurately simulate the behavior of a single particle as well as several hundreds of thousands of particles with the same degree of precision.

In the DEM, particles are discrete bodies owning singular properties, including size, position, and velocity (angular and translational). The movement of particles is represented by Newton's and Euler's Second Law of Motion, which is integrated in time to obtain their velocities and positions at each interval[2].

$$m_i \frac{\mathrm{d}\boldsymbol{x}_i}{\mathrm{d}t} = \boldsymbol{v}_i \tag{2.2a}$$

$$m_i \frac{\mathrm{d}\boldsymbol{v}_i}{\mathrm{d}t} = \sum_{j=1, j\neq i}^{N_p} (\boldsymbol{f}_{c,ij}) + \sum_{w=1}^{N_w} (\boldsymbol{f}_{c,iw}) + \sum_{k,k\neq i}^{N_p} (\boldsymbol{f}_{cn,ik}) + \boldsymbol{f}_{g,i} \tag{2.2b}$$

$$I_i \frac{\mathrm{d}\boldsymbol{\omega}_i}{\mathrm{d}t} = \sum_{j,j\neq i}^{N_p} (\boldsymbol{M}_{t,ij} + \boldsymbol{M}_{r,ij}) + \sum_{w=1}^{N_w} (\boldsymbol{M}_{t,iw} + \boldsymbol{M}_{r,iw}) + \boldsymbol{M}_{Ext}, \tag{2.2c}$$

where the subscripts i refer to the i^{th} particle interacting with the j^{th} and the k^{th} particle or the w^{th} wall, while $\boldsymbol{x}$ is the position, m is the mass, N_p is the number of particles, N_w is the number of walls, $\boldsymbol{v}$ and $\boldsymbol{\omega}$ represent translational and angular velocities, $\boldsymbol{f}_c$ the contact forces, $\boldsymbol{f}_{cn}$ the non-contact forces, and $\boldsymbol{f}_g$ the gravity force, while $\boldsymbol{M}_t$ and $\boldsymbol{M}_r$ are the tangential and rolling friction torques, and lastly $\boldsymbol{M}_{Ext,}$ are external torques. I is the moment of inertia, calculated for a sphere with radius R by:

$$I = \frac{2}{5} m R^2 \tag{2.3}$$

Non-contact forces were not accounted for in this work due to their typical negligible magnitude compared to contact forces in granular flows (and mainly interphase forces in multiphase flows). Those forces include electrostatic, van der Walls, liquid-bridge, and lubrication forces(ZHU *et al.*, 2007)[3]. External torques, such as viscous dissipative torque, were also neglected.

Colliding particles can be approached as hard and soft bodies. Those approaches are also called hard-sphere/impulse-based/event-driven and soft-sphere/force-based/time-driven, respectively. The hard-sphere approach was first introduced by Campbell and Brennen (1985) to simulate granular systems. The most important underlying hypothesis in this approach is that the duration of particles' collisions is so insignificant that one can approach it to zero. Additionally, properties are evaluated only when a collision happens. Therefore, this approach is suitable for very dilute systems of particles with a high coefficient of restitution, and it is more computationally efficient in most cases (NOROUZI *et al.*, 2016).

On the other hand, the soft-sphere approach, first introduced by Cundall and Strack (1979), computes collision forces based on the virtual overlap between bodies. As the overlap changes with time, the force needs to be calculated at each time step and, naturally, the method requires a way smaller time step compared to the hard-sphere,

[2] The adopted notation presents scalars as non-bold letters and symbols represent scalars, and bold letters represent vectors. For coherence, the same notation is adopted in the entire work.

[3] Probably, the highest magnitude non-contact force occurring in the studied problem is the lubrication force. However, the research produced in the group by Campos *et al.* (2022) has demonstrated little influence of the lubrication force on the liquid-solid fluidized bed.

augmenting the computational cost of simulations. Nevertheless, it accurately reproduces actual collisions, even if they happen simultaneously.

In this work, we applied the DEM to simulate the motion of particles using the soft-sphere formulation detailed by Tsuji *et al.* (1992), Zhu *et al.* (2007), Norouzi *et al.* (2016), Blais *et al.* (2019), Golshan *et al.* (2022)[4]. A schematic illustration of the method is presented in Figure 2.2.

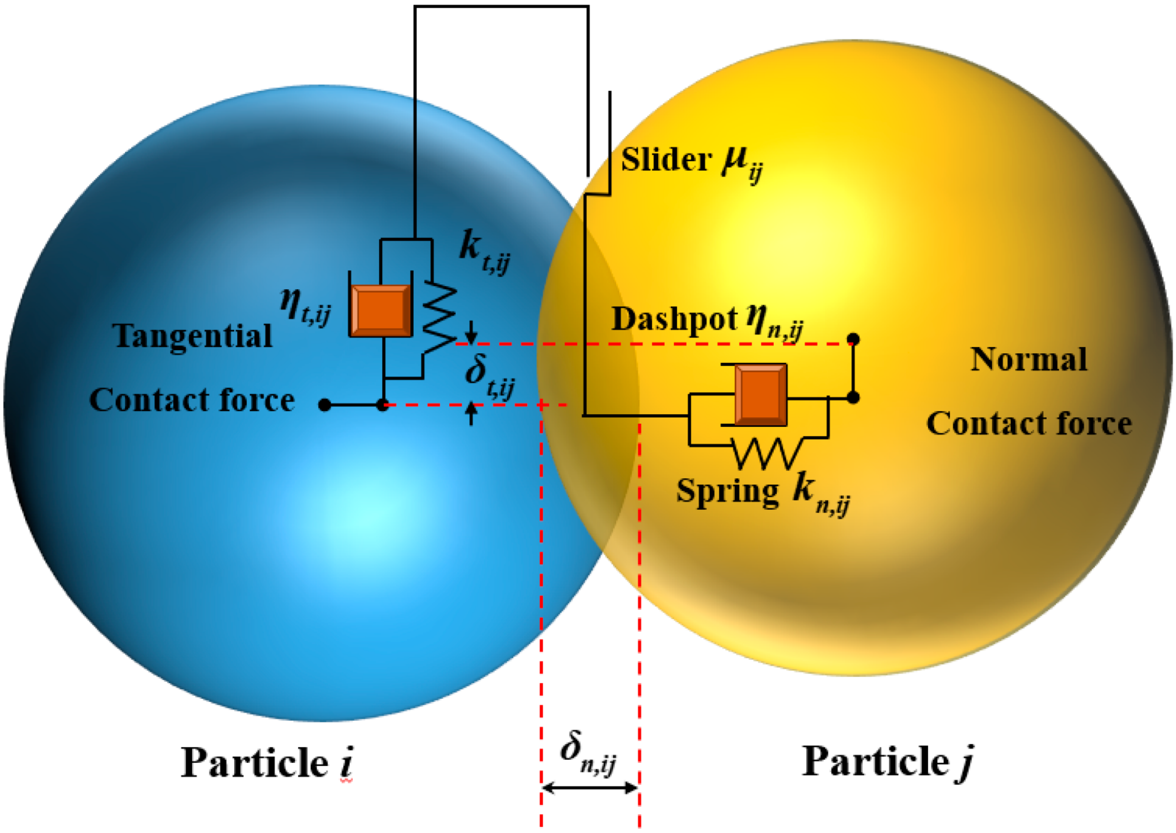


Figure 2.2 – Schematic illustration of a collision in the soft-sphere model. Reproduced from Li *et al.* (2022).

First, the contact forces are decomposed into normal and tangential components:

$$\boldsymbol{f}_{c,ij} = \boldsymbol{f}_{cn,ij} + \boldsymbol{f}_{ct,ij} \tag{2.4}$$

As shown in Figure 2.2, the soft-sphere approach consists of representing the colliding particles i and j as springs and dashpots associated in parallel. The deformation of the springs is given by the normal and tangential overlaps between particles ($\delta_{n,ij}$ and $\delta_{t,ij}$, respectively). The model proposed by Cundall and Strack (1979), called the spring-dashpot model or viscoelastic model, represents the normal and the tangential

[4] We approximate particles from spheres, but the DEM has already been extended to simulate the dynamics of non-spherical particles by several authors, e.g., Matsushima *et al.* (2009), Garcia *et al.* (2009), Ferellec and McDOWELL (2010), and Gao *et al.* (2012).

components of the contact forces by (TSUJI *et al.*, 1992; ZHU *et al.*, 2007):

$$\boldsymbol{f}_{cn,ij} = -k_{n,ij}\delta_{n,ij}\boldsymbol{n}_{ij} - \gamma_{n,ij}\boldsymbol{v}_{n,ij} \tag{2.5a}$$

$$\boldsymbol{f}_{ct,ij} = -k_{t,ij}\boldsymbol{v}_{t,ij} + \gamma_{t,ij}\boldsymbol{v}_{t,ij} \tag{2.5b}$$

where $\boldsymbol{n}_{ij}$ is the contact normal unit vector, calculated for particles i and j at positions $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ by:

$$\boldsymbol{n}_{ij} = \frac{\boldsymbol{x}_j - \boldsymbol{x}_i}{|\boldsymbol{x}_j - \boldsymbol{x}_i|} \tag{2.6}$$

The normal and tangential overlaps are calculated by (GOLSHAN *et al.*, 2022):

$$\delta_{n,ij} = R_i + R_j - |\boldsymbol{x}_j - \boldsymbol{x}_i| \tag{2.7a}$$

$$\delta_{t,ij}^{t} = \delta_{t,ij}^{t-dt} + \boldsymbol{v}_{t,ij}dt \tag{2.7b}$$

where dt is the time interval (time step) between the present time t and the previous $t - dt$.

The normal and tangential relative velocities can be defined as (KLOSS; GONIVA, 2011; LIGGGHTS, 2016; GOLSHAN *et al.*, 2022; LETHE, 2023):

$$\boldsymbol{v}_{n,ij} = (\boldsymbol{v}_{rel,ij} \cdot \boldsymbol{n}_{ij})\, \boldsymbol{n}_{ij} \tag{2.8a}$$

$$\boldsymbol{v}_{t,ij} = \boldsymbol{v}_{rel,ij} - \boldsymbol{v}_{n,ij} \tag{2.8b}$$

where the relative velocity $\boldsymbol{v}_{rel,ij}$ is given by:

$$\boldsymbol{v}_{rel,ij} = (\boldsymbol{v}_j - \boldsymbol{v}_i) + (R_i\boldsymbol{\omega}_i + R_j\boldsymbol{\omega}_j) \times \boldsymbol{n}_{ij} \tag{2.9}$$

In Eqs. (2.5), the terms k and γ represent the spring and the damping constants, respectively. These terms condense the mechanical properties of the particles and the interaction properties of the pair, being respectively responsible for representing the energy restitution and dissipation. Cundall and Strack (1979) first proposed that the particle collision should resemble linear springs and dashpots. Nonetheless, better accuracy was found by other authors (KUWABARA; KONO, 1987; TSUJI *et al.*, 1992; ZHOU *et al.*, 1999) for the non-linear viscoelastic collision model proposed by Mindlin (1949) and Mindlin and Deresiewicz (1953) based on the Hertzian theory.

The equations of the non-linear viscoelastic model used in this work to represent the spring and damping constants are presented in Table 2.1 (KLOSS; GONIVA, 2011; GOLSHAN *et al.*, 2022).

The tangential component $\boldsymbol{f}_{ct,ij}$ is limited by the friction coefficient μ_t via Coulomb's criterion (CUNDALL; STRACK, 1979; GOLSHAN *et al.*, 2022). If the condition $\boldsymbol{f}_{ct,ij} \leq \mu_f \boldsymbol{f}_{cn,ij}$ is violated, the tangential overlap is calculated with no damping:

$$\delta_{t,ij} = -\frac{\hat{\boldsymbol{f}}_{ct,ij} + \gamma_{t,ij}\boldsymbol{v}_{t,ij}}{k_{t,ij}} \tag{2.10}$$

Table 2.1 – DEM equations

Property	**Equation**
Mass of particle a i	m_i
Radius of particle a i	R_i
Poisson ratio of a particle i	ν_i
Young Modulus of a particle i	Y_i
Coefficient of restitution of a pair ij	e
Distance between a particle i and contact point	$\boldsymbol{r}_i$
Equivalent mass	$\frac{1}{m^*_{ij}} = \frac{1}{m_i} + \frac{1}{m_j}$
Equivalent radius	$\frac{1}{R^*_{ij}} = \frac{1}{R_i} + \frac{1}{R_j}$
Equivalent Young's modulus	$\frac{1}{Y^*_{ij}} = \frac{1-\nu_i^2}{Y_i} + \frac{1-\nu_j^2}{Y_j}$
Equivalent shear modulus	$\frac{1}{G^*_{ij}} = \frac{2(2-\nu_i)(1+\nu_i)}{Y_i} + \frac{2(2-\nu_j)(1+\nu_j)}{Y_j}$
Normal stiffness	$k_{n,ij} = \frac{4}{3} Y^*_{ij} \sqrt{R^*_{ij} \delta_{n,ij}}$
Tangential stiffness	$k_{t,ij} = 8 G^*_{ij} \sqrt{R^*_{ij} \delta_{n,ij}}$
Normal damping	$\gamma_{n,ij} = -2\sqrt{\frac{5}{6}} \frac{\ln(e)}{\sqrt{\ln^2(e+\pi^2)}} \sqrt{\frac{2}{3} k_{n,ij} m^*_{ij}}$
Tangential damping	$\gamma_{t,ij} = -2\sqrt{\frac{5}{6}} \frac{ln(e)}{\sqrt{\ln^2(e+\pi^2)}} \sqrt{k_{t,ij} m^*_{ij}}$

and $\hat{\boldsymbol{f}}_{ct,ij}$ is the limited tangential force, given by:

$$\hat{\boldsymbol{f}}_{ct,ij} = \mu_t \left|\boldsymbol{f}_{cn,ij}\right| \frac{\boldsymbol{f}_{ct,ij}}{\left|\boldsymbol{f}_{ct,ij}\right|} \tag{2.11}$$

The rolling friction torque can be modeled as a constant torque or a viscous torque. The viscous torque accounts for $\boldsymbol{v}_\omega$ while the constant torque does not. We applied the viscous rolling friction torque, calculated by (GOLSHAN *et al.*, 2022):

$$\boldsymbol{M}_{r,ij} = \mu_r R^*_{ij} \left|\boldsymbol{f}_{cn,ij}\right| \left|\boldsymbol{v}_\omega\right| \bar{\omega}_{ij} \tag{2.12}$$

where μ_r is the rolling friction coefficient of the pair, and $\bar{\omega}_{ij}$ and $\boldsymbol{v}_\omega$ are respectively calculated by:

$$\bar{\omega}_{ij} = \frac{\boldsymbol{\omega}_i - \boldsymbol{\omega}_j}{\left|\boldsymbol{\omega}_i - \boldsymbol{\omega}_j\right|} \tag{2.13}$$

$$\boldsymbol{v}_\omega = (\boldsymbol{\omega}_i \times R_i \boldsymbol{n}_{ij} - \boldsymbol{\omega}_j \times R_j \boldsymbol{n}_{ij}) \tag{2.14}$$

Lastly, the tangential torque is given by:

$$\boldsymbol{M}_{t,ij} = R_i \boldsymbol{n}_{ij} \times \boldsymbol{f}_{c,ij} \tag{2.15}$$

2.1.1.3 Particles and collision pair properties

Combining granular mechanical properties such that they are representative of the system is among the main issues with granular dynamics modeling. The DEM soft-sphere model approaches particles from a spring-dashpot system, which provides a very good approximation of actual collisions. As a drawback, several sensitive-case empirical parameters need to be determined, namely:

Young's modulus: Solid matter submitted to compression or traction will reorganize its atomic (in case of ceramic or metallic materials) or molecular (in case of polymers) structure to compensate for the stress. This rearrangement is called deformation. Positive axial deformation will result in a negative orthogonal compensation and vice-versa (a phenomenon called Poisson effect). A bulk material submitted to a slowly increasing tensile effort will suffer an elastic (or reversible) deformation up to a point of non-return, where a plastic (or irreversible) deformation regime takes place. When the material can no longer resist the stress, its structure is completely split; in other words, the material is broken. The integration of the tensile versus deformation curve gives the amount of energy absorbed by the material prior to failing (CALLISTER; RETHWISCH, 2020).

The elastic region in a stress versus deformation graphic is linear. The reason for this is that the microstructure of the material is intact and the orthogonal deformation fully compensates for the axial deformation. The angular coefficient of the linear relation between stress and strain is called Young's modulus. Ceramic materials tend to have steep stress vs. strain elastic behavior and a very small energy capacity in the plastic region. Polymers tend to behave oppositely. Metals are in the middle between ceramics and polymers. As such, it is a material's property (and not a pairwise property, such as the coefficient of restitution) (CALLISTER; RETHWISCH, 2020).

Collisions between particles in granular flows tend to occur in the elastic region. Consequently, the spring model is a good approximation for fluidized beds, mixing tanks, fluid-particle reactors, and other contact-dominated particulate systems (NOROUZI *et al.*, 2016).

Poisson's ratio: It is the ratio between the axial and orthogonal deformation suffered by a material under a tensile effort. It is a dimensionless number representing the Poisson effect, calculated for a body i by:

$$\nu_i = \frac{\mathrm{d}L_{\perp}}{\mathrm{d}L_{\parallel}} \tag{2.16}$$

where $L_{\perp}$ and $L_{\parallel}$ are the transversal (orthogonal) and the axial deformations, respectively. The Poisson ratio usually lies between 0.0 and 0.5 (CALLISTER; RETHWISCH, 2020).

Coefficient of restitution: Dimensionless number representing the amount of the kinetic energy that remains in the kinetic form after a collision, calculated by:

$$e = \frac{|E_{k,a} - E_{k,b}|}{\left|E^0_{k,a} - E^0_{k,b}\right|} \equiv \frac{|\boldsymbol{v}_a - \boldsymbol{v}_b|}{|\boldsymbol{v}^0_a - \boldsymbol{v}^0_b|} \tag{2.17}$$

where a and b stand for two colliding bodies with kinetic energy $E_{k,a}$ and $E_{k,b}$, and velocities $\boldsymbol{v}_a$ and $\boldsymbol{v}_b$, respectively. The superscript 0 denotes "instantaneously before the collision" and the symbols without it represent values instantaneously after the event. As such, this is a pairwise empirical parameter that depends on the shape, the material, the relative velocity, and several other factors related to both bodies (MELO *et al.*, 2021a; MELO *et al.*, 2021b). Usually, the actual value applied to the soft-sphere model is the effective (or "wet") coefficient of restitution (e_{eff}), which is measured under the influence of a surrounding fluid[5].

Coefficient of (sliding) friction: The ratio between the normal and the frictional force. It is also a pairwise property that will depend on several factors such as the shape and the roughness of the interacting materials.

Coefficient of rolling friction: The ratio between the rolling friction force and the gravity resultant force on an object. It is also a pairwise property dependent on the roughness of the interacting materials.

2.1.2 Liquid dynamics

Liquids are a condensed state of matter (KOHN, 1999; ZANOTTO; MAURO, 2017), meaning that their molecules are strongly attached. Consequently, λ is very low and the continuum hypothesis is reasonably adopted. Additionally, the high interaction forces between molecules due to their attachment result in negligible compressibility. As a result, liquid flows can be modeled using Navier-Stokes equations under the incompressible hypothesis with a very high degree of certainty.

2.1.2.1 Navier-Stokes equations

Navier-Stokes equations represent the mass and momentum continuity laws from an Eulerian reference. The two equations originate from the continuity equation (mass conservation) and the Cauchy equation (momentum conservation) (BIRD *et al.*, 2007)[6]:

$$\frac{\partial \rho_f}{\partial t} + \nabla \cdot (\rho_f \boldsymbol{u}) = 0 \tag{2.18a}$$

$$\frac{\partial \rho_f \boldsymbol{u}}{\partial t} + \nabla \cdot (\rho_f \boldsymbol{u} \otimes \boldsymbol{u}) = \nabla \cdot \boldsymbol{\sigma} + \rho_f \boldsymbol{g} \tag{2.18b}$$

where ρ_f is the density of the fluid, $\boldsymbol{u}$ is the fluid velocity vector, $\boldsymbol{g}$ is the gravity acceleration vector, and $\boldsymbol{\sigma}$ is Cauchy's stress tensor, given by:

$$\boldsymbol{\sigma} = -p\boldsymbol{\mathcal{I}} + \boldsymbol{\tau} \tag{2.19}$$

[5] It is usually assumed that the effective coefficient of restitution represents the energy dissipation due to the collisional and non-contact forces.

[6] The adopted notation presents scalars as non-bold letters and symbols represent scalars, bold alphabet letters represent vectors and bold Greek symbols represent tensors. For coherence, the same notation is adopted in the entire work.

where p stands for pressure, $\boldsymbol{\mathcal{I}}$ for identity matrix, and $\boldsymbol{\tau}$ for the deviatoric stress tensor.

The complete deviatoric stress tensor is written as:

$$\boldsymbol{\tau} = \mu \left(\nabla \boldsymbol{u} + (\nabla \boldsymbol{u})^T \right) + \left(\frac{2}{3}\mu - \kappa \right) (\nabla \cdot \boldsymbol{u}) \boldsymbol{\mathcal{I}} \tag{2.20}$$

where κ is the dilatational viscosity and μ is the dynamic viscosity. Incompressible flows present constant density in space and time, thus, from the continuity Eq. (2.18a), the divergence of the velocity vector $\nabla \cdot \boldsymbol{u}$ is 0. Consequently, $\boldsymbol{\tau}$ is simplified to:

$$\boldsymbol{\tau} = \mu \left(\nabla \boldsymbol{u} + (\nabla \boldsymbol{u})^T \right) \tag{2.21}$$

Applying Eq. (2.21) on Eq. (2.19) and then on Eq. (2.18b), and assuming that the flow is incompressible and the fluid is Newtonian, we obtain the Navier-Stokes equations:

$$\nabla \cdot \boldsymbol{u} = 0 \tag{2.22a}$$

$$\rho_f \frac{\partial \boldsymbol{u}}{\partial t} + \rho_f \nabla \cdot (\boldsymbol{u} \otimes \boldsymbol{u}) = -\nabla p + \mu \nabla^2 \boldsymbol{u} + \rho_f \boldsymbol{g} \tag{2.22b}$$

The terms of the momentum equation represent transience, advection, driving force, diffusion, and gravity (source term); respectively. While the velocity field is determined by the momentum equation, the set of equations does not include an independent equation for pressure. For incompressible flows, this equation is derived from continuity. Consequently, the pressure becomes the mass conservation constraint.

Pressure is determined by the Poisson equation obtaining velocity from Navier-Stokes. The Poisson equation is obtained by applying the divergence on both equations and combining the results (FERZIGER *et al.*, 2020):

$$\nabla^2 p = \rho_f \nabla \cdot (\boldsymbol{g} - \boldsymbol{u} \cdot \nabla \boldsymbol{u}) \tag{2.23}$$

Apart from very specific problems underlined by strong assumptions, Navier-Stokes equations do not have a global analytical solution[7]. For this reason, numerical methods are often implemented on computers to solve more complex fluid dynamics problems. This family of methods together form the field of Computational Fluid Dynamics (CFD), briefly introduced in Section 2.2.2. Further details about the Finite Elements Method (FEM), applied in this work, are presented in Section 2.2.3.

2.1.2.2 Turbulence

Turbulence is found in the majority of flows (DAVIDSON, 2015). It originates from a disturbance in the flow due to a wide variety of sources, including interactions with

[7] In fact, the Clay Mathematics Institute offers a USD 1,000,000.00 prize to the person capable of solving or at least proving the existence and smoothness of an analytical solution for the Navier-Stokes equations. The challenge, proposed by Fefferman (2000), is the sixth among the seven Millennium Prize Problems. The prize was not claimed until this day.

solids and other fluids, heat and mass transport, momentum advection, and so on. As a consequence, the flow is very unpredictable. Those features can be observed by looking at fluid velocity field $\boldsymbol{u}(\boldsymbol{x}, t)$ (position vector $\boldsymbol{x} \in \mathbb{R}^3$). If one isolates an infinitesimal element in a space percolated by a turbulent flow, $\boldsymbol{u}(\boldsymbol{x}, t)$ in this space element will be far from periodic for a wide range of time intervals and varying direction and magnitude at each instant (POPE, 2015).

Osborne Reynolds (ROTT, 1990) is often referred to as the starter of the field of turbulence. His experimental demonstration of turbulence (REYNOLDS, 1883) has led to the development of his famous dimensionless number (REYNOLDS, 1895 apud POPE, 2015):

$$\mathrm{Re} = \frac{\mathcal{U}\mathcal{L}}{\nu} \tag{2.24}$$

where ν is the kinematic viscosity of the fluid, $\mathcal{U}$ is the characteristic velocity and $\mathcal{L}$ the characteristic length of the flow. The dimensionless number expresses the ratio between inertial and diffusive effects on the fluid. Highly inertial flows tend to behave more chaotically, while diffusion-driven flows are laminar. The limits of Re where the flow regime changes from laminar to turbulent depends on the system. For pipes, flows with $\mathrm{Re} > 3800 - 4000$ are considered turbulent while the limit of laminar flows lies around $\mathrm{Re} < 2200 - 2300$. The region between the laminar upper limit and the turbulent lower limit lies the transition, which resembles a laminar flow with perturbations increasing according to Re. Figure 2.3 represents these three flow patterns.

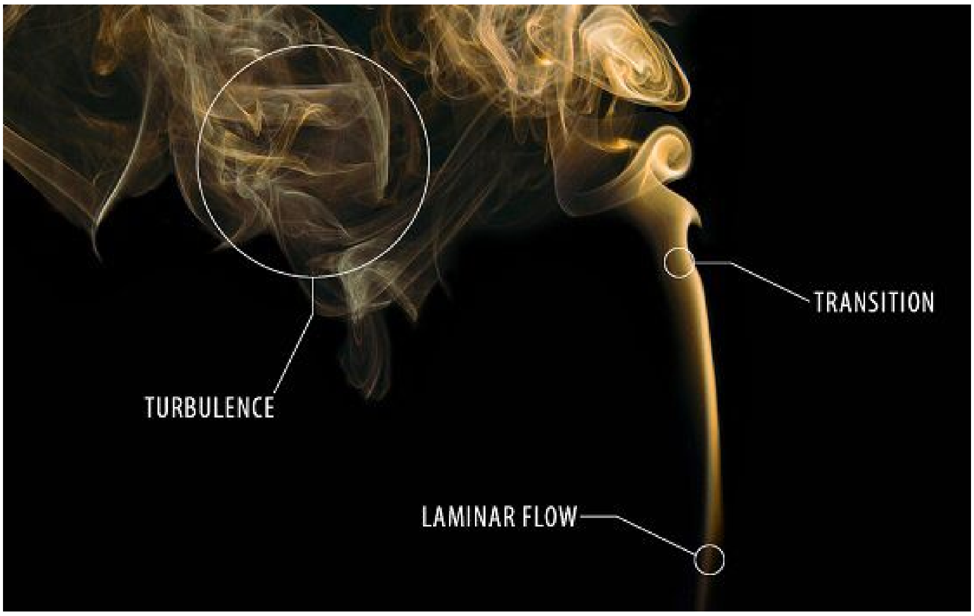


Figure 2.3 – Representation of turbulence. Reproduced from Overmeen (2021).

Turbulence is characterized by vortices and eddies. Those are circular structures in the fluid field that chaotically propagate momentum. The agitation provoked by these structures allows for better mixing and increases convective effects, enhancing heat and mass transfer when compared to laminar flows. Turbulence is also responsible for a higher interaction between fluids and solids. Vortices also provoke a “domino effect” on the flow, creating other vortices along their pathway (GEORGE, 2013).

The interscale cascade (Kolmogorov cascade) effect of turbulence is among the most complex fluid dynamics phenomena. Large turbulent structures create turbulence at smaller scales via inviscid dissipation processes (POPE, 2015). This process is repeated up to the smallest scale (Komolgorov scale, $\eta = (\nu^3/\epsilon)^{1/4}$, where ϵ is the rate of dissipation of turbulent kinetic energy) where all kinetic energy is dissipated by viscous stress (RICHARDSON; LYNCH, 2007; POPE, 2015). An illustration of this process is shown in Figure 2.4 (RODRIGUEZ, 2019).

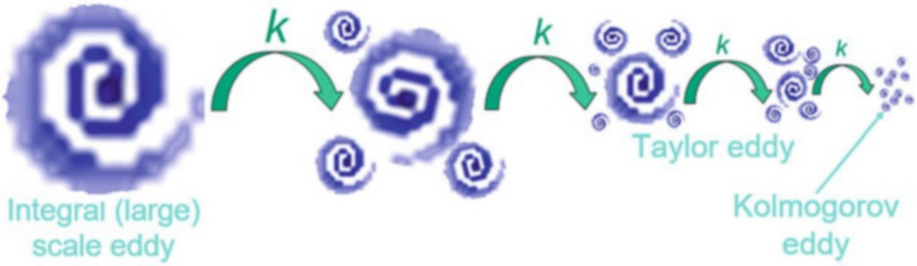


Figure 2.4 – Komolgorov cascade. Reproduced from Rodriguez (2019).

Ideally, Navier-Stokes equations should be resolved up to the Komolgorov scale to guarantee all turbulent effects are taken into account. In CFD, the name of this approach is Direct Numerical Simulation (DNS) (ORSZAG, 1970). Nonetheless, this approach is impractical for almost all problems in fluid mechanics, even for the most powerful of the supercomputers. Instead, alternative methods reduce the number of degrees of freedom of the phenomenon to more manageable equations. Those are the Reynolds-Averaged Navier-Stokes (RANS) and the Large Eddy Simulation (LES) approaches.

This approach is based on the procedure introduced by Reynolds (1895), who approached turbulence from a statistical point of view. According to Wilcox (2010), Reynolds averaging can be done in several ways: the time average, which is adequate for steady-state flows; the spatial average, for homogeneous turbulence; and the ensemble average, the more general among them. The method introduces a velocity fluctuation term in the Navier-Stokes equations to represent the Kolmogorov cascade, which can be represented by several models. Models include $\kappa - \epsilon$ (most popular), $\kappa - \omega$ (and derivatives), Spallart-Almaras, and others. The conjunct of RANS methods includes the Unsteady RANS (URANS), which applies frequency-basis temporal fluctuations. Ultimately, RANS produce a coarser approximation to the actual phenomenon (BLAIS, 2016).

The LES approach, on the other hand, implements a low-pass convolution wavenumber filter to remove small-scale eddies, while larger eddies are directly resolved

(SAGAUT, 2006). According to Sagaut (2006), the conventional approach to apply LES consists of first applying the filter on Navier-Stokes equations and then applying filtered equations on the CFD mesh. Applying this method can be done explicitly, by adding terms to the equations, or implicitly, by arranging the truncation error such that the numerical method will induce the desired effect. Since turbulence still needs to be resolved at larger scales, the mesh is usually finer compared to the one used with RANS.

In this work, turbulence is represented by the implicit LES (ILES) method. Details on how turbulence is approached in this work are provided in Section 2.2.3.

2.1.3 Multiphase flows

Phase stands for the state of matter. Multiphase flows are essentially flows that have at least two states involved: liquid, solid, and vapor or gas. Those phases can both contain the same component, such as water-water vapor, or different components, such as fluidized beds of ceramic particles in air. Additionally, the level of separation between phases needs to be of a scale way above the molecular. Most flows in nature and industry are multiphase and multicomponent. Industrial applications include all sorts of physical and chemical reactions, from drying goods to energy generation through liquid fuel combustion (CROWE *et al.*, 2011; YEOH; TU, 2016).

The multiphase flow theory discussed in this work is focused on binary granular-liquid (or solid-liquid) flows. Therefore, the principles can be applied to solid-gas flows (CROWE *et al.*, 2011)[8]. As in granular flows, the modeling frame in fluid-solid flows can be either Eulerian-Eulerian or Eulerian-Lagrangian, depending on the phase elementary unit-to-geometry size relation and the need for discrete information.

2.1.3.1 Eulerian-Eulerian approach

The work by Anderson and Jackson (1967) is usually referred to as the first benchmark on fluidization modeling. The authors applied the Eulerian-Eulerian (or multifluid) frame and derived averaged equations from the fraction of particle and gas components in a fluidized bed. Their theory was later reproduced by other authors, namely Buyevich (1971), Nigmatulin (1979), Drew (1983), Jenkins and Savage (1983), for example. In the early 1980s, with the advance of computational power, authors such as authors, such as Gidaspow *et al.* (1991), Gidaspow (1994), started to simulate those systems.

The method consists of representing the flow of the mixture with the regular Navier-Stokes equations. However, all quantities represent spatial averages and the contribution

[8] For information about liquid-gas and liquid-liquid flows, readers are referred to the books by Clift *et al.* (1978) and Crowe *et al.* (2011)

of each phase is accounted for by their volume fraction in the region. The set of equations originating from this approach is called Volume-Averaged Navier-Stokes (VANS) equations, which include equations for the mixture (resembling Eqs. (2.22a) and (2.22b)) and their single components.

Let us consider the fluid phase f in a binary fluid-particle mixture. In the VANS approach, the continuity equation for the component is (GIDASPOW, 1994; ZHOU *et al.*, 2010; BéRARD *et al.*, 2020):

$$\frac{\partial \varepsilon_f \rho_f}{\partial t} + \nabla \cdot (\varepsilon_f \rho_f \boldsymbol{u}) = 0 \tag{2.25}$$

where $\boldsymbol{u}$ is the fluid velocity vector, rho_f is the density of the fluid, and ε_f is the void (fluid volume) fraction ($\varepsilon_f = 1 - \varepsilon_p$, where ε_p is the particles/solids fraction). The continuity equation for the solid phase in the Eulerian-Eulerian approach resembles Eq. (2.25) but with solids properties (ρ_p and ε_p).

There are two different approaches to the momentum equation. The two approaches are commonly referred to as VANS model types A and B (or simply model A and B) (GIDASPOW, 1994; ZHOU *et al.*, 2010; BéRARD *et al.*, 2020; GEITANI *et al.*, 2023a)[9]. If one considers that each phase has its own pressure and shear stress, i.e., pressure and shear stress of the mixture result from contributions of both phases, then model A (Eq. (2.26a)) is being applied. Oppositely, if both pressure and shear stress of the mixture are equivalent to the one of the fluid, i.e., solids do not have pressure and shear stress, VANS model B (Eq. (2.26b)) is applied.

$$\frac{\partial \varepsilon_f \rho_f \boldsymbol{u}}{\partial t} + \nabla \cdot (\varepsilon_f \rho_f \boldsymbol{u} \otimes \boldsymbol{u}) = -\varepsilon_f \nabla p + \varepsilon_f \nabla \cdot \boldsymbol{\tau} + \varepsilon_f \rho_f \boldsymbol{g} - \boldsymbol{F}_{pf}^{A} \tag{2.26a}$$

$$\frac{\partial \varepsilon_f \rho_f \boldsymbol{u}}{\partial t} + \nabla \cdot (\varepsilon_f \rho_f \boldsymbol{u} \otimes \boldsymbol{u}) = -\nabla p + \nabla \cdot \boldsymbol{\tau} + \varepsilon_f \rho_f \boldsymbol{g} - \boldsymbol{F}_{pf}^{B} \tag{2.26b}$$

In Eqs. (2.26), $\boldsymbol{F}_{pf}^{A}$ and $\boldsymbol{F}_{pf}^{B}$ are source terms accounting for interphase (coupling) forces. Note that the difference between the momentum equations of models A and B is that model A has the void fraction multiplying both the pressure gradient and the divergence of the shear stress tensor of the mixture, while model B does not.

The technique uses CFD (Section 2.2.2) to solve both phases using Eqs. (2.25) and (2.26) (A or B) is called Two-Fluid Model (TFM). This method has been applied to several fluid-solid systems, including gas-solid circulating fluidized beds (WANG *et al.*, 2010), spouted beds (BéTTEGA *et al.*, 2009; BATISTA *et al.*, 2018), liquid-solid fluidized

[9] Zhou *et al.* (2010) also show an alternative a less frequently used nomenclature. The authors refer to model A as VANS model II and model B as VANS model I. The different nomenclature does not have any impact on the model.

beds (KOERICH *et al.*, 2018; METOLINA; LOPES, 2019), flows with gas-solid reactions (ZHONG *et al.*, 2016), and others.

As mentioned in Section 2.1.1, the main challenge in this approach is to determine a valid expression for granular rheology and hydrodynamics incorporating interaction characteristics such as particles' roughness. Additionally, particles' discrete behaviors are not accounted for, so results need to be interpreted from a space-average perspective.

2.1.3.2 Eulerian-Lagrangian approach

The Eulerian-Lagrangian approach borrows the volume-averaging concepts from the Eulerian-Eulerian approach to express the fluid phase while the particles are seen as discrete entities. If one takes the same fluid f as before and assumes its flow is incompressible, ρ_f vanishes from Eq. (2.25), and continuity becomes (ZHOU *et al.*, 2010; BéRARD *et al.*, 2020):

$$\frac{\partial \varepsilon_f}{\partial t} + \nabla \cdot (\varepsilon_f \boldsymbol{u}) = 0 \tag{2.27}$$

The same can be done for momentum. Disregarding hydrostatic pressure momentum VANS equations become (models A and B, respectively):

$$\rho_f \left[\frac{\partial \varepsilon_f \boldsymbol{u}}{\partial t} + \nabla \cdot (\varepsilon_f \boldsymbol{u} \otimes \boldsymbol{u}) \right] = -\varepsilon_f \nabla p + \varepsilon_f \nabla \cdot \boldsymbol{\tau} - \boldsymbol{F}_{pf}^{A} \tag{2.28a}$$

$$\rho_f \left[\frac{\partial \varepsilon_f \boldsymbol{u}}{\partial t} + \nabla \cdot (\varepsilon_f \boldsymbol{u} \otimes \boldsymbol{u}) \right] = -\nabla p + \nabla \cdot \boldsymbol{\tau} - \boldsymbol{F}_{pf}^{B} \tag{2.28b}$$

Since particles are Lagrangian entities, they can be modeled by Eqs. (2.2). However, a new term accounting for the interphase (coupling) forces on the particles ($\boldsymbol{f}_{pf}$) needs to be added to Eq. (2.2b) such that:

$$m_i \frac{\mathrm{d}\boldsymbol{v}_i}{\mathrm{d}t} = \sum_{j=1, j\neq i}^{N_p} (\boldsymbol{f}_{c,ij}) + \sum_{w=1}^{N_w} (\boldsymbol{f}_{c,iw}) + \sum_{k, k\neq i}^{N_p} (\boldsymbol{f}_{cn,ik}) + \boldsymbol{f}_{g,i} + \boldsymbol{f}_{pf,i} \tag{2.29}$$

This approach allows for a particle-scale description of the solid phase. Consequently, no solid distribution nor particle-to-domain size ratio restrictions are imposed *a priori*. Nonetheless, in practice, bottlenecks arise in computing. In this work, the Eulerian-Lagrangian approach was extensively applied. The choice is laid over the premise that the particles used in this work are too large and the flow geometry too restrictive (as shown in 6Article 2), thus, approaching particles' flow from the Eulerian perspective would be a coarse approximation. So the method is discussed in further detail in Section 2.2.4 and comments on the state-of-art are provided in Section 2.3.3.

2.1.3.3 Interphase (coupling) forces

Coupling forces are commonly applied to both Eulerian-Eulerian and Eulerian-Lagrangian approaches. Those forces are also volume-averaged, but they come from the force balance on a single particle, extrapolated to a granular system. This work applies the Eulerian-Lagrangian approach to represent liquid-solid flows with spherical particles, so it is convenient to start from a force balance on a single spherical particle with diameter d_p.

2.1.3.3.1 Creeping flow

When a discrete particle i is surrounded by an Eulerian fluid, the momentum coupling between the two phases can be characterized by the following aspects (CROWE *et al.*, 2011):

1. Volume fraction of fluid and particle (ε_f and ε_p, respectively)[10].
2. Apparent density of the particle $\rho_p - \rho_f$.
3. Particle spacing, which is the ratio between the fluid characteristic length and the particle size L/D
4. Response times.

Response times are the time interval between a perturbation and a reaction to this perturbation. For this project, only the momentum response time is introduced[11]. The ratio between the effect of gravitational and viscous forces can be expressed by:

$$\mathrm{Ar} = d_p^3 \frac{\rho_l \left(\rho_p - \rho_f\right) |\boldsymbol{g}|}{\mu} \tag{2.30}$$

where Ar is the Archimedes number. When $(\rho_p - \rho_l) \to 0$, the particle is subject to viscous effects only. Those effects can be expressed by the drag force $\boldsymbol{f}_{D,i}$, which represents the totality of $\boldsymbol{f}_{pf,i}$ in this case. For a creeping flow, i.e., $\varepsilon_f \to 1.0$ and $L/D \to 0$, Eq. (2.29) becomes:

$$m_i \frac{\mathrm{d}\boldsymbol{v}_i}{\mathrm{d}t} = \boldsymbol{f}_{pf,i} \approx \boldsymbol{f}_{D,i} \tag{2.31}$$

The drag force can be expressed in terms of a drag coefficient $C_{D,i}$ as:

$$\boldsymbol{f}_{D,i} = \frac{1}{2} \rho_f C_{D,i} A_i \left|(\boldsymbol{u} - \boldsymbol{v}_i)\right| (\boldsymbol{u} - \boldsymbol{v}_i) \tag{2.32}$$

[10] For a binary flow, $\varepsilon_f + \varepsilon_p = 1$.

[11] The reader is referred to Crowe *et al.* (2011) for temperature and mass transport response times.

where A_i the projected area (for a sphere, πd_p^2), and $(\boldsymbol{u} - \boldsymbol{v}_i)$ the relative velocity between the fluid and the particle. The flow regime of the particle can be expressed by the Particle Reynolds number, defined by (YANG, 2003b; CROWE *et al.*, 2011):

$$\mathrm{Re}_{p,i} = \frac{\rho_f \left|(\boldsymbol{u} - \boldsymbol{v}_i)\right| d_p}{\mu} \tag{2.33}$$

The relation between $\mathrm{Re}_{p,i}$ and C_D is well established for spherical particles. It can be expressed by Figure 2.5.

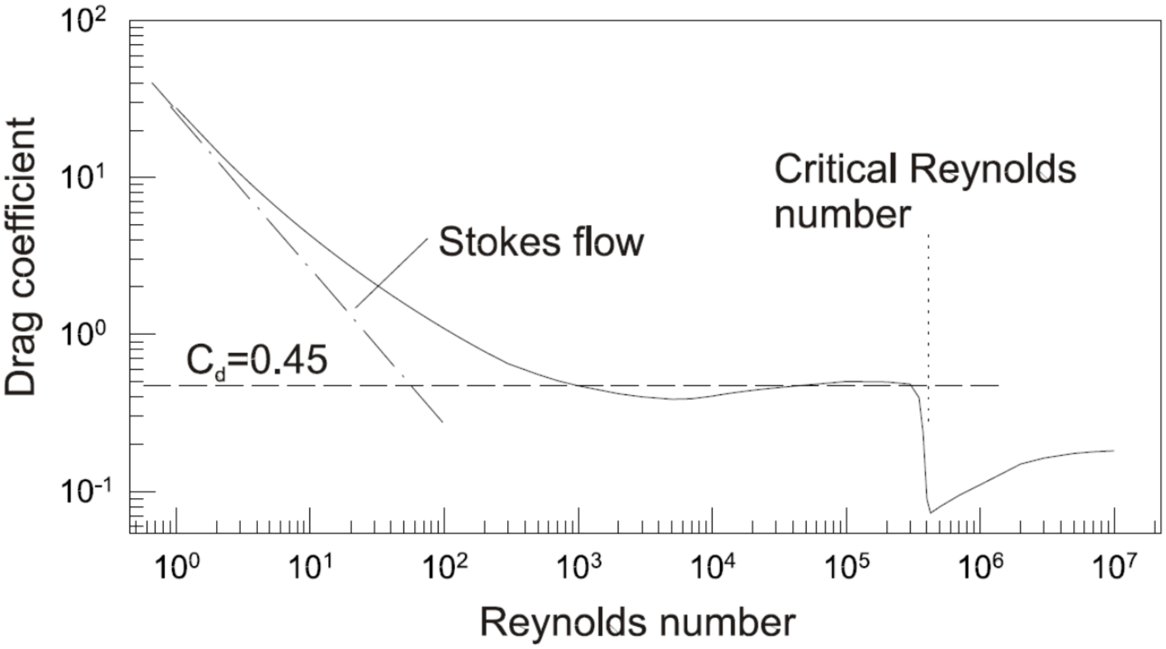


Figure 2.5 – Drag coefficient as a function of particle Reynolds number. Reproduced from Crowe *et al.* (2011).

The drag coefficient C_D results from the integration of the pressure and the vorticity on the particle surface (CLIFT *et al.*, 1978). The analytical solution for the creeping flow ($\varepsilon_f \to 1.0$) results on the Stokes equation (CLIFT *et al.*, 1978; CROWE *et al.*, 2011):

$$C_{D_{\varepsilon_f \to 1.0}} = \frac{24}{\mathrm{Re}_{p,i}} \tag{2.34}$$

The Stokes regime (or Stokes flow), where Eq. (2.34) is valid, is limited to $\mathrm{Re}_{p,i} < 0.1$. Applying the definitions given by Eqs. (2.33) and (2.34), one can derive the momentum response time (CROWE *et al.*, 2011):

$$\tau_m = \frac{\rho_p d_p^2}{18\mu} \tag{2.35}$$

The Stokes number of the particle is then defined as:

$$St = \tau_m / \tau_{flow} \tag{2.36}$$

where τ_{flow} is the characteristic flow time, defined as:

$$\tau_{flow} = L/U \tag{2.37}$$

where L is the characteristic length of the flow (the diameter of a pipe, for example) and U is the fluid velocity. The Stokes number is a measurement of how perturbations on the fluid field will affect particles. For the limit where $St \to 0$, the particle is fully carried by the fluid flow, while for $St \to \infty$, the particle does not feel the movement of the fluid (CROWE *et al.*, 2011).

2.1.3.3.2 Concentrated flow

The creeping flow helps define dimensionless numbers to characterize the fluid-particle flow. Even so, the approximations assumed are very restrictive. In concentrated fluid-particle flows, particles and fluid exchange momentum through several mechanisms. Maxey and Riley (1983) decomposed $\boldsymbol{f}_{pf,i}$ into (CROWE *et al.*, 2011):

$$\begin{aligned}\boldsymbol{f}_{pf,i} = &\underbrace{\boldsymbol{f}_{\nabla p,i} + \boldsymbol{f}_{\nabla\cdot\boldsymbol{\tau},i}}_{\text{Undisturbed flow}} + \underbrace{\boldsymbol{f}_{\mathrm{Ar},i}}_{\text{Archimedes}} + \underbrace{\boldsymbol{f}_{D,i}}_{\text{Drag}} + \underbrace{\boldsymbol{f}_{Basset,i}}_{\text{History}} + \\ &+ \underbrace{\boldsymbol{f}_{vm,i}}_{\text{Virtual mass}} + \underbrace{\boldsymbol{f}_{\mathrm{Saffman},i} + \boldsymbol{f}_{\mathrm{Magnus},i}}_{\text{Lift}}\end{aligned} \tag{2.38}$$

As a consequence of the approach to the pressure and shear stress terms in models A and B (Eqs. (2.28a) and (2.28b), respectively), the coupling terms on the fluid side ($\boldsymbol{F}_{pf}^{A}$ and $\boldsymbol{F}_{pf}^{B}$) are defined as:

$$\boldsymbol{F}_{pf}^{A} = \frac{1}{V_\Omega}\sum_{i=1}^{N_p}\left(\boldsymbol{f}_{pf,i} - \boldsymbol{f}_{\nabla p,i} - \boldsymbol{f}_{\nabla\cdot\boldsymbol{\tau},i}\right) \tag{2.39a}$$

$$\boldsymbol{F}_{pf}^{B} = \frac{1}{V_\Omega}\sum_{i=1}^{N_p}\left(\boldsymbol{f}_{pf,i}\right) \tag{2.39b}$$

where Ω is the domain with volume V_Ω containing N_p particles. The components of Eq. $\boldsymbol{f}_{pf,i}$ are:

Undisturbed flow: Forces due to fluid's pressure gradient and shear stress, respectively represented by:

$$\boldsymbol{f}_{\nabla p,i} = -V_p\nabla p \tag{2.40}$$

$$\boldsymbol{f}_{\nabla\cdot\boldsymbol{\tau},i} = V_p\nabla\cdot\boldsymbol{\tau} \tag{2.41}$$

Archimedes (buoyancy): Force due to the fluid displaced by the particle, calculated by[12]:

$$\boldsymbol{f}_{\mathrm{Ar},i} = -V_p\rho_f\boldsymbol{g} \tag{2.42}$$

[12] If ∇p incorporates the hydrostatic pressure, $\boldsymbol{f}_{\mathrm{Ar},i}$ is implicit in $\boldsymbol{f}_{\nabla p,i}$ and the term can be removed from Eq. (2.38) (ZHOU *et al.*, 2010).

Drag: Force associated with the resistance felt by an object that moves at a velocity different from the surrounding fluid. A common definition for drag in fluid-particle flows is given by (GIDASPOW, 1994; ZHOU *et al.*, 2010; BéRARD *et al.*, 2020):

$$\boldsymbol{F}_D^A = \sum_{i=1}^{N_p} \beta_i \left(\boldsymbol{u} - \boldsymbol{v}_i\right) \tag{2.43a}$$

$$\boldsymbol{F}_D^B = \frac{1}{\varepsilon_f} \sum_{i=1}^{N_p} \beta_i \left(\boldsymbol{u} - \boldsymbol{v}_i\right) \tag{2.43b}$$

where β_i is a drag parameter represented by correlations[13]. A wide variety of correlations were proposed to represent β_i for concentrated fluid-solid flows (GIDASPOW, 1994; FELICE, 1994; BENYAHIA *et al.*, 2006; BEETSTRA *et al.*, 2007; MAZZEI; LETTIERI, 2007; RONG *et al.*, 2013; JAJCEVIC *et al.*, 2013). Gidaspow (1994) developed a correlation based on Ergun (1952) and Wen and Yu (1966) equations for the pressure drop in particle-fluid flows. Considering perfectly spherical particles:

$$\beta_i = \begin{cases} 150\frac{\left(1-\varepsilon_f\right)^2\mu}{(\varepsilon_f d_p)^2} + 1.75\frac{\rho_f|\boldsymbol{u}-\boldsymbol{v}_i|(1-\varepsilon_f)}{d_p\varepsilon_f}, \text{ for } \varepsilon_f < 0.8 \\ \frac{3}{4}C_{D,i}\frac{\rho_f(1-\varepsilon_f)|\boldsymbol{u}-\boldsymbol{v}_i|}{d_p}, \text{ for } \varepsilon_f \geq 0.8 \end{cases} \tag{2.44}$$

where $C_{D,i}$ is the drag coefficient for a single particle, calculated by:

$$C_{D,i} = \begin{cases} \frac{24}{\mathrm{Re}_{p,i}}\left(1 + 0.15\mathrm{Re}_{p,i}^{0.687}\right), \text{ for } \mathrm{Re}_{p,i} < 1000 \\ C_{D,i} = 0.44, \text{ for } \mathrm{Re}_{p,i} \geq 1000 \end{cases} \tag{2.45}$$

Following a similar approach, other authors proposed correlations for β_i mainly finding a function $G(\varepsilon_f, \mathrm{Re}_{p,i})$ that represents the ratio between the drag coefficient $C_{D,i}$ and drag coefficient for a single isolated particle C_{D0}:

$$\frac{C_{D,i}}{C_{D0}} = G(\varepsilon_f, \mathrm{Re}_{p,i}) \tag{2.46}$$

In this approach, β_i is:

$$\beta_i = \frac{1}{2}C_{D0}\frac{\pi d_p^2}{4}\rho_f\left|\boldsymbol{u} - \boldsymbol{v}_i\right| G(\varepsilon_f, \mathrm{Re}_{p,i}) \tag{2.47}$$

Felice (1994) and Rong *et al.* (2013) proposed a correlation for $G(\varepsilon_f, \mathrm{Re}_{p,i})$ based on C_{D0} proposed by DallaValle (1948):

$$C_{D0} = \left(0.63 + \frac{4.8}{\sqrt{\mathrm{Re}_p}}\right)^2 \tag{2.48}$$

[13] The parameter β_i is often referred to as "interphase momentum transfer coefficient" (GIDASPOW, 1994) but drag does not correspond to the totality of the coupling between phases.

The correlation proposed by Di Felice for $G(\varepsilon_f, \mathrm{Re}_{p,i})$ is:

$$G(\varepsilon_f, \mathrm{Re}_{p,i}) = \varepsilon_f^{\,2-\left\{3.7-0.65exp\left[-\frac{\left(1.5-log_{10}\mathrm{Re}_{p,i}\right)^2}{2}\right]\right\}} \tag{2.49}$$

This equation assumes the strong hypothesis that the exponent is a function only of Re_p and not of ε_f. As shown by several authors (BENYAHIA *et al.*, 2006; MAZZEI; LETTIERI, 2007; BEETSTRA *et al.*, 2007; CELLO *et al.*, 2010; RONG *et al.*, 2013), this hypothesis leads to errors in the estimation of drag. The Rong correlation is similar to Di Felice but tries to account for the effect of the void fraction on the exponent, written as:

$$G(\varepsilon_f, \mathrm{Re}_{p,i}) = \varepsilon_f^{\,2-\left\{2.65\left(\varepsilon_f+1\right)-\left(5.3-3.5\varepsilon_f\right)\varepsilon_f^2 exp\left[-\frac{\left(1.5-log_{10}\mathrm{Re}_{p,i}\right)^2}{2}\right]\right\}} \tag{2.50}$$

An alternative approach was used by Beetstra *et al.* (2007). It determines a normalized drag force $F(\varepsilon_f, \mathrm{Re}_{p,i})$ by correcting the drag force using the Stokes-Einstein drag ($3\pi d_p \boldsymbol{u}$), the latter representing the drag experienced by a single isolated particle ($\varepsilon_f \to 1$) in the limit of $\mathrm{Re}_{p,i} \to 0$:

$$F(\varepsilon_f, \mathrm{Re}_{p,i}) = \frac{\boldsymbol{F}_d}{3\pi d_p(\boldsymbol{u} - \boldsymbol{v}_i)} \tag{2.51}$$

The authors proposed correlations for both mono and poly-dispersed beds. The correlation for mono-dispersed flows is:

$$\begin{aligned} F(\varepsilon_f, \mathrm{Re}_{p,i}) = {} & \frac{10\left(1-\varepsilon_f\right)}{\varepsilon_f^2} + \varepsilon_f^2\left(1+1.5\varepsilon_f^{0.5}\right) \\ & + \frac{0.413\mathrm{Re}_{p,i}}{24\varepsilon_f^2}\left[\frac{(1/\varepsilon_f) + 3\varepsilon_f\left(1-\varepsilon_f\right) + 8.4\mathrm{Re}_{p,i}^{-0.343}}{1+10^{3\left(1-\varepsilon_f\right)}\mathrm{Re}_{p,i}^{-\left(1+4\left(1-\varepsilon_f\right)\right)/2}}\right] \end{aligned} \tag{2.52}$$

History (Basset): Cumulative force due to the delay in the formation of the boundary layer over the particle's surface. The complete term including the effect of the initial relative velocity $(\boldsymbol{u} - \boldsymbol{v}_i)_0$ deduced by Reeks and McKee (1984) is (CROWE *et al.*, 2011):

$$\boldsymbol{f}_{Basset,i} = \frac{3}{2}d_p^2\sqrt{\pi\rho_f\mu}\left[\int_{t_0}^{t}\frac{\frac{\mathrm{d}(\boldsymbol{u}-\boldsymbol{v}_i)}{\mathrm{d}t'}}{\sqrt{t-t'}}\,dt' + \frac{(\boldsymbol{u}-\boldsymbol{v}_i)_0}{\sqrt{t}}\right] \tag{2.53}$$

where t_0 is the moment of perturbation and t is the present time.

Virtual mass: Force due to the perturbation caused by the acceleration of the particle. It is often called "added mass" or "apparent mass" because it results in an inertial-like effect on the particle. It can be calculated by[14]:

$$\boldsymbol{f}_{vm,i} = \frac{V_p\rho_f}{2}\left[\frac{\mathrm{D}\boldsymbol{u}_i}{\mathrm{D}t} - \frac{\mathrm{d}\boldsymbol{v}_i}{\mathrm{d}t}\right] \tag{2.54}$$

[14] Crowe *et al.* (2011) mentions that correlations were proposed in the literature to represent the virtual mass force. Since virtual mass is not applied to this work, the correlations are not discussed.

where $\frac{\mathrm{D}\boldsymbol{u}_i}{\mathrm{D}t}$ is the substantial derivative (BIRD *et al.*, 2007) of the fluid velocity at the particle's position.

Lift force: Force acting perpendicular to the relative velocity. Lift is associated with rotation. It can be decomposed into Saffman and Magnus lift forces. The Saffman lift force (SAFFMAN, 1965; SAFFMAN, 1968) is caused by the velocity distribution on the particle's surface. It can be represented by combining the correlation proposed by Saffman (1965), Saffman (1968) and the lift coefficient proposed by Mei (1992), usually referred to as the Saffman-Mei lift force model, and is calculated by (CROWE *et al.*, 2011):

$$\boldsymbol{f}_{\text{Saffman},i} = 1.61 C_{\text{Saffman},i} d_p^2 (\mu \rho_f)^{1/2} \left|\boldsymbol{\omega}_{f,i}\right|^{-1/2} \left[(\boldsymbol{u} - \boldsymbol{v}_i) \times \boldsymbol{\omega}_{f,i}\right] \tag{2.55}$$

where the vorticity $\boldsymbol{\omega}_{f,i}$ corresponds to the curl of the vector field, calculated as:

$$\boldsymbol{\omega}_{f,i} = \nabla \times \boldsymbol{u} \tag{2.56}$$

In Eq. (2.55), $C_{\text{Saffman},i}$ is the Saffman lift coefficient, calculated by:

$$C_{\text{Saffman},i} = \begin{cases} \left(1 - 0.3314\alpha^{1/2}\right) \exp\left(\frac{-\text{Re}_{p,i}}{10}\right) + 0.3314\alpha^{1/2}, \text{ for } \text{Re}_{p,i} \leq 40 \\ 0.0524(\alpha_l \text{Re}_{p,i})^{1/2}, \text{ for } \text{Re}_{p,i} > 40 \end{cases} \tag{2.57}$$

where α is:

$$\alpha = \frac{d_p}{2\left|\boldsymbol{u} - \boldsymbol{v}_i\right|} \left|\boldsymbol{\omega}_{f,i}\right| \tag{2.58}$$

While Saffman happens due to rotational motion in the fluid field, Magnus is the lift force due to the particle's rotation. The particle's angular motion causes a pressure gradient on the fluid field, pushing the fluid and delocating the particle sideways. The force is calculated by:

$$\boldsymbol{f}_{\text{Magnus},i} = \frac{1}{2} \rho_f \pi d_p^2 C_{\text{Magnus},i} \left[(\bar{\boldsymbol{\omega}}_i \times (\boldsymbol{u} - \boldsymbol{v}_i))\right] \left|(\boldsymbol{u} - \boldsymbol{v}_i)\right| \tag{2.59}$$

where $\bar{\boldsymbol{\omega}}_i$ is the normalized angular velocity ($\boldsymbol{\omega}_i / \left|\boldsymbol{\omega}_i\right|$) and $C_{\text{Magnus},i}$ is the Magnus lift coefficient. The Magnus coefficient depends on the spin parameter, defined by[15]:

$$\Theta_i = \frac{d_p \left|\boldsymbol{\omega}_i\right|}{2\left|(\boldsymbol{u} - \boldsymbol{v}_i)\right|} \tag{2.60}$$

and the rotational Reynolds number, defined by:

$$\text{Re}_{r,i} = \frac{\rho_f \left|\boldsymbol{\omega}_i\right| d_p^2}{4\mu} \tag{2.61}$$

The Magnus coefficient is (OESTERLé; DINH, 1998; CROWE *et al.*, 2011):

$$C_{\text{Magnus},i} = \begin{cases} 2\Theta, \text{ for } 1 < \Theta_i \text{ or } \Theta_i > 60 \text{ and } 10 < \text{Re}_{r,i} \text{ or } \text{Re}_{r,i} > 140 \\ 0.45 + \left(2\Theta_i - 0.45\right) e^{-0.075 \Theta_i^{0.4} \text{Re}_{r,i}^{0.7}}, \text{ else.} \end{cases} \tag{2.62}$$

[15] Note that Magnus does not include a viscous dissipation term.

2.2 Numerical methods

The set of complex differential equations describing the phenomena previously introduced requires robust numerical strategies. The physical meaning and the efficiency of the solving process rely on carefully implementing those methods in computers. This work's simulations were carried out in Lethe (BLAIS *et al.*, 2020; GOLSHAN *et al.*, 2022; GEITANI *et al.*, 2023b; LETHE, 2023), an open-source CFD/DEM/CFD-DEM/Multiphysics software based on the robust finite element library deal.II (ARNDT *et al.*, 2021; ARNDT *et al.*, 2022; ARNDT *et al.*, 2023)[16]. In this section, we introduce the concepts behind the numerical computational methods applied to this work.

2.2.1 Integration schemes on Discrete Elements Method

As shown in Section 2.1.1.2, the discrete elements method describes the particles' dynamics as discrete entities to which Newton's and Euler's Second Law of Motion is applied (Eqs. (2.2)). Those are ordinary differential equations (ODE's) that need to be integrated in time. Several methods are available to do so, but the method of choice can vary with the particles (are they all spherical and of the same diameter?), their distribution (how frequent collisions are?), the geometry of the domain (are there concave or convex walls?), the collision model (hard-sphere or soft-sphere?), the computational resources (is parallelization available?), and other factors (NOROUZI *et al.*, 2016).

According to Norouzi *et al.* (2016), a robust general-purpose DEM software needs to implement explicit integration methods because they allow for higher time steps and are more stable compared to implicit schemes. Since we are applying the soft-sphere method, the choice of time step needs to be small enough to account for the collision time. The criterium is given by the critical Rayleigh time step, which is calculated by (LI *et al.*, 2005):

$$\Delta t_{critical} = \frac{\pi R}{\mathcal{X}} \sqrt{\frac{\rho_p}{G}} \tag{2.63}$$

where R is the radius of the particles, G is the shear modulus, and $\mathcal{X}$ is approximately:

$$\mathcal{X} = 0.1631\nu_p + 0.8766 \tag{2.64}$$

As a general rule, Norouzi *et al.* (2016) recommends a maximum time step in the range of $10^{-7} \leq \Delta t \leq 10^{-5}$. For such a small time step and given the large amount of particles granular systems have, the choice of a fast algorithm and an adequate implementation is key to good results in a reasonable time. Contact detection algorithms, for example, can be an important performance bottleneck.

[16] A project this humble candidate is proud of contributing to.

According to Blais *et al.* (2019), the Naive approach to contact detection, i.e., search through all pairs, implies a processing time of $\mathcal{O}\left(N_p^2\right)$, meaning that the processing time increases with the number of particles to the power of two. Thus, other more scalable approaches are necessary. The authors and Norouzi *et al.* (2016) present two other families of algorithms for more efficient contact detection: cells-based and Verlet lists. The cell-based algorithms use mesh structures to find potential contact. Usually, this approach is limited to a broader search, requiring a coarse mesh to work properly. Verlet lists (VERLET, 1967), oppositely, depends on particles' position only. Contact candidates are stored in lists and tracked for a certain number of time steps.

In Lethe, contact detection is divided into broad and fine search steps (GOLSHAN *et al.*, 2022). The broad search is a mesh-based step, in which particle's collision candidates are selected if they are in the same cell or in a cell in the neighborhood. The particles stored in the broad search list are contact candidates for the fine search, which uses Verlet lists. This combination results in a more complex contact detection algorithm but saves a lot of computational cost[17].

After calculating all forces and torques, the method requires the integration of Eqs. (2.2). Lethe implements two explicit integration methods on DEM simulations: explicit Euler (RENZO; MAIO, 2004) and Verlet (FRAIGE; LANGSTON, 2004). Explicit Euler's algorithm can be represented by (GOLSHAN *et al.*, 2022; LETHE, 2023):

$$\boldsymbol{v}_i^{t+\Delta t} = \boldsymbol{v}_i^t + \frac{\mathrm{d}\boldsymbol{v}_i}{\mathrm{d}t}\Delta t \tag{2.65a}$$

$$\boldsymbol{x}_i^{t+\Delta t} = \boldsymbol{x}_i^t + \boldsymbol{v}_i^t\Delta t \tag{2.65b}$$

and Verlet, by:

$$\boldsymbol{v}_i^{t+\Delta t/2} = \boldsymbol{v}_i^t + \frac{\mathrm{d}\boldsymbol{v}_i}{\mathrm{d}t}\frac{\Delta t}{2} \tag{2.66a}$$

$$\boldsymbol{x}_i^{t+\Delta t} = \boldsymbol{x}_i^t + \boldsymbol{v}_i^{t+t/2}\Delta t \tag{2.66b}$$

$$\boldsymbol{v}_i^{t+\Delta t} = \boldsymbol{v}_i^{t+\Delta t/2} + \frac{\mathrm{d}\boldsymbol{v}_i}{\mathrm{d}t}\frac{\Delta t}{2} \tag{2.66c}$$

The integration process consists of applying the selected integration scheme to all particles at each time step until the end of the simulation is reached. The algorithm implemented on Lethe is summarized in Figure 2.6.

2.2.2 Computational Fluid Dynamics

The family of numerical computational methods applied to fluid dynamics problems is called Computational Fluid Dynamics (CFD). Those methods are designed to solve the

[17] Notes on performance and scalability are provided by Golshan *et al.* (2022).

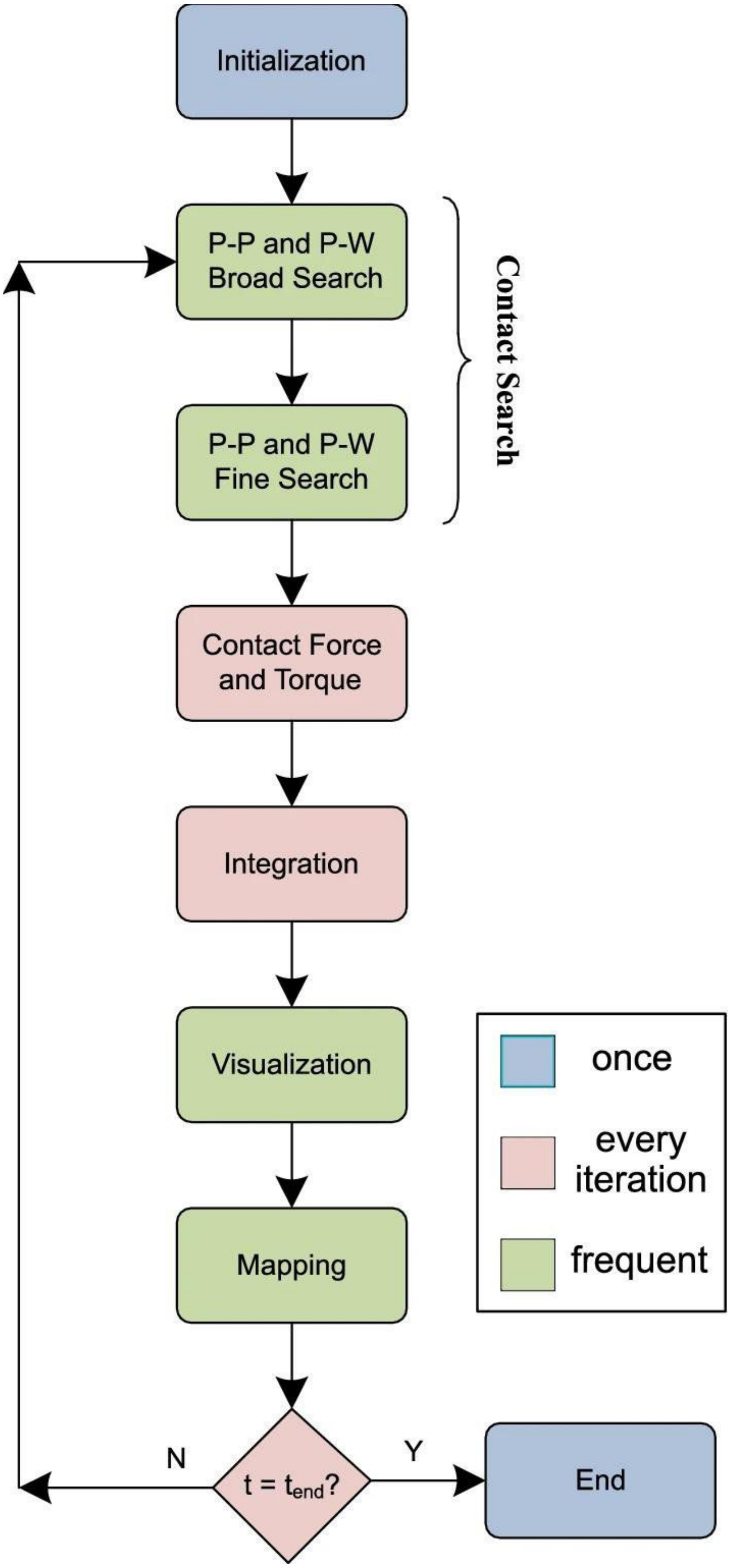


Figure 2.6 – DEM algorithm in Lethe. Reproduced from Golshan *et al.* (2022).

partial differential equations (PDE's), such as the ones presented in Section 2.1.2, using discretization. As explained, the incompressible Navier-Stokes equations represent the fluid field from the Eulerian frame. Discretization consists of "breaking" the domain into subdomains (cells). The cells will represent portions of the continuum to which the models are to be solved iteratively (FERZIGER *et al.*, 2020).

According to Ferziger *et al.* (2020), the components of a CFD solution are:

1. Mathematical model: Set of equations, hypotheses, initial value, and boundary conditions. The hypotheses adopted in this work include, for example:

   a) The fluid flow can be approached from an Eulerian frame.

   b) The flow is incompressible and isothermal.

   c) The system is chemically stable and mass transfer is negligible.

   The incompressible Navier-Stokes momentum equations are non-linear first-order in time second-order in space partial differential. Consequently, each equation requires at least an initial value and two boundary conditions. Additionally, at least one of the boundary conditions needs to be of Dirichlet type.

2. Discretization method: Method applied to solve the model equations. The most used discretization methods in fluid dynamics problems are finite difference (FDM), finite volume (FVM), and finite element (FEM)[18]. Other methods include spectral schemes, boundary element methods, and lattice-Boltzmann. The method of choice depends on the use but the first three are the more versatile.

3. Numerical mesh (grid): Discretizing the domain implies representing it as a set of discrete entities. Those entities are called cells while the conjunct of them is called mesh (or grid). Meshes can be classified according to their dimensionality (1D, 2D, or 3D), and cells' distribution (structured, block-structured, composite, unstructured) and shapes (triangular, quadrangular, pyramidal, hexahedral) (FERZIGER *et al.*, 2020).

4. Finite approximation: How the element will represent the solution. In FDM, for example, the approximation is done by applying derivatives at cells' vertices. In FVM, surfaces and volume integrations are applied. In FEM, we use shapes and weighting functions.

5. Solution algorithm: The previous steps result in a system of algebraic equations that need to be solved iteratively by a solution algorithm.

6. Convergence criteria: The level of approximation required for the solution.

[18] Used in this work.

2.2.2.1 Discretization numerical methods

The simplest and most commonly implemented method applied to fluid dynamics problems is the FDM. It consists of approximating the differential terms from finite differences, defined similarly as derivatives. Those finite differences are defined at elements' vertices (nodes). The differences can be obtained from Taylor series expansion or polynomial approximations. As such, increasing the solution order is a matter of selecting the order truncation term. Nonetheless, the method requires the use of structured meshes, since it depends on the interconnection between neighbor nodes positions (LEVEQUE, 2007; FERZIGER *et al.*, 2020). Moreover, applying gradient-type boundary conditions requires additional approximation of the boundary data (REDDY, 2015).

The FVM is also very popular on CFD. It is the method behind popular CFD software such as Open∇FOAM (OPENFOAM, 2021), Fluent by Ansys®(ANSYS, 2013), MFIX by NETL Multiphase Flow Science (PORCU *et al.*, 2023), and STAR-CCM+ (SOFTWARE, 2021). It consists of solving fluxes at mesh elements. In contrast with the FDM, the nodes are defined at cells' centroids. The net flux in the cell (control volume) is obtained by solving the integral of the fluxes at cells' faces using the divergence theorem. The flux equations can then be interpolated cell-by-cell applying several interpolation schemes, such as upwind interpolation (UDS), linear interpolation (CDS), and Quadratic Upwind Interpolation (QUICK). The method also supports higher-order interpolation schemes (FERZIGER *et al.*, 2020).

FEM, the method applied in this work, traditionally consists of representing domains by a conjunct of subdomains (finite elements) where phenomena are approximated to linear combinations of approximation functions and undetermined parameters (REDDY, 2015). These parameters are the unknowns of the linear system and represent the values of the properties at the nodes, defined as the vertices of the elements. The constraints to the parameters are given by the physics of the problem (mathematical model) (REDDY, 2015). The approximation functions are polynomial functions of any order. According to Malan *et al.* (2002), the linear FEM is very similar to the vertex-centered FVM. The method is explained in further detail in Section 2.2.3.

2.2.3 Finite Elements Method

The finite elements method is classified as a variational method, which comprises methods of finding the Maxima or Minima of a function by finding a defined integral expression involving the function. The derivatives of this expression give the Maxima or Minima values (BYERLY, 1917). Hence, applying the FEM requires deriving the model's variational form, i.e., the integral form of the mathematical model to be resolved multiplied by a weight function (LARSON; BENGZON, 2013). The integral form includes boundary and initial conditions, governing equations, constraints, etc (REDDY, 2018). Another name for the variational form is "weak form". The weak form of a mathematical model is bounded by the domain where it was defined for incorporating constraints into the model, consequently, it is restricted to the problem to which it was defined. On the contrary, the strong form of a model is still valid when boundaries change (LARSON; BENGZON, 2013; REDDY; GARTLING, 2010; REDDY, 2015; REDDY, 2018).

According to Reddy and Gartling (2010), the finite elements method consists of three basic steps:

1. Divide the domain into finite elements.
2. Develop algebraic relations between pairs of dual variables, called primary and secondary. The authors define dual variables in a cause-consequence logic. Examples are displacement and force, temperature and heat, pressure and fluid flow, fluid velocity and shear stress, and so forth.
3. Putting together the algebraic relations among elements (assembly). The algebraic system is then solved using linear algebra methods.

Those steps are followed on any application on FEM. For brevity, the procedure is explained conceptually. For a more didactic approach, the reader is referred to the Appendix A where the weak form of the Poisson equation is derived, and for a more complete explanation, to Larson and Bengzon (2013), Reddy (2015), Reddy (2018).

2.2.3.1 Discretization of the domain into finite elements

Representing the domain as finite elements implies taking a domain Ω with boundaries Γ and splitting it into parts Ω_e with boundaries Γ_e. In general, the nodes of the elements are then defined at cells' vertices, where the values representing the continuum field will be resolved (REDDY; GARTLING, 2010). The unknowns, also known as degrees of freedom, can also be defined at elements boundaries or centroids (DEAL.II, 2023).

The finite elements take shapes of simple geometric structures. The most popular are triangles and quadrangles in 2D, and pyramids and hexahedra in 3D. In general, when the domain does not have the same shape as the elements the volume of Ω is greater than the summation of Ω_e elements' volume (REDDY; GARTLING, 2010). The mesh can be adaptive, i.e., redistributed and locally refined depending on the problem, using algorithms such as the Kelly error estimator (KELLY *et al.*, 1983; BARBEAU *et al.*, 2022). The meshes applied to this work are not adaptive, so the explanation that follows is for a static mesh.

2.2.3.2 Weak form

Our goal in FEM is to obtain an approximation such that (REDDY, 2015):

$$\boldsymbol{u}\left(\boldsymbol{x},t\right) \approx \sum_{j=1}^{N_{nodes}} \boldsymbol{u}_{e,j}(t)\psi_{e,j}(\boldsymbol{x}) \tag{2.67}$$

where N_{nodes} is the number of j nodes in a finite element e. Note that the approximation $\boldsymbol{u}_{e,j}(t)$ at the node j is a function of time only, while the approximation function $\psi_{e,j}(\boldsymbol{x})$ depends on the position $\boldsymbol{x}$.

The weighted-residual finite element model of the PDE is obtained by multiplying it by a weight function w and integrating it in space. Supposing that $\boldsymbol{u}$ is twice differentiable in $\boldsymbol{x}$, the term with $w\nabla\boldsymbol{u}\cdot(\nabla\boldsymbol{u})$ is then integrated by parts, resulting in a weighted integral term with $\nabla w \cdot \nabla\boldsymbol{u}$. Before this step, the approximation function $\psi_{e,j}$ of choice needed to be at least twice differentiable (a second-degree polynomial, for example). The integration by parts weakens the continuity requirement transferring part of the differentiation from $\boldsymbol{u}$ to w, allowing for lower-order approximations (REDDY; GARTLING, 2010).

The last step consists of applying the divergence theorem on the fluxes within the cells. By doing so, the integral of the volumetric fluxes becomes a cyclic integral of the fluxes through the boundaries Γ_e. The resulting integrodifferential equation is called the variational form or weak form of the PDE.

If one chooses to represent the w by the same function as $\psi_{e,j}$, the obtained model is going to be the weak form Galerkin (REDDY; GARTLING, 2010). If continuity is imposed throughout the finite elements, the continuous Galerkin method is being used, while the opposite results in the application of the discontinuous Galerkin method (LI, 2006). The implications of each method are explained in Section 2.2.3.8.

2.2.3.3 Initial and boundary conditions

The transient problem will require an initial condition $\boldsymbol{u}(\boldsymbol{x}, t = 0)$, which is applied directly to all $\boldsymbol{u}_{e,ij}$ at the first time step $i = 0$. Boundary conditions can be of the two types (REDDY; GARTLING, 2010; REDDY, 2015):

$$\text{Dirichlet (essential):}\ \ \boldsymbol{u}|_{\Gamma_u} = f_u(\boldsymbol{x}|_{\Gamma_u}, t) \text{ on } \Gamma_u \tag{2.68a}$$

$$\text{Neumann (natural):}\ \ \left.\left(\frac{\partial \boldsymbol{u}}{\partial t}\right)\right|_{\Gamma_f} \boldsymbol{n}(\boldsymbol{x}|_{\Gamma_f}) \equiv f_f(\boldsymbol{x}|_{\Gamma_f}, t) \text{ on } \Gamma_f \tag{2.68b}$$

where f_u and f_f are expressions representing the value and the flux prescribed at the boundaries Γ_u and Γ_f, respectively, and $\boldsymbol{n}$ is the unit vector normal to the boundary Γ_f at $\boldsymbol{x}|_{\Gamma_f}$. Boundary conditions should enclose the entire domain Ω such that $\Gamma = \Gamma_u \cup \Gamma_f$ (REDDY; GARTLING, 2010). The same procedure applied to the weak form of the PDE is applied to the boundary conditions.

2.2.3.4 Assembly of the linear system

The weak form Galerkin model in the matrix form can be represented by:

$$\mathbf{M}\dot{\mathbf{u}} + \mathbf{A}\mathbf{u} = \mathrm{F} \tag{2.69}$$

where the unknowns in the vector $\mathbf{u}$ are multiplied by the coefficient matrix $\mathbf{A}$, the rates in the vector $\dot{\mathbf{u}}$ (composed of the time derivative terms $\partial \boldsymbol{u}_{e,ij}/\partial t$) are multiplied by the mass matrix $\mathbf{M}$, and the right-hand side vector is represented by F (REDDY; GARTLING, 2010). The initial condition is applied to Eq. (2.69) for the time step $i = 0$ and boundary conditions are applied directly to the nodes at Γ.

2.2.3.5 Integration in space

The numerical integration in space will also require a coordinate transformation:

$$\boldsymbol{x} \approx \sum_{j=1}^{N_{nodes}} \boldsymbol{x}_{e,j}\varphi_{e,j} \tag{2.70}$$

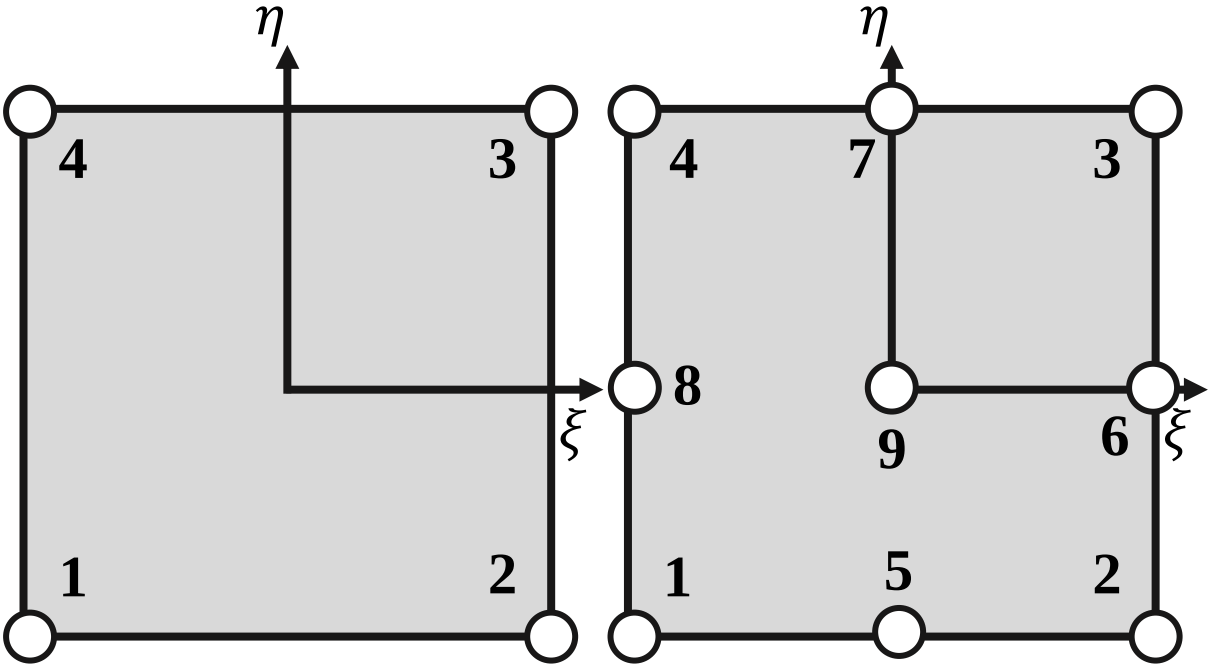
where $\varphi_{e,j}$ are geometry approximation functions. The finite element formulations can be classified as (REDDY, 2015):

1. Superparametric: $\varphi_{e,j}$ is represented by a higher-order function than $\psi_{e,j}$.
2. Isoparametric: $\varphi_{e,j}$ and $\psi_{e,j}$ are represented by functions of the same degree.
3. Subparametric: $\varphi_{e,j}$ is represented by a lower-order function than $\psi_{e,j}$.

The higher-order methods, such as the ones applied in Lethe, are the ones using functions of order greater than 2 to represent the geometry approximations $\varphi_{e,j}$. The use of higher-order elements allows for a virtually finer resolution in space (ELMAN *et al.*, 2014; REDDY, 2015). In Lethe, these functions are Lagrange polynomials. The order of the scheme n is given by the degree of the polynomial p such that $n = p + 1$. Tensor elements (quadrilaterals in two-dimensional meshes and hexahedra in three-dimensional) are represented by Qp.

For instance, a quadrilateral element of any geometric proportion can be transformed into rectangular finite elements. Examples of transformed elements of orders two (Q1) and three (Q2) are presented in Figure 2.7 (REDDY, 2015), in which ξ and η represent arbitrary coordinates. Given $\boldsymbol{x}$ in 2D $(= (x, y))$, the equivalent pair $-1 \leq (\xi, \eta) \leq$ can be established such that:

$$x = \sum_{j=1}^{N_{nodes}} x_{e,j}\varphi_{e,j}(\xi,\eta), y = \sum_{j=1}^{N_{nodes}} y_{e,j}\varphi_{e,j}(\xi,\eta) \tag{2.71}$$



(a) Q1 finite element

(b) Q2 finite element.

Figure 2.7 – Schematic representation of Q_p finite elements. Adapted from Reddy (2015).

By taking the Jacobian matrix J between the pairs (x, y) and (ξ, η) and applying the chain rule to the weak formulation, the entire problem will be expressed in terms of approximation functions $\psi_{e,j}$ and $\varphi_{e,j}$. Conveniently, integrals over rectangular (or hexahedron) master elements with side 2 can be evaluated by the Gaussian quadrature rule (REDDY, 2015). In practice, the system for the finite element e is assembled such that:

$$\mathbf{A}_e \equiv \mathbf{G}_e = \int_{\hat{\Omega}_e} G_{e,ij} d\boldsymbol{\xi} \tag{2.72}$$

where $\boldsymbol{\xi}$ is the transformation vector, $\hat{\Omega}_e$ is the transformed finite element, and $\mathbf{G}_e$ is the new coefficient matrix. For a direction ξ, the Gaussian quadrature formula is:

$$\int_{\hat{\Omega}_e} G d\xi = \int_{-1}^{1} G d\xi \approx \sum_{J=1}^{N_q} G W_J \tag{2.73}$$

where N_q is the number of quadrature points ($= int\left[(p+1)/2\right]$) represented by the white numerated dots in Figure 2.7 and W_J are the quadrature weights, which are tabulated. After solving the resulting linear system, the results are transformed back to the incognito base $\boldsymbol{u}$.

2.2.3.6 Solvers

For a steady-state solution, $\dot{\mathbf{u}}$ vanishes, resulting in a simpler linear system. Under this hypothesis, $\mathbf{u}$ is determined by applying linear solvers such as Gaussian elimination.

Nonetheless, the method is computationally expensive, in general, $\mathcal{O}\left(n^3\right)$ with n unknowns. On the other hand, $\mathbf{A}$ is a sparse matrix, that is, several of its entries are null. This feature allows for faster iterative methods that are more computationally efficient than the Gaussian elimination. For instance, in this work, the Generalized Minimum Residual Method (GMRES) method (SAAD; SCHULTZ, 1986) was applied, using the incomplete LU factorization (ILU) as the preconditioner. GMRES is very popular for asymmetric sparse matrices arising from PDEs (ZOU, 2023). For further details, the reader is referred to the books by Saad (2003) and Elman *et al.* (2014).

For a transient problem, the integration in time is most frequently done using the α-family of approximation (REDDY, 2015). This method consists of expressing solutions for consecutive time steps t and $t + \Delta t$ by[19]:

$$(1-\alpha)\dot{\boldsymbol{u}}^t + \alpha\dot{\boldsymbol{u}}^{t+\Delta t} \approx \frac{\boldsymbol{u}^{t+\Delta t} - \boldsymbol{u}^t}{\Delta t} \text{ for } 0 \leq \alpha \leq 1 \tag{2.74}$$

Isolating $\boldsymbol{u}^{t+\Delta t}$ in Eq. (2.74) leads to:

$$\boldsymbol{u}^{t+\Delta t} = \boldsymbol{u}^t + \Delta t\left[(1-\alpha)\dot{\boldsymbol{u}}^t + \alpha\dot{\boldsymbol{u}}^{t+\Delta t}\right] \text{ for } 0 \leq \alpha \leq 1 \tag{2.75}$$

Different values for α yield different α-family of approximation schemes. The names of the schemes, their numerical stability, and their order of accuracy in time are (REDDY, 2015):

$$\alpha = \begin{cases} 0 \text{ - forward difference scheme - conditionally stable - } \Delta t \\ 1/2 \text{ - Crank-Nicolson scheme - stable - } \Delta t^2 \\ 2/3 \text{ - Galerkin method - stable - } \Delta t^2 \\ 1 \text{ - backward difference scheme - stable - } \Delta t \end{cases} \tag{2.76}$$

In this work, the backward difference scheme (BDF) was applied. Since BDF are implicit schemes, it is unconditionally stable (HAY *et al.*, 2015). Explicit schemes, otherwise, need to satisfy the Courant–Friedrich–Lewy (CFL) (COURANT *et al.*, 1967) condition for stability (LI, 2006):

$$CFL = \frac{|\boldsymbol{u}|\,\Delta t}{|\Delta \boldsymbol{x}|} \leq CFL_{max} \tag{2.77}$$

where $CFL_{max} = 1$ is typically used. Higher-order backward difference schemes can be achieved using the solutions before the immediate previous time step. The time step Δt can be adaptive using the CFL criterion in Eq. (2.77).

[19] This method yields the definition of the finite differences method.

2.2.3.7 Weak form of the Navier-Stokes equations

The weak form of the Navier-Stokes equations can be expressed by (ELMAN *et al.*, 2014)[20]:

$$\int_\Omega q\,(\nabla\cdot\boldsymbol{u})\;d\boldsymbol{x} = 0 \tag{2.78a}$$

$$\nu\int_\Omega \nabla\boldsymbol{u} : \nabla\boldsymbol{w}\,d\boldsymbol{x} + \int_\Omega (\boldsymbol{u}\cdot\nabla\boldsymbol{u})\cdot\boldsymbol{w}\,d\boldsymbol{x} - \int_\Omega p\,(\nabla\cdot\boldsymbol{w})\;d\boldsymbol{x} - \int_\Omega (\boldsymbol{f}\cdot\boldsymbol{w})\;d\boldsymbol{x} = 0 \tag{2.78b}$$

where ν $(= \mu/\rho_f)$ is the fluid's kinematic viscosity, $\boldsymbol{w}$ is the weight function vector related to the velocity vector $\boldsymbol{u}$ and q is the weight fuction related to pressure p, while $\boldsymbol{f}$ represents the body forces. Note that the choice of functions to represent q is limited by the existence of a meaningful value for $\int_\Omega q\,d\boldsymbol{x}$, that is, the Lebesque function space $\mathcal{L}^2$. The same constraint is applied to $\boldsymbol{w}$, but the function also needs to possess a generalized first derivative, in other words, be in the Sobolev function space $\mathcal{H}^1$ (ELMAN *et al.*, 2014; BLAIS *et al.*, 2020). In function notation, the spaces are defined as (REDDY, 2015):

$$\mathcal{L}^2(\Omega) = \left\{u : \int_{x_1}^{x_2} |u|\,dx < \infty\right\} \tag{2.79}$$

$$\mathcal{H}^1(\Omega) = \left\{u : u, \frac{\mathrm{d}u}{\mathrm{d}x} \in \mathcal{L}^2(x_1,x_2), x_1 < x < x_2\right\} \tag{2.80}$$

2.2.3.8 Galerkin methods, stabilization strategy and turbulence

According to Tezduyar (1991), the computation of the solution for the incompressible Navier-Stokes equations suffers from severe instability. The author points out that the two main sources of instability are the advection term and the wrong choice of interpolation functions, generating oscillations primarily in the velocity and in the pressure field, respectively. The first source is potentiated in the case of advection-dominated flows due to the rapid changes in the velocity profile at small scales. This happens due to the assembly of the Eqs. (2.78) leads to a saddle-point problem (LARSON; BENGZON, 2013).

Li (2006) proposes that a suitable alternative to remedy the instability is to use a discontinuous Galerkin method. Opposed to the continuous approach, the method consists of relaxing the continuity at the boundaries of the elements, such that the velocity propagated from one finite element to the other as if the flux was being resolved at the boundaries. In essence, the same node j would have two solutions $\boldsymbol{u}$:

$$\boldsymbol{u} \approx \begin{cases} \boldsymbol{u}_{e,j}^{+} = \lim\limits_{\boldsymbol{x}\downarrow\boldsymbol{x}_j^{+}} \boldsymbol{u}_{e,j}^{h}(\boldsymbol{x}) \\ \boldsymbol{u}_{e,j}^{-} = \lim\limits_{\boldsymbol{x}\uparrow\boldsymbol{x}_j^{-}} \boldsymbol{u}_{e,j}^{h}(\boldsymbol{x}) \end{cases} \tag{2.81}$$

[20] The : symbol represents the double dot product operation. Given the tensors $\boldsymbol{U}$ and $\boldsymbol{T}$, $\boldsymbol{U} : \boldsymbol{T} = T_{..ij}U_{ji...}$. The operation is equivalent to $\otimes$ in the case of second-rank tensors (vectors).

where the superscript h denotes a space function. The weakly imposed continuity throughout the elements would be resolved by embedding continuous discretization methods (FDM, FEM, or FVM).

Instead, the method of choice in Lethe is the continuous Galerkin method. The advantage is that the continuity is strongly imposed throughout the finite elements. Nevertheless, stabilization strategies and the proper choice of the velocity and pressure finite element spaces need to be adopted to remedy the oscillations due to the advection term and ensure that the Ladyzhenskaya–Babuska–Brezzi (LBB) inf–sup condition is met (BLAIS *et al.*, 2020). The stabilization strategies adopted in Lethe are the Galerkin/least-squares (GLS), the streamline-upwind/Petrov-Galerkin (SUPG), the pressure-stabilizing/Petrov-Galerkin (PSPG) formulations (TEZDUYAR, 1991; BLAIS *et al.*, 2020; LETHE, 2023), and the grad-div block formulation (HEISTER; RAPIN, 2013).

For brevity, we refer the reader to Tezduyar (1991), Roos *et al.* (1996), and Blais *et al.* (2020) for further information in the GLS, SUPG, and PSPG terms, and to Heister and Rapin (2013) for the grad-div block formulation[21]. Here, we bring brief remarks about each of them (REDDY; GARTLING, 2010):

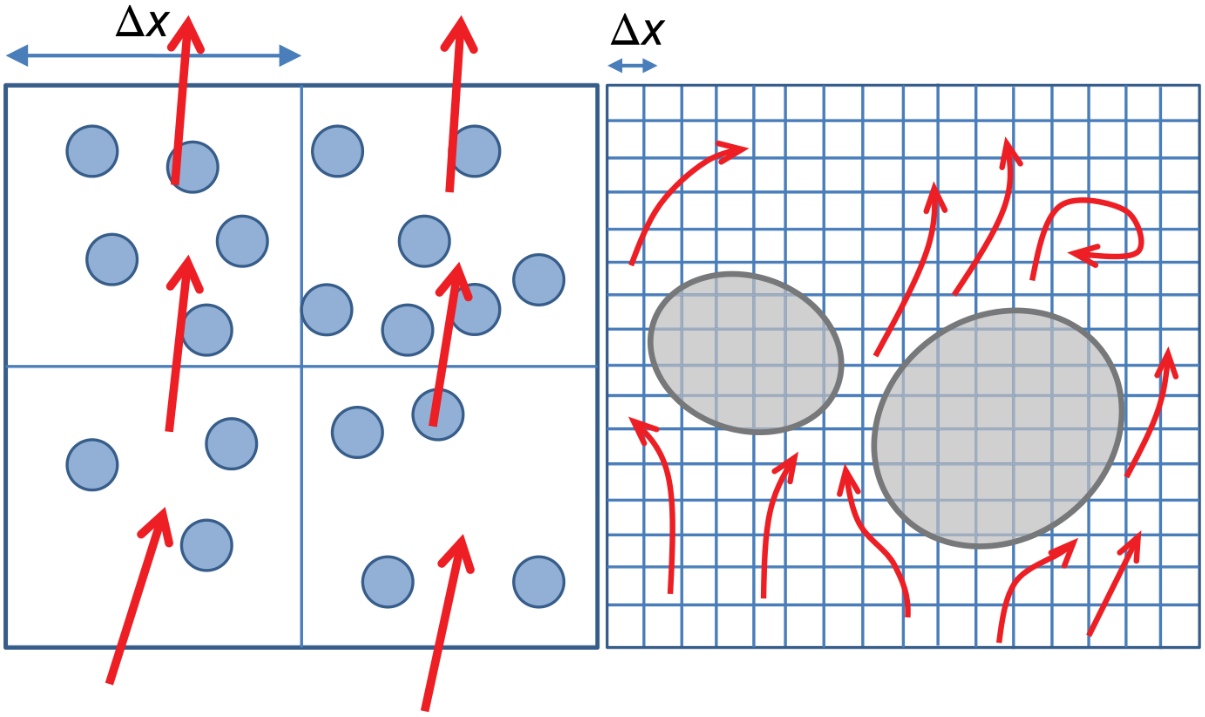
1. SUPG: Responsible for preventing oscillations on the velocity field due to high advection-driven gradients at boundary layers.
2. PSPG: Allows for pressure stabilization with equal order elements used for pressure and velocity by relaxing the LBB inf-sup condition (such as Q1-Q1, Q2-Q2, ...).
3. grad-div block stabilization: Helps prevent oscillations in the velocity field in advection-dominated problems by improving mass conservation (HEISTER; RAPIN, 2013).
4. GLS: This is a generalization of the SUPG/PSPG formulation using the residuals originating from the application of the Least-Squares FEM approach. It implies the addition of weighted element residual (strong residual) back to the momentum and continuity equations, helping with mass conservation.

The application of the stabilization strategy results in an alleviation of local gradients at the elements' scale, introducing a local "viscous-like" dissipation to the fluid field. This virtual increment in the local viscosity mimics turbulence inside the elements. Hence, high-order models are notably appropriate for LES due to their low dissipation (BLAIS *et al.*, 2020). The stabilization-based implicit LES (ILES) approach implemented in Lethe has been validated by Saavedra *et al.* (2022) for flow over periodic hills. The authors also point out that the approach can reduce computational efforts by allowing for larger mesh elements compared to traditional LES results.

[21] Lethe's documentation (LETHE, 2023) also has a brief explanation on the stabilization terms.

2.2.4 CFD-DEM coupling

The CFD-DEM coupling comprises two coupling strategies, named resolved and unresolved CFD-DEM coupling. Although they are of the same family, they imply completely different approaches. A schematic representation of these methods is presented in Figure 2.8.



(a) Unresolved CFD-DEM

(b) Resolved CFD-DEM

Figure 2.8 – Schematic representation of the CFD-DEM methods. In the figure, the fluid flow is represented by the red arrows. Adapted from Norouzi *et al.* (2016).

In the resolved CFD-DEM method, the momentum coupling between the phases is done by imposing the no-slip boundary condition at the particles' surface. The fluid field, modeled by the Navier-Stokes equations, is finely resolved around the particles. The microscale resolution of the fluid field results in accurate fluid-particle interactions, implicitly including liquid-bridge effects and fluid-particle forces without explicit detachment between phenomena. Additionally, the method allows for a better approach to turbulence. Nonetheless, to this moment, the method's need for a fine space discretization imposes limits on its application to fluid-particle flows with thousands of particles due to the high computational cost (BéRARD *et al.*, 2020).

Diversely, the unresolved CFD-DEM method consists of discretizing the fluid field into broader regions where fraction-weighted properties are represented as volume averages by the VANS equations (explained in Section 2.1.3). The coupling between phases is done explicitly and each fluid-particle interaction is calculated using individual expressions (as in Eq. (2.38)). Most of the interphase forces are expressed by correlations developed using experimental data (GIDASPOW, 1994; FELICE, 1994) or resolved CFD-DEM methods (BEETSTRA *et al.*, 2007; RONG *et al.*, 2013). The coarser discretization of the fluid field and the correlation-based coupling entail imprecision compared to resolved CFD-DEM methods (BéRARD *et al.*, 2020). Nonetheless, the computational efficiency is significantly enhanced, allowing for simulations of industrial-scale equipment with several hundreds of thousands of particles. Additionally, it provides higher accuracy than Eulerian-Eulerian methods such as TFM due to the higher resolution of the granular behavior and the fact that the Lagrangian frame is applied (NOROUZI *et al.*, 2016; BéRARD *et al.*, 2020).

In this work, we applied the unresolved CFD-DEM method to the representation of the liquid-solid fluidized bed. For this reason, further details on its appliance's practical aspects and limitations are provided. Additionally, since the resolved CFD-DEM coupling will no longer be discussed until the end of this section, for brevity, the term "CFD-DEM" will imply unresolved CFD-DEM.

2.2.4.1 Overview of the algorithm

Figure 2.9 depicts the algorithm of the unresolved CFD-DEM solver in Lethe (GEITANI *et al.*, 2023b).

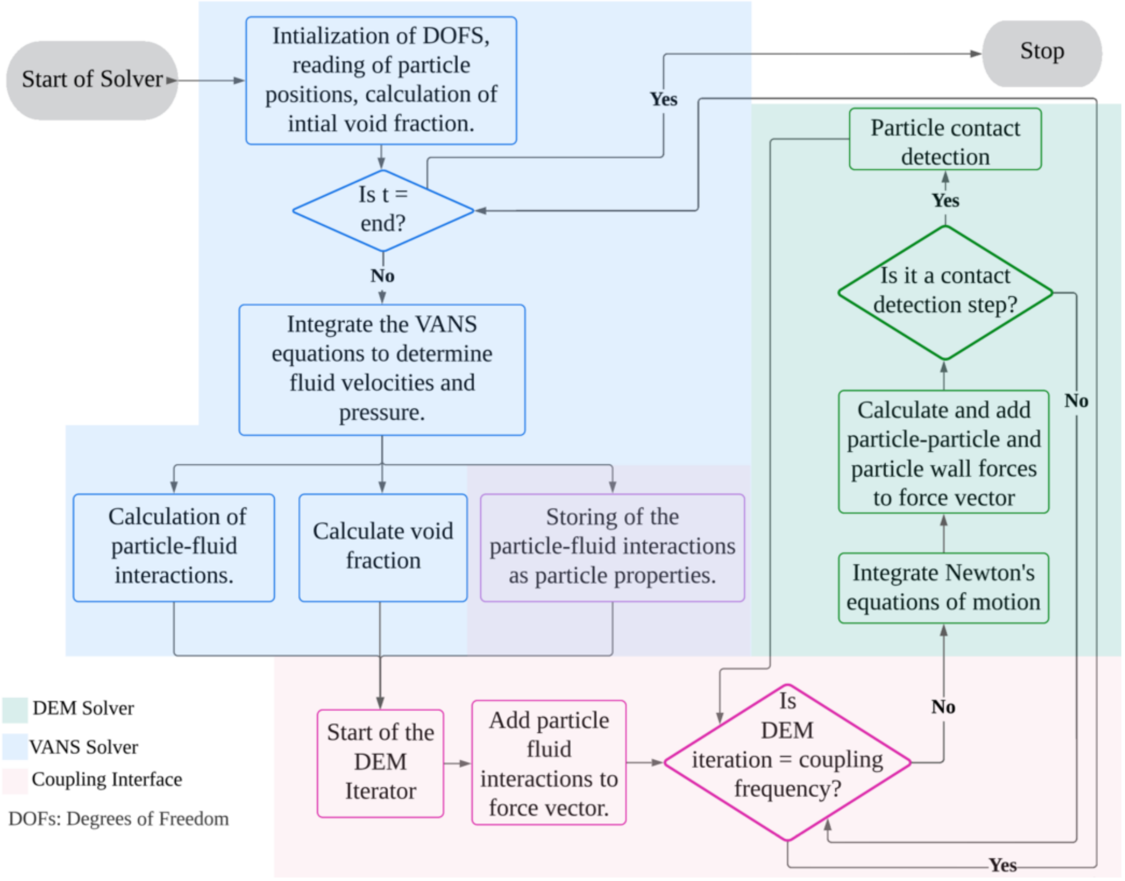

Start of Solver
Intialization of DOFS, reading of particle positions, calculation of intial void fraction.
Stop
Yes
Is t = end?
No
Integrate the VANS equations to determine fluid velocities and pressure.
Calculation of particle-fluid interactions.
Calculate void fraction
Storing of the particle-fluid interactions as particle properties.
Particle contact detection
Yes
Is it a contact detection step?
No
Calculate and add particle-particle and particle wall forces to force vector
Integrate Newton's equations of motion
No
Start of the DEM Iterator
Add particle fluid interactions to force vector.
Is DEM iteration = coupling frequency?
Yes
DEM Solver
VANS Solver
Coupling Interface
DOFs: Degrees of Freedom

Figure 2.9 – Unresolved CFD-DEM algorithm. Reproduced from Geitani *et al.* (2023b).

Before the CFD-DEM simulation is started, particles need to be inserted into the domain. In Lethe, this is done with a pure DEM simulation. The method starts with reading the particles' position and calculating the void fraction. The VANS equations are then integrated in space. With information from the fluid field, the particle-fluid interactions are calculated. The interactions are incorporated into the DEM and the steps of the DEM (Figure 2.6) are followed. Comments on each of the most important steps are provided.

2.2.4.2 Void fraction calculation

The determination of the local void fraction (ε_f) is an essential step. This parameter is present in the VANS equations and the fluid-particle force expressions (see Section 2.1), impacting mass conservation and the precision of the interphase forces. Void fraction is explicitly determined by calculation schemes based on the particles' position relative to the mesh (BéRARD *et al.*, 2020). The scheme of choice needs to be suitable to the discretization method applied to the fluid field. Cell-centered FVM CFD-DEM solvers, for example, require schemes that calculate in-cell void fraction values. Nodal methods such as FEM need a projection step of the void fraction to the node (BéRARD *et al.*, 2020; GEITANI; BLAIS, 2023b).

According to Yang (2003a), the maximum voidage achieved after prolonged shaking is about 0.395. Consequently, fluid-particle force correlations that take the fluid portion into account are approximately limited to the interval $0.40 \leq \varepsilon_f \leq 1.00$. Since particles overlap in the DEM, it is important to use large cell elements to prevent unphysical coupling. As a drawback of the increase in the cell size, the fluid field becomes poorly represented. The limiting cell size for stability and accuracy depends on the particle and domain sizes, the void fraction scheme, the discretization method, the order of the mesh elements, the time step, and the integration methods. This number of factors and possible combinations is the reason behind the absence of a robust universal rule for the mesh-to-particle guideline.

The void fraction calculation schemes can be divided into two groups: analytical and non-analytical (NOROUZI *et al.*, 2016). Analytical methods are the ones that use trigonometric functions to determine the exact particle fraction in the subdomain. In the cell-centered FVM context, the analytical method by Peng *et al.* (2014) has demonstrated very good continuity of the fluid field. However, the method is limited to rectangular-based hexahedral structured meshes, requiring further adaptation for other cases. The method by Wu *et al.* (2009) suits unstructured pyramidal meshes. Besides the precision, this class of methods is computationally intensive, limiting their application to very small systems. More recently, Geitani and Blais (2023b) proposed a mesh-independent analytical method for FEM-DEM simulations that takes advantage of the nodal discretization of the

FEM to create a virtual spherical subdomain where the void fraction is calculated. Since determining the overlap between spheres is way simpler than the one between a sphere and a hexahedral and given the nodal nature, the method is very efficient compared to other analytical methods in the literature[22].

The family of non-analytical schemes comprises several methods. The vast majority of CFD-DEM works in the literature apply at least one of the methods in this class. By far, the particle centroid method is the most popular due to its simplicity. The PCM consists of two steps: count the number of particles with centroid in the CFD cell ($N_{p,Cell}$) and calculate (NOROUZI *et al.*, 2016):

$$\varepsilon_f = 1 - \frac{1}{V_{Cell}} \sum_{i=1}^{N_{p,\Omega_e}} V_{p,i} \tag{2.82}$$

Note that the method is highly discontinuous in time and space. Nonetheless, its implementation is straightforward and its computational cost is the lowest possible for a discrete method. Additionally, the discontinuity can be remedied if particles are uniformly distributed along the system and the mesh elements are considerably coarse compared to the particles. The rule of thumb for this method, proposed by Peng *et al.* (2014) and followed by most of the works using PCM (NOROUZI *et al.*, 2016; BéRARD *et al.*, 2020), is to use mesh cells with $S_c/d_p > 3 - 4$, where S_c is the mesh cells characteristic length given by:

$$S_c = \sqrt[3]{\bar{V}_{Cell}} \tag{2.83}$$

where $\bar{V}_{Cell}$ is the average volume among cells. One way to circumvent the discontinuity is numerically smoothing the void fraction. In Lethe, this is done by applying a length smoothing gradient factor on the $\mathcal{L}^2$ projection of the void fraction to the nodes (GEITANI *et al.*, 2023b). This strategy was adopted for all simulations in this work. Other non-analytical methods include the satellite-point method (BLAIS *et al.*, 2016), statistical methods based on probabilistic distribution functions, adaptations to the traditional PCM, and others (NOROUZI *et al.*, 2016; CLARKE *et al.*, 2018; BéRARD *et al.*, 2020).

2.2.4.3 Coupling

The coupling between phases is done through the terms $\boldsymbol{F}_{pf}$ in Eq. (2.28) and $\boldsymbol{f}_{pf}$ in Eq. (2.29), where $\boldsymbol{f}_{pf}$ is calculated by the Eq. (2.38) and the relation between the terms is given by Eq. (2.39). In practice, since all forces are explicitly calculated term by term, some of them can be neglected. The choice of the model will depend mainly on the characteristic porosity of the system. A general rule is given by Norouzi *et al.* (2016), that classifies the coupling strategies according to the criteria in Figure 2.10[23].

[22] Available in Lethe (LETHE, 2023).

[23] Although it is a popular classification, it is worth pointing out that other authors use the terms one-way and two-way coupling to all coupling strategies, removing the collisions from the coupling classification.

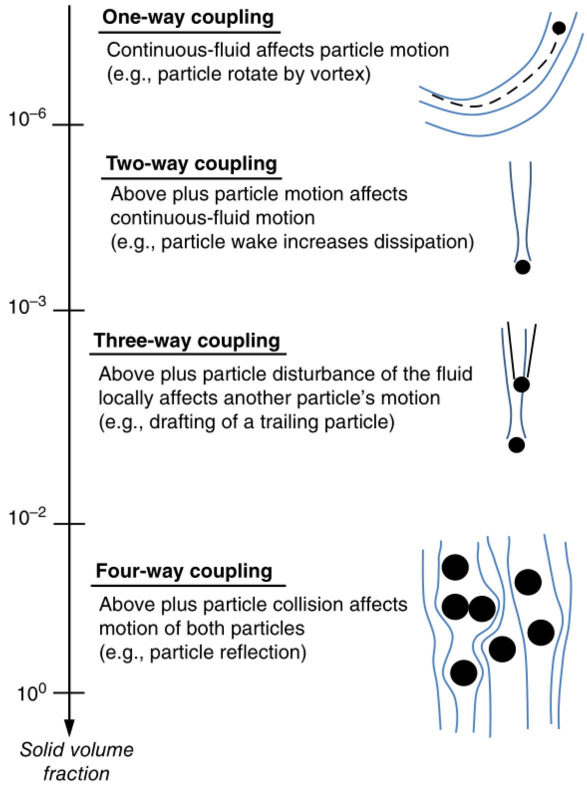

One-way coupling
Continuous-fluid affects particle motion
(e.g., particle rotate by vortex)
10^{-6}
Two-way coupling
Above plus particle motion affects
continuous-fluid motion
(e.g., particle wake increases dissipation)
10^{-3}
Three-way coupling
Above plus particle disturbance of the fluid
locally affects another particle's motion
(e.g., drafting of a trailing particle)
10^{-2}
Four-way coupling
Above plus particle collision affects
motion of both particles
(e.g., particle reflection)
10^{0}
Solid volume
fraction

Figure 2.10 – Coupling strategies. Reproduced from Norouzi *et al.* (2016).

In this classification, coupling strategies would imply:

1. One-way coupling: Forces applied to the particles due to the fluid motion are not applied back to the fluid. It is as if the fluid did not "feel" the presence of the particles. Additionally, collisions between particles are not considered, i.e., the motion of the particles is completely driven by the fluid motion. Very dilute systems with light particles transported by fluid motion can be simulated using this strategy since the pressure felt by the fluid due to the presence of the particles is negligible.

2. Two-way coupling: Both phases feel each other's presence. Still, the particle contact is neglected. Additionally, particle-induced fluid disturbance affects the fluid only. Dilute systems with heavy large particles being carried by the fluid traveling in the same direction can be simulated using this strategy.

3. Three-way coupling: Same as the two-way coupling, however, particle-induced disturbances on the fluid field also affect the particles' motion. Dilute systems with heavy large particles being carried by the fluid with considerable particle-induced turbulence, both traveling in the same direction, can be simulated with the three-way coupling scheme.

The present author agrees with this point of view but here the classic classification is presented.

4. Four-way coupling: Particle contacts are now considered. Suitable for concentrate contact-driven systems with non-negligible collision events. Fluidized beds, such as the one in this work, are usually simulated using this strategy.

Naturally, the greatest magnitude forces in fluidized beds are gravity and drag (YANG, 2003a). In gas-solid fluidized beds, the density of the fluid is so low that buoyancy can be (and is often) completely removed (HOEF *et al.*, 2006). Under this hypothesis, the simple and incomplete model type III in Zhou *et al.* (2010) can be used. Oppositely, the apparent density of the particles is considerably lower in liquids and buoyancy can no longer be left behind. Additionally, as explained in Section 2.1.3.3, if the VANS equations incorporate the hydrostatic pressure through the gravity term, buoyancy is directly incorporated into the pressure gradient force and an explicit expression is not required (ZHOU *et al.*, 2010).

The virtual mass force is often neglected for two main reasons: it is a source of instability due to the requirement for an implicit integration in time and space, and the frequent collisions in contact-driven systems. The implicit integration implies adding more non-linearity to the VANS equations. The frequent collisions are responsible for preventing particles from reaching unphysical high velocities that can be observed in dilute systems without virtual mass. An alternative approach to virtual mass is to use a factor of 0.5 on the density (HOEF *et al.*, 2006). The reasons for neglecting the Basset force are similar. The relaxation of the particle's history in time is considered nullified by the collisions in concentrated contact-driven systems (NOROUZI *et al.*, 2016).

Among the most important problems in CFD-DEM is the prediction of a physical particle rotation. The classical model for the particle-fluid coupling does not incorporate a torque dissipation mechanism. Consequently, the particles' angular velocity is contact-driven. However, this is not necessarily true for liquid-solid systems due to the viscous effect. In the absence of a model for such an effect, the Magnus force can be neglected to avoid an unphysical response (RENZO *et al.*, 2011; BLAIS *et al.*, 2016). The Saffman lift force is usually neglected as well under the hypothesis of a way lower magnitude compared to drag, nonetheless, further discussion on the subject is provided in Article 6.

Lastly, drag can be expressed implicitly or explicitly in the VANS equations. Implicit drag implies integrating the VANS momentum equations with the term in Eq. (2.43) in the present time. The relative velocity imposes a non-linear source term to the integration that can lead to numerical instability in the pressure field. Conversely, the pressure is more stable when the term is calculated to the present step based on the previous step's relative velocity. This was demonstrated by Blais *et al.* (2016) for the case of a fluidized bed. In Lethe, drag is explicit.

As shown in Figure 2.9, the fluid-particle forces are calculated once after a given number of DEM iterations. This is an efficient way of dealing with the fact that the DEM simulations require integration time steps way lower than CFD. The fluid-particle forces are stored and replicated while the CFD solver waits for the next step (NOROUZI *et al.*, 2016; BéRARD *et al.*, 2020; GEITANI *et al.*, 2023b).

2.2.4.4 Stabilized weak form of the VANS equations

The stabilized weak form of the incompressible VANS equations was derived by Geitani *et al.* (2023b), Geitani *et al.* (2023a). Given the mass accumulation term m', the strong form of the continuity can be expressed by:

$$\frac{\partial \varepsilon_f}{\partial t} + \varepsilon_f \nabla \cdot \boldsymbol{u} + \boldsymbol{u} \nabla \varepsilon_f = \frac{m'}{\rho_f} \tag{2.84}$$

Applying the accumulation term back in the momentum VANS equations, the authors obtained (GEITANI *et al.*, 2023a):

Continuity:

$$\int_\Omega \left(\frac{\partial \varepsilon_f}{\partial t} + \varepsilon_f \nabla \cdot \boldsymbol{u} + \boldsymbol{u} \nabla \varepsilon_f \right) q d\boldsymbol{x} + \underbrace{\mathcal{SR} \cdot (\tau_u \nabla q)\, d\boldsymbol{x}}_{\text{PSPG}} = 0 \tag{2.85a}$$

VANS model A:

$$\begin{aligned}
&\int_\Omega \left(\varepsilon_f \frac{\partial \boldsymbol{u}}{\partial t} + \varepsilon_f \boldsymbol{u} \nabla \cdot \boldsymbol{u} \right) \cdot \boldsymbol{w} d\boldsymbol{x} + \frac{1}{\rho_f} \left\{ \int_\Omega \left[\varepsilon_f \nu \left(\nabla^2 \boldsymbol{u} \right) + \nu \nabla \varepsilon_f \right] \cdot \boldsymbol{w} d\boldsymbol{x} + \right. \\
&+ \left. \int_\Omega m' \boldsymbol{u} \cdot \boldsymbol{w} d\boldsymbol{x} - \int_\Omega \left(\varepsilon_f p \cdot \nabla \boldsymbol{w} + p \nabla \varepsilon_f \cdot \boldsymbol{w} \right) d\boldsymbol{x} + \int_\Omega \frac{\boldsymbol{F}_{pf}^A}{V_\Omega} \cdot \boldsymbol{w} d\boldsymbol{x} \right\} + \\
&+ \underbrace{\sum_K \int_{\Omega_K} \mathcal{SR} \cdot (\tau_u \boldsymbol{u} \cdot \nabla \boldsymbol{w})\, d\boldsymbol{x}}_{\text{SUPG}} + \underbrace{\sum_K \int_{\Omega_K} \gamma \left[\frac{\partial \varepsilon_f}{\partial t} + \nabla \cdot (\varepsilon_f \boldsymbol{u}) \right] (\nabla \cdot \boldsymbol{w})\, d\boldsymbol{x}}_{\text{grad-div}} = 0
\end{aligned} \tag{2.85b}$$

VANS model B:

$$\begin{aligned}
&\int_\Omega \left(\varepsilon_f \frac{\partial \boldsymbol{u}}{\partial t} + \varepsilon_f \boldsymbol{u} \nabla \cdot \boldsymbol{u} \right) \cdot \boldsymbol{w} d\boldsymbol{x} + \frac{1}{\rho_f} \int_\Omega \left[\nu \left(\nabla \boldsymbol{u} \right) \nabla \boldsymbol{w} + m' \nabla \boldsymbol{u} \cdot \boldsymbol{w} + \right. \\
&\left. - p \nabla \cdot \boldsymbol{w} + \frac{\boldsymbol{F}_{pf}^B}{V_\Omega} \cdot \boldsymbol{w} \right] d\boldsymbol{x} + \underbrace{\sum_K \int_{\Omega_K} \mathcal{SR} \cdot (\tau_u \boldsymbol{u} \cdot \nabla \boldsymbol{w})\, d\boldsymbol{x}}_{\text{SUPG}} \\
&\underbrace{+ \sum_K \int_{\Omega_K} \gamma \left[\frac{\partial \varepsilon_f}{\partial t} + \nabla \cdot (\varepsilon_f \boldsymbol{u}) \right] (\nabla \cdot \boldsymbol{w})\, d\boldsymbol{x}}_{\text{grad-div}} = 0
\end{aligned} \tag{2.85c}$$

In the equations, the strong residual $\mathcal{SR}$ associated with the PSPG and SUPG terms represent the GLS stabilized terms:

$$\mathcal{SR} = \varepsilon_f \frac{\partial \boldsymbol{u}}{\partial t} + \varepsilon_f \boldsymbol{u} \nabla \cdot \boldsymbol{u} + \frac{1}{\rho_f} \left[m' \boldsymbol{u} + \nabla p - \nu \left(\nabla^2 \boldsymbol{u} \right) + \frac{\boldsymbol{F}_{pf}}{V_\Omega} \right] \tag{2.86}$$

ν $(= \mu/\rho_f)$ is the fluid's kinematic viscosity, $\boldsymbol{w}$ is the weight function vector associated to the velocity vector $\boldsymbol{u}$, and q is the weight function associated to pressure p. The stabilization parameter τ_u is calculated for the transient simulation by (TEZDUYAR; SATHE, 2003; BLAIS *et al.*, 2020; GEITANI *et al.*, 2023a):

$$\tau_u = \left[\left(\frac{1}{\Delta t_{CFD}} \right)^2 + \left(\frac{2\,|u|}{h_{conv}} \right)^2 + 9 \left(\frac{4\nu}{h_{diff}^2} \right)^2 \right]^{-1/2} \tag{2.87}$$

where Δt_{CFD} is the CFD time step, h_{conv} is the size of the element related to the convection transport, and h_{diff} is the size of the element related to the diffusion mechanism. In Lethe, both element sizes correspond to the diameter of a sphere of a volume equivalent to that of the cell (BLAIS *et al.*, 2020). Additionally, γ is the grad-div stabilization weight factor, given by (OLSHANSKII *et al.*, 2009):

$$\gamma = \nu + c * \boldsymbol{u}_{f_{\Gamma_e}} \tag{2.88}$$

where $c*$ depends on the pressure behavior.

2.3 Liquid-solid fluidized beds

In this section, concepts of liquid fluidization are introduced. Emphasis is given to fluid and particle dynamics since it is the main subject of the project. An overview of the trends in liquid-solid fluidized beds research is also provided.

2.3.1 Design of liquid-solid fluidized beds

Liquid-solid fluidized beds are simple equipment. The fluidized bed equipment can be divided into 4 components, namely: calming section, distributor, fluidization section, and outlet section (EPSTEIN, 2003b). In Figure 2.11, a schematic drawing of the structure of the liquid-solid fluidized bed (LSFB) is provided.

As gas-solid fluidized beds, LSFBs are equipment designed to maximize heat and/or mass exchange between the phases. To do so, the particles are percolated by a fluid with a velocity above the minimum fluidization (U_{min}, discussed in Section 2.3.2). The high relative velocity between particles and fluid agitates the particles and the bad is expanded. This drag-driven motion at a high fluid inlet velocity makes particles stay in contact with fresh influent throughout the operation, impacting positively the effectiveness of the convective exchange between phases while preventing flow defects such as dead zones and preferential paths. In addition, a remarkable difference between gas-solid and liquid-solid fluidized beds is that the latter presents a uniformly distributed bed of particles which makes it more predictable (EPSTEIN, 2003b).

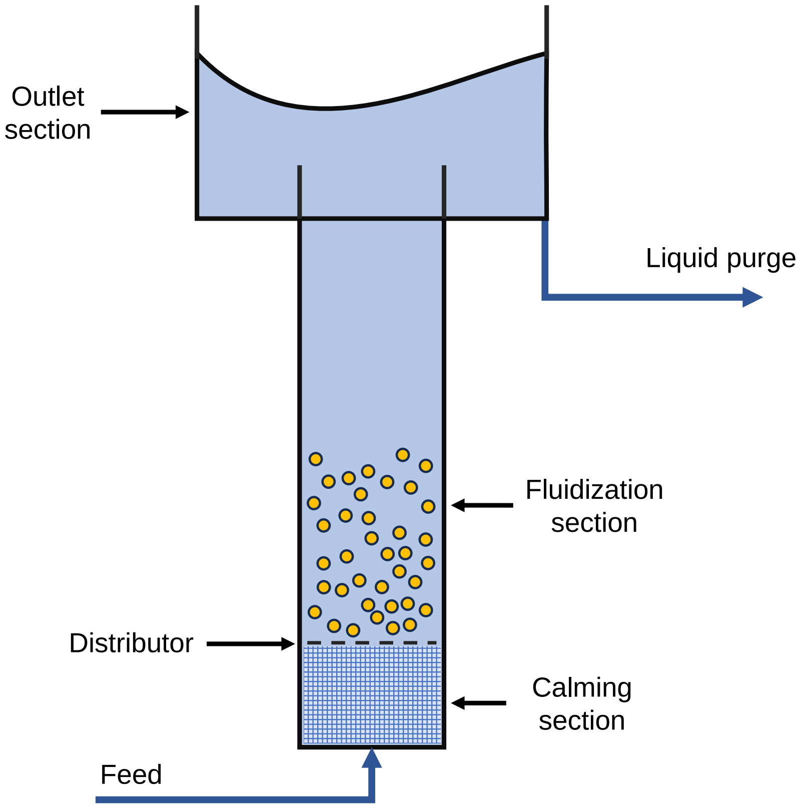


Figure 2.11 – Schematic representation of liquid-solid fluidized bed equipment.

The flow past a LSFB is controlled by a pump and valves. Before reaching the bed, the flow is uniformized by a calming section, which can be assembled using a fixed bed of particles, saddles, rings, or flow straighteners. The distributor, positioned right above the calming section, is designed to physically separate the calming section from the fluidization section. It is a mesh-like structure with orifices small enough to hold particles above it and large enough so that the flow is not interrupted by it. Past the distributor, the flow percolates the bed of particles at the fluidization section. After this, the liquid is purged out of the equipment at the outlet section, which can be closed or opened to the environment. For opened outlets, the diameter of the section needs to be sufficiently large to prevent the fluid from flowing out of the equipment. After leaving the outlet section, the liquid can be purged (one-time operation) or directed back to the same equipment (recirculating LSFB) (EPSTEIN, 2003b).

The particles and the fluid used in LSFBs will depend on the application. Further discussion on LSFB applications is presented in Section 2.3.3

2.3.2 Minimum fluidization, terminal velocity, and Pressure drop

LSFBs operate at a flow inlet velocity between the minimum fluidization (U_{min}) and the particle's terminal velocity (U_0). The terminal velocity is the maximum relative velocity achieved by a free-falling body immersed in a fluid. Starting from rest, the body stops accelerating because the vertical resultant force is null, i.e., buoyancy and drag fully compensate for the gravity force. The terminal velocity can be estimated for know-shape materials using drag correlations for the creeping flow or direct correlations in the literature. Lopes *et al.* (2018), for example, obtained a good fit between experimental data and the correlation by Turton and Clark (1987) on the prediction of terminal velocity of spherical particles. The direct correlations are a function of the characteristics of the fluid and the body, usually given by the Archimedes number (YANG, 2003b).

Starting from rest, the minimum fluidization is achieved when a bed of particles starts to expand due to a gradual increase in the inlet velocity. From this moment, the drag force is strong enough to overcome the apparent mass of the bed $(N_pV_p(\rho_p - \rho_f)g)$. Visualizing this state is not easy, however, it is identifiable by the effect of the pressure drop between the bottom and the top of the fluidization section. From a force balance in a packed bed of uniformly distributed particles in the vertical direction z, one can derive the following expression for pressure p (EPSTEIN, 2003b):

$$-\frac{\mathrm{d}p}{\mathrm{d}z} = \left(1 - \bar{\varepsilon}_f\right)\left(\rho_p - \rho_f\right) g \tag{2.89}$$

where g is the norm of the gravity acceleration vector $\boldsymbol{g}$ and $\bar{\varepsilon}_f$ is the average void fraction of the bed.

Between the states of complete rest and minimum fluidization, the pressure drop due to the fixed bed of particles $-\Delta p$ is given by Ergun's equation (ERGUN, 1952). For spherical particles:

$$-\frac{\mathrm{d}p}{\mathrm{d}z} = \frac{150U\mu\left(1 - \bar{\varepsilon}_f\right)^2}{d_p^2\bar{\varepsilon}_f} + \frac{1.75U^2\rho_f(1 - \bar{\varepsilon}_f)}{d_p\bar{\varepsilon}_f} \tag{2.90}$$

where U is the inlet velocity of the fluid. The minimum fluidization velocity can be calculated by passing all terms in Eq. (2.90) to the right-hand side and substituting $-\frac{\mathrm{d}p}{\mathrm{d}z}$ by the right-hand side of Eq. (2.89). In the resulting equation, the minimum fluidization bed voidage $\bar{\varepsilon_f}_{min}$ for spherical particles is approximated to 0.415 (CHEN, 1987). The resulting quadratic equation can be resolved and the single meaningful root is going to be the minimum fluidization velocity (EPSTEIN, 2003b).

In the fluidization region, since the mass of particles is constant, the pressure drop $-\Delta p$ does not vary with the inlet velocity. It is one of the great advantages of fluidization over fixed or packed beds. This can be proved by integrating the Eq. (2.89) from the bottom ($z = 0$) to the top of the bed of particles ($z = H_b$):

$$-\Delta p = H_b(1 - \bar{\varepsilon}_f)(\rho_p - \rho_f) \tag{2.91}$$

where H_b is given by:

$$1 - \bar{\varepsilon}_f = \frac{M_p}{\rho_p A_b H_b} \tag{2.92}$$

where M_p is the total mass of particles and A_b is the cross-section area of the bed region. Combining Eqs. (2.91) and (2.92), we have (EPSTEIN, 2003b):

$$H_b(1-\bar{\varepsilon}_f) = \frac{-\Delta p}{(\rho_p - \rho_f)g} = \frac{M_p}{\rho_p A} = \text{Constant} \tag{2.93}$$

Interestingly, the fluidization and defluidization curves behave slightly differently. The reason for this is the inertial effect. Taking particles from rest to the fluidization state demands more energy than doing the reverse. Hence, the characteristic $-\Delta p$ as a function of U curve, can be represented by Figure 2.12 (EPSTEIN, 2003b):

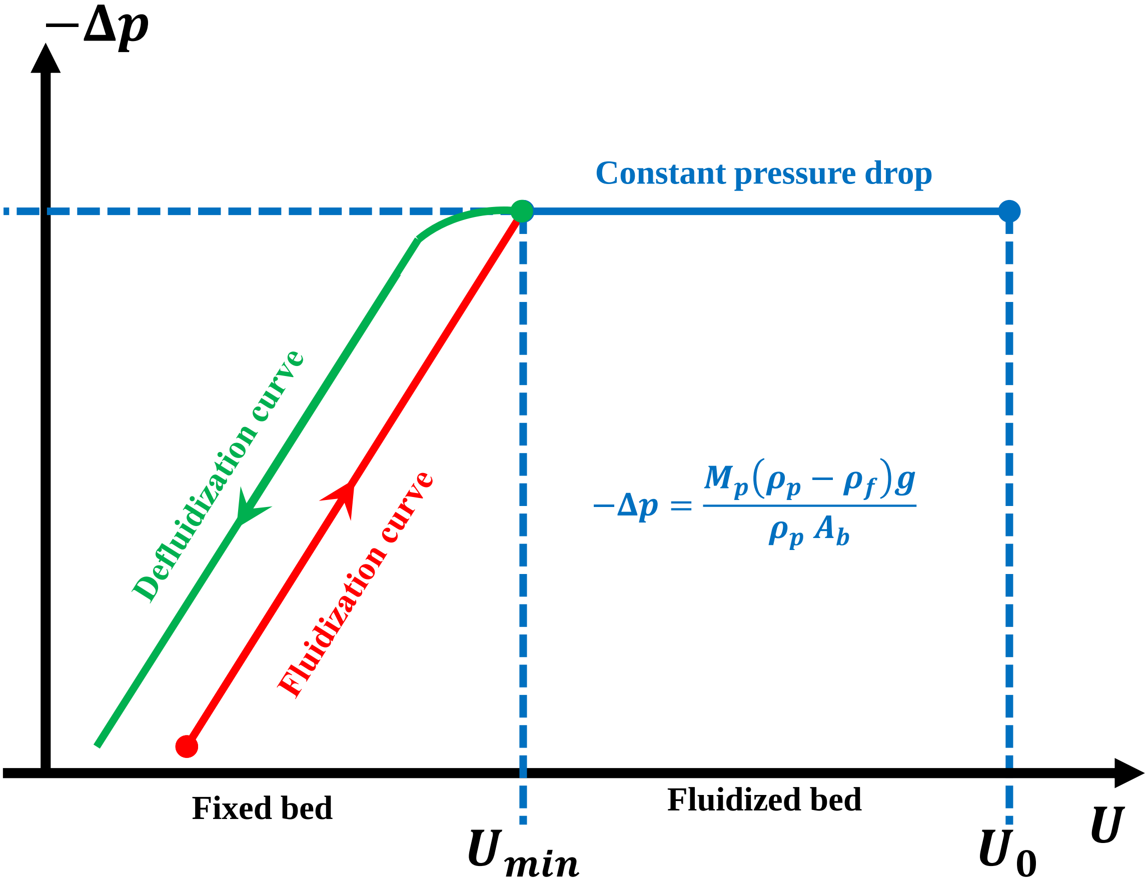


Figure 2.12 – Schematic representation of the fluidization and defluidization curves.

2.3.2.1 Bed expansion

As said, in LSFBs particles are uniformly dispersed throughout the bed. Hence, the pressure profile along the bed height has a linear behavior. Additionally, if the inlet velocity is kept constant, the bed height will reach a pseudo-steady state. Consequently, $-dp/dz$ is constant at the bed region. From these assumptions, it is possible to determine the bed height using Eq. (2.89) by measuring the pressure drop along z. Since bed height has an inverse relation with $\bar{\varepsilon}_f$ (Eq. (2.92)), the bed expansion can be conveniently expressed by the bed void fraction (EPSTEIN, 2003b).

In the absence of experimental data, the pressure drop can only be estimated. The straightforward way of doing so is to use correlations in the literature. The most popular among them is the Richardson-Zaki equation (RICHARDSON; ZAKI, 1954). The correlation arises from the same hypothesis adopted by Felice (1994) and Rong *et al.* (2013) on the development of a correlation for the drag force, i.e., that drag in a bulk granular material has an exponential relation with the average porosity. In its complete form, the Richardson-Zaki equation (or R-Z) is expressed by (RICHARDSON; ZAKI, 1954):

$$\frac{U}{U_0} = k\bar{\varepsilon}_f^n \tag{2.94}$$

where n is the R-Z parameter and k is a correction parameter originally accounting for wall effects. To estimate $\bar{\varepsilon}_f$, the terminal velocity (measured or estimated) needs to be provided. The parameter indirectly accounts for characteristics of the creeping flow. n can be calculated from empirical correlations in the literature. There is good agreement in the literature about the precision of the correlation by Khan and Richardson (1989), which is given by:

$$\frac{4.8 - n}{n - 2.4} = 0.043\mathrm{Ar}^{0.57} \tag{2.95}$$

The advantage of Eq. (2.95) over other correlations is that it is explicit in the number of Archimedes, which, as opposed to correlations based on the terminal Reynolds number Re_0 ($= d_p U_0 \rho_f/\mu$) does not depend on the terminal velocity of the particles (EPSTEIN, 2003b). Conversely, the literature diverges on the determination of k. Most of the literature uses a value of $k = 1$. Discussions about this are provided in the Article 5.

The precise prediction of bed voidage is key to safely applying LSFBs in liquid-particle processes. It is among the crucial variables to be determined during the design because physical and chemical reactions are dependent on it. For instance, according to Epstein (2003b), the heat transfer between phases in a pseudo-steady state LSFBs is given by:

$$\rho_f c_p U \frac{\mathrm{d}T}{\mathrm{d}z} = (\lambda + \lambda_{eddy})\frac{\mathrm{d}^2T}{\mathrm{d}z^2} + h_h S_v (T_p - T) \tag{2.96}$$

where c_p is the specific heat capacity of the bulk, T is the temperature of the bulk, T_s is the temperature of the solid, h_h the heat-transfer coefficient, λ is the thermal conductivity of the particles, λ_{eddy} is the interphase thermal conductivity and S_v is the spatial characteristic of the bulk, given by:

$$S_v = 6\frac{(1 - \bar{\varepsilon}_f)}{d_p} \tag{2.97}$$

Similarly, the mass transfer between phases of a species A at a molecular concentration at the bulk of C_A is given by:

$$U\frac{\mathrm{d}C_A}{\mathrm{d}z} = D_A\frac{\mathrm{d}^2C_A}{\mathrm{d}z^2} + h_m S_v (C_{A,p} - C_A) \tag{2.98}$$

where S_v is defined by Eq. (2.97), D_A is the molecular diffusivity, and h_m is the mass-transfer coefficient. Note that $\bar{\varepsilon}_f$ is present in both equations at the predominant convection term regardless of the fluid and solid pair.

2.3.3 Remarkable applications and trends in liquid fluidization

LSFBs can be applied to a wide range of processes. Some of them are mentioned in this section (EPSTEIN, 2003b; EPSTEIN, 2003a).

2.3.3.1 Particles classification

According to Epstein (2003b), it is a more than a century-old application. As shown, the Archimedes number of a fluid-solid pair has a direct impact on its bed expansion response to the inlet flow rate. Hence, classifying particles with different Archimedes numbers is a rather trivial operation.

2.3.3.2 Backwashing

This application includes any sort of recovery of inert particles by exposing them to intensive agitation in fresh water. Wastewater particulate filters benefit from this method since the agitation and collisions help with the removal of the softer material from the hard particles' surface (CLEASBY *et al.*, 1977). More recently, Aouf and Dounit (2023) used this technique to remedy the toxicity of hydrocarbon-contaminated soil.

2.3.3.3 Crystallization

Seeded crystal growth consists of inducing the crystallization in a metastable supersaturated solution over a smaller crystal (seed) of the saturated substance. This process benefits from the agitation in LSFBs because the kinetic energy breaks the metastability condition. At the same time, an *in-situ* size classification inherent to the fluidization process helps achieve the desired crystal size. Gui *et al.* (2022) presents several applications of crystallization in liquid-solid fluidization, ranging from inorganic salts to organic acid crystals, in various industry segments, from effluent treatment to pharmaceutical.

2.3.3.4 Adsorption

LSFBs can be very effective in the removal of easy-adsorption contaminants in effluents. The mass exchange coefficient benefits from the high relative velocity between phases. For fast removal operations, the LSFBs allow for higher fluid flow rates than packed beds while not increasing the pressure drop. Lv *et al.* (2021) used this method to effectively remove copper from wastewater using a bed of active coal. The authors demonstrate that the technique was more effective than the classical packed bed.

2.3.3.5 Ion exchange

Ion exchange works similarly to adsorption. It consists of removing ions from the liquid through higher chemical affinity with the particles. Ion exchange-packed bed columns used in deionization processes can also be washed out with distilled water using LSFBs. According to Amirtharajah (1978), the optimal conditions for this operation are at $\bar{\varepsilon}_f = 0.65 - 0.7$.

2.3.3.6 Flocculation and floculation-coagulation

Flocculation consists of separating solid coagulated material from liquid. Fluidization can help separate solids with different diameters and facilitate the induced coagulation of colloidal agglomerates. For instance, Cheknane *et al.* (2005) used LSFBs for flocculation-induced coagulation intensification to remove organic matter from seawater. The authors report a 70% removal on the total pollutant with a residence time way lower than the classic batch operation.

2.3.3.7 Electrolysis

Electrolysis consists of inverting the natural oxidation-reduction reaction direction by applying an electrical current between the reactants. Fluidized bed electrodes have been used to enrich zinc, copper, nickel, cobalt, and other metallic material particles (CHENG *et al.*, 2020). The process benefits from the uniform distribution of temperature and the high mass exchange due to the agitation and the high flow rates in LSFBs. Although FBEs have many advantages, the electrical conductivity of particulate electrodes needs improvement.

2.3.3.8 Heat exchange and thermal energy storage

Naturally, the convective effect of the particles' agitation can be applied to enhance the interphase heat exchange. One special application of this feature that has been attracting attention is the combination of phase change materials encapsulation with fluidized beds for a more efficient energy storage process. In the operation, encapsulated phase change materials are used to store energy due to their high latent heat capacity.

The main application of these materials is in the solar energy industry as a sustainable alternative to regular batteries. Although LSFBs have been tested for this application, gas-solid fluidized beds are still a preferable alternative (ALMENDROS-IBáñEZ *et al.*, 2019). However, there is an important gap in the literature.

2.3.3.9 Bioreactors and wastewater treatment

Bioreactors and wastewater treatment are the most important applications of the technology at this moment. LSFBs have been extensively used as aerobic and anaerobic bioreactors, ranging from wastewater treatment applications to alcohol production (EPSTEIN, 2003a; EPSTEIN, 2003b). The most prominent among those applications is wastewater treatment through digestion. In this application, carrier particles made of any inert material are coated with a biofilm composed of bacteria, particulates, extracellular polymers, and gels growing on a support media (NELSON *et al.*, 2017). The bacteria feeds itself on the organic nutrients in the wastewater, breaking down the complex large compounds into more soluble molecules. The advantage of this process as opposed to the classic strong acid or strong base exposure is that the effluent is not as aggressive. Furthermore, according to Fraia *et al.* (2018), wastewater treatment plants can be highly energy-consuming industries. Since the process does not add value to any product, the industrial interest is very low. However, anaerobic digestors produce biogases that can be used as fuel, which can be sold or reused. The anaerobic bioreactors can decrease external energy consumption since the produced fuel can be used in energy generators, making the process cheap and self-sustainable (NADAIS *et al.*, 2011; JAMALI *et al.*, 2017).

The fluidization technology helps intensify the process because the mass transfer between the fresh influent and the bioactive particles is higher compared to slurry or fixed bed bioreactors. The reason for this is the higher nutrient removal through the enhanced exchange momentum between phases and the interactions of the particles with themselves and the equipment walls. In such a process, the composition of the products can be controlled by varying the microorganisms and the influents. Wastewater treatment with biogas production using LSFB bioreactors is particularly popular due to its versatility, low cost of maintenance (BELLO *et al.*, 2017), self-sustainability, and low environmental impact in comparison with traditional techniques using inorganic reactants (usually strong bases or strong acids). It produces well-treated non-aggressive effluent and biogases (*e.g.*, methane, hydrogen). Additionally, the effectiveness of the influent conversion is sensitive to the carrier particle, which can be made of different polymers and ceramics (ELREEDY *et al.*, 2016; NELSON *et al.*, 2017).

CHAPTER 3

OBJECTIVES AND THESIS STRUCTURE

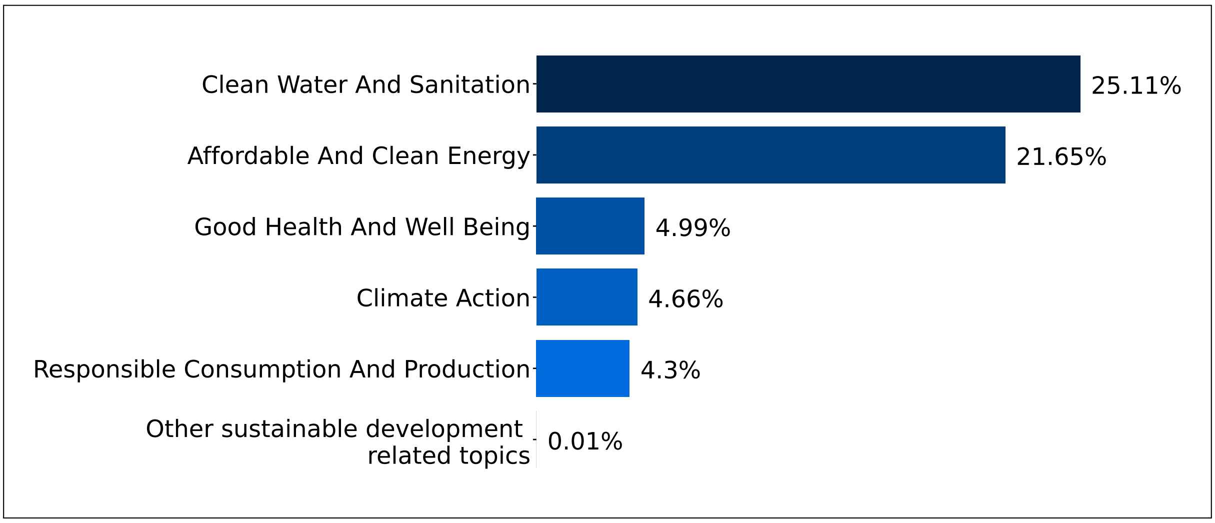
The literature review shows that essential knowledge has been accumulated in the last decades about the dynamics of multiphase flows, the computer implementation of numerical methods to represent Lagrangian and Eulerian flows, and the potential for process intensification of liquid-solid fluidized beds. Nonetheless, much is yet to be done to assess the full capability of the fields combined. In this section, gaps in the literature are identified and the ones this research intends to address are pointed out. The objectives of the work were designed so that the key steps to fill these gaps are achieved, and they are also presented in this section. Lastly, a comment on the thesis structure is provided.

3.1 Gaps in the literature and project scope

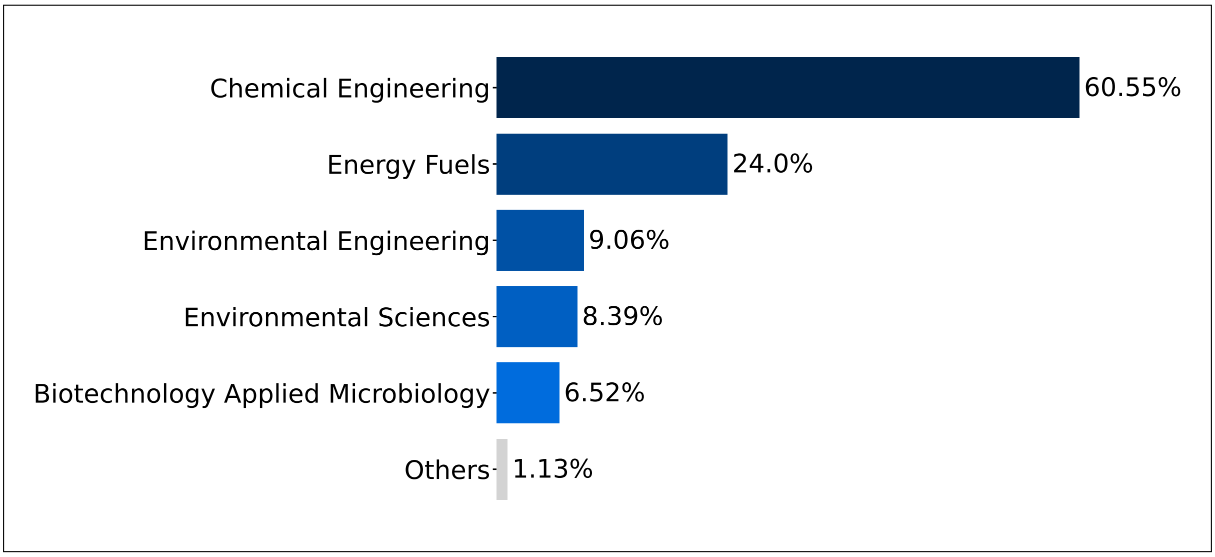
The mathematical models describing multiphase flows have significantly evolved in the past decades. The main reason is that robust numerical algorithms using discretization methods can now be implemented in high-performance computers to simulate the behavior of real systems. These methods have the potential to produce reliable physics-based estimations with reasonable cost and time. Academic researchers, R&D teams, and engineers throughout the industry apply these technologies to design, optimize, and safely implement technologies in all fields.

Unresolved CFD-DEM is a thorough tool that has attracted much attention for combining a detailed description of particle-fluid flows with computational efficiency. On the other hand, a lot of effort is yet to be applied to validate the method against real systems. This is especially important given the black-box nature of the interphase momentum coupling mathematical expressions describing drag and lift forces. Additionally, it is not clear to what extent frequently applied hypotheses, such as neglecting lift (Magnus and Saffman), virtual mass, and Basset, can affect the precision of the predictions.

On the other hand, in Section 2.3, several possible applications of the liquid-solid fluidized beds on the intensification of industrial processes were presented. Most of those applications are closely related to very sensitive research areas, with a direct impact on the environment and society. In Figure 3.1, this fact is illustrated by the literature records in liquid-solid fluidized beds (source data available on www.webofknowledge.com, accessed on September 25th, 2023).



(a) Sustainability-related subjects.



(b) Research fields.

Figure 3.1 – Percentage of the publications on a) sustainability-related subjects and b) research fields combining the topic keywords "Liquid" and "Fluidized Bed". Source: www.webofknowledge.com, accessed on September 25th, 2023.

In Figure 3.1a it is shown that about 25% of the research on LSFBs is related to the topic “Clean Water and Sanitation” and about 22% is related to “Affordable and Clean Energy”. The main reason is related to the very favorable application of LSFBs in wastewater treatment plants using anaerobic digestion. The stunning increase in the human population at urban centers brings up new public health and resource management challenges. Wastewater management and sanitation are key to the former, while sustainable energy resources are urgent to the latter. Anaerobic digestion technologies offer both. It is a cheap, self-sustainable, environmentally friendly, and compact process that can immensely benefit from LSFB features (CAYETANO *et al.*, 2022).

The intensive application of this promising technology, however, is still dragged back by the lack of predictability. LSFBs designing equations are empirical black-box correlations, which allow for the prediction of global behavior, while temporal and spatial variability are not contemplated. Nevertheless, the research interest in liquid-solid fluidization dynamics is far from ideal, especially compared to gas-solid fluidization. The numbers of publications per year combining the keywords “Fluidized Bed” with “Gas” and “Liquid” are presented in Figure 3.2 (source data available on www.webofknowledge.com, accessed on September 25^{th}, 2023).

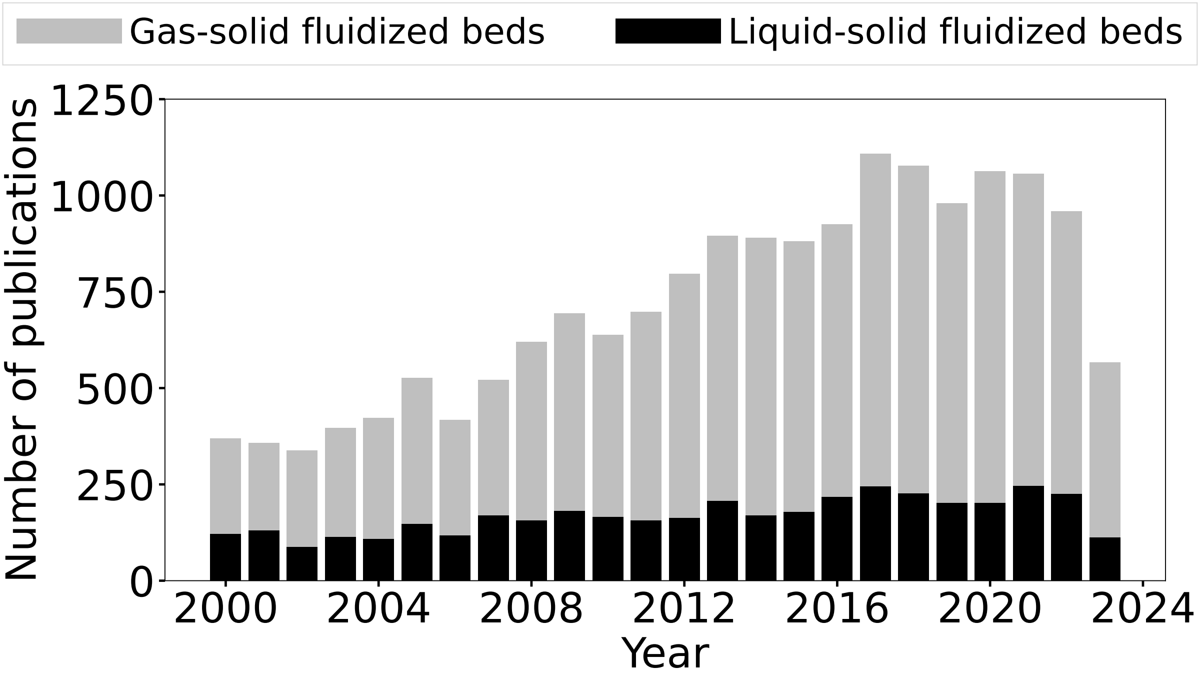


Figure 3.2 – Number of publications per year in gas-solid and liquid-solid fluidization. Source: www.webofknowledge.com, accessed on September 25^{th}, 2023.

More specifically, the literature reveals that most fluidization studies in the novel unresolved CFD-DEM methods are mostly aimed at gas-solid applications. Challenges related to the liquid-solid physics can be the reason for this, namely:

1. Dry pairwise collisional properties are more easily determined than wet, since non-contact forces, usually neglected on DEM and CFD-DEM simulations, can start to play a significant role (MELO *et al.*, 2023). At dry systems, collisions are naturally

more restitutive, and higher values for the coefficient of restitution are acceptable. This is not necessarily true for liquids.

2. The fluid density is way higher and viscous effects become more significant (lower Ar), influencing not only the collisional behavior but also the interphase momentum exchange. In particular forces other than drag, usually negligible in gas-solid fluidization, are possibly more important (CROWE *et al.*, 2011).

3. Drag correlations are determined using static beds and to a limited range of Re_p (GIDASPOW, 1994; FELICE, 1994; BEETSTRA *et al.*, 2007; RONG *et al.*, 2013). According to Rong *et al.* (2013), those correlations can produce up to 30%+ errors. In gas-solid fluidization, particles travel at velocities way faster than in liquid and the damping of this effect in time is naturally faster. LSFBs, especially at lower porosity, will be more susceptible to this error, especially given the homogeneous nature of the particles' distribution (YANG, 2003b; EPSTEIN, 2003b).

4. From the experimental perspective, it can be very challenging to do *in-situ* measurements, limiting validation.

For the provided reasons, the high potential of LSFBs as a process-intensification strategy remains covered by uncertainty, and the extensive application of the unresolved CFD-DEM methods on LSFBs, is still underexplored. This research aims to contribute to the literature on the subject and reduce the uncertainties by combining these two phenomenal technologies. Under this premise, the following objectives were designed.

3.2 Research objectives

The main objective of this research is to assess the dynamics of liquid-solid fluidized beds through experimental and unresolved CFD-DEM methods. Ergo, the following specific objectives were outlined:

1. Assemble a pilot-scale liquid-solid fluidized bed setup and implement Arduino sensors to measure the bed expansion, liquid flow rate, and temperature during fluidization experiments.

2. Select, synthesize, and characterize particles with significantly different properties to be used in liquid-solid fluidization experiments.

3. Calibrate the sensors implemented on the experimental setup and measure the bed expansion for all particles at several inlet flow rates.

4. Test and validate the predictions of the bed expansion behavior against experimental data.

5. Implement interphase force models in the open-source FEM-based CFD/DEM/CFD-DEM software called Lethe. Use it to simulate the liquid-solid fluidized bed and validate the unresolved CFD-DEM method against experimental data.

6. Extrapolate the validated simulation setup to assess the internal dynamics of particles in liquid-solid fluidized beds using mixing indices based on Lagrangian information.

3.3 Thesis structure

The upcoming chapters describe how the previously mentioned objectives were achieved. They were separated in a logical order that does not necessarily resemble the chronological. The choice of presenting results as a compilation of articles was taken following the best international practices (GUSTAVII, 2012). Although it is not the usual structure of Brazilian Theses, the choice was motivated by this author's belief that the efforts of compiling results and publishing can be combined to bring more relevance to the research.

In Chapter 4, the methods applied to this work are introduced. More specifically, details that were omitted in the articles (for brevity) are presented, including particle synthesis and characterization, assembly of the pilot-scale LSFB setup, data acquisition and treatment, and comments on Lethe (related to specific objectives 1, 2, and 3).

In Chapter 5 (Article 1), the experimental results are compared to estimations using correlations in the literature. Discussions on the precision of the classic Richardson-Zaki equation are provided. An alternative method using drag correlations for the estimation of the bed expansion was proposed, tested, and validated (related to specific objectives 3 and 4).

In Chapter 6 (Article 2), the unresolved CFD-DEM method is tested and validated. The simulations accurately reproduce the expansion behavior of the pilot scale liquid-solid fluidized bed experiments. Discussions on the importance of the Saffman lift force on the particle dynamics, the sensitivity of the method to mesh topologies, the difference between drag correlations, and other aspects concerning the simulation of the LSFB are provided (related to specific objectives 4 and 5).

In Chapter 7 (Article 3), the validated model is used to assess the mixing dynamics of particles in the liquid-solid fluidized bed. The mixing is measured by the Lagrangian position of the particles, using the nearest neighbors method (NNM) (FAN *et al.*, 1970; GODLIEB *et al.*, 2007) and Doucet's mixing index (DOUCET *et al.*, 2008). The indices are applied and compared to a wide range of regimes and for particles with a wide range of properties (related to specific objective 6).

In Chapter 8, a general discussion about the findings of the present research is provided. Chapter 9 brings conclusions based on these findings. The appendices include the derivation of the weak form of the Poisson equation (AAppendix A); a sample of the parameters file of the simulations (Appendix B); the source code of the post-processing tool developed in Python and used to process the simulation data of this work (Appendix C); and the conference paper in which the author of this thesis is the second author, concerning the sensitivity of the bed expansion to parameters influenced by the temperature of the liquid (Appendix D). The main author of the paper in Appendix D is Daniel Silva Junior, an intern (*estudante de iniciação científica*) and colleague researcher who worked extensively on the liquid-solid fluidized bed experimental setup and coauthored Articles 5 and 6.

CHAPTER 4

METHODOLOGY

The methodology applied to this work is presented in this chapter. First, the experimental methods are described, followed by the simulation strategies. The experimental section is focused on the assembling and operating of the pilot-scale liquid-solid fluidized bed unity, while the simulation section presents Lethe, the CFD/DEM/CFD-DEM software used in simulations. Specific details on the application of each method in the articles are suppressed in this section to avoid redundancy.

4.1 Experimental methods

The experiments of this work were performed at the Chemical Engineering Department of the Federal University of São Carlos.

4.1.1 Particles selection, synthesis and characterization

The experiments started with particle selection, acquisition, synthesis, and characterization. Snapshots of the groups of particles are shown in Figure 4.1. The first four groups of particles, namely ABS, Alumina (6.37 mm), Alumina (3.09 mm), and Porcelain; were used as received, while the remaining three, namely Alginate (3.48 mm), and Alginate (2.66 mm); were synthesized.

4.1.1.1 Alginate particles synthesis

The synthesis of the alginate particles follows the procedure developed and detailed in previous works by the research group, including Lopes *et al.* (2018), Melo *et al.* (2021a),

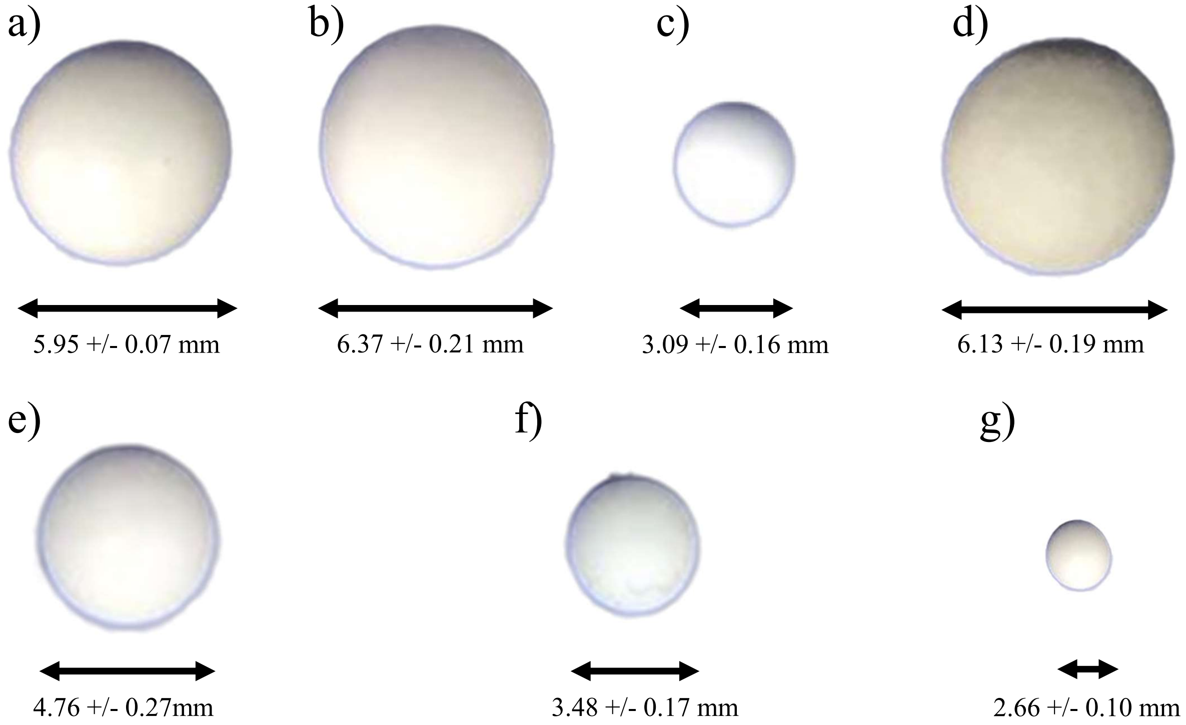


Figure 4.1 – Snapshots of a) ABS, b) Alumina (6.37 mm), c) Alumina (3.09 mm), d) Porcelain, e) Alginate (4.76 mm) f) Alginate (3.48 mm), and g) Alginate (2.66 mm). Adapted from Ferreira *et al.* (2023b).

and Article 5. Sodium alginate ($NaC_6H_7O_6$) is a chemical powder product extensively applied in the food industry as a thickener, due to its capability of becoming a gel when in contact with calcium, barium, and other cation. The chemical reaction between the cation and sodium alginate produces a very lightweight gel with a high liquid content.

The synthesis consisted of dripping a sodium alginate solution in a barium chloride solution. The 15 $\text{g}\,\text{L}^{-1}$ barium chloride solution and the 20 $\text{g}\,\text{L}^{-1}$ sodium alginate is prepared using distilled water. Titanium dioxide powder is added to the sodium alginate mixture at a mass ratio of 25:2 to increase the weight of the resulting particles (which otherwise would be almost identical to water and fluidization experiments would be unfeasible).

The formation of alginate particles is illustrated in Figure 4.2. It consists of dripping the sodium alginate solution on the barium chloride solution. The sodium alginate solution was pumped through silicon tubes using a peristaltic pump. The diameter of the particles was controlled both by the pump flow rate and the tip attached to the end of the tubes. Alginate (3.48 mm), and Alginate (2.66 mm) were produced using 1/4" diameter plastic tube, a cut ball pump needle (1.5 cm diameter, approximately), and a 0.7 cm diameter injection needle, respectively.

During the synthesis, the barium chloride solution was constantly agitated using a magnet agitator and was kept at a low temperature using an ice and water bath. The agitation was constantly controlled to prevent particles from sticking to each other

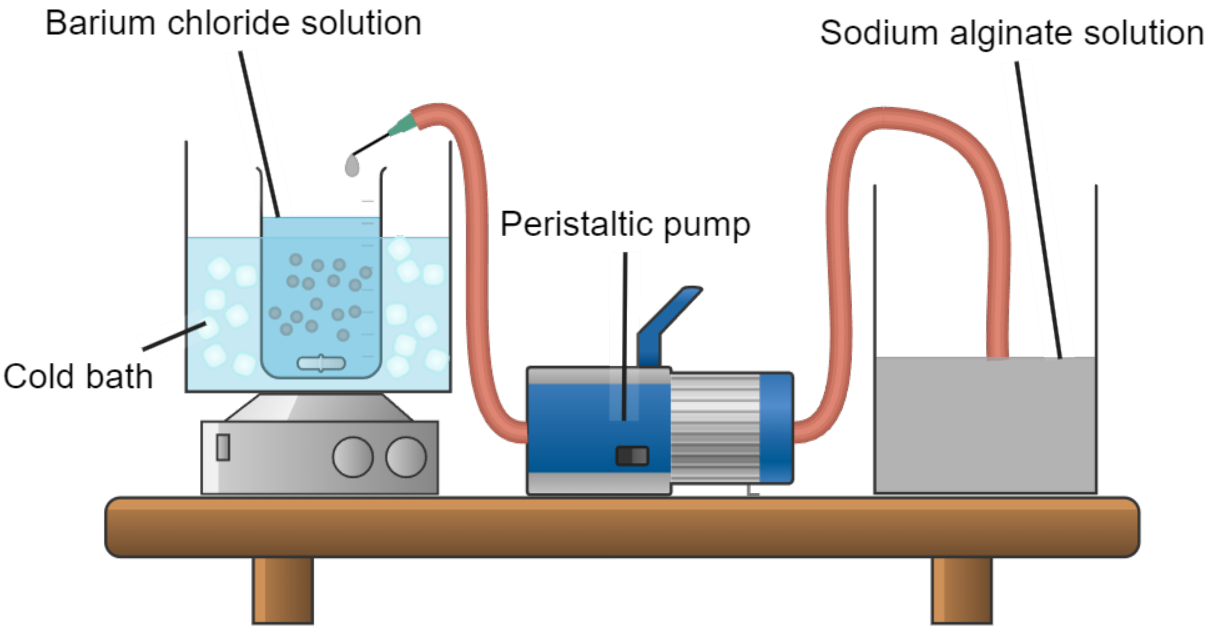


Figure 4.2 – Schematic representation of alginate particle synthesis (produced using Chemix, available on chemix.org).

and avoid deformation due to over-agitation. Additionally, lowering the temperature of the barium chloride solution during the synthesis produced visually rounder particles in preliminary tests.

4.1.1.2 Particles characterization

The density of particles was measured using pycnometry, with a 25 mL pycnometer. The method consists of measuring the mass of the systems represented in Figure 4.3, which are: empty pycnometer (m_{pyc}), pycnometer filled with distilled water (m_{pyc+H_2O}), pycnometer with particles (m_{pyc+p}), and pycnometer with particles and distilled water (m_{pyc+H_2O+p}). The particles are filled up to about 60% of the pycnometer height.

First, we determine the real capacity (volume) of the pycnometer according to the temperature by:

$$V_{pyc} = \frac{m_{pyc+H_2O} - m_{pyc}}{\rho_{H_2O}} \tag{4.1}$$

where ρ_{H_2O} is the density of distilled water. We measured the temperature of the water throughout the density measurements and calculated the density for each particle, according to the data by Wagner and Pruß (2002 apud PERRY; GREEN, 2008).

The volume of particles in the pycnometer is calculated by:

$$V_p = V_{pyc} - \frac{m_{pyc+H_2O+p} - m_{pyc+p}}{\rho_{H_2O}} \tag{4.2}$$

and the density of particles is calculated by:

$$\rho_p = \frac{m_{pyc+p} - m_{pyc}}{V_p} \tag{4.3}$$

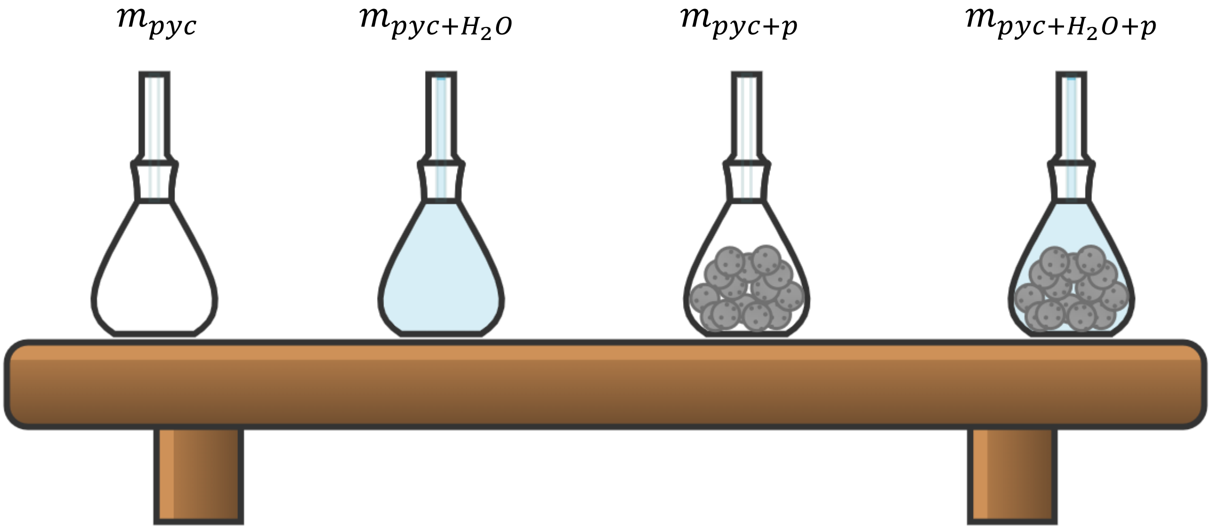
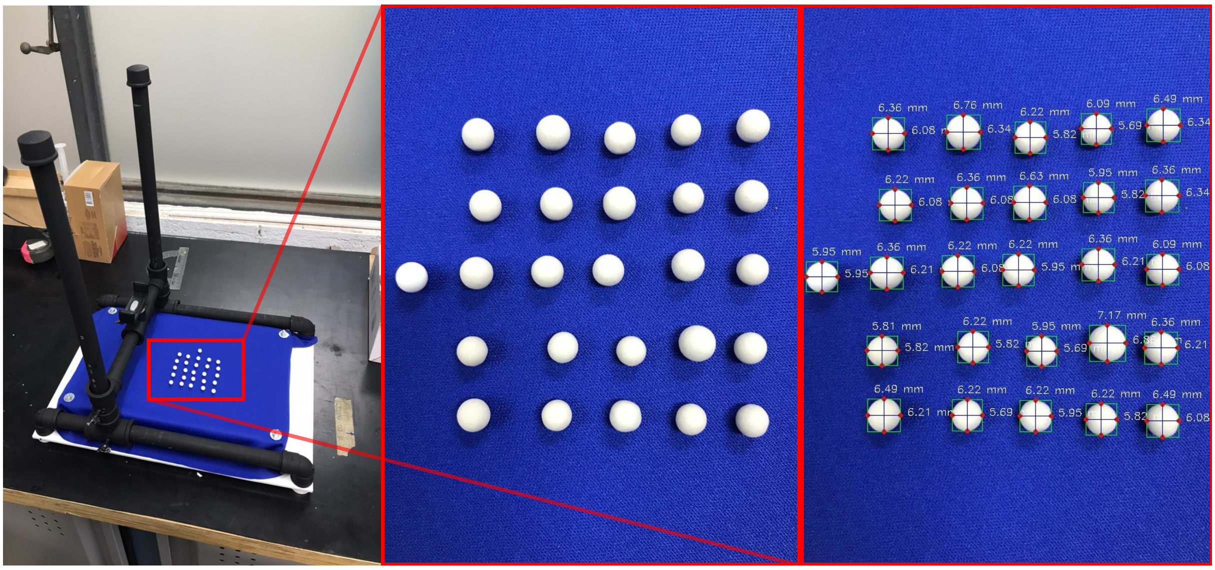


Figure 4.3 – Schematic representation of particle density measurement using pycnometry (produced using Chemix, available on chemix.org).

The particle density used in all results of this work corresponds to the average among five measurements. Each measurement is done with different randomly sampled conjuncts of particles. After each measurement, the water in the pycnometer was discharged and the pycnometer was cleaned and dryer. All weights were measured with a calibrated semi-analytical balance. To guarantee that porosity would not affect ceramic particle density, alumina and porcelain particles were submerged in distilled water for at least 24 hours prior to the measurement and superficially dried with a paper towel right before entering the pycnometer.

The diameter of the particles was measured using the apparatus in Figure 4.4a, which comprises a cellphone holder and a table with a blue chromakey where particles are positioned. The pictures were taken using a cellphone (brand Samsung, model Galaxy S21+). Pictures of the particles were taken after leveling the table and the cellphone, and maintaining the same distance between them and the camera. The resulting images are similar to the illustration in Figure 4.4b.

The diameter of the particles was measured using a Python script written with the OpenCV image processing library (BRADSKI, 2000). The calibration of the software was done using the ABS particles as reference ($d_p = 5.95 \pm 0.01$ mm). The calibration is done on all images separately, using the same ABS particle. The diameter of each particle corresponds to the average between the vertical and the horizontal diameters illustrated in Figure 4.4c. The resulting diameters for each group of particles are equivalent to the surface diameter of a set of 20 samples



(a) Apparatus. (b) Particles. (c) Particles with diameters.

Figure 4.4 – Illustration of the measurement of particles' diameter.

Terminal settling velocities of the particles were measured by tracking their falling trajectories inside a square base 20 cm wide water tank, as illustrated in Figure 4.5. The particle motion was recorded using a high-speed camera (DSC-RX100 M, brand Sony) at 960 frames per second. The position tracking was done with the Tracker open-source image processing tool (BROWN *et al.*, 2022). The calibration of the distance is done using the rules in Figure 4.5. The position of the particles is tracked by calibrating the software using each ruler individually. The velocity for a single particle is the average among the two measurements. The terminal settling velocity of a group of particles corresponds to the average among ten particle launches.

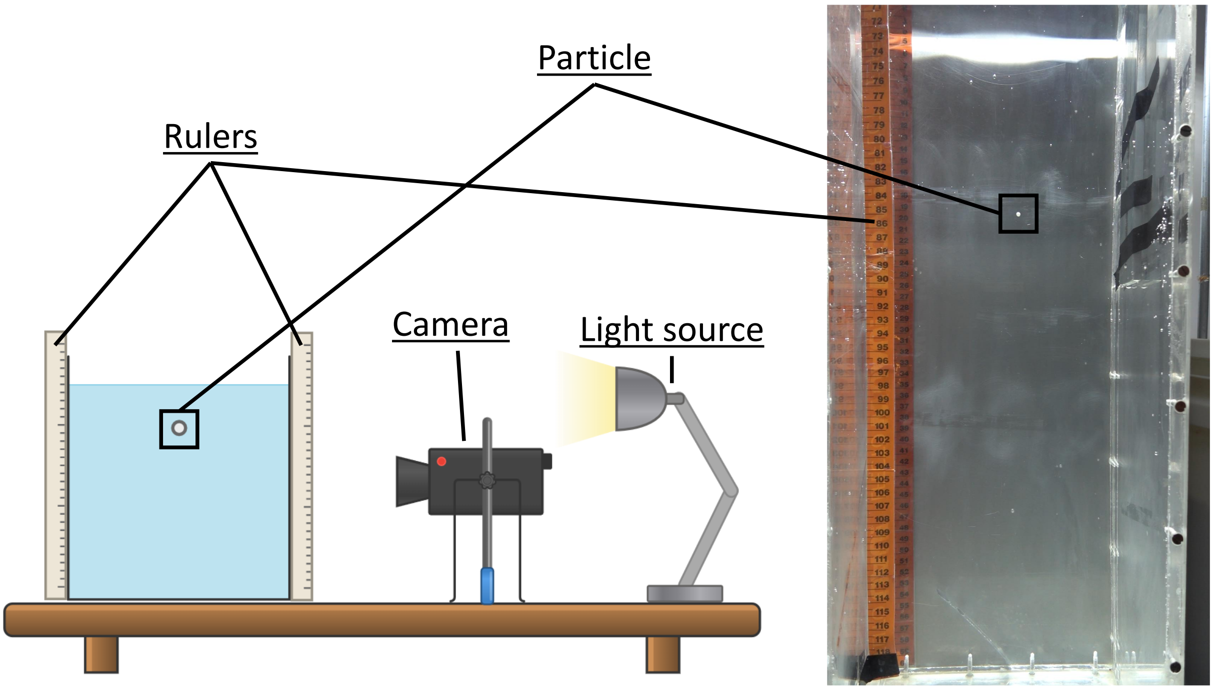
4.1.2 Liquid-solid fluidized bed experiments

The first objective of this work was to assemble a pilot-scale liquid-solid fluidized bed experimental setup. A general description of the equipment can be found in Chapters 5 and 6 (Article 1 and Article 2). All experiments were carried out with water. Here, more details about the assembly of the experimental apparatus and the sensors are provided.

4.1.2.1 Pilot-scale equipment

All experiments were carried out on the same fluidized bed setup, presented in Figure 4.6. Figure 4.6a presents a full view of the experimental setup. Figure 4.6b shows the parts of the equipment that are used while in operation, comprising:

– A: Stainless steel structure holding the entire setup, including the centrifuge pump, the



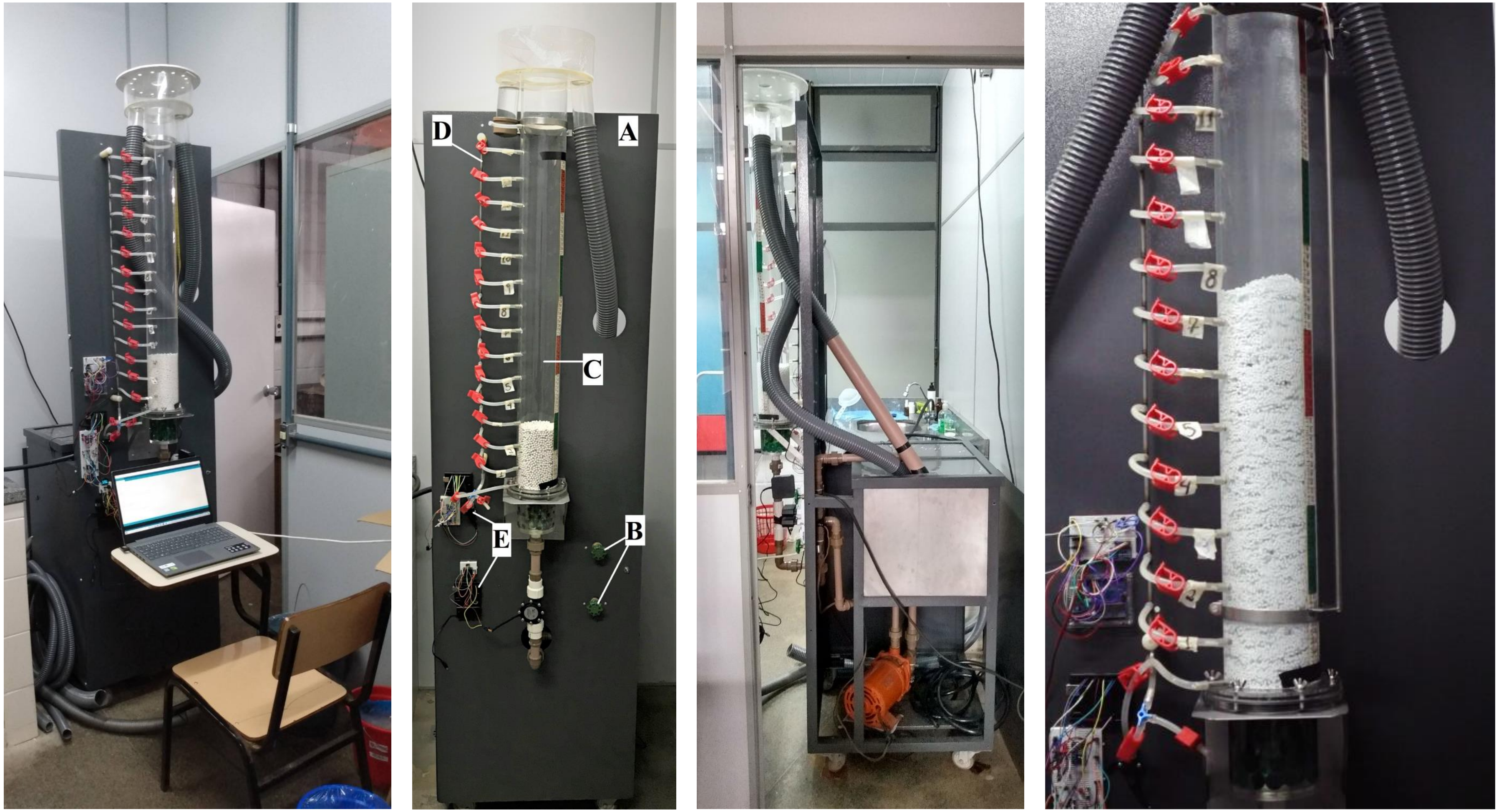
(a) Schematic drawing (produced using Chemix, available on chemix.org).

(b) Snapshot of experiment.

Figure 4.5 – Schematic representation of the free falling experiment for terminal velocity measurement.

water reservoir, the Arduino circuits, and the fluidization setup.

- B: Globe valves, applied to control the direct (lower) and bypass (upper) flows.
- C: Fluidized bed, including:
  a) A 10 cm height, 10 cm diameter calming section made of acrylic filled with 2.5 cm glass beads.
  b) A pierced plate distributor made of aluminum.
  c) A 1 m height, 10 cm diameter fluidization region made of acrylic.
  d) A 18 cm outlet portion that redirects the flow back to the reservoir by a 6 cm diameter flexible tube.
- D: Pressure plugs leading to the pressure sensor. From the bottom up, the first pressure plug is fixed and directly attached to the pressure sensor. The remaining 14 pressure plugs are connected to the other terminal of the pressure sensor. The distance between the first pressure plug and the first among the other 14 remaining is 4.5 cm, while the distance between the remaining plugs is 6 cm from one another.
- E: Arduino circuits. The upper measures the pressure and the water temperature in the reservoir while the bottom measures the fluid inlet flow rate.



(a) Full setup. (b) Front view. (c) Side view. (d) Under operation.

Figure 4.6 – Pilot-scale liquid-solid fluidized bed experimental setup.

From the side view (Figure 4.6c) it is possible to see the reservoir right above the orange 1/2 horsepower centrifugal pump. In Figure 4.6d the equipment under operation is shown. During the experiments, water is pumped from the reservoir and divided into a bypass flux, that redirects part of the flow back to the reservoir, and the direct flow that leads to the bed. The inlet flow rate is controlled by closing and opening both valves. The direct flow gets past the flow rate sensor and enters the calming section, goes through the distributor, percolates the bed of particles, and goes back to the reservoir. The maximum inlet flow rate the current setup is capable of achieving is 75 $\mathrm{L\,min^{-1}}$.

4.1.3 Arduino sensors

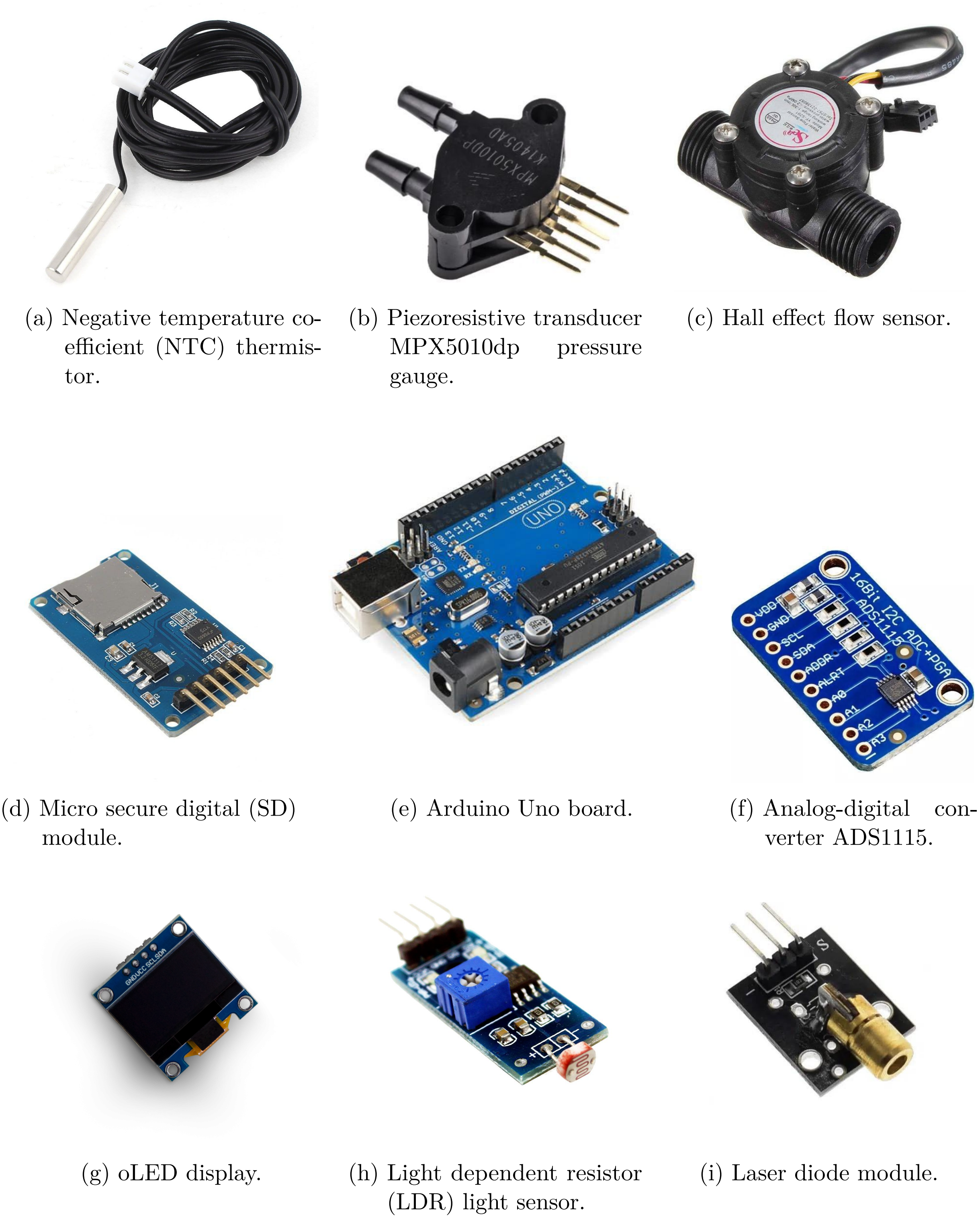
The sensors used in this work were developed using Arduino Uno microcontrollers. The board has 16 digital and 5 analog pins. The digital pins can send and interpret binary digital signals while the analog pins are capable of measuring tension between 0 and 5 V with a resolution of 10 bits (1024 binaries, being 1 for the sign and 1023 for values). Resistive sensors, such as the temperature, the laser, and the pressure sensors of this work are plugged into the analog pins and the electrical tension signal is converted to the physical quantity by a correlation. Digital sensors, such as the flow rate sensor, work by sending binary signals to the board, and the physical quantity is calculated by using the frequency of the signal. Figure 4.7 illustrates the Arduino board and the components used in this work. In this section, each sensor is detailed.

4.1.3.1 Temperature sensor

The temperature of the water was measured throughout the experiments using the negative temperature coefficient (NTC) thermistor in Figure 4.7a. The average temperature during the experiments was 30 °C. This value was used to calculate the viscosity and the density of the water using the data available on Wagner and Pruß (2002 apud PERRY; GREEN, 2008). The effect of the variability in the temperature throughout the experiments on the bed expansion was investigated as part of the research project. This study, presented in the Appendix D, shows that the impact of these variances is negligible. The conference paper of this author's co-authorship was written mainly by Daniel Silva Junior, co-author of Articles 1 and 2.

4.1.3.2 Flow rate sensor

The flow rate was measured using Hall effect flow sensors similar to the one presented in Figure 4.7c. The sensor sends a digital signal (pulse) to the microcontroller at every full revolution of its internal helices. The pulses have a direct proportional relation



(a) Negative temperature coefficient (NTC) thermistor.

(b) Piezoresistive transducer MPX5010dp pressure gauge.

(c) Hall effect flow sensor.

(d) Micro secure digital (SD) module.

(e) Arduino Uno board.

(f) Analog-digital converter ADS1115.

(g) oLED display.

(h) Light dependent resistor (LDR) light sensor.

(i) Laser diode module.

Figure 4.7 – Arduino board and components. Reproduced from www.makerhero.com, accessed in September 23th, 2023.

with the flow rate, and the conversion factor depends on the sensor. In this work, the YF-G1 ($2-100$ L min^{-1}) and the YF-S201 ($1-30$ L min^{-1}) were used for the higher and lower inlet flow rate experiments.

The measurement of the flow rate was done using a dedicated Arduino board. The assembled circuit does not require a connection with the computer. The results are displayed in a small oLED screen, depicted in Figure 4.7g. The full flow rate measurement setup works in two function modes, one dedicated to the adjustment of the flow rate and the other to the measurement during the experiments. The latter, along with the instantaneous measurement shown in the former, calculates the average and the standard deviation among the last ten measurements. The working setup is shown in Figure 4.8.

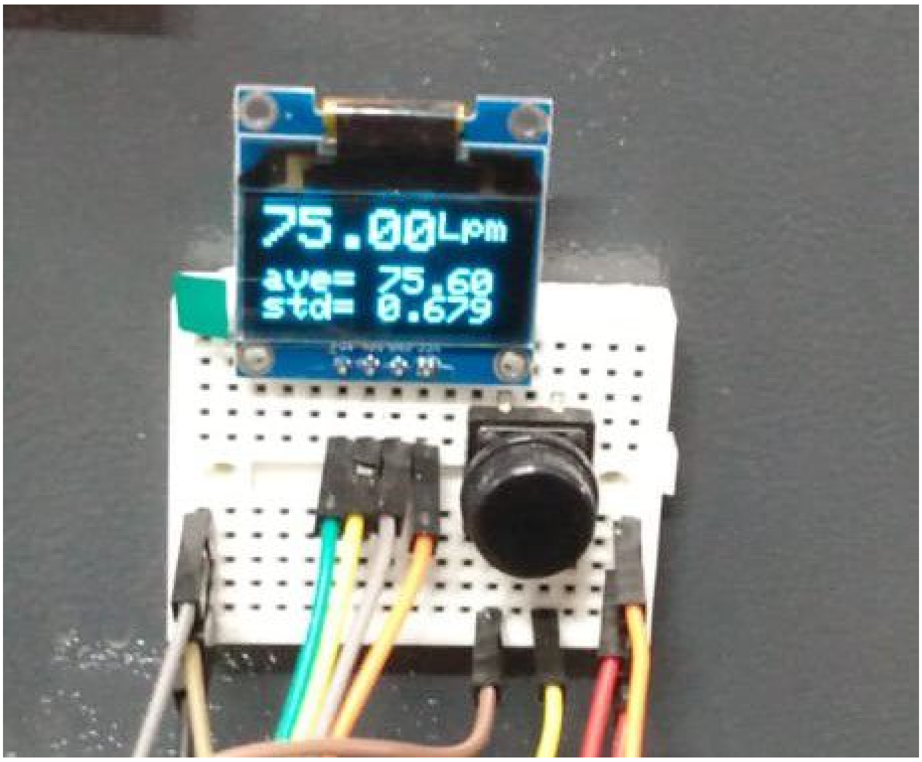


Figure 4.8 – Flow rate measurement setup.

4.1.3.3 Pressure sensor

As shown in Section 2.3.2, it is possible to measure the bed expansion using its direct relation with the pressure profile. The pressure was measured using the MPX5010dp differential pressure gauge presented in Figure 4.7b. The pressure is measured by the deformation of a membrane to which a strain gauge is attached. The simple connection between MPX5010dp with the analog pins of the Arduino Uno allows for a measurement of the differential pressure up to 10 kPa, with a measurement resolution of approximately 9.78 Pa.

The highest total pressure drop achieved by the particles of this work is below 4 kPa, and the measurement resolution is not sufficiently fine to capture the way smaller pressure differences between the plugs. To circumvent this problem, an external analog-digital converter module ADS1115 presented in Figure 4.7f was applied. The module offers 16 bits (65536 binaries, with 1 sign and 65535 values) of measurement resolution for analog signals. Additionally, the analog signal measurement limit is adjustable, allowing for higher resolution of signals within limits below the maximum standard of 5 V. Nevertheless, the natural oscillations of the MPX5010dp (due to the physical pulse of the pressure and the natural instability of the 5 V Arduino feeding) are incorporated in the more sensitive setup with a higher relative magnitude.

The oscillations were damped by a resistor-capacitor (RC) hardware low-pass filter, implemented in the connection between the analog output of the MPX5010dp and the ADS1115 analog interpreter, and a moving average software filter, with each point in time corresponding to an average among itself and the previous 10 points. Additionally, the setup was programmed to register the raw analog data, while the actual pressure was calculated using a Python program. The transfer function needed to be adapted to incorporate the ADS1115 modifications and the calibration of this adaptation was done using the raw data in bits.

The post-processing program calculates the time average among 30 seconds of measurements at 5 measurements per second (150 samples). Each experimental run consisted of measuring the differential pressure for each of the 14 pressure plugs, which took approximately 35 seconds per plug (30 seconds of measurement and 5 additional seconds to switch between plugs). Hence, each experimental journey took around 10 minutes (including adjustments on the inlet flow rate) per flow rate, per particle. For 8 to 10 flow rates per particle, the total experimental time was of about 2 hours per particle per replicate. The experiments were done in triplicates, changing the group of particles at each replicate to avoid biased statistical errors. For the same purpose, the order of particles was randomly determined at the beginning of each replicate.

The use of the ADS1115 with the RC circuit, the moving average, and the descriptive statistics approach allowed for the measurement of the differential pressure profile of the higher-density particles. However, the oscillations were too high, which did not allow for the same precision at very low pressure drops in the alginate beds. Hence, the alternative measurement of the bed expansion by the bed height was more extensively applied for the low-density particles.

4.1.3.4 Bed height measurement

The bed height was measured using a setup developed by Daniel Silva Junior with this author's collaboration. A picture of the setup is presented in Figure 4.9

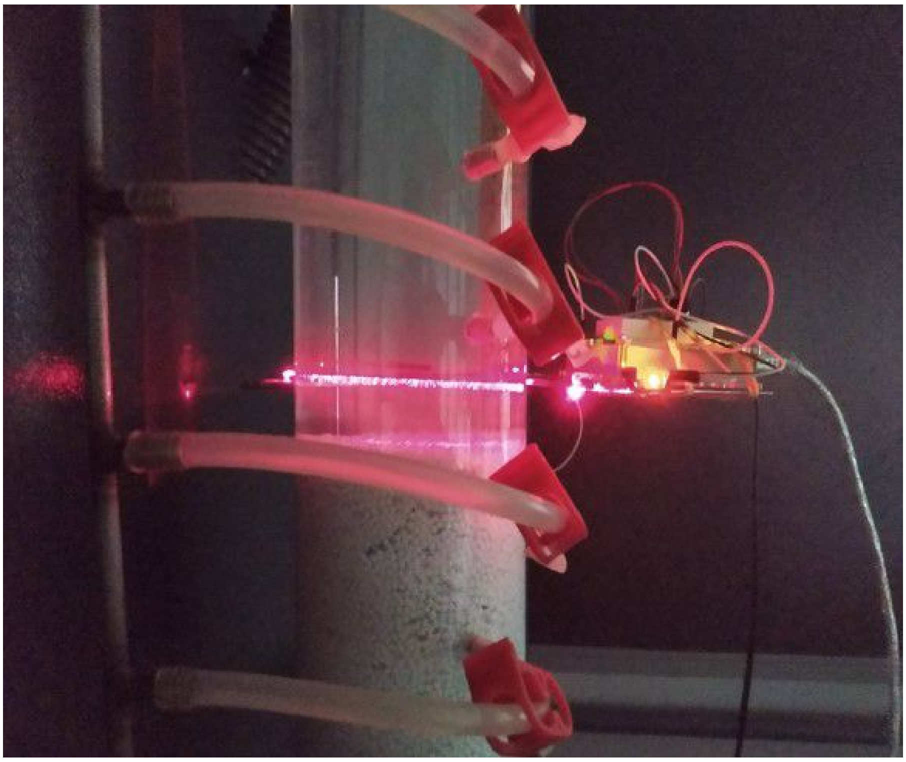


Figure 4.9 – Laser height sensor setup.

It comprises a 5 V laser diode and a Light Dependent Resistor light sensor (LDR), pointed to each other at opposite sides of the equipment. Both the beam source and sensor are fixed to a small platform, which can be moved along the height of the equipment. The sensor measures the received light intensity and sends a proportional analog signal to an Arduino. Once the microcontroller is turned on, it starts registering the analog signal. After 15 seconds of measurement, the microcontroller calculates an average among the registered values. In this work, this average value is referred to as laser (or beam) intensity.

The calibration of the sensor is done with the column full of water and with the particles in it. Before the experiments, the platform is positioned above the bed of particles so that the laser beam reaches the sensor without interruptions. This returns a high beam intensity to the microcontroller, which is the intensity in the absence of particles. After this value is registered, the platform is positioned in the middle of the bed of particles. Because the particles are opaque, the light from the laser does not reach the sensor, and the measured beam intensity is low. The average between both the high and low values is considered the threshold value. This value is used to identify the bed height.

During the fluidization experiments, the height of the laser was increased until the measured beam intensity was within 10% of the threshold value. This defined the height of the bed (H). This measurement method was highly reproducible, presenting a maximum standard deviation of 2% of the average. With the bed height H, the bed porosity ($\bar{\varepsilon}_f$) was calculated using Equation 2.93.

4.1.3.5 Data storage system

The data collection was done using the micro secure disk (SD) module in Figure 4.7d. The module allowed for a faster, safer, and more independent measurement of the analog signals captured by the Arduino compared to the regular Monitor Serial.

4.2 Unresolved CFD-DEM simulations

All simulations of this work were carried out in Lethe (BLAIS *et al.*, 2020; GOLSHAN *et al.*, 2022; GEITANI *et al.*, 2023b; GEITANI *et al.*, 2023a), which is an open-source CFD, DEM, and CFD-DEM software based on the powerful Finite Element Method library differential equations analysis library (deal.II) (ARNDT *et al.*, 2021; ARNDT *et al.*, 2022). Deal.II is highly optimized and uses Trilinos (TEAM, 2020) for sparse linear algebra routines and p4est (BURSTEDDE *et al.*, 2011) for distributed adaptive quadtrees and octrees (image data structures). Lethe applies high-order continuous Galerkin formulations to numerically solve flow equations such as the incompressible Navier-Stokes equations and others (e.g. VANS).

Comments on the algorithms adopted by Lethe to simulate granular, fluid, and granular-fluid flows are provided in Sections 2.2.1, 2.2.3, and 2.2.4. Examples of the DEM and CFD-DEM parameters files used in this work can be found in Appendix B. An in-depth validation of Lethe's unresolved CFD-DEM feature on the simulation of liquid-solid fluidized beds is shown in Article 2. Other unresolved CFD-DEM verification and validation cases include works by Geitani and Blais (2023b), Geitani *et al.* (2023a), Geitani *et al.* (2023b), Geitani and Blais (2023a). Additionally, the DEM code was validated by Golshan *et al.* (2022), and the implicit LES approach to turbulence was validated by Saavedra *et al.* (2022).

The post-processing of the simulation data of this work was done with a module written in Python using the PyVista library (SULLIVAN; KASZYNSKI, 2019). The source code of the module and samples of its applications are provided in Appendix C. The module and Lethe's source code are fully available on www.github.com/lethe-cfd/lethe, and fully documented on Lethe's official website www.lethe-cfd.github.io/.

CHAPTER 5

ARTICLE 1: PREDICTION OF THE BED EXPANSION OF A LIQUID FLUIDIZED BED BIOREACTOR APPLIED TO WASTEWATER TREATMENT AND BIOGAS PRODUCTION

Victor Oliveira Ferreira, Daniel Silva Junior, Karla Raphaela Braga de Melo, Bruno Blais, Gabriela Cantarelli Lopes

Published in Energy Conversion and Management, volume 290, 2023

5.1 Abstract

The high potential of water treatment and biogas production systems using liquid fluidization is still underexplored. The design of this equipment is usually done using the simple Richardson-Zaki equation for bed expansion predictions, which is powerful but overlooks the interactions between fluid and particles. In this work, an alternative method based on a force balance on the bed and drag correlations to estimate the bed porosity was proposed. The accuracy of the methods was assessed by comparing bed porosity estimations with experiments carried out in a wide range of regimes (Reynolds numbers between 498 and 18664), using 7 different particles with various diameters (2.66 to 6.37 mm) and densities (1022 to 3585 $\mathrm{kg\,m^{-3}}$). On average, the fitting between the Richardson-Zaki equation and the experimental results was improved by 60% when the wall effects were considered. The alternative approach using the drag correlations showed promising results, presenting a coefficient of determination higher than 80% for all particles, and better precision in 4 out of the 7 particles compared to Richardson-Zaki. The results show that, in general, both the Richardson-Zaki equation and the drag correlation approach can be used to predict the liquid fluidized bed expansion. However, the proposed approach using drag correlations showed more reliable results, especially when the Rong drag model

was applied, with an average coefficient of determination of 0.92. Comparisons with other results in the literature confirm the extent and the scalability of the method. The use of the proposed method to estimate the liquid fluidized bed expansion allows for easier and safer applicability of the technology in areas such as wastewater treatment and biogas production.

5.2 Introduction

Liquid-solid fluidized beds (LSFB) are applied in many processes such as drinking water production (KRAMER *et al.*, 2020), crystallization (ALDACO *et al.*, 2007), particles classification (MUKHERJEE *et al.*, 2009), heat transfer (DIAS *et al.*, 2018), advanced oxidation processes (BELLO *et al.*, 2017), bioartificial devices (NAGHIB *et al.*, 2017), wastewater treatment with biogas production (GARCíA-CALDERóN *et al.*, 1998) and others (EPSTEIN, 2003a). In this type of equipment, the solids are lifted by the uprising liquid injected at velocities between the minimum fluidization and the terminal velocity of the particles. Consequently, the particles are held up in a uniformly dispersed, contact-driven dynamic. Among the main advantages of LSFBs compared to fixed beds is that the porosity of the bed can be easily increased by incrementing the inlet flow rate at a constant pressure drop (EPSTEIN, 2003b), making the operation conditions more manageable.

Wastewater treatment with biogas production using LSFB bioreactors is particularly popular due to its versatility, low cost of maintenance (BELLO *et al.*, 2017), self-sustainability, and low environmental impact in comparison with traditional techniques using inorganic reactants (usually strong bases or strong acids). The process consists of fluidizing particles carrying bioactive material, which consumes the organic matter present in the influent. It produces well-treated non-aggressive effluent and biogases (*e.g.*, methane and hydrogen) (JAMALI *et al.*, 2017). In such a process, the composition of the products can be controlled by varying the microorganisms and the influents. Additionally, the effectiveness of the influent conversion is sensitive to the carrier particle, which can be made of different polymers and ceramics (ELREEDY *et al.*, 2016).

According to Fraia *et al.* (2018), wastewater treatment plants can be highly energy-consuming industries. However, the LSFB anaerobic bioreactors can decrease external energy consumption since the biogas product from the bioreactions can be used as fuel for energy generators, making the process cheap and self-sustainable (NADAIS *et al.*, 2011). The fluidization can help in the process because it increases the mass transfer between the fresh influent and the bioactive particles compared to slurry or fixed bed bioreactors.

Significant constraints regarding the LSFB bioreactors are related to the limited understanding of their fluid dynamics, leading to operational issues (NADAIS *et al.*, 2011) and design limitations (METOLINA; LOPES, 2019). Most of the designs rely upon empirical equations based on experiments with hard, heavy particles. Among the design equations, the Richardson-Zaki equation (R-Z) (RICHARDSON; ZAKI, 1954) is particularly popular due to its simplicity and precision in predicting the LSFBs expansion. This equation (in its extended version) is

$$\frac{U}{U_0} = k\bar{\varepsilon}_f^n \tag{5.1}$$

where U is the superficial velocity of the fluid, U_0 the terminal settling velocity of the particles, and the fluid fraction inside the bed region, also referred to as bed voidage or porosity.

In Equation (5.1), k and n are empirical parameters. The parameter n is related to the characteristics of the flow, varying between 4.60 and 4.90 for particles in the Stokes regime ($\mathrm{Re}_{p,0} < 0.2$, $\mathrm{Ar} < 4$) and between 2.39 and 2.40 in the Newton regime ($\mathrm{Re}_{p,0} > 500$, $\mathrm{Ar} > 8500$). For the region between the Stokes and Newton regimes, the correlation proposed by Khan and Richardson (1989) can be used:

$$\frac{4.8 - n}{n - 2.4} = 0.043Ar^{0.57} \tag{5.2}$$

where Ar is the Archimedes number, given by:

$$\mathrm{Ar} = gd_p^3\rho\left(\rho_p - \rho\right)/\mu^2 \tag{5.3}$$

There are other accurate correlations in the literature to determine n, such as the ones proposed by Wallis (2020), Garside and Al-Dibouni (1977), and Rowe (1987). Notwithstanding, Equation (5.2) is significantly more convenient since it does not depend on experimental efforts or other correlations to determine the terminal Reynolds of the particle-fluid pair. Several authors have discussed the extension of R-Z to a wide range of particle characteristics and, until the present moment, there is good agreement in the literature about the estimation of n.

The parameter k represents the effect of the walls in the bed expansion. In theory, for narrower fluidized beds, the walls of the equipment should start to play a role in the bed expansion, while for larger equipment, the wall effect should be negligible, that is, k should be equal to 1. Nonetheless, the use of k and its dependence on the characteristics of the bed is still a divergence in the literature.

Richardson and Zaki (1954) and Khan and Richardson (1989), for example, proposed different correlations to calculate k as a function of the ratio between the equipment diameter and particle size (d_p/D). In contrast, several authors achieved good agreement between the experimental and estimated bed porosities neglecting the wall effects, that is, assuming $k = 1$. Miura *et al.* (2001), for example, have shown that, although the wall effects were not considered in their work, the proposed correlations for the bed voidage obtained for Newtonian and non-Newtonian liquid–solid fluidized beds were not affected by this simplification. Moreover, Kramer *et al.* (2020) found that wall effects can be neglected in case $\mathrm{Re}_{p,0} > 1 - 10$. Still, Tang *et al.* (2016) have shown that, when the particle size distribution is wide, the wall effect can decrease or increase the value of n depending on the particle size distribution. They observed, however, that the wall effect becomes more significant with decreasing bed diameter.

Alternatively, other authors, such as Rapagnà *et al.* (1989) and Epstein (2003b), proposed other correlations for k as a function of the terminal settling Reynolds number ($\mathrm{Re}_{p,0}$) instead of d_p/D. According to Epstein (2003b), k can be estimated for a wide range of particle characteristics using the equation proposed by Khan and Richardson (1989):

$$k = 1 - 1.15\left(\frac{d_p}{D}\right)^{0.6} \tag{5.4}$$

Lopes *et al.* (2018), for example, obtained good agreement between prediction and experiments using the extended version R-Z by estimating k with Equation 5.4. Despite that, taking or not the wall effects into account, *i.e.*, the use of a calculated k in the R-Z equation and how to calculate it is still under debate.

Although the R-Z equation is widely used to estimate the bed expansion of liquid-solid fluidized beds, there is no consensus on what is the best way to estimate its parameters. Most of the studies found in the literature use very restrictive conditions to validate their correlations, hampering their applicability. Additionally, very few popular alternatives to R-Z have been proposed so far, and most of them are direct correlations that take only the macro scale into account.

Aiming to fill these gaps, the present work investigates the extent of R-Z to 7 particles with a wide range of physical properties (d_p, Ar, $\mathrm{Re}_{p,0}$). All particles are fluidized in water using several different superficial velocities. The objective is to find the possible limitations of R-Z, using the convenient correlations for k and n proposed by Khan and Richardson (1989). Since most details regarding the fluid-particle interactions at a meso scale are considered by drag correlations, an alternative to the bed porosity estimation using the correlations proposed by Felice (1994), Beetstra *et al.* (2007), and Rong *et al.* (2013) is provided. To do this, the absolute difference between the Drag force calculated by the force balance for a single particle and the one estimated by each of the correlations was minimized to find the bed expansion, as explained further. Additionally, to assess the extent of the method, it was applied and compared to experimental data by Lopes *et al.* (2018).

5.3 Materials and Methods

To investigate the prediction of the bed expansion of liquid fluidized beds using R-Z and the proposed alternative method, experiments were performed using different particles. The materials and methods applied in the present work are described below.

5.3.1 Materials

The particles used in this work are divided into two big groups: heavy and light. Each group was characterized in terms of diameter, density, and terminal settling velocity. All heavy particles were used as acquired, while the light particles were synthesized. The heavy particles group comprises Acrylonitrile Butadiene Styrene (ABS, 5.95 mm diameter), porcelain (6.13 mm diameter), and two alumina (6.37 and 3.09 mm diameter) particles. The 3 light particles are all made of Barium Alginate (referred to as alginate particles).

The light particles were synthesized by dropping a 20 $\mathrm{g\,L^{-1}}$ sodium alginate (brand Fisher Chemical) solution into a 15 $\mathrm{g\,L^{-1}}$ barium chloride solution. When the sodium alginate droplets react with the barium chloride solution, the Barium Alginate particles are formed due to the gelification process. Titanium dioxide powder (mass ratio 25:2) was added to the sodium alginate solution to increase the weight of the particles. Three different diameters of particles were obtained by changing the tip of the tubes and the flow rate of the sodium alginate solution. For a detailed description of the synthesis, the authors refer to Lopes *et al.* (2018) and Melo *et al.* (2021a).

Densities of particles were measured by the pycnometry method, using a 25 mm pycnometer and distilled water as standard fluid. The diameter and circularity of the particles were measured using the OpenCV image processing library (BRADSKI, 2000). Pictures of the particles were taken maintaining the distance between them and the camera. Calibration of the software was done using the ABS particles as reference ($d_p = 5.95 \pm 0.01$ mm). The resulting diameters for each group of particles are equivalent to the surface diameter of a set of 20 samples.

Terminal settling velocities of the particles were measured by tracking their falling trajectories inside a square base 20 cm wide water tank. The particle motion was recorded using a high-speed camera (DSC-RX100 M, brand Sony) at 960 frames per second. The position tracking was done with the Tracker open-source image processing tool (BROWN *et al.*, 2022). The terminal settling velocity corresponds to the average of ten particle launches. The average diameter, density, terminal settling velocity (± their respective standard deviations), and terminal Reynolds number of each particle are shown in Table 5.1. Images of the particles are also presented in Figure 5.1.

Table 5.1 – Properties of particles.

Particle	**Density,** ρ_p **(kg/m^3)**	**Diameter,** d_p **(mm)**	**Terminal settling velocity,** U_0 **(cm/s)**	**Terminal Reynolds number,** $\mathrm{Re}_{p,0}$ **(-)**
ABS	1823 ± 5	5.95 ± 0.07	39.45 ± 0.01	2840
Alumina (6.37 mm)	3573 ± 10	6.37 ± 0.21	71.12 ± 0.03	5485
Alumina (3.09 mm)	3586 ± 10	3.09 ± 0.16	48.02 ± 0.02	1795
Porcelain	2407 ± 3	6.13 ± 0.19	51.13 ± 0.02	3798
Alginate (4.76 mm)	$1033 + 2$	4.76 ± 0.27	6.04 ± 0.00	348
Alginate (3.48 mm)	1023 ± 2	3.48 ± 0.17	3.36 ± 0.00	142
Alginate (2.66 mm)	1029 ± 1	2.66 ± 0.10	3.03 ± 0.00	98

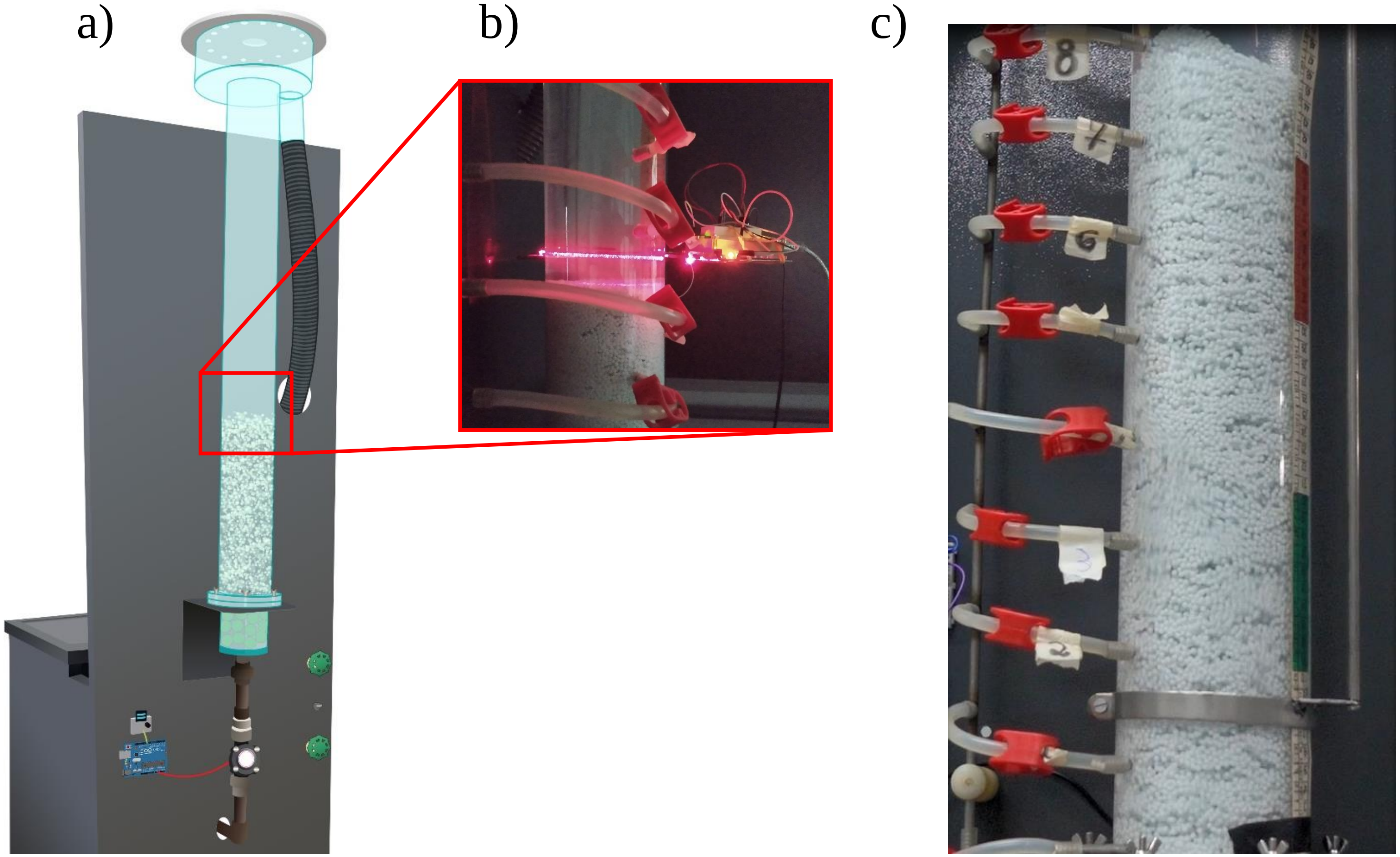


Figure 5.1 – Schematic representation of a) the experimental setup, b) the laser apparatus, and c) the fluidization experiment.

All experiments were carried out on the same fluidized bed setup, presented in Figure 5.1. The experimental setup, Figure 5.1a), comprises a liquid reservoir, a distributor, and the 10 cm diameter ($D = 10$ cm) vertical tube made of acrylic. A 1/2 Horsepower (HP) centrifugal pump pumps the liquid from the reservoir to the distributor. Below the distributor, there is a 10 cm height flow homogenizing portion full of packed glass beads. The particles are fluidized right above the distributor, and the fluid flows throughout the 1 m height columns and back to the reservoir behind it.

Two different Arduino-based Hall effect sensors are used to measure the inlet flow rate of the fluid. For the experiments with the heavy particles, the YF-G1 (2 – 100 $\mathrm{L\,min^{-1}}$) was used, while the YF-S201 (1 – 30 $\mathrm{L\,min^{-1}}$) was applied for the light ones. The inlet flow rate was controlled by an inlet and a bypass valve. Before each experiment, the inlet flow rate was set to the given values, and the sensor registered both the time average and standard deviation of the measurements.

Additionally, the fluid temperature was monitored using a Negative Temperature Coefficient sensor (NTC). The average temperature in the experiments was 30 °C. This value was used to find the viscosity (μ) and density (ρ_f) of the fluid. The same characteristics of the liquid were used in the entire data.

5.3.2 Measurement of the bed expansion

At each replicate, a known mass of particles (M_p) was introduced into the column, and fluidization was established at a given liquid flow rate. The mass of particles, inlet flow rate range, and the superficial Reynolds number ($\mathrm{Re} = \rho_f U D/\mu$) used in the fluidization experiments are shown in Table 5.2.

Table 5.2 – Experimental conditions.

Particle	**Total Mass of particles,** M_p (kg)	**Number of liquid flow rates (-)**	**Liquid flow rate range** ($\mathrm{L\,min^{-1}}$)	**Inlet Reynolds number, Re (-)**
ABS	2.0	11	25 – 75	6221-18664
Alumina (6.37 mm)	4.0	8	40 – 75	9954-18664
Alumina (3.09 mm)	4.0	8	40 – 75	9954-18664
Porcelain	2.7	11	25 – 75	6221-18664
Alginate (4.76 mm)	1.0	8	2 – 7	498 – 1742
Alginate (3.48 mm)	1.1	7	2 – 5	498 – 1244
Alginate (2.66 mm)	1.0	7	2 – 5	498 – 1244

Bed heights were measured using the apparatus shown in Figure 5.1b). It comprises a 5 V laser diode and a Light Dependent Resistor light sensor (LDR), pointed to each other at opposite sides of the equipment. Both the beam source and sensor are fixed to a small platform, which can be moved along the height of the equipment. The sensor measures the received light intensity and sends a proportional electric signal to an Arduino microcontroller, called analogic signal. Once the microcontroller is turned on, it starts registering the analogic signal. After 15 seconds of measurement, the microcontroller calculates an average among the registered values. In this work, this average value is referred to as laser (or beam) intensity.

The calibration of the sensor is done with the column full of water and with the particles in it. Before the experiments, the platform is positioned above the bed of particles so that the laser beam reaches the sensor without interruptions (as depicted in Figure 5.1b). This returns a high beam intensity to the microcontroller, which is the intensity in the absence of particles. After this value is registered, the platform is positioned in the middle of the bed of particles. Because the particles are opaque, the light from the laser does not reach the sensor, and the measured beam intensity is low. The average between both the high and low values is considered the threshold value. This value is used to identify the bed height.

During the fluidization experiments, the height of the laser was increased until the measured beam intensity was within 10% of the threshold value. This defined the height of the bed (H). This measurement method was highly reproducible, presenting a maximum standard deviation of 2% of the average. With the bed height H, the bed porosity ($\bar{\varepsilon}_f$) was calculated using Equation 5.5:

$$\bar{\varepsilon}_f = 1 - \frac{M_p/\rho_p}{\pi D^2 H} \tag{5.5}$$

The procedures adopted, the parameters measured and the techniques used in the present work are summarized in the workflow presented in Figure 5.2.

5.3.3 Bed expansion estimation

Expansion of the bed was estimated for all experimental conditions by two different methods. These values were further compared to the bed expansions obtained experimentally.

5.3.3.1 Richardson-Zaki equation (R-Z)

R-Z (Equation 5.1) was used to estimate the bed expansion given the superficial velocity of the fluid U and the experimental terminal settling velocity of the particles

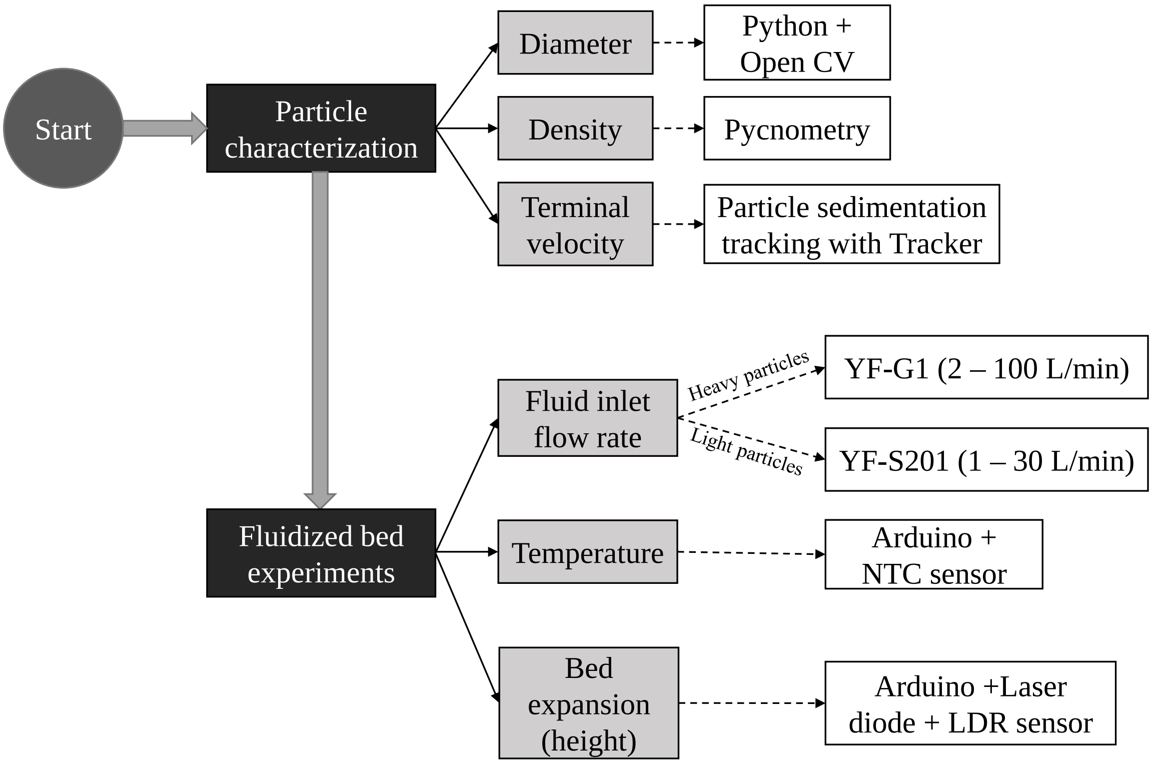


Figure 5.2 – Workflow of the experimental procedure.

U_0. In this equation, the values of k and n were determined by the correlations proposed by Khan and Richardson (1989) (Equations 5.2 and 5.4). Since some approaches to this equation neglect the parameter k, the expansion was also estimated for $k = 1$.

5.3.3.2 Drag correlations

An alternative to the traditional R-Z was applied in the present work using drag correlations in the literature. Since the drag force acting over a single particle with volume V_p is

$$F_{D0} = V_p \left(\rho_p - \rho\right) g \tag{5.6}$$

and since the average drag force F_D can be estimated as a function of $\bar{\varepsilon}_f$ and Re_p using drag correlations in the literature, ε_f can be numerically estimated by minimizing $F_D - F_{D0}$. In the present work, this procedure was tested for the drag correlations proposed by Felice (1994), Beetstra *et al.* (2007), and Rong *et al.* (2013) (referred to as Di Felice, Beetstra, and Rong). Both Di Felice and Rong correlations follow a more traditional approach to the drag, which considers that, for a single particle inside the bed

$$F_D = \frac{1}{8}\rho U^2 C_{D0} \pi d_p^2 G(\varepsilon_f, Re_p) \tag{5.7}$$

where C_D is the drag coefficient. According to DallaValle (1948):

$$C_{D0} = \left(0.63 + \frac{4.8}{Re_p^{0.5}}\right)^2 \tag{5.8}$$

In Equation 5.7, $G(\varepsilon_f, Re_p) = \varepsilon_f^{-\beta}$. Di Felice and Rong proposed different correlations to calculate β, respectively written as

$$\beta = 3.7 - 0.65\left[-\frac{(1.5 - \log Re_p)^2}{2}\right] \tag{5.9}$$

$$\beta = 2.65(\varepsilon_f + 1) - (5.3 - 3.5\varepsilon_f)\varepsilon_f^2\left[-\frac{(1.5 - \log Re_p)^2}{2}\right] \tag{5.10}$$

As a counterpart, Beetstra follows an alternative approach, proposing the following correlation.

$$\begin{aligned} F_D = {} & \frac{10\,(1-\varepsilon_f)}{\varepsilon_f^2} + \varepsilon_f^2\left(1 + 1.5\varepsilon_f^{0.5}\right) \\ & + \frac{0.413 Re_p}{24\varepsilon_f^2}\left[\frac{(1/\varepsilon_f) + 3\varepsilon_f\,(1-\varepsilon_f) + 8.4 Re_p^{-0.343}}{1 + 10^{3\left(1-\varepsilon_f\right)} Re_p^{-\left(1+4\left(1-\varepsilon_f\right)\right)/2}}\right] \end{aligned} \tag{5.11}$$

In the present work, these 3 drag correlations are applied to find $\bar{\varepsilon}_f$ such that $|F_D - F_{D0}| < 10^{-10}$.

5.4 Results and Discussion

The precise description of the fluidized bed expansion through simple equations is crucial to designing, maintaining, and optimizing LSFBs. For instance, the digestion of the biomass in wastewater and the yield of biogas production in LSFB bioreactors are directly related to the liquid fraction at the bed region. Overpredicting the bed expansion can lead to low conversions and problems related to controlling the biofilm thickness, while the underestimation of this parameter will lead to the washout of the particles. The bed porosity is directly related to velocity control. Using the described methodology, the bed porosity as a function of the inlet velocity was measured and estimated. The results for each particle are presented in Figure 5.3.

5.4.1 Bed expansion estimation using R-Z

Figure 5.3 presents the bed expansion as a function of the dimensionless superficial velocity U/U_0. The logarithmic scale highlights the linear trend amongst the experimental data. The error bars correspond to one standard deviation. As shown in Figure 5.3, the standard deviation was very small, highlighting the reproducibility of the experiments. The highest standard deviation among the experiments was found for the Alginate (2.66 mm) particles in the experiments with the lowest inlet flow rate (inlet velocity of 0.008644 $\mathrm{m\,s^{-1}}$), corresponding to 1.89% of the average measured porosity.

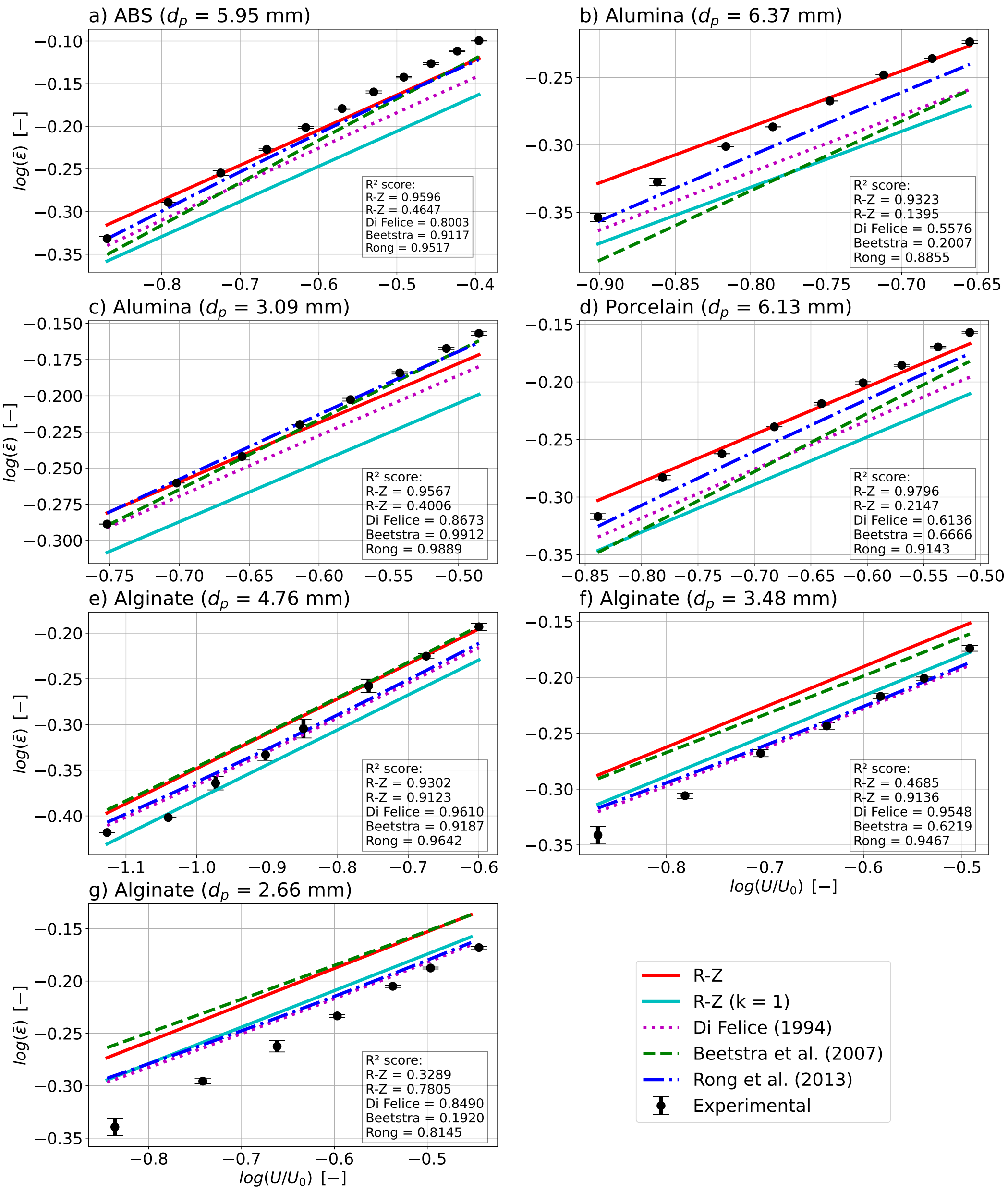


Figure 5.3 – Bed voidage vs. U/U_0 in logarithmic scale ± standard deviation for a) ABS, b) Alumina (6.37 mm), c) Alumina (3.09 mm), d) Porcelain, e) Alginate (4.76 mm) Alginate (3.48 mm), and g) Alginate (2.66 mm) particles.

According to Epstein (2003b), the correlation should be accurate for particles with $0.01 < \mathrm{Re}_{p,0} < 7000$ and $0.001 < d_p/D < 0.2$; that is, all particles in this work. Nonetheless, as shown by the coefficient of determination (R^2), the R-Z using Khan and Richardson (1989) k presented better fitting for the higher diameter particles (Figure 5.3a), b), and d), respectively). This suggests that the k may be less sensitive to d_p/D than indicated by Khan and Richardson (1989), which could justify the better fitting between the experiments and R-Z with $k = 1$ for small and intermediary alginate beads (Figure 5.3f) and g), respectively).

However, comparing the R^2 for R-Z in Figures 5.3c) and f), it is noted that R-Z fitted better to the smaller alumina (3.09 mm) than to the slightly larger alginate particle (3.48 mm). Since the density ratio between the alumina and alginate is 3 to 1, this shows that the value of k is also dependent on the pair fluid-particle characteristics, **i.e.**, Ar or $\mathrm{Re}_{p,0}$, for instance. Following the principle of Fidleris and Whitmore (1961) for a single particle, Dharmarajah and Cleasby (1986) showed that the wall effect is not negligible for pilot-scale experiments, mainly if scale-up is intended. The authors highlight the importance of $\mathrm{Re}_{p,0}$ together with d_p/D to the wall effect. More recently, Akgiray and Soyer (2006) highlighted the same feature based on new experiments and data in the literature. The correlations proposed in both works are detached from the traditional R-Z correlation and are currently not as prominent.

By contrast, Rapagnà *et al.* (1989) and Epstein (2003b) proposed two correlations to estimate k based exclusively on $\mathrm{Re}_{p,0}$ with good fitting to the experimental data, but with limited ranges of $\mathrm{Re}_{p,0}$. According to Epstein (2003b), Equation 5.4 can be used for a wide range of $\mathrm{Re}_{p,0}$. The author recommends the use of Rapagnà *et al.* (1989) correlation when it gives a lower value than Equation 5.4 for particles presenting $100 < \mathrm{Re}_{p,0} < 1000$. The same for the equation proposed by the author for $35 < \mathrm{Re}_{p,0} < 100$, whereas this was not the case with the particles in this work.

To find an experimental value for k, a linear regression was applied to the experimental data on a logarithmic scale. In Figure 5.4, the values of k obtained from the regression and calculated using the Khan and Richardson (1989) correlation are shown.

The approach to R-Z proposed by Epstein (2003b) was recently applied by Lopes *et al.* (2018) to predict the expansion of a fluidized bed of particles like the one used in the present work. According to the authors, the lack of agreement between the experimental and the predictions using the k of Khan and Richardson (1989) could be explained by the mechanical properties of the particles, *i.e.*, the soft and light particles tend to agglomerate due to their low effective coefficient of restitution and Stokes number, which would increase the drag acting over the particles. The results in the present work bring up a new hypothesis for such deviations, since, as can be seen in Figure 5.4, k was also higher than predicted for the rigid porcelain, ABS, and larger alumina particles.

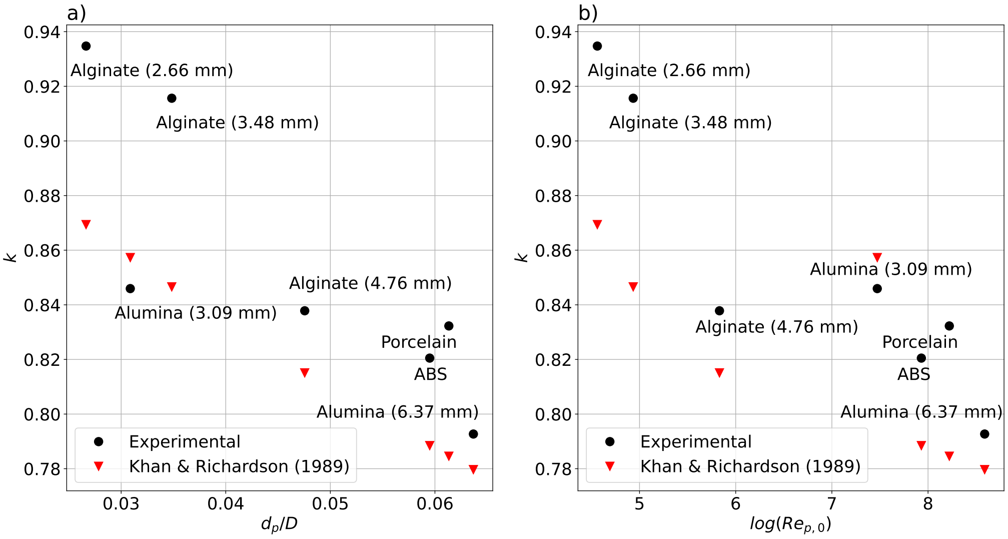


Figure 5.4 – k as a function of a) d_p/D, and b) $ln(\mathrm{Re}_{p,0})$

As can be noted from Figure 5.4, the Khan and Richardson (1989) correlation is more likely to accurately predict k for higher values of $\mathrm{Re}_{p,0}$. On the other hand, the highest relative deviation between the experimental and predicted k is 6.99% for the smaller alginate particle. It shows that the estimation of the bed porosity using R-Z is notably sensitive to parameter estimation, especially considering lower values of d_p/D. This lack of robustness makes the pilot-scale experiments less likely to represent full-scale equipment and the use of R-Z correlation may lead to incorrect prediction of the bed expansion.

5.4.2 Bed expansion estimation using drag correlations

R-Z is particularly powerful with $k = 1$ to represent the drag acting over the particles at given porosities. As discussed in the Introduction section, several authors choose to neglect the effect of k (= 1) for various applications, either by justifying that the wall effect is negligible for full-scale equipment or to reduce the number of parameters. In all works, good fittings are achieved by the correlations proposed to n. For instance, Felice (1994), Beetstra *et al.* (2007), Rong *et al.* (2013), and Mazzei and Lettieri (2007) developed drag correlations by decomposing forces and isolating the effect of the drag from the summation of particle-fluid forces (buoyancy, pressure gradient, shear stress, and others). More recently, with the advance of computational power, these and other correlations based on the same approach are applied in numerical simulations of multiphase systems (such as cyclones, fluidized beds, and others), presenting good agreement with empirical data.

In the present work, the numerical approach was simplified to determine the expansion of the bed based on consolidated drag correlations. In this approach, $\bar{\varepsilon}_f$ was estimated by minimizing the objective function $F_D - F_{D0}$, that is, the absolute difference between Equation 5.7 and the drag correlations. The results of this approach for the Felice (1994), Beetstra *et al.* (2007), and Rong *et al.* (2013) drag correlations (called from this point Di Felice, Beetstra, and Rong correlations, respectively) are presented in Figure 5.3. A better comparison can be done using the R^2 of the adjustments, which is presented in Figure 5.5.

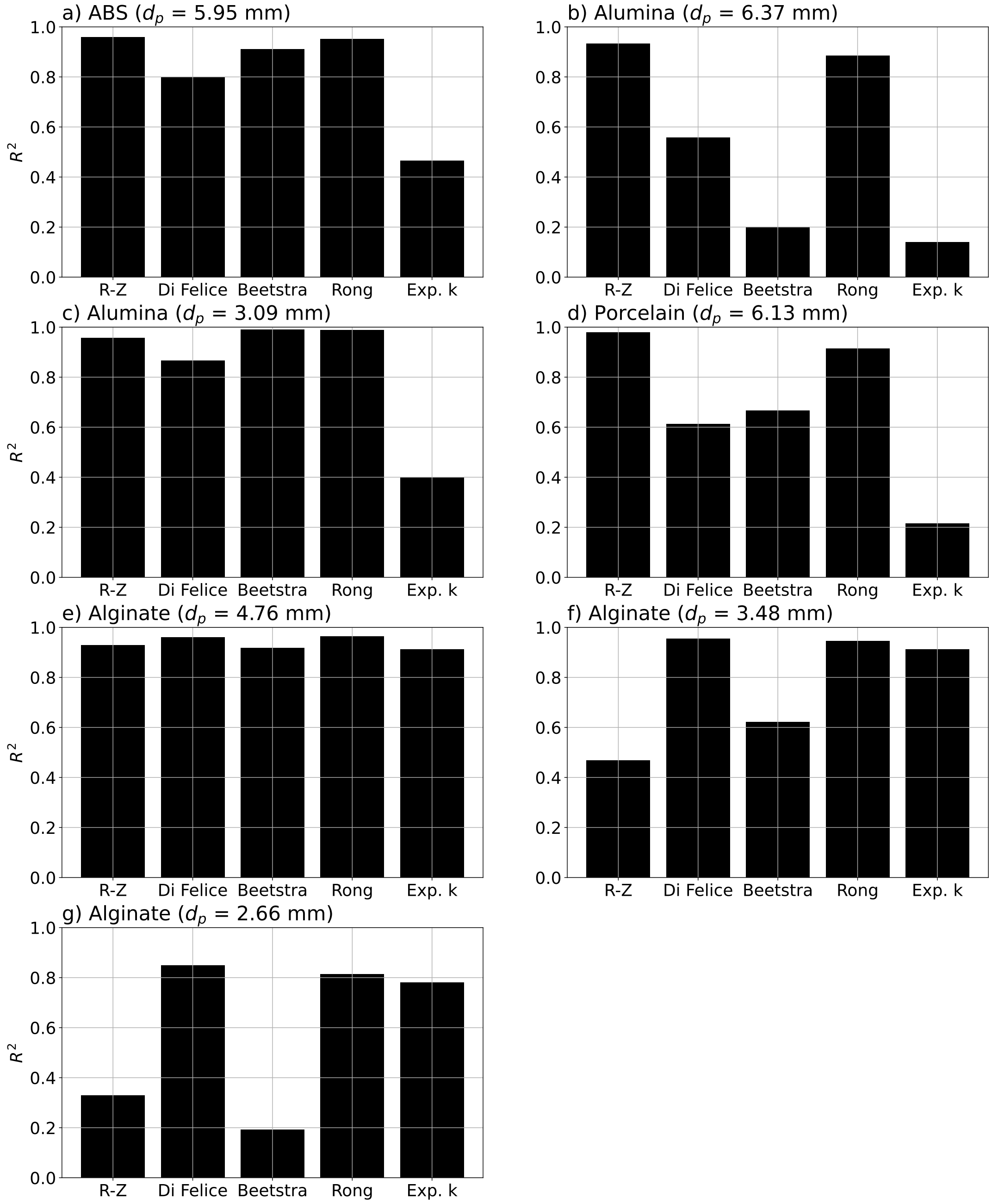


Figure 5.5 – R^2 of the different methods of bed expansion estimation for a) ABS, b) Alumina (6.37 mm), c) Alumina (3.09 mm), d) Porcelain, e) Alginate (4.76 mm) Alginate (3.48 mm), and g) Alginate (2.66 mm) particles.

The difference between Di Felice and Rong is attributed to the way $\bar{\varepsilon}_f$ is calculated for each correlation. Di Felice and Rong share the same formulation for F_D, with C_{D0} calculated by the well-established correlation proposed by DallaValle (1948). In Di Felice correlation, β is not a function of ε_f, which was shown by many authors, including Beetstra *et al.* (2007) and Rong *et al.* (2013), to lead to significant uncertainties on the average drag estimation. Rong correlation includes the effect of the porosity on ε_f by fitting data obtained by Lattice-Boltzmann simulations. In verification with experimental results, Rong correlation reduced the overall deviation by half (around 15%) compared to the other correlations tested in this work (25 to 30%). In the case of Beetstra correlation, besides accounting for both the Re_p and ε_f, the completely different formulation for C_{D0}, which is different from the one of Dallavalle, may be the cause for the additional inaccuracies.

It is important to note that the Rong drag correlation does not take the wall effect into account. Since it has better agreement than R-Z, Rong correlation is more capable of capturing the dependence of the expansion with the fluid-particle pair characteristics (i.e., Ar or $\text{Re}_{p,0}$) than the Khan and Richardson (1989) correlation for k. This difference between Rong correlation and R-Z highlights the dependence between k and the fluid-particle interactions beyond wall effects. Most likely, both d_p/D and Ar (or $\text{Re}_{p,0}$) should be combined to find better correlations for k, as shown by Dharmarajah and Cleasby (1986) and Akgiray and Soyer (2006). Since such a correlation is not yet established and the Rong correlation has been tested for a wide range of regimes, the fast and computational cheap minimization approach using the Rong correlation is a good alternative based on the present results.

5.4.3 Extrapolation of the proposed method

To assess its extent, the proposed numerical method was applied and compared to results obtained by Lopes *et al.* (2018). The work was chosen due to the similarity between the experimental setups. In their study, particles with different characteristics (Table 5.1) were fluidized by water in a 19-cm-diameter column (almost twice the diameter of the column used in the present work). The liquid fluidized bed expansion was also investigated for a variety of particles and regimes, with inlet Reynolds numbers between 2940 and 11779.

Table 5.3 – Experimental conditions (LOPES *et al.*, 2018).

Particle	**Density,** ρ_p ($\mathrm{kg\,m^{-3}}$)	**Diameter,** d_p (mm)	**Terminal settling velocity,** U_0 ($\mathrm{cm\,s^{-1}}$)	**Terminal Reynolds number,** $\mathrm{Re}_{p,0}$ **(-)**
Glass	2500 ± 10	1.85 ± 0.10	25.7 ± 1.8	357
ABS (uncoated)	1965 ± 2	5.87 ± 0.01	38.4 ± 2.8	1859
ABS (uniformly coated)	1913 ± 10	5.96 ± 0.01	38.7 ± 1.6	1898
ABS (non-uniformly coated)	1919 ± 12	5.94 ± 0.01	36.3 ± 2.0	1777
Alginate (Calcium)	1228 ± 8	4.88 ± 0.32	14.0 ± 1.2	647
Alginate (Cobalt)	1110 ± 5	5.83 ± 0.45	12.1 ± 1.2	619
Starch Pearls	1080 ± 6	$12.18\pm$ 0.56	14.3 ± 0.6	1673

For this comparison, the Rong correlation was chosen for presenting the best fitting with the experimental data of this work. Figure 5.6 shows the coefficient of determination R^2 of the adjustment between the experimental results of Lopes *et al.* (2018) and estimations using the proposed approach and R-Z (also extracted from Lopes *et al.* (2018)).

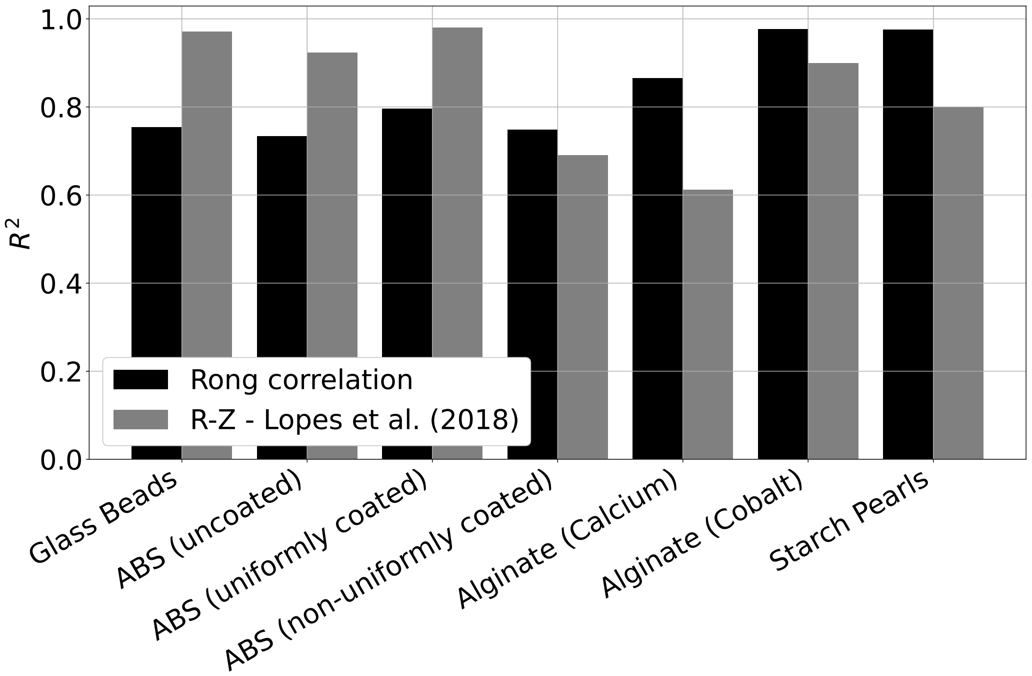


Figure 5.6 – R^2 of the fitting between experiments by Lopes *et al.* (2018) and estimations using R-Z and Rong drag correlation.

The average R^2 for R-Z and the numerical approach using Rong correlation were 0.8393 and 0.8356, respectively. Although R-Z was better to represent the data obtained for the glass beads and some of the ABS particles, the proposed approach presented better results for most of the particles (4 out of the 7) investigated. Additionally, the lowest R^2 found using the proposed drag correlation approach was 0.7339 (uncoated ABS), while for R-Z, it was 0.6118 (Alginate made with calcium chloride). These results highlight not only the precision but also the scalability of the method, given that the diameter of the equipment (0.1905 m) is almost two times the one described in the present work.

5.5 Conclusions

The results of bed expansion estimated from the Richardson-Zaki equation were compared to experimental data obtained for several inlet velocities, using 7 particles with significantly different diameters, densities, Archimedes numbers, and terminal velocities. The aim was to determine if R-Z can predict the bed expansion for a wide variety of particles and flow regimes. Since the literature presents good agreement about the R-Z index n, the present work focuses the discussion on the parameter k. It was shown that the use of the parameter k is strongly case dependent, with no specific criterium, and still needs further investigation.

Apart from that, an alternative approach to predict bed expansion was tested, presenting promising results. The minimization approach using the drag correlations showed better agreement than R-Z for 4 out of the 7 particles. Among the drag correlations, the one proposed by Rong showed the best fitting, being consistently accurate for the wide range of particles considered in the present work. This approach was also validated against the experimental data by Lopes *et al.* (2018).

The main drawback of the proposed method is that the force balance is applied to the entire bed, without taking the particles distribution into account. This means that it will be more precise the more uniform the bed is, which is true for most monodispersed liquid fluidized beds. Additionally, the proposed method is iterative, meaning that it is not as directly applicable as R-Z. Nonetheless, the use of fully black box correlations to calculate the parameters n and k leads to unpredictability, which is not the case in the proposed method.

It is important to note that none of the drag correlations take the wall effects into account. Considering that the Rong correlation presented better accuracy amongst the results, the correction due to the wall effects using the parameter k may not be the key to the prediction of the bed expansion in pilot scale experiments or even the right correction for the scale-up. This fact highlights the need for more robust methods to predict bed expansion, and the minimization approach presented in this paper can be a good alternative for this due to its negligible computational cost and high accuracy. Additionally, since the method is based on the force balance applied to the bed of particles and it was validated for a wide range of regimes, this approach can be easily and reliably applied to pairs other than particle-water, especially in the broad investigated range of Reynolds and Archimedes numbers. The proposed method using the Rong drag correlation improves the predictability of the liquid fluidized bed operating conditions, consequently spreading the fluidization technologies to industrial applications in which they have a high potential such as wastewater treatment and biogas production plants.

5.6 Acknowledgments

Thanks are due to São Paulo Research Foundation (FAPESP, grant #2019/19173-9, grant #2020/14567-6, and grant #2019/25146-4); the Funding Authority for Studies and Projects of the Ministry of Science, Technology and Innovation (FINEP/MCTI) by means of the Program of Human Resources of the Brazilian National Petroleum Agency (PRH-ANP/MCTI); and the Brazilian National Council for Scientific and Technological Development (CNPq, process number 408618/2018-3) for financial support. The authors also thank Federal University of São Carlos (UFSCar) for the infrastructure and Polytechnique Montréal for the technical support. This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Finance Code 001. Bruno Blais acknowledges the financial support from the Natural Sciences and Engineering Research Council of Canada (NSERC) through the RGPIN-2020-04510 Discovery grant.

CHAPTER 6

ARTICLE 2: IN-DEPTH VALIDATION OF UNRESOLVED CFD-DEM SIMULATIONS OF LIQUID-SOLID FLUIDIZED BEDS

Victor Oliveira Ferreira, Toni El Geitani, Daniel Silva Junior, Bruno Blais, Gabriela Cantarelli Lopes

Published in Powder Technology, volume 426, 2023

6.1 Abstract

In this work, we assess the accuracy of the unresolved CFD-DEM method by confronting simulated and experimental results of a pilot-scale cylindrical liquid-solid fluidized bed. We used two types of particles with significantly different densities (1029 and 3586 $\mathrm{kg\,m^{-3}}$), allowing for testing a wide range of flow regimes. Comparisons between experimental and simulated particles' dynamics show that the Saffman lift force is essential in predicting the physical dispersion of particles, preventing unphysical plumes. Simulations precisely reproduced the overall force balance in the system. The bed expansion as a function of the inlet velocity results shows excellent agreement between simulations and experiments. High agreement between experiments and simulations is observed for the drag models proposed by Di Felice, Rong, and Beetstra but not for Gidaspow. The results validate the use of unresolved CFD-DEM to simulate the liquid-solid fluidized bed.

6.2 Introduction

Liquid-solids fluidized beds (LSFB) are applied in a wide range of unit operations, including heterogeneous reactions, crystallization, and classification (EPSTEIN, 2003a). Its advantages compared to fixed beds include the agitation of particles and the possibility of increasing operational flow rates (way above the minimum fluidization velocity) at a constant pressure drop. Together, the agitation of particles and high flow rates increase the heat and/or mass transfer between phases, implying, for example, better temperature control and higher conversion. Additionally, LSFB tend to form well-dispersed beds, which, together with the high degree of agitation, preserve the uniformity of products.

The design of LSFB relies mainly on empirical correlations with little or no phenomenological background. For example, one popular correlation in the design of LSFB is the Richardson-Zaki equation (R-Z) (RICHARDSON; ZAKI, 1954):

$$\frac{U}{U_0} = \bar{\varepsilon}_f^n \tag{6.1}$$

where U is the fluid inlet velocity, U_0 is the free settling terminal velocity of the particles, $\bar{\varepsilon}_f$ is the fluid (void) fraction in the bed of particles, and n is an empirical parameter of the equation (RICHARDSON; ZAKI, 1954). The R-Z equation predicts the global porosity of the bed, given the characteristics of both flow and particles. However, the precision of predictions is closely dependent on n and U_0, determined by correlations in the literature (KHAN; RICHARDSON, 1989; RAPAGNà *et al.*, 1989; EPSTEIN, 2003b) or experimental efforts, both limited to a given range of application. Additionally, the correlation does not predict bed dispersion, preferential path generation, or other localized phenomena. This lack of predictability restricts the applicability of LSFB.

In this sense, Computational Fluid Dynamics (CFD) plays an essential role in understanding such particle-fluid flows. The Volume-Averaged Navier-Stokes (VANS) equations (ANDERSON; JACKSON, 1967) can be numerically solved to provide detailed descriptions of multiphase flows. Authors such as Gidaspow (1994) used this concept to model both solid and fluid phases as continuums. This technique is called the Two-Fluid Model (TFM) and has been applied by many authors since then (CORNELISSEN *et al.*, 2007; REDDY; JOSHI, 2009; KOERICH *et al.*, 2018; METOLINA; LOPES, 2019; ISLAM; NGUYEN, 2021). The main disadvantage of this technique is that the solid phase is modeled as a continuous phase, in which all particles' movement and interactions are represented by spatial averages. Consequently, it does not evaluate particles' individual behavior.

Alternatively, the unresolved coupling between CFD and the Discrete Elements Method (CFD-DEM) (ZHOU *et al.*, 2010; BLAIS *et al.*, 2016; BéRARD *et al.*, 2020) describes the solid phase with significantly more details. This is because the DEM (ZHU *et al.*, 2007; GOLSHAN *et al.*, 2022) models the motion of the solid phase at the scale of individual particles. This means that each particle has its own movement described by Newton's second law, individually accounting for the collisional, non-collisional (such as gravity), and other forces. In the unresolved CFD-DEM coupling, the mesh elements are larger than particles (preferably three to four times the size of particles) (PENG *et al.*, 2014), and the fluid properties are described as volume averages within averaging volume (generally mesh elements). The use of volume-averaged properties to describe the fluid and its coupling with a very detailed description of the particle dynamics is key to obtaining accurate results at a reasonable computational cost.

Several authors applied unresolved CFD-DEM (RENZO; MAIO, 2007; RENZO *et al.*, 2011; BLAIS *et al.*, 2016; BLAIS *et al.*, 2017; BLAIS; BERTRAND, 2017; MAESTRI *et al.*, 2019; NIJSSEN *et al.*, 2020; CAMPOS *et al.*, 2022; PICABEA *et al.*, 2022) to simulate different particle-liquid systems. These studies present promising results, showing that the technique has great potential for these systems. For example, Renzo and Maio (2007) report realistic homogeneity in glass beads fluidized by water simulation using unresolved CFD-DEM. They also obtained good agreement between the fluid velocity and estimation of the velocity propagation for the transient bed expansion.

One significant challenge is the choice of models for the solid-fluid interaction. In the unresolved CFD-DEM approach, we use correlations to represent drag, lift, (Basset) history, and virtual mass forces. The accuracy of the simulations depends on the accuracy of these correlations. Nijssen *et al.* (2020) showed that forces other than drag can have a non-negligible impact on particles' dynamics of a liquid-solid fluidized bed simulated using unresolved CFD-DEM. In their case, the solids mixing was 20% slower due to the damping caused by the dissipation forces. Nevertheless, the lack of experimental techniques with enough resolution to accurately measure each force separately reduces the reliability of such correlations. That is an important reason why works using VANS frequently neglect forces other than drag. Another reason for the simplification is that most works using unresolved CFD-DEM involve gas-solid systems. In these systems, drag generally dominates other forces. Nonetheless, for liquid-solid systems, this assumption is debatable (LIANG; MICHAELIDES, 1992; NIJSSEN *et al.*, 2020).

Although significant results have been reported for solid-liquid systems, only a few liquid-solid fluidized bed studies comparing experiments and unresolved CFD-DEM simulations are found in the literature (HAGER *et al.*, 2014; PICABEA *et al.*, 2022). The present research aims to validate the unresolved CFD-DEM method on the simulation of LSFB using the finite elements method (FEM). To do so, we confront simulation results with experiments of a pilot-scale cylindrical liquid-solid fluidized bed. The experiments were carried out for alginate and alumina particles, chosen because of their density difference (1029 and 3586 $\mathrm{kg\,m^{-3}}$, respectively), which allowed for the study of a wide range of fluidization regimes. First, we discuss the importance of the Saffman lift force and the smoothing of the void fraction field on the dispersion of particles and flow pattern. After this, we investigate how different mesh topologies may affect this result. Then, we assess the accuracy of the simulation predictions through pressure drop and bed expansion results. Finally, we vary drag force correlations and compare them to experimental results to assess the impact of each aspect on the accuracy of the simulations.

6.3 CFD-DEM Formulation

The present section briefly describes the equations used within the unresolved CFD-DEM model. For a more detailed description, the authors refer the reader to Zhou *et al.* (2010), Bérard *et al.* (2020), and Gidaspow (1994) for the VANS equations, and to Zhu *et al.* (2007), Blais *et al.* (2019), and Golshan *et al.* (2022) for the DEM modeling.

The simulations were carried out using Lethe (BLAIS *et al.*, 2020; GOLSHAN *et al.*, 2022; GEITANI; BLAIS, 2023b), an open-source CFD and DEM software with resolved and unresolved CFD-DEM coupling capabilities. We refer the reader to Geitani and Blais (2023b), Geitani *et al.* (2023a) for further details about the numerical strategies, including derivation of the weak form of VANS equations, stabilization strategies, and void fraction smoothing.

6.3.1 Solid phase modeling

In the DEM approach, we apply Newton's second law of motion on each moving particle to calculate both its linear and angular momentum, respectively represented by Equations 6.2 and 6.3:

$$m_i \frac{\mathrm{d}\boldsymbol{v}_i}{\mathrm{d}t} = \sum_{j=1, j\neq i}^{N_p} (\boldsymbol{f}_{c,ij}) + \sum_{j, j\neq i}^{N_p} (\boldsymbol{f}_{nc,ij}) + \boldsymbol{f}_{pf,i} + \boldsymbol{f}_{g,i} \tag{6.2}$$

$$I_i \frac{\mathrm{d}\boldsymbol{\omega}_i}{\mathrm{d}t} = \sum_{j, j\neq i}^{N_p} (\boldsymbol{M}_{t,ij} + \boldsymbol{M}_{r,ij}) \tag{6.3}$$

where the subscripts i refer to the i^{th} particle interacting with the j^{ith} particle, N_p is the number of particles, $\boldsymbol{v}$ represents the velocity of the particle, $\boldsymbol{f}_c$ the contact forces, $\boldsymbol{f}_{nc}$ the non-contact forces, $\boldsymbol{f}_g$ the gravity force and $\boldsymbol{f}_{pf}$ the sum of forces exerted over particles by the fluid phase, while $\boldsymbol{M}_t$ and $\boldsymbol{M}_r$ are the tangential and rolling friction torques. In this work, the non-contact forces are neglected due to their small magnitude compared with the contact and coupling forces.

We modeled the collisions using the soft-sphere collision model proposed by Cundall and Strack (1979). In Equation (6.2), the contact forces can be decomposed in normal ($\boldsymbol{f}_{cn,ij}$) and tangential ($\boldsymbol{f}_{ct,ij}$) contact forces, such that:

$$\boldsymbol{f}_{c,ij} = \boldsymbol{f}_{cn,ij} + \boldsymbol{f}_{ct,ij} = -k_{n,ij}\boldsymbol{\delta}_{n,ij} - \gamma_{n,ij}\dot{\boldsymbol{\delta}}_{n,ij} - k_{t,ij}\boldsymbol{\delta}_{t,ij} - \gamma_{t,ij}\dot{\boldsymbol{\delta}}_{t,ij} \tag{6.4}$$

in which the subscript n and t refer to normal and tangential directions, respectively. The overlap ($\boldsymbol{\delta}$) and its derivative with respect to time ($\dot{\boldsymbol{\delta}}$) are both used to calculate the contact forces. In Equation (6.4), the tangential overlap ($\boldsymbol{\delta}_{t,ij}$) is limited by Coulomb's law, written as:

$$\boldsymbol{\delta}_{t,ij} \leq -\mu_{s,ij} \left|\boldsymbol{f}_{cn,ij}\right| \frac{\boldsymbol{\delta_{n,ij}}}{\left|\boldsymbol{\delta_{n,ij}}\right|} \tag{6.5}$$

For both normal and tangential directions, k and γ are the stiffness and the damping coefficients of the colliding pair, calculated based on the mechanical properties of the particles, which are the effective Young's Modulus (Y), the coefficient of restitution (e) and Poisson ratio (ν). Table 6.1 summarizes the Equations for stiffness and damping calculation.

Using $\boldsymbol{f}_{cn,ij}$ and $\boldsymbol{f}_{ct,ij}$, the tangential and rolling friction torques ($\boldsymbol{M}_{t,ij}$ and $\boldsymbol{M}_{r,ij}$, respectively) can be calculated by:

$$\boldsymbol{M}_{t,ij} = \boldsymbol{r}_i \times (\boldsymbol{f}_{ct,ij}) \tag{6.6}$$

$$\boldsymbol{M}_{r,ij} = -\mu_{r,ij} \left|\boldsymbol{f}_{ct,ij}\right| \frac{\boldsymbol{\omega}_{ij}}{\left|\boldsymbol{\omega}_{ij}\right|} R^*_{ij} \tag{6.7}$$

where the coefficient of rolling friction ($\mu_{r,ij}$) and the equivalent radius (R^*_{ij}) are calculated by the Equations in Table 6.1.

In this work, we consider large particles ($d_p > 1\,\text{mm}$), and we can neglect the non-contact forces $\boldsymbol{f}_{nc,ij}$ since they are small compared to the contact $\boldsymbol{f}_{c,ij}$ and particle-fluid forces $\boldsymbol{f}_{pf,i}$. The source of the latter is the interaction between the liquid and the solid particles, which is calculated by:

$$\boldsymbol{f}_{pf,i} = \boldsymbol{f}_{d,i} + \boldsymbol{f}_{\nabla p,i} + \boldsymbol{f}_{\nabla\cdot\boldsymbol{\tau},i} + \boldsymbol{f}_{Ar,i} + \boldsymbol{f}''_i \tag{6.8}$$

for particle i, where $\boldsymbol{f}_{d,i}$ represents the drag, $\boldsymbol{f}_{\nabla p,i}$ is the force due to the pressure gradient, $\boldsymbol{f}_{\nabla\cdot\boldsymbol{\tau},i}$ is the force due to the shear stress with the fluid, and $\boldsymbol{f}''_i$ stands for the rest of forces less frequently considered due to their magnitude, represented by:

$$\boldsymbol{f}''_i = \boldsymbol{f}_{\text{vm},i} + \boldsymbol{f}_{\text{B},i} + \boldsymbol{f}_{\text{Mag},i} + \boldsymbol{f}_{\text{Saff},i} \tag{6.9}$$

Table 6.1 – DEM equations

Property	**Equation**
Mass of particle i	m_i
Radius of particle i	R_i
Poisson ratio of a particle i	ν_i
Young Modulus of a particle i	Y_i
Coefficient of restitution of a pair ij	e
Distance between a particle i and contact point	$\boldsymbol{r}_i$
Equivalent mass	$\frac{1}{m^*_{ij}} = \frac{1}{m_i} + \frac{1}{m_j}$
Equivalent radius	$\frac{1}{R^*_{ij}} = \frac{1}{R_i} + \frac{1}{R_j}$
Equivalent Young's modulus	$\frac{1}{Y^*_{ij}} = \frac{1-\nu_i^2}{Y_i} + \frac{1-\nu_j^2}{Y_j}$
Equivalent shear modulus	$\frac{1}{G^*_{ij}} = \frac{2(2-\nu_i)(1+\nu_i)}{Y_i} + \frac{2(2-\nu_j)(1+\nu_j)}{Y_j}$
Normal stiffness	$k_{n,ij} = \frac{4}{3} Y^*_{ij} \sqrt{R^*_{ij} \boldsymbol{\delta_{n,ij}}}$
Tangential stiffness	$k_{t,ij} = 8 G^*_{ij} \sqrt{R^*_{ij} \boldsymbol{\delta_{n,ij}}}$
Normal damping	$\gamma_{n,ij} = -2\sqrt{\frac{5}{6}} \frac{\ln(e)}{\sqrt{\ln^2(e+\pi^2)}} \sqrt{\frac{2}{3} k_{n,ij} m^*_{ij}}$
Tangential damping	$\gamma_{t,ij} = -2\sqrt{\frac{5}{6}} \frac{ln(e)}{\sqrt{\ln^2(e+\pi^2)}} \sqrt{k_{t,ij} m^*_{ij}}$

where the terms of $\boldsymbol{f}_i''$ represent virtual mass, Basset, Magnus lift, and Saffman lift forces, respectively. In the present work, we evaluated the influence of the Saffman lift force in the LSFB. Magnus lift, Basset, and virtual mass forces were also neglected due to the same assumptions used by Renzo and Maio (2007), namely that it is not possible to extrapolate the expression for these forces, which were mathematically derived assuming very dilute systems, to a highly concentrated contact-driven system. The remaining terms of Eq. 6.8 are described in Section 6.3.3.

6.3.2 Liquid phase modeling

The equations for fluid-particle systems were first developed considering the two-fluid model, which assumes two continuous phases. The idea is that each phase has a fraction of the cell properties proportional to the volume it occupies. In this sense, the Navier-Stokes equations represent not only the fluid phase but a volume average of the mixture. The Volume-Averaged Navier-Stokes Equations are used in TFM and unresolved CFD-DEM to represent the fluid phase.

The VANS equations can be expressed in two forms as described by Zhou *et al.* (2010), referred to as model A and B (Sets II and I, respectively), to represent the fluid phase. In both forms, the continuity equation for an incompressible flow is written as:

$$\frac{\partial \varepsilon_f}{\partial t} + \nabla \cdot (\varepsilon_f \boldsymbol{u}) = 0 \tag{6.10}$$

where $\boldsymbol{u}$ is the fluid velocity vector, and ε_f is the void (fluid volume) fraction, calculated by:

$$\varepsilon_f = 1 - \frac{\sum_i^{N_{p,C}} V_{p,i}}{\Delta V_{\Omega_C}} = 1 - \varepsilon_p \tag{6.11}$$

where $N_{p,C}$ is the number of particles inside the cell Ω_C with volume ΔV_{Ω_C}

The difference between models A and B comes from the way we represent pressure and shear stress. In model A, both terms are assumed to be in both fluid and solid phases, which makes the momentum equation take the form of:

$$\rho_f \left[\frac{\partial \varepsilon_f \boldsymbol{u}}{\partial t} + \nabla \cdot (\varepsilon_f \boldsymbol{u} \otimes \boldsymbol{u}) \right] = -\varepsilon_f \nabla p + \varepsilon_f \nabla \cdot \boldsymbol{\tau} - \boldsymbol{F}_{pf}^A \tag{6.12}$$

where $\boldsymbol{\tau}$ is the viscous shear stress tensor, and $\boldsymbol{F}_{pf}^A$ is the fluid-particle momentum exchange (source) term for model A:

$$\boldsymbol{F}_{pf}^A = \frac{1}{\Delta V_{\Omega_C}} \sum_i^{N_{p,C}} (\boldsymbol{f}_{d,i} + \boldsymbol{f}_i'') = \frac{1}{\Delta V_{\Omega_C}} \sum_i^{N_{p,c}} (\boldsymbol{f}_{pf,i} - \boldsymbol{f}_{\nabla p} - \boldsymbol{f}_{\nabla \cdot \boldsymbol{\tau}} - \boldsymbol{f}_{Ar}) \tag{6.13}$$

where the index $N_{p,C}$ stands for the number of particles inside the cell Ω_C in which the averaging is applied. On the other hand, in model B, the pressure and shear stress are entirely in the fluid phase. This way, we can write the momentum equation for model B as:

$$\rho_f \left[\frac{\partial \varepsilon_f \boldsymbol{u}}{\partial t} + \nabla \cdot (\varepsilon_f \boldsymbol{u} \otimes \boldsymbol{u}) \right] = -\nabla p + \nabla \cdot \boldsymbol{\tau} - \boldsymbol{F}_{pf}^B \tag{6.14}$$

where $\boldsymbol{F}_{pf}^B$ is the interaction term for model B, written as:

$$\boldsymbol{F}_{pf}^B = \frac{1}{\Delta V_{\Omega_C}} \sum_i^{N_{p,c}} \boldsymbol{f}_{pf,i} + \boldsymbol{f}_i'' \tag{6.15}$$

The shear stress tensor is expressed as:

$$\boldsymbol{\tau} = \mu \left[(\nabla \cdot \boldsymbol{u}) + (\nabla \cdot \boldsymbol{u})^T - \frac{2}{3} (\nabla \cdot \boldsymbol{u} \boldsymbol{I}) \right] \tag{6.16}$$

where μ is the dynamic viscosity and $\boldsymbol{I}$ is the identity matrix.

6.3.3 Interphase momentum transfer modeling

The interphase momentum coupling between numerical solutions for the two phases is done through the interphase momentum exchange term $\boldsymbol{F}_{pf}$. The components of this term are represented in what follows.

6.3.3.1 Pressure and buoyancy forces

Pressure and buoyancy (or Archimedes) forces are treated separately in the present work. As explained in Section 6.3.2, in Lethe the fluid pressure gradient ∇p does not account for the hydrostatic pressure. This means that the force due to the pressure gradient takes into account only the undisturbed pressure. As a consequence, the buoyancy force needs to be explicitly applied in Eq. (6.8). The expressions that represent the pressure gradient and buoyancy forces are, respectively:

$$\boldsymbol{f}_{\nabla p,i} = V_{p,i} \nabla p \tag{6.17}$$

$$\boldsymbol{f}_{Ar,i} = V_{p,i} \rho_f \boldsymbol{g} \tag{6.18}$$

where $V_{p,i}$ is the volume of the particle i, ρ_f is the density of the fluid, p stands for dynamic pressure, $\boldsymbol{g}$ is the gravity acceleration vector.

6.3.3.2 Shear force

The force due to the fluid viscous shear stress is:

$$\boldsymbol{f}_{\nabla\cdot\boldsymbol{\tau},i} = V_{p,i} \boldsymbol{\nabla} \cdot \boldsymbol{\tau} \tag{6.19}$$

6.3.3.3 Drag force

Drag stands for the friction between a body and the surrounding fluid. In fluidization, drag and buoyancy counterbalance gravity, lifting and holding particles up in the fluidized state. Drag is defined as:

$$\boldsymbol{F}_d = \sum_i^{N_{p,C}} \boldsymbol{f}_{d,i} = \sum_i^{N_{p,C}} \beta_i \left(\boldsymbol{u} - \boldsymbol{v}_i\right) \tag{6.20}$$

In unresolved CFD-DEM, since the fluid is described at a mesoscale, drag is conveniently defined as a function of a coefficient β_i, usually referred to as the "interphase momentum transfer coefficient". However, this is only true for model A because, in the absence of $\boldsymbol{f}_i''$, $\boldsymbol{F}_{pf}^A = \boldsymbol{F}_d$ (see Eq. (6.13)).

Several authors tried to determine β_i for particle-fluid systems. Gidaspow (1994) developed a correlation based on Ergun (1952) and Wen and Yu (1966) equations for the pressure drop in particle-fluid flows. Considering perfectly spherical particles:

$$\beta_i = \begin{cases} 150\frac{\left(1-\varepsilon_f\right)^2 \mu}{(\varepsilon_f d_p)^2} + 1.75\frac{\rho_f |\boldsymbol{u}-\boldsymbol{v}_i|(1-\varepsilon_f)}{d_p \varepsilon_f}, \text{ for } \varepsilon_f < 0.8 \\ \frac{3}{4} C_{D,i} \frac{\rho_f (1-\varepsilon_f)|\boldsymbol{u}-\boldsymbol{v}_i|}{d_p}, \text{ for } \varepsilon_f \geq 0.8 \end{cases} \tag{6.21}$$

where $C_{D,i}$ is the drag coefficient for a single particle, calculated by:

$$C_{D,i} = \begin{cases} \frac{24}{\mathrm{Re}_{p,i}} \left(1 + 0.15\mathrm{Re}_{p,i}^{0.687}\right), \text{ for } \mathrm{Re}_{p,i} < 1000 \\ C_{D,i} = 0.44, \text{ for } \mathrm{Re}_{p,i} \geq 1000 \end{cases} \tag{6.22}$$

where the Reynolds number of the particles $\mathrm{Re}_{p,i}$ is:

$$\mathrm{Re}_{p,i} = \varepsilon_f \rho_f \left| \boldsymbol{u} - \boldsymbol{v}_i \right| d_p / \mu \tag{6.23}$$

Following a similar approach, other authors proposed correlations for β_i mainly finding a function $G(\varepsilon_f)$ that represents the ratio between the drag coefficient $C_{D,i}$ and drag coefficient for a single isolated particle C_{D0}:

$$\frac{C_{D,i}}{C_{D0}} = G(\varepsilon_f, \mathrm{Re}_{p,i}) \tag{6.24}$$

In this approach, β_i is:

$$\beta_i = \frac{1}{2} C_{D0} \frac{\pi d_p^2}{4} \rho_f \left| \boldsymbol{u} - \boldsymbol{v}_i \right| G(\varepsilon_f, \mathrm{Re}_{p,i}) \tag{6.25}$$

Felice (1994) and Rong *et al.* (2013) proposed a correlations for $G(\varepsilon_f, \mathrm{Re}_{p,i})$ based on C_{D0} proposed by DallaValle (1948):

$$C_{D0} = \left(0.63 + \frac{4.8}{\sqrt{\mathrm{Re}_p}} \right)^2 \tag{6.26}$$

The correlation proposed by Di Felice for $G(\varepsilon_f, \mathrm{Re}_{p,i})$ is:

$$G(\varepsilon_f, \mathrm{Re}_{p,i}) = \varepsilon_f^{\,2 - \left\{ 3.7 - 0.65 exp \left[- \frac{\left(1.5 - log_{10} \mathrm{Re}_{p,i}\right)^2}{2} \right] \right\}} \tag{6.27}$$

This equation assumes the strong hypothesis that the exponent is a function only of Re_p and not of ε_f. As shown by several authors (BENYAHIA *et al.*, 2006; MAZZEI; LETTIERI, 2007; BEETSTRA *et al.*, 2007; CELLO *et al.*, 2010; RONG *et al.*, 2013), this hypothesis leads to errors in the estimation of drag. The Rong correlation is similar to Di Felice but tries to account for the effect of the void fraction on the exponent, written as:

$$G(\varepsilon_f, \mathrm{Re}_{p,i}) = \varepsilon_f^{\,2 - \left\{ 2.65 \left(\varepsilon_f + 1\right) - \left(5.3 - 3.5 \varepsilon_f\right) \varepsilon_f^2 exp \left[- \frac{\left(1.5 - log_{10} \mathrm{Re}_{p,i}\right)^2}{2} \right] \right\}} \tag{6.28}$$

An alternative approach was used by Beetstra *et al.* (2007). It determines a normalized drag force $F(\varepsilon_f, \mathrm{Re}_{p,i})$ by correcting the drag force using the Stokes-Einstein drag ($3\pi d_p \boldsymbol{u}$), the latter representing the drag experienced by a single isolated particle ($\varepsilon_f \rightarrow 1$) in the limit of $\mathrm{Re}_{p,i} \rightarrow 0$:

$$F(\varepsilon_f, \mathrm{Re}_{p,i}) = \frac{\boldsymbol{F}_d}{3\pi d_p (\boldsymbol{u} - \boldsymbol{v}_i)} \tag{6.29}$$

The authors proposed correlations for both mono and poly-dispersed beds. The correlation for mono-dispersed flows is:

$$\begin{aligned} F(\varepsilon_f, \mathrm{Re}_{p,i}) = \; & \frac{10\left(1-\varepsilon_f\right)}{\varepsilon_f^2} + \varepsilon_f^2\left(1+1.5\varepsilon_f^{0.5}\right) \\ & + \frac{0.413\mathrm{Re}_{p,i}}{24\varepsilon_f^2}\left[\frac{\left(1/\varepsilon_f\right)+3\varepsilon_f\left(1-\varepsilon_f\right)+8.4\mathrm{Re}_{p,i}^{-0.343}}{1+10^{3\left(1-\varepsilon_f\right)}\mathrm{Re}_{p,i}^{-\left(1+4\left(1-\varepsilon_f\right)\right)/2}}\right] \end{aligned} \tag{6.30}$$

6.3.3.4 Saffman lift force

The lift force can be separated into two components, usually referred to as Saffman and Magnus. Both forces act perpendicular to the relative velocity between phases, but the first happens due to the gradient of the fluid velocity field along the particle, and the latter is caused by the particles' rotation (CROWE *et al.*, 2011). Since the rotation of the particles is dissipated by the presence of water and collisions in the LSFB, the Magnus force was neglected in the present work.

The Saffman component of the lift forces acting over a particle i can be represented by combining the correlation proposed by Saffman (1965), Saffman (1968) and the lift coefficient proposed by Mei (1992), usually referred to as Saffman-Mei lift force model, and is calculated by:

$$\boldsymbol{f}_{\mathrm{Saff},i} = 1.61 C_{\mathrm{Saff},i} d_p^2 (\mu\rho_f)^{1/2} \left|\boldsymbol{\omega}_{c,i}\right|^{-1/2} \left[(\boldsymbol{u}-\boldsymbol{v}_i)\times\boldsymbol{\omega}_{c,i}\right] \tag{6.31}$$

where the vorticity $\boldsymbol{\omega}_{c,i}$ corresponds to the curl of the vector field, calculated as:

$$\boldsymbol{\omega}_{c,i} = \nabla\times\boldsymbol{u} \tag{6.32}$$

In equation (6.31), $C_{\mathrm{Saff},i}$ is the Saffman lift coefficient, calculated by:

$$C_{\mathrm{Saff},i} = \begin{cases} \left(1-0.3314\alpha^{1/2}\right)\exp\left(\frac{-\mathrm{Re}_{p,i}}{10}\right)+0.3314\alpha^{1/2}, \text{ for } \mathrm{Re}_{p,i}\le 40 \\ 0.0524(\alpha_l \mathrm{Re}_{p,i})^{1/2}, \text{ for } \mathrm{Re}_{p,i} > 40 \end{cases} \tag{6.33}$$

where α is:

$$\alpha = \frac{d_p}{2\left|\boldsymbol{u}-\boldsymbol{v}_i\right|}\left|\boldsymbol{\omega}_{c,i}\right| \tag{6.34}$$

6.4 Fluidized Bed Experiment

In the present section, we describe the materials and experimental setup used in the validation campaign.

6.4.1 Particles

We used two groups of particles in this work: alginate and alumina. The alumina particles were used as received. We synthesized the alginate particles by dropping a 20 $\mathrm{g\,L^{-1}}$ sodium alginate solution (Fisher Chemical) impregnated with titanium dioxide powder (mass ratio 25:2) into a 15 $\mathrm{g\,L^{-1}}$ barium chloride solution. The sodium alginate gelifies when in contact with the barium chloride, forming barium alginate particles. We added titanium dioxide powder to increase the weight of the particles, making them heavier than water. We refer the reader to works by Lopes *et al.* (2018) and Melo *et al.* (2021a) for more reference on manufacturing these particles.

We characterized both groups of particles in density, diameter, and terminal settling velocity. The diameter of the particles was measured by capturing images with 20 random samples and processing the images using the image processing library called OpenCV (BRADSKI, 2000). The software was calibrated using Acrylonitrile Butadiene Styrene (ABS) commercial particles ($d_p = 5.95 \pm 0.01$ mm) as reference. We maintained the distance between the camera and the particles for all samples. In this work, we considered the diameter of all particles equivalent to the superficial diameter of the set of 20 samples.

To measure the particles' apparent densities, we applied the pycnometry method using a 25 mL pycnometer and distilled water as reference fluid. Since the alginate particles present a high water content and alumina is a porous material, we measured their densities after soaking them in water for at least 24 hours.

For the terminal settling velocity, we tracked the falling trajectory of ten particles inside a 20 cm wide 1 m height square-based tank using a high-speed camera (correlation DSC-RX100 M, brand Sony) at 960 frames per second. We used the open-source image processing software Tracker (BROWN *et al.*, 2022) to measure their velocities.

The particles' samples were weighed before every fluidization experiment. For all experiments with alginate, the total weight of the particles was 1.1 kg, and for the alumina, 4.0 kg.

The characterization results are presented in Table 6.2.

Table 6.2 – Summary of particles' characterization results.

Characteristic	**Alginate**	**Alumina**
Total mass (M_p)	1.1 kg	4.0 kg
Density (ρ_p)	1029 ± 1 $\mathrm{kg\,m^{-3}}$	3586 ± 10 $\mathrm{kg\,m^{-3}}$
Diameter (d_p)	2.66 ± 0.10 mm	3.09 ± 0.16 mm
Terminal settling velocity (U_0)	3.03 ± 0.01 $\mathrm{cm\,s^{-1}}$	48.02 ± 0.02 $\mathrm{cm\,s^{-1}}$

6.4.2 Fluidization experiments

We used water as the fluid in all fluidization experiments. The experimental setup consists of a 1 m height, 10 cm diameter cylindrical fluidized bed column. A schematic representation of the experimental setup is presented in Figure 6.1a.

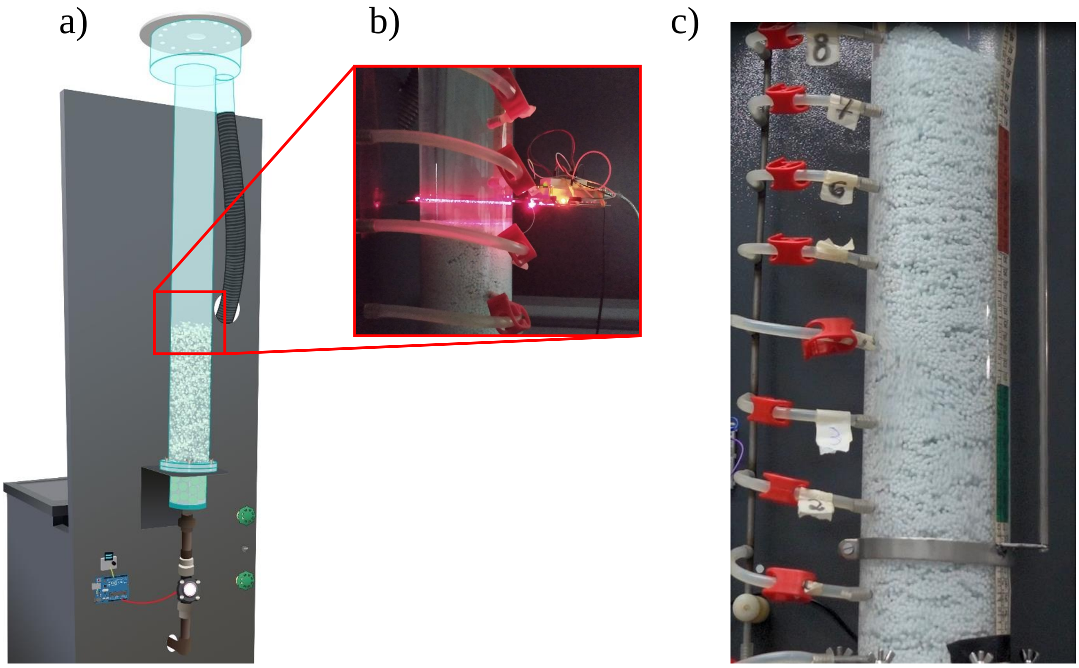


Figure 6.1 – Schematic representation of a) the equipment and b) the laser apparatus used for bed height measurement.

We used Arduino-based sensors in all experimental measurements. Since we conducted the experiments using two different ranges of inlet flow rates, we applied different Hall effect flow rate sensors according to the particles: for the experiments with alginate particles, we used the YF-S201 ($1 - 30\ \mathrm{L\,min^{-1}}$); while for the alumina we used the YF-G1 ($2 - 100\ \mathrm{L\,min^{-1}}$). The Arduino system registered the instantaneous inlet flow rate and calculated the time average during the experiments. The inlet conditions of the experiments are described in Table 6.3.

Table 6.3 – Summary of fluidization experiments inlet conditions.

Fluid inlet condition	**Alginate**	**Alumina**
Number of inlet flow rates	7	8
Flow rate range	2 - 5 $\mathrm{L\,min^{-1}}$	40 - 75 $\mathrm{L\,min^{-1}}$
Inlet Reynolds number range	498 - 1244	9954 - 18664
Range of U/U_0	0.14 - 0.35	0.18 - 0.33

We used an analog Negative Temperature Coefficient (NTC) thermistor to monitor the water temperature. The average temperature among the experiments was 30 °C. We used this value to estimate the water's density and viscosity in the simulations.

6.4.2.1 Pressure drop

In the fluidization regime, the total pressure drop along the bed height, $-\Delta p$ is constant, regardless of the inlet flow rate (EPSTEIN, 2003b). It can be calculated using the following equation:

$$-\Delta p = \frac{M_p\left(\rho_p - \rho_f\right)g}{\rho_p A_b} \tag{6.35}$$

where M_p is the total mass of the particles inside the bed and A_b is the cross-section area of the equipment. The total pressure drop for the alginate particles is 41.64 Pa, and for the alumina, it is 3608.55 Pa.

6.4.2.2 Bed porosity

The fluidized bed porosity $\bar{\varepsilon}_f$ is a convenient way of representing the bed expansion since it is dimensionless, and both quantities have a direct relation. To determine the experimental bed expansion, we measure the bed porosity using two techniques, depending on the particles' characteristics.

For the alginate particle, we measure the porosity using a direct measurement of the bed height since the top of the bed is calm and flat. We used a laser apparatus comprising a semi-circular platform holding a 5 V laser diode, a Light Dependent Resistor light sensor (LDR), and an Arduino microcontroller. The platform is free to move along the equipment's height and keeps the laser beam pointing to the LDR. The microcontroller registers the signal returned by the LDR and calculates the moving average of this value at 15-second intervals. Figure 6.1b shows a picture of the laser apparatus.

During the fluidization experiments, we measured the bed height by positioning the laser apparatus at the height where the moving average of the analog signal is within a threshold value ±10% and measured the laser height. The threshold value corresponds to the average between the LDR analog signal measured at a particle-free zone and an utterly interrupted beam.

With the experimental bed height, we calculated $\bar{\varepsilon}_f$ using:

$$\bar{\varepsilon}_f = 1 - \left(\frac{M_p}{\rho_p A_b H_b}\right) \tag{6.36}$$

Since the alumina particles are heavier and tend to form a more unstable bed top, the laser technique could not be applied. Instead, we measured the bed expansion using the differential pressure profile along the bed height. As shown in Figure 6.1c, pressure outlets are positioned along the height of the bed, and we measured the differential pressure to each point. With the differential pressure ($-\Delta p$) and the height of each pressure outlet (Δz), we found the slope of the differential pressure curve along the bed height ($-dp/dz$)

and calculated the bed porosity using the following relation (EPSTEIN, 2003b):

$$\bar{\varepsilon}_f = 1 - \left[\frac{-\mathrm{d}p/\mathrm{d}z}{(\rho_p - \rho_f)g}\right] \tag{6.37}$$

For comparison reasons, we estimated the bed porosity using R-Z (Equation 6.1). The index n was calculated using the correlation proposed by (KHAN; RICHARDSON, 1989). To evaluate the fitting between the estimated and simulated results with respect to the experiments, we apply the root-mean-squared error (RMSE), calculated by:

$$RMSE = \sqrt{\frac{1}{N}\sum_{i}^{N}\left(y_i' - y_{exp,i}\right)^2} \tag{6.38}$$

where N is the number of inlet conditions (i.e., 7 for the alginate and 8 for the alumina particles), y_i' is the estimated or simulated result, and $y_{exp,i}$ is the experimental result. The RMSE is especially useful for such analysis for highlighting outliers, which can be helpful in cases where the results are close to each other.

6.5 Simulation Setup

The simulations were carried out on Lethe (BLAIS *et al.*, 2020; GOLSHAN *et al.*, 2022; GEITANI; BLAIS, 2023b; GEITANI *et al.*, 2023a), a Finite Elements Method (FEM) open-source software with CFD, DEM, and CFD-DEM coupling capabilities. We used the setup described in Table 6.4 as a benchmark for all simulations. Unless mentioned, the parameters are the same for all simulations.

Table 6.4 – Simulation parameters.

Parameter	**Alginate**	**Alumina**
Total real-time	35 s	20 s
Time integration method	BDF1	BDF1
CFD time-step and coupling interval (Δt_{CFD})	0.001 s	0.001 s
DEM time-step (Δt_{DEM})	0.000 01 s	0.000 01 s
Output time-step (Δt_{DEM})	0.25 s	0.10 s
Diameter of the cylinder (D_b)	10 cm	10 cm
Height of the cylinder (H_b)	1.10 m	1.10 m
Mesh ($n_r \times n_\theta \times n_z$)	$6 \times 16 \times 132$	$6 \times 16 \times 132$
Liquid density (ρ_f)	996.78 kg m^{-3}	996.78 kg m^{-3}
Liquid dynamic viscosity (μ)	8.352×10^{-4} Pa s	8.352×10^{-4} Pa s
Number of particles (N_p)	107960	72400
Diameter of the particles (d_p)	2.66 mm	3.09 mm
Density of the particles (ρ_p)	1029 kg m^{-3}	3586 kg m^{-3}
Young's modulus (Y)	10 MPa	10 MPa
Coefficient of restitution (e)	0.9	0.9
Poisson ratio (ν)	0.3	0.3
Coefficient of rolling friction (μ_r)	0.2	0.2
Coefficient of sliding friction (μ_f)	0.1	0.1
VANS model type	A	A
Void fraction calculation scheme	PCM	PCM
Void fraction smoothing length	$2.0 \times d_p$	$2.0 \times d_p$
Boundary conditions at the walls	free slip	free slip
Drag model	Rong et al. (RONG *et al.*, 2013)	Rong et al.(RONG *et al.*, 2013)
Gravity (g)	9.81 m s^{-2}	9.81 m s^{-2}

The DEM and CFD time steps were selected such that one CFD time step occurred every 100 DEM time steps. The coupling time step was equal to the CFD time step for all simulations. We applied the first-order Backward Differentiation (BDF1) method for the time-stepping. Lethe addresses turbulence implicitly by applying an implicit Large Eddy Simulation (LES) approach, demonstrating high accuracy in describing turbulent systems (GEITANI; BLAIS, 2023b; SAAVEDRA *et al.*, 2022).

Apart from the number of particles (N_p), density (ρ_p), and diameter(d_p), we used the same DEM parameters for both alumina and alginate particles. The collisional properties, such as Young's modulus and coefficient of restitution, were kept the same for both walls and particles.

Figure 6.2 is a schematic representation of the mesh and initial packing used in the simulations. Details about the mesh refinement are provided in Table 6.4.

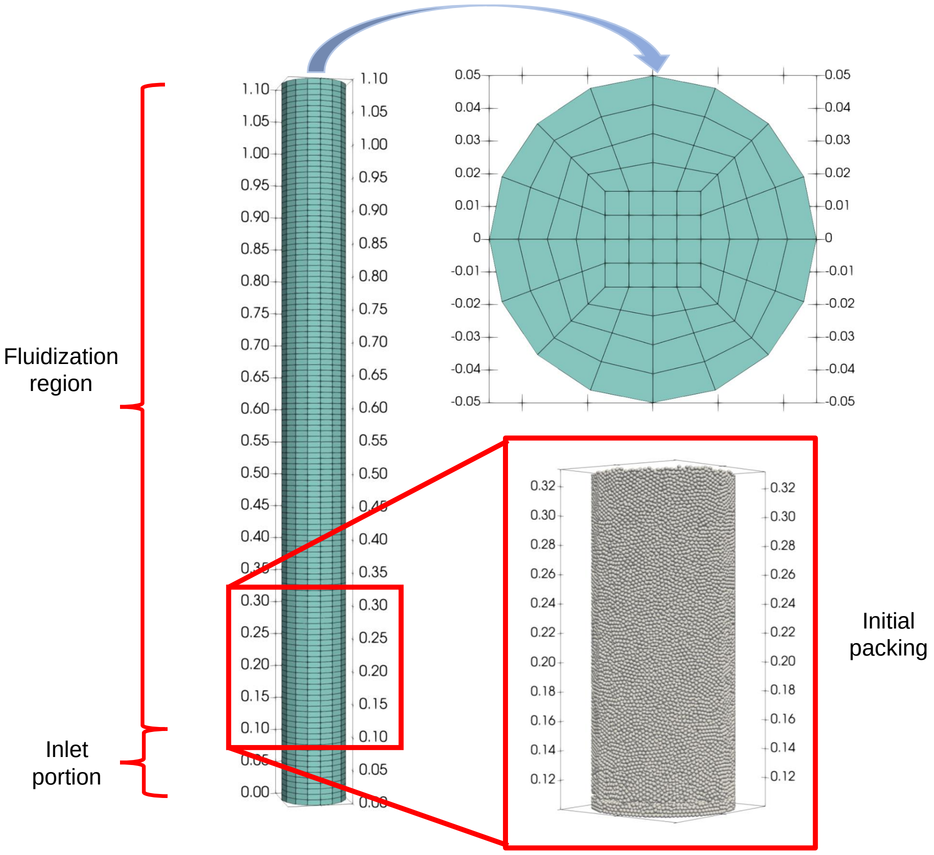


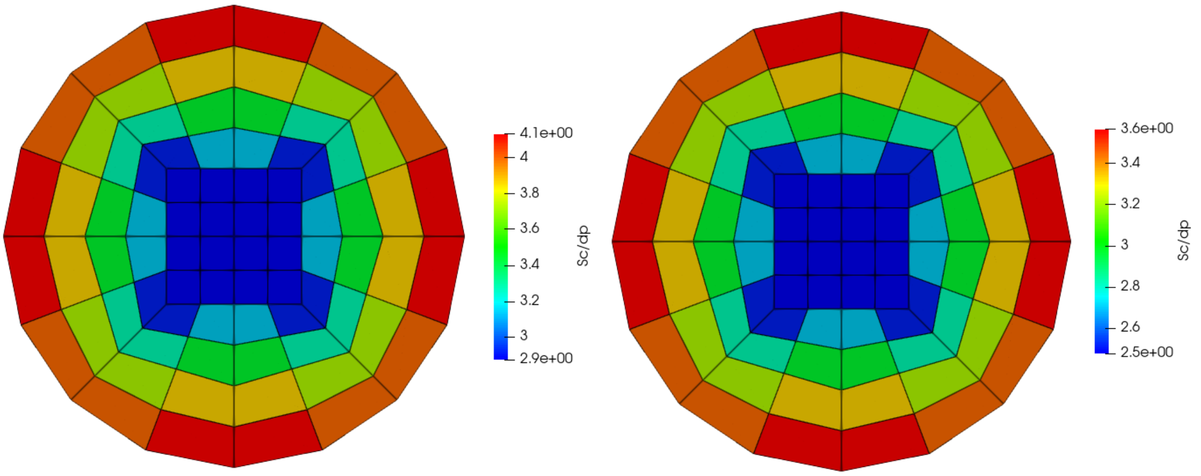
Figure 6.2 – Schematic representation of the mesh and initial packing of particles (units in meters).

As presented in Figure 6.2, the inlet portion corresponds to a 10 cm height flow calming region. This region without particles helps stabilize the flow below the bed of particles.

The precise calculation of the void fraction is key to obtaining good simulation results. To achieve such precision, the ratio between the size of cells and particles should be high enough so that the void fraction values are representative. One popular way to represent this ratio is using the ratio between the characteristic length of the cells (S_c) and the diameter of the particles. The characteristic length of the cells is calculated based on the cells' volume (V_c) as:

$$S_c = V_c^{1/3} \tag{6.39}$$

Since the domain is cylindrical, the mesh elements are not all of the same sizes. This implies a variation in S_c/d_p within the domain. Figure 6.3 shows the distribution of S_c/d_p with the cross-section of the domain for both particles.



(a) Alginate

(b) Alumina

Figure 6.3 – S_c/d_p for (a) alginate and (b) alumina particles.

We applied the PCM void fraction scheme for the void fraction calculation. To avoid sharp discontinuities and enhance numerical stability, we smoothed the void fraction for a length 2 times the diameter of the particles. This is done by adding $\int_\Omega L^2 \nabla\varphi_i \nabla\varphi_j d\Omega$ to the left-hand side of the $\mathcal{L}^2$ projection equation for the void fraction, resulting in the following:

$$\int_\Omega \varphi_i \alpha_{f,j} \varphi_j d\Omega + \int_\Omega L^2 \nabla\varphi_i \nabla\varphi_j d\Omega = \int_\Omega \varepsilon_{f,i} \varphi_i d\Omega \tag{6.40}$$

where α_f is the projected void fraction, L is the smoothing length, Ω is the domain in which the equation is being resolved (CFD cell), and φ is the basis (hat) function. We refer the reader to the article by Geitani *et al.* (2023a) for details about the smoothing strategy and its effect on numerical stability.

Initially, all simulations start with the particles completely static and sedimented. The inlet velocity is imposed at t = 0, and after some time, the fluidized bed reaches a pseudo-steady state. To find when the pseudo-steady state is reached, we calculated the instantaneous porosity of the bed at each output time-step. We considered that the pseudo-steady state was reached when the instantaneous porosities stabilizes around the average among the last 5 seconds of the simulation. For the alginate and the alumina particles, it was reached at around 10 and 4 seconds in real-time, respectively. To guarantee that all results were captured in the pseudo-steady state and are not influenced by the initial bed expansion, all the comparisons with the experimental results are made 5 seconds after the moment when the pseudo-steady state was reached. We present the porosity results as a function of the ratio between the fluid inlet velocity and the terminal velocity of the particles (U/U_0) so that the results follow the shape of Equation 6.1.

6.6 Results and Discussion

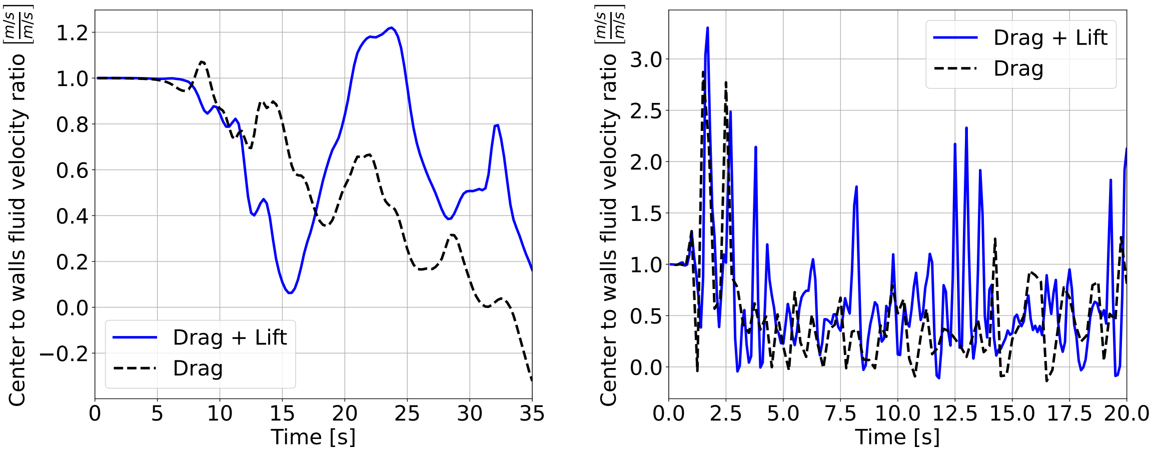
The main objective of the present work is to prove the validity of the unresolved CFD-DEM method on the simulation of the LSFB. To do so, we start with a qualitative analysis of the particle dynamics inside the equipment, and then the quantitative analysis proceeds with the comparison between experimental data and simulations.

In this sense, we first discuss the importance of void fraction smoothing and the Saffman lift force to obtain accurate particle distribution. To do so, we qualitatively compare particles' experimental and simulated behaviors. Then, since we have a restrictive geometry (small equipment to particle size), we discuss the mesh choice. After this, we proceed to the quantitative validation of the simulations, comparing them with experimental results for pressure drop and bed expansion. Finally, we compare different drag correlations to investigate the robustness.

6.6.1 Flow structures, void fraction, and the importance of the Saffman lift force

For all simulations, we observed critical channeling at the walls. Clear evidence of this channeling is presented in Figure 6.4, where lines represent the ratio between two fluid velocities: the one at the center of the bed of particles and the spatial average of the velocities close to the walls (both at the same height).

As shown in Figure 6.4, the velocity close to the walls is higher than in the center for almost the entire simulated time. The reason for this relies upon two factors: firstly, the void fraction is naturally higher closer to the walls due to the arrangement of the particles, and secondly, S_c/d_p decreases towards the center of the equipment as shown in Figure 6.3, virtually distorting the distribution of the void fraction field. The first phenomenon responsible for the void fraction distortion cannot be avoided since it is inherent to the particles' distribution. However, the second effect significantly diminished with void fraction smoothing.



(a) Alginate

(b) Alumina

Figure 6.4 – Center to walls velocity ratio for (a) alginate and (b) alumina particles, at $U/U_0 = 0.3595$ and $U/U_0 = 0.3270$, respectively.

Since the drag correlations are highly dependent on the void fraction, thus, distortion of the void fraction virtually increments drag in regions with smaller cells. As a reaction, the fluid feels a higher resistance in the center of the equipment than the walls, consequently being pushed toward the walls, producing channeling. If no void fraction smoothing strategy is used, the difference between the drag force experienced in the middle of the equipment and close to the walls would be even higher.

In real life, two things reduce this channeling effect: no-slip at the walls and interphase forces in the radial direction. Since the mesoscale of the fluid discretization is not fine enough to represent the boundary layer formed by the fluid friction with the walls, particles are pushed far from the channeling. With nothing to counterbalance this effect, the pseudo-steady state of the bed is unstable. The consequence of this set of phenomena is the unrealistic pluming of particles, easily visible at the top of the bed, as shown in Figures 6.5 and 6.6.

This pluming effect is significantly diminished with Saffman lift forces since the channeling is not as persistent. The Saffman lift force pushes the particles towards and away from the channeling regions, which breaks the flow pattern, spreading the fluid and creating a more realistic fluid velocity distribution in the pseudo-steady-state. As a consequence, bed height and uniformity of particle distribution are increased, and the void fraction distortion is compensated. With the more uniform distribution of particles, drag distribution is also more uniform, and the pseudo-steady-state is more stable. Uniformity is a crucial aspect to be taken into account since uniformity is one of the main characteristics of LSFB. These conclusions agree with previous studies by Koreich et al. (KOERICH *et al.*, 2018) and Zbib et al. (ZBIB *et al.*, 2018). For this reason, we apply the Saffman lift force in all subsequent simulations in this work.

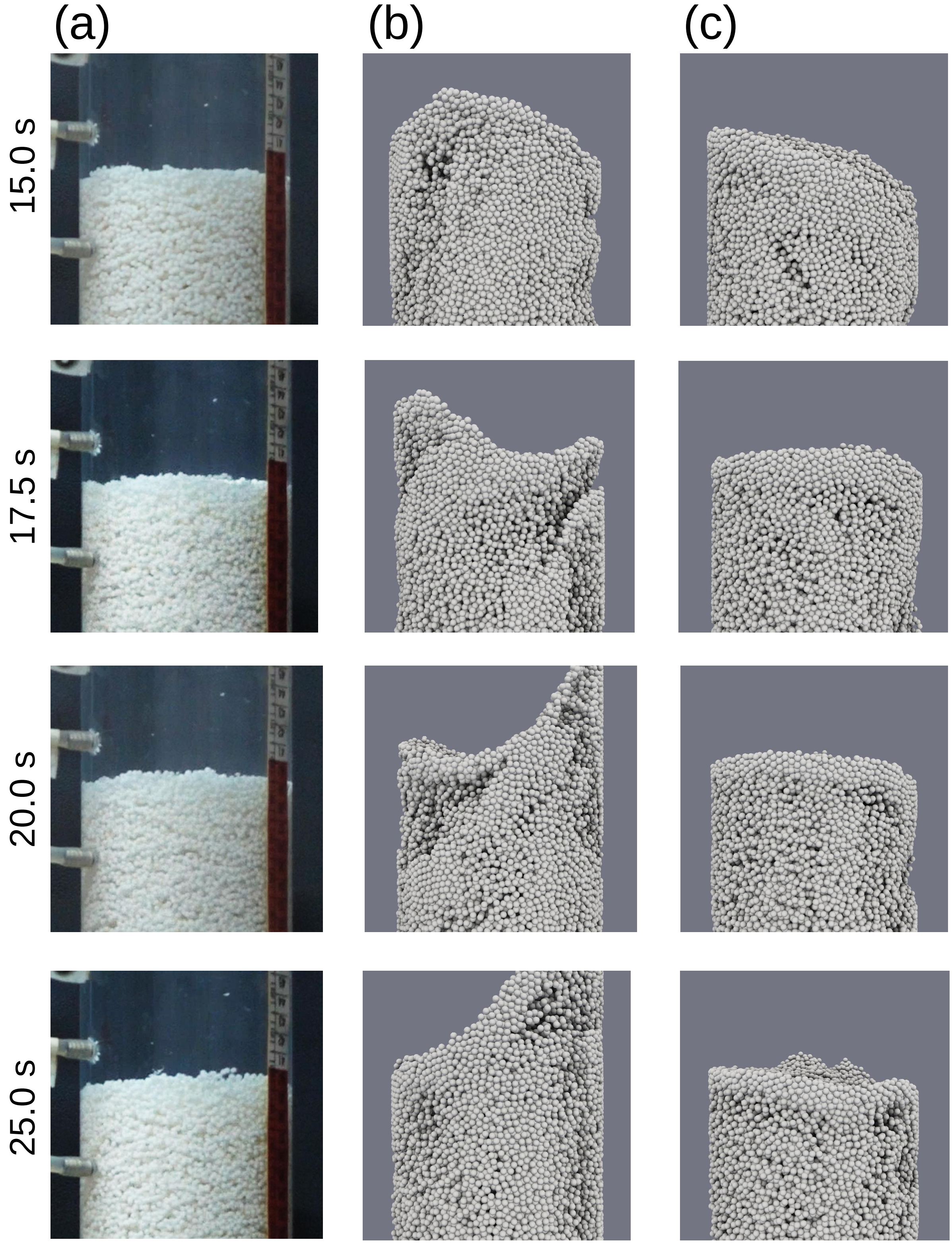


Figure 6.5 – Images of the top of the pseudo-steady state (reached around 10 seconds of simulation)alginate particles bed with 2.5 seconds intervals for (a) experimental, (b) simulation with drag force only, and (c) simulation with drag and lift forces results at the highest inlet velocity ($U/U_0 = 0.3595$).

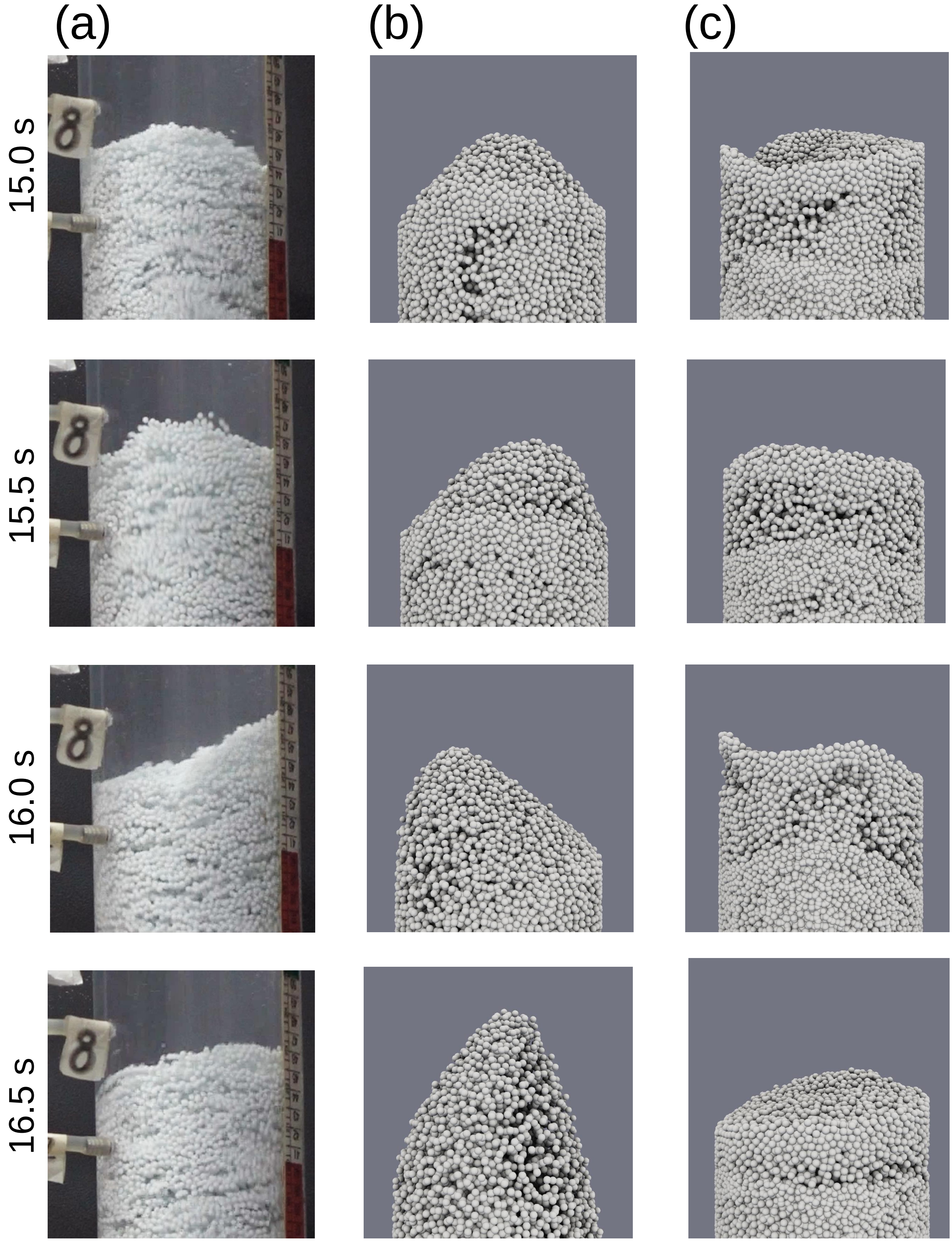


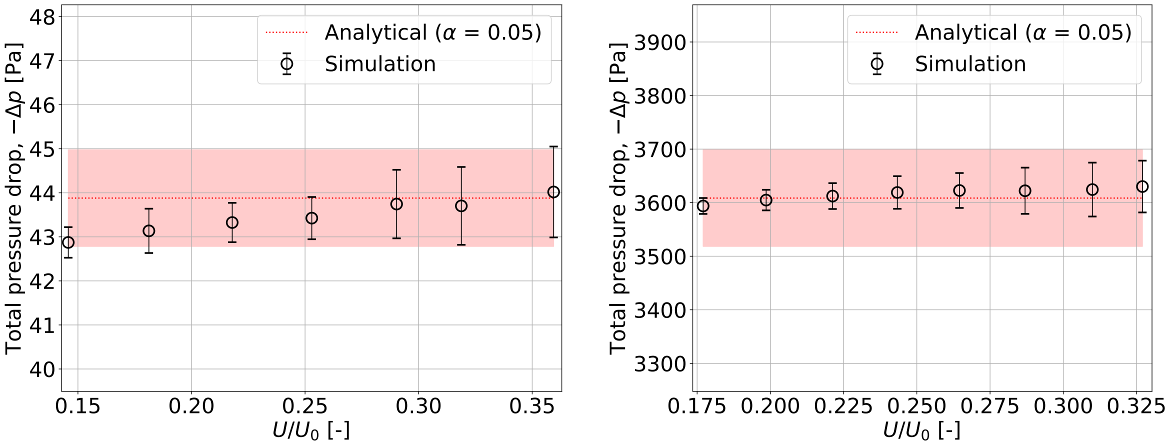
Figure 6.6 – Images of the top of the pseudo-steady-state (reached around 4 seconds of simulation) alumina particles bed with 0.5 seconds intervals for (a) experimental, (b) simulation with drag force only, and (c) simulation with drag and lift forces results at the highest inlet velocity ($U/U_0 = 0.3270$).

6.6.2 Validation of the simulations

The simulations were validated by comparing the pressure drop and the bed porosity with experimental results.

6.6.2.1 Pressure drop

We compare the simulated total pressure difference with the one calculated by Equation 6.35 with a 0.95 two-tailed confidence interval ($\alpha = 0.05$). The comparative results are presented in Figure 6.7. For the simulation results, the error bars correspond to a temporal standard deviation.



(a) Alginate

(b) Alumina

Figure 6.7 – Average of the total pressure difference for the (a) alginate and (b) alumina particles

For almost all simulations, the total pressure difference was precisely reduced, standing within the 95% confidence interval region, even accounting for the temporal oscillations represented by the standard deviation. This means that the numerical results precisely represent the pressure difference, correctly accounting for the mass of all particles in the force balance.

Few simulated results, mainly the alginate particles, especially at low inlet velocities, did not reach the expected pressure drop. This is caused by the smoothing of the void fraction, which virtually increases the void fraction of cells right above the bottom wall of the equipment. This slight increase in the void fraction reduces the drag force sufficiently to prevent particles from fluidizing. Consequently, the equipment's bottom wall carries the weight of a few particles and this reduces the pressure drop. This effect can be observed in Figure 6.8.

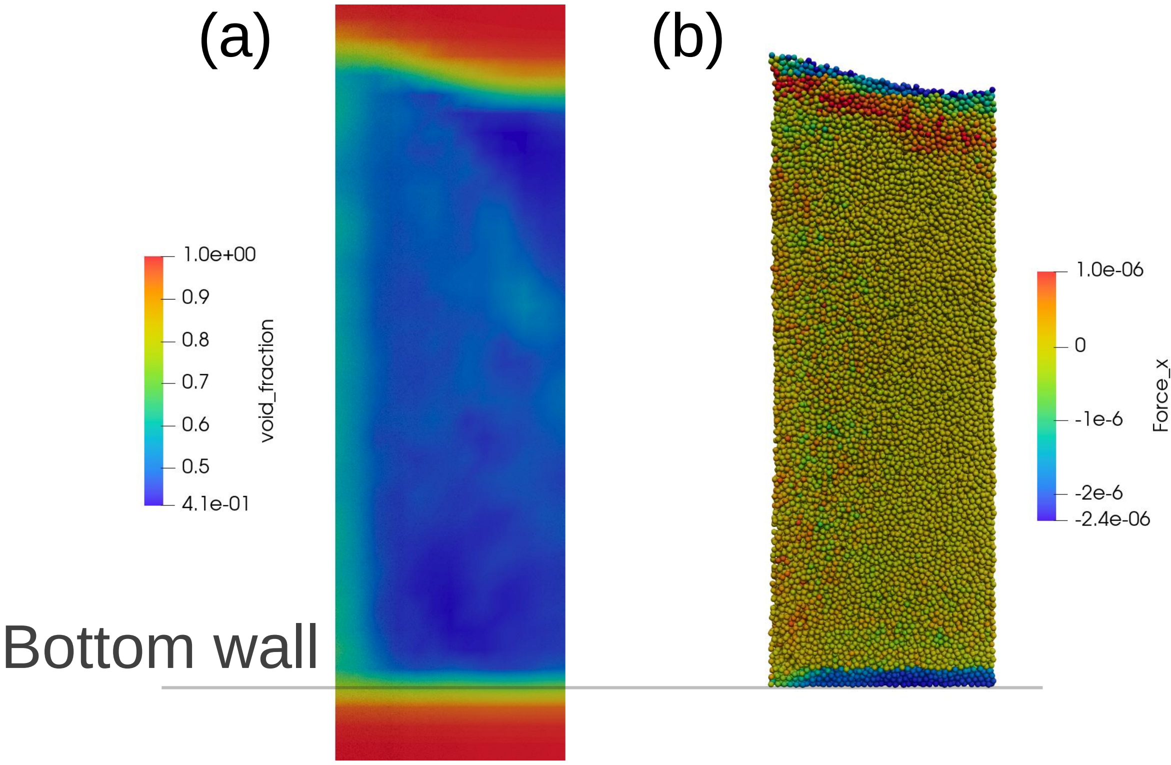


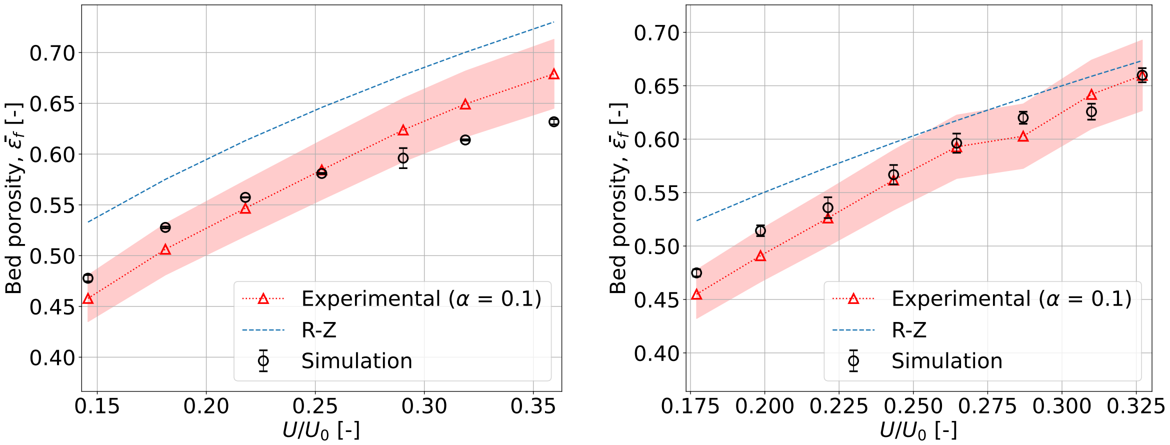
Figure 6.8 – Snapshot of the $U/U_0 = 0.1430$ alginate particles simulation at 35 seconds (last time-step), showing (a) the void fraction profile and (b) the total force acting over particles in Newtons. Results are clipped in the axial direction to show the middle of the equipment. It can be clearly seen that the first few layers of particles are supported by the bottom wall.

The small fluctuations in the pressure are within realistic values. They are mainly related to the dynamics of the bed, which attests to numerical stability throughout the simulations, especially considering all results collected inside the pseudo-stationary regime. Other authors found higher oscillations of pressure (BLAIS *et al.*, 2016) in the simulation of fluidized beds using Finite Volume codes. We usually expect higher variations when PCM is applied due to the coarse approximation of the void fraction field (PENG *et al.*, 2014). In this sense, since we are using PCM, the stability of the pressure field is attributed to void fraction smoothing, which improves the continuity of the field, and the grad-div stabilization, which enhances mass conservation (GEITANI *et al.*, 2023a).

6.6.2.2 Bed expansion

The bed expansion is usually represented by the spatial average of the void fraction along the bed's volume (or bed porosity) $\bar{\varepsilon_f}$. In the design of LSFB reactors, porosity is among the essential variables because of its use in predicting mass/heat transfer and conversion of reactants. Unlike gas-solids fluidized beds, it is well established that LSFBs tend to present a uniform porosity.

In Figure 6.9 we present the numerical and experimental results of the porosity measurement for both particles. We also present the results of the estimation of $\bar{\varepsilon_f}$ by the R-Z Equation (Equation 6.1), for comparison.



(a) Alginate

(b) Alumina

Figure 6.9 – Bed porosity for (a) alginate and (b) alumina particles.

Almost all simulations fall into the experimental results region with a two-tail confidence interval of 90% ($\alpha = 0.1$). The agreement between simulations and experiment results is better assessed when comparing the results using R-Z. For the alginate, the RMSE is 78% higher than R-Z estimations (RMSE = 0.0618) compared to simulations (RMSE = 0.0347). The difference was even higher for alumina, reaching 168% (RMSE = 0.0417 for R-Z and 0.0156 for simulations). In the case of the alumina particles, the bed expansion was fully captured by the simulations, with all results staying within the confidence interval. The lower fitting in the case of the alginate particles can be associated with the diameter distribution of particles, which is not considered in the simulations. It is also important to note that, for all results, the bed expansion was stable with time, with a very low standard deviation (as noted in Figure 6.9).

These results highlight the reliability of unresolved CFD-DEM on the representation of particle-liquid fluidized beds. Furthermore, since the alginate particle's density is about a third of the alumina particles, it also reveals that the technique is reliable for a wide range of regimes, from the lowest fluid inlet velocity ($Re = 534$) to the highest ($Re = 19021$).

6.6.3 Mesh topology

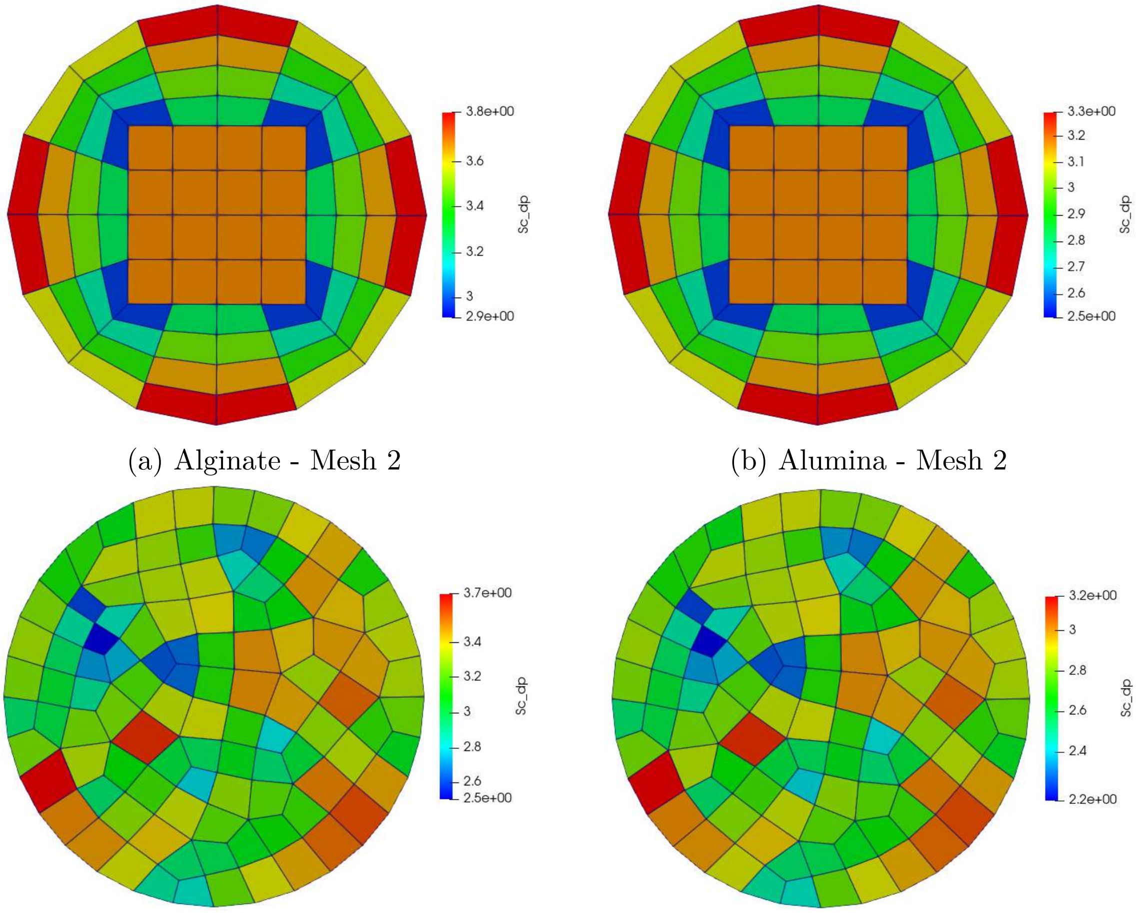
As discussed by several authors (PENG *et al.*, 2014; CLARKE *et al.*, 2018; WANG *et al.*, 2019), the mesh resolution of the fluid representation is limited by the size of the particles. This is a limitation of the volume-averaging approach applied to this work, which is not mesh-independent. As a rule of thumb, when PCM is applied, $S_c/d_p > 3$ is recommended, which, in the case of coarse particles like the ones in this work, imposes an important limitation on the mesh. In the case of finer meshes, the mesh is too fine to have an accurate void fraction profile (even when smoothing is applied), affecting the precision of the drag force calculation. On the other hand, coarser meshes have an inadequate resolution to coherently capture the fluid dynamics.

One way to mitigate this limitation is to use a more continuous void fraction calculation method. In the present work, we did so by smoothing the void fraction field. Alternatives include computationally intensive analytical methods (WU *et al.*, 2009; FREIREICH *et al.*, 2010) and more efficient but less precise non-analytical methods (KHAWAJA *et al.*, 2012; CLARKE *et al.*, 2018). In the context of the Finite Element Method, the Quadrature-Centered Method by Geitani *et al.* (2023b), which is a computationally efficient analytical method for void fraction calculation, is a good alternative. It is worth mentioning that there are other Eulerian-Lagrangian alternatives to the unresolved CFD-DEM approach adopted in this work such as strategies that employ volume filtering (CAPECELATRO; DESJARDINS, 2013) , two-grid formulation (DEB; TAFTI, 2013), and kernel-based approximation to the fluid field (WANG *et al.*, 2019), which allow for particle-to-cell size ratios close to 1 but involve the implementation of more complex algorithms.

Although the mesh resolution independency test was not carried out, we assessed the sensitivity of the method to the mesh topology. As pointed out, one crucial reason for applying void fraction smoothing is to compensate for the mesh elements' size unequal distribution. To test whether a more uniform cell size distribution along the cylinder radius impacts the dynamics of the particles, we tested two other mesh topologies. These meshes are presented in Figure 6.10.

The difference between Mesh 1 (Figure 6.2) and 2 is that the length of the square side in Mesh 1 is way smaller, but the distribution of cells is the same. Mesh 3 is completely unstructured in the radial direction but has the same cell distribution as Meshes 1 and 2 along its height.

Besides the better distribution of cell sizes in Meshes 2 and 3, no changes in the overall behavior of the particles were observed when compared to Mesh 1. A slightly worse representation of the behavior of the particles is observed for the higher inlet flow rates using Mesh 2, even using the same smoothing strategy and applying the Saffman lift force. This can be attributed to its distortion in the outer cells, i.e., since the shape of the outer cells in Mesh 2 are irregular (sides do not present similar lengths), the void fraction profile in the radial direction (discussed in Section 6.6.1) is even more non-uniform. Since drag is highly dependent of the void fraction, the plumbing effect represented in Figures 6.5 and 6.6 are even more prominent. However, Mesh 3 produced results similar to the original mesh presented in Figure 6.2 for all simulation conditions and particles.



(a) Alginate - Mesh 2

(b) Alumina - Mesh 2

(c) Alginate - Mesh 3

(d) Alumina - Mesh 3

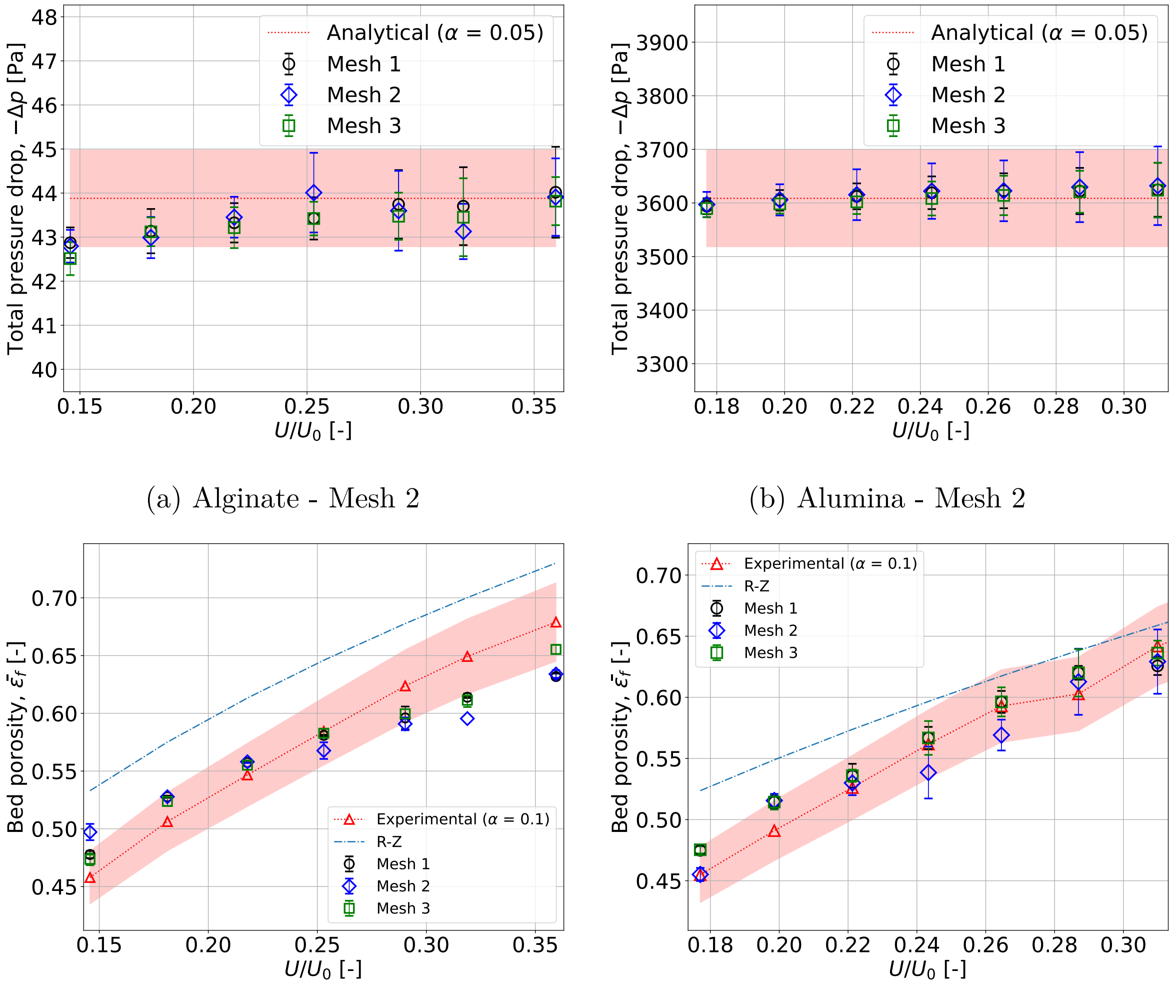
Figure 6.10 – S_c/d_p for the different mesh and particle combinations.

In terms of quantitative analysis, the total pressure drop and bed expansion for all simulations with the different meshes stood close to one another. The quantitative results of the mesh test are presented in Figure 6.11.

As observed in Figure 6.11, the quantitative results for all meshes and both particles present high similarity. Through the deviation, we observed slightly higher oscillations on the pressure field for Mesh 2 compared to the others, which can be attributed to the previously mentioned oscillatory behavior of the particles due to the distribution of mesh element sizes. Such oscillations lead to higher variability of the bed expansion in the tests with Mesh 2. Even so, the results stood within the experimental tolerance.

6.6.4 Drag model comparison

One way to assess the precision of drag correlations in representing liquid-solid fluidized beds is by evaluating its capability of reproducing the bed expansion according to the inlet velocity. The results in Figure 6.12 demonstrated that Rong, Di Felice, and Beetstra drag correlations precisely reproduced the observed behavior of the bed expansion



(a) Alginate - Mesh 2

(b) Alumina - Mesh 2

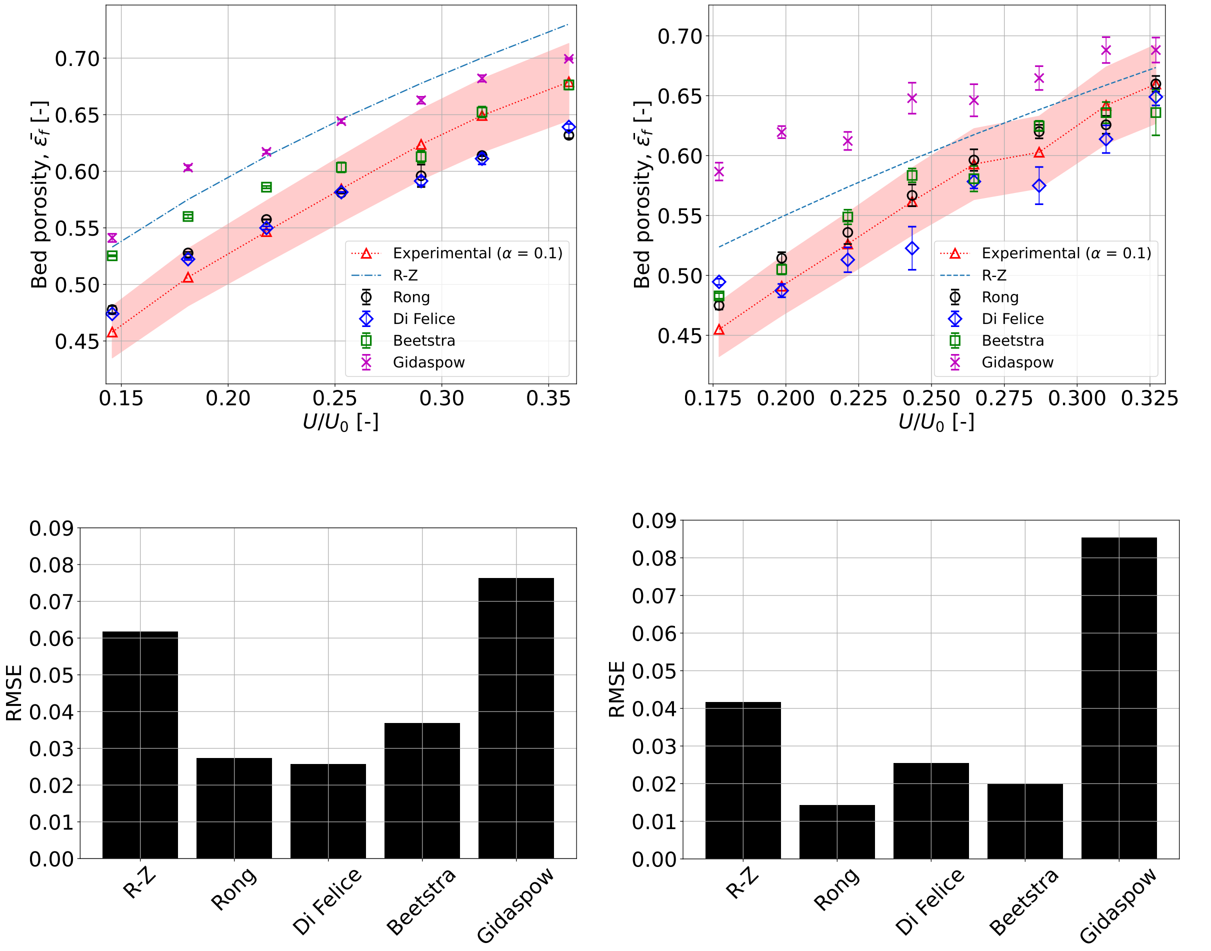
(c) Alginate - Mesh 3

(d) Alumina - Mesh 3

Figure 6.11 – Bed expansion and pressure drop for the different mesh and particle combinations.

as a function of the inlet velocity. Almost all points stood within the confidence interval, except for some alginate results. In the case of Beetstra, the highest distance between the simulated and experimental results is for the lowest inlet velocity (simulation 14% higher than experiment). It is important to note that the structure of the correlations is significantly different from one to another, especially considering Beetstra, so the results are reassuring from a validation point of view since they show that relatively accurate results can be obtained with unresolved CFD-DEM.

According to several authors (BENYAHIA *et al.*, 2006; MAZZEI; LETTIERI, 2007; BEETSTRA *et al.*, 2007; CELLO *et al.*, 2010; RONG *et al.*, 2013), neglecting ε_f in the exponent of $G(\varepsilon_f, \mathrm{Re}_{p,i})$ (Equation 6.27) can lead to important lack of accuracy. However, in the case of this work, the correlation proposed by Di Felice was the one with the highest similarity with the alginate experimental results. Since all models are, in fact, correlations, i.e., come from fitting an equation to a given data, the lack of fitting with



(a) Alginate

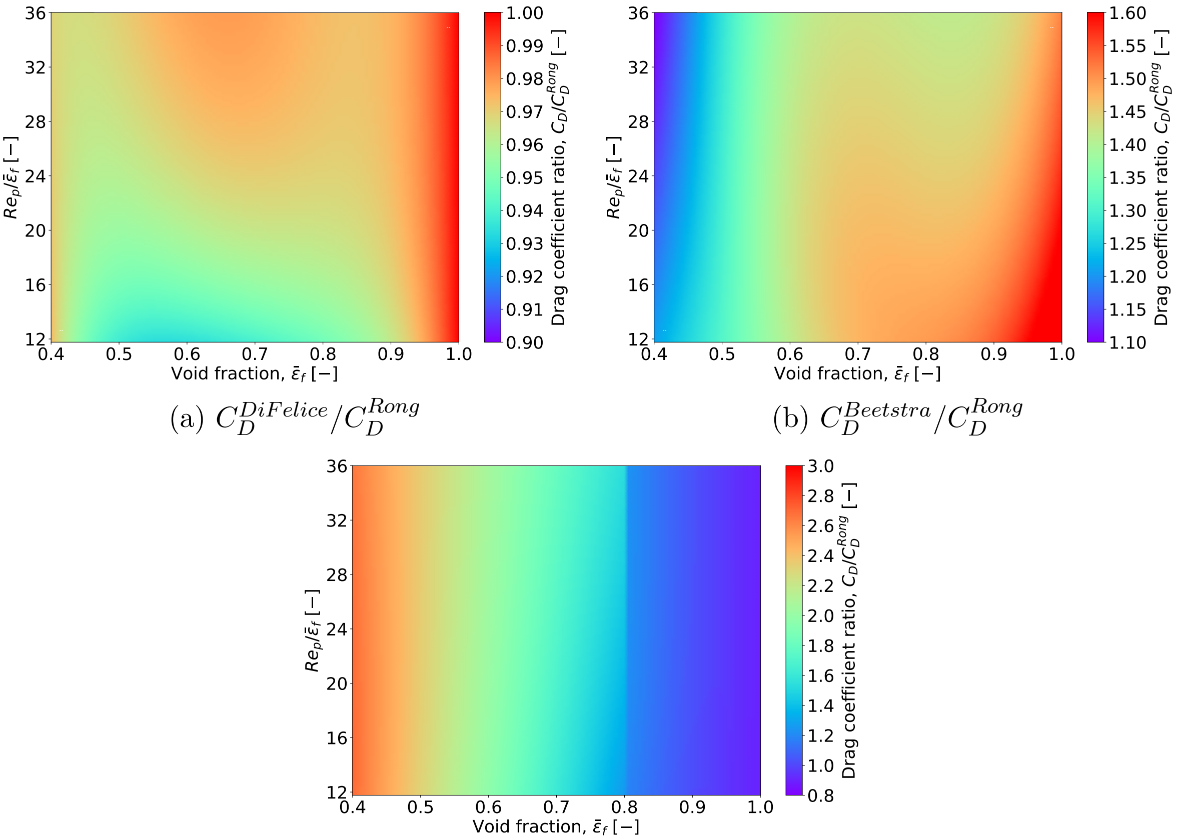
(b) Alumina

Figure 6.12 – Bed porosity with different drag correlations for (a) alginate and (b) alumina particles.

different data is expected. In the case of Rong and Beetstra drag correlations, since the models are highly complex, overfitting ε_f is a hypothesis for the slightly lower accuracy compared to Di Felice. Apart from that, all models could capture the experimental behavior of the bed.

Lack of fitting is found in Gidaspow drag correlation results. For both particles, the last points are the only ones within the 90% double tail confidence interval. The RMSE obtained using Gidaspow was 23% and 104% higher than the R-Z estimations for the alginate and alumina, respectively. The Gidaspow drag model is the only one that severely underperforms among the drag models tested.

In Figures 6.13 and 6.14, we compare the C_D calculated using all correlations in this work with C_D^{Rong} (calculated using the Rong drag model) for the alginate and alumina particles, respectively. This comparison is done by presenting a surface plot of C_D/C_D^{Rong} as a function of $\mathrm{Re}_p/\varepsilon_f$ and ε_f.



(a) $C_D^{DiFelice}/C_D^{Rong}$

(b) $C_D^{Beetstra}/C_D^{Rong}$

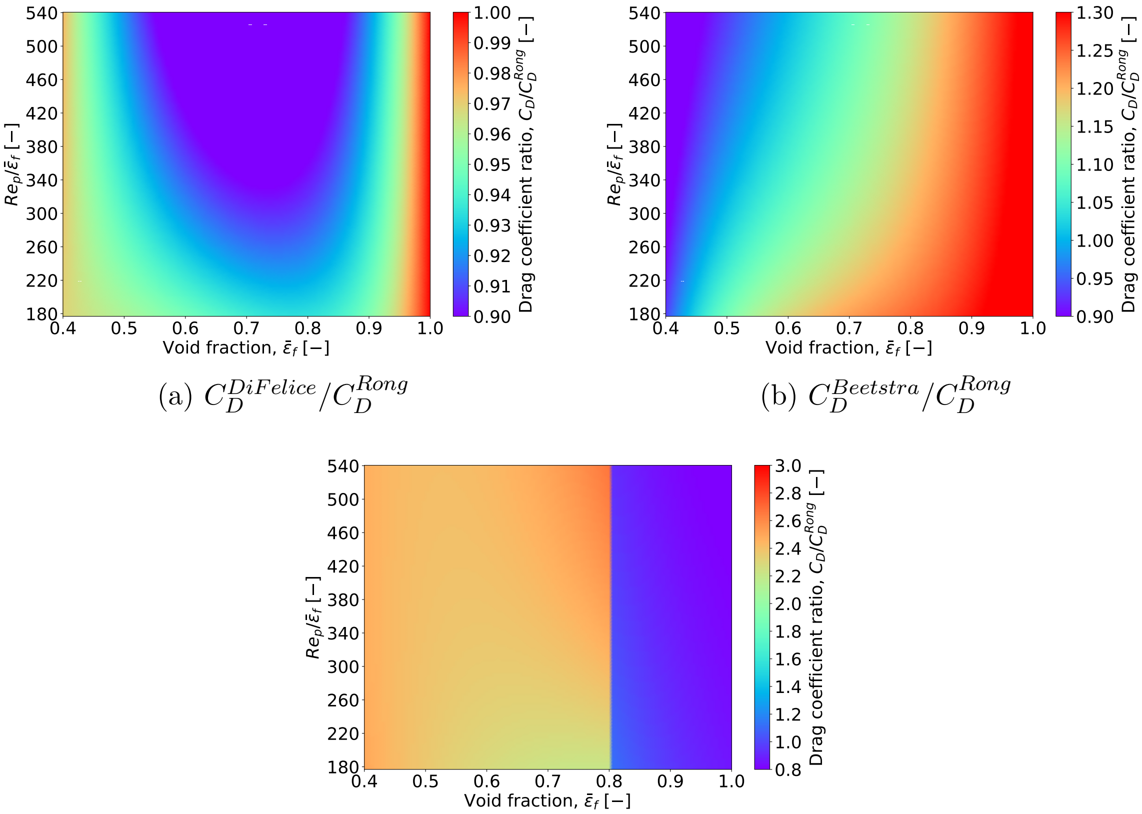
(c) $C_D^{Gidaspow}/C_D^{Rong}$

Figure 6.13 – C_D/C_D^{Rong} comparison among correlations for the alginate particles, showing the ratios a) $C_D^{DiFelice}/C_D^{Rong}$, b) $C_D^{Beetstra}/C_D^{Rong}$, and c) $C_D^{Gidaspow}/C_D^{Rong}$ for Di Felice, Beetstra, and Gidaspow, respectively. Color scales are different for each correlation due to their high discrepancy.

As shown in Figures 6.13a and 6.14a, $C_D^{DiFelice}/C_D^{Rong}$ is very close to 1.0 in most of the studied region. For the alginate particles (Figure 6.13a), the drag coefficient is almost identical for most void fractions and particle Reynolds numbers, which explains the similarity between the results of both correlations in Figure 6.12a. As for the alumina particles, the center of the graphic in Figure 6.14a deviates from 1, which also has an effect on the porosity trend observed in Figure 6.12b.

$C_D^{Beetstra}/C_D^{Rong}$ (Figures 6.13a and 6.14a) highlights the higher sensitivity of Beetstra drag correlation to the flow regime. In the case of the alumina particles, the average $C_D^{Beetstra}/C_D^{Rong}$ for the studied region was 1.1, showing a considerably higher drag coefficient only at very dilute regions ($\varepsilon_f > 0.8$), and significantly lower at very concentrated regions with high Re_p; both rarely occurring in the case of fluidized beds. On the other hand, for alginate, the average ratio was 1.4, not reaching 1.0 in any region of the graphic, which explains the detachment between Rong and Beetstra in Figure 6.12a.

As for the Gidaspow drag correlation, $C_D^{Gidaspow}/C_D^{Rong}$ was, on average, 1.7 and 1.9 for alginate and alumina particles, respectively. The Gidaspow drag correlation is



(a) $C_D^{DiFelice}/C_D^{Rong}$

(b) $C_D^{Beetstra}/C_D^{Rong}$

(c) $C_D^{Gidaspow}/C_D^{Rong}$

Figure 6.14 – C_D/C_D^{Rong} comparison among correlations for the alumina particles, showing the ratios a) $C_D^{DiFelice}/C_D^{Rong}$, b) $C_D^{Beetstra}/C_D^{Rong}$, and c) $C_D^{Gidaspow}/C_D^{Rong}$ for Di Felice, Beetstra, and Gidaspow, respectively. Color scales are different for each correlation due to their high discrepancy.

considerably less sensitive to ε_f, with a slightly better agreement with the experimental data at dilute regions, close to the limit when the model by Wen and Yu (1966) takes place. However, the important discontinuity in this region can lead to highly unphysical drag force distribution in those regions. Similar features on $C_D^{Gidaspow}$ were observed by Marchelli *et al.* (2020).

6.7 Conclusions

In the present work, we used Lethe (BLAIS *et al.*, 2020; GOLSHAN *et al.*, 2022), a FEM-based CFD and DEM code with CFD-DEM coupling capabilities, to simulate the behavior of a pilot-scale liquid-solid fluidized bed. The numerical results were compared with experiments with water and for two distinct particle types: alginate ($\rho_p = 1029 \mathrm{kg\,m^{-3}}$) and alumina ($\rho_p = 3586 \mathrm{kg\,m^{-3}}$). We made all comparisons in the pseudo-steady-state regime for a considerable interval of real-time and multiple fluidization velocities.

Comparisons between simulations with and without Saffman lift force show that the application of the force creates more realistic dynamics of particles, avoiding channeling close to the walls caused by inherent distortions on the void fraction distribution along the radius. This result is in agreement with other results in the literature (KOERICH *et al.*, 2018; ZBIB *et al.*, 2018; NIJSSEN *et al.*, 2020)

The comparison between experimental and simulated total pressure drop shows that the simulations could accurately represent the interphase momentum change of the real system for all the inlet conditions and both particles. Furthermore, this result shows that the magnitude of drag was successfully achieved.

Bed average porosity comparisons also show that the numerical results agree with the experiments in bed expansion. Bed porosities predicted by the simulations showed better agreement with experiments than the Richardson-Zaki equation (RICHARDSON; ZAKI, 1954) with the n index calculated by the correlation proposed by Khan and Richardson (1989). It is worth emphasizing the consistency of all tested conditions.

Tests with meshes with different topologies presented a high agreement with each other and the experiments. Higher unwanted oscillations in the dynamics of the particles are observed when the outer cell layers of the cylinder have a lower aspect ratio, which negatively impacts the void fraction radial distribution and drag. However, the quantitative results indicate a close fitting between experiments and simulations with all meshes.

Comparison between the drag correlations proposed by Rong *et al.* (2013), Felice (1994), Beetstra *et al.* (2007), and Gidaspow (1994) shows that the Di Felice, Rong, and Beetstra drag correlations are capable of reproducing the experimental behavior, presenting better precision on the reproduction of the experimental bed expansion than the Richardson-Zaki equation. The only exception was Gidaspow.

The results of the present work highlight the capability of the unresolved CFD-DEM to simulate a variety of solid-fluid systems in a wide range of flow regimes with high precision. Since the method is phenomenological, extrapolations of the case for other liquid-particle systems may be valid. However, additional work on solid-fluid force submodels is required to maximize the accuracy of this approach. We recommend the application of Saffman lift forces and void fraction smoothing liquid fluidized beds unresolved CFD-DEM simulations. Cells with high aspect ratios are also recommended to avoid the need for void fraction smoothing lengths larger than $2d_p$.

6.8 Acknowledgments

Thanks are due to São Paulo Research Foundation (FAPESP, grant #2019/19173-9, and grant #2020/14567-6); the Funding Authority for Studies and Projects of the Ministry of Science, Technology and Innovation (FINEP/MCTI) by means of the Program of Human Resources of the Brazilian National Petroleum Agency (PRH-ANP/MCTI); and the Brazilian National Council for Scientific and Technological Development (CNPq, process number 408618/2018-3) for financial support. The authors also thank Federal University of São Carlos (UFSCar) and Polytechnique Montréal for the infrastructure and all the technical support. This study was financed in part by the Coordenação de Aperfeicoamento de Pessoal de Nível Superior – Brasil (CAPES) – Finance Code 001.

We 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.

CHAPTER 7

ARTICLE 3: CFD-DEM STUDY OF MIXING IN A LIQUID FLUIDIZED BED

Victor Oliveira Ferreira, Bruno Blais, Gabriela Cantarelli Lopes

Under preparation.

7.1 Abstract

In this work, we assess the internal dynamics of particles in a liquid-solid fluidized bed using a validated unresolved CFD-DEM model. To do so, we applied the Nearest Neighbors Method (NNM) and the mixing index based on the principal component analysis proposed by Doucet *et al.* (2008) on the quantification of bed mixing for several regimes and particle properties. Discussions on the advantages and drawbacks of each method are provided. Using results for several inlet flow rates, we show that the mixing speed reaches a *plateau* and does not increase significantly with the fluid speed. The proposed frequency-based principal component analysis shows that the axial direction (z) is predominant over the radial and azimuthal (r and θ) in the particles' mixing, especially for looser beds ($\bar{\varepsilon}_f > 0.5$). Additionally, we demonstrate that, except for the sliding friction coefficient, the interaction properties have almost negligible influence on the mixing behavior. Despite this, concentrated beds may be more resistant to mixing when higher sliding friction coefficient particles are used.

7.2 Introduction

Liquid-solid fluidized beds (LSFB) are very versatile equipment, applicable to various processes in which heat and/or mass exchange between particles and liquids are desired (EPSTEIN, 2003a). In those pieces of equipment, the drag force counterbalances the apparent mass of the particles, agitating particles in a dynamically stable and uniform fluidized state. Though the interphase momentum exchange plays the most important role in the fluidization state, the collisions between particles are inheriting the fluidization dynamics and should be taken into account. Anaerobic fluidized bed reactors (AFBRs), for example, rely on uniform fluid-particle and particle-particle interactions through the equipment's volume to control the removal rate of the treated matter from the particle's surface, hence, mixing plays an important role in the process's effectiveness.

Despite its importance, measuring the degree of mixing of particles *in-situ* is not trivial, especially in liquid-particle systems. It is challenging to track the movement of individual particles, and, consequently, detect abnormal distributions of particles through the equipment's volume, leading to unpredictability. The experimental methods applied to the measurement of mixing are divided into two categories: intrusive and non-intrusive.

The intrusive methods are mainly sampling and probing. Sampling material from the equipment can be an important inconvenience and, in many applications, impossible to be done without affecting the process. Probing is very limiting in terms of resolution and can only be used on very dilute systems, in which the presence of particles does not affect the detection of the others.

Non-intrusive methods involve various techniques, from rudimentary visual observation, which is frequently applied to real systems, to more robust solutions such as Radioactive Particle Tracking (RPT) (JAIN *et al.*, 2017) and Electrical Resistance Tomography (RAZZAK *et al.*, 2007). The latter techniques are usually precise, however, not practical at an industrial scale, being limited to scientific applications. Other examples of non-intrusive techniques are Particle Image Velocimetry (PIV) (CHEN; FAN, 1992), Laser Doppler Velocimetry (LDV) (LI *et al.*, 2017), and Positron Emission Particle Tracking (PEPT) (HENSLER *et al.*, 2015).

One alternative to studying the mixing behavior in liquid-solids systems avoiding the experimental limitations is to use computational simulation techniques. Its advantages compared to experimental methods are, for example, higher time and space resolution, lower cost, versatility, and lower risk. The unresolved Computational Fluid Dynamics-Discrete Elements Method coupling method (CFD-DEM) (ZHOU *et al.*, 2010; BéRARD *et al.*, 2020), validated to several fluid-particle systems (RENZO *et al.*, 2011; PEPIOT; DESJARDINS, 2012; BLAIS *et al.*, 2016; GEITANI; BLAIS, 2023b; FERREIRA *et al.*, 2023a), stands out for being able to simulate fluid-particle and particle-particle interactions in systems with hundreds of thousands of particles at a reasonable computational cost (BéRARD *et al.*, 2020).

The unresolved CFD-DEM is particularly powerful in the study of particles' mixing compared to other approaches, such as the Two-Fluid Model (TFM), since the

DEM allows for individual particle tracking. Assessing particles' position with time allows for the use of several methods for mixing measurement, including variance reduction ratio (DANCKWERTS, 1953), nearest neighbors method (NNM), neighbor distance method, Average height method (FAN *et al.*, 1970), mixing entropy (ARNTZ *et al.*, 2008), mixing segregation index (EITZLMAYR; KHINAST, 2015), Lacey mixing index (LACEY, 1954), Doucet mixing index (DOUCET *et al.*, 2008), and others (BHALODE; IERAPETRITOU, 2020).

In such methods, mixing is expressed as an index varying between 0 (unmixed) and 1 (fully mixed). Usually, the system is considered fully mixed when the mixing index is > 0.95, and the time it takes for the value to be achieved (t_{95}) is used to express mixing effectiveness, allowing for the comparison between mixing methods, regimes, and equipment designs. Recently, we have validated the unresolved CFD-DEM method (FERREIRA *et al.*, 2023a) against liquid-solid fluidized bed experiments (FERREIRA *et al.*, 2023b) at a wide range of regimes. However, the model was not yet applied to the study of particle mixing. Additionally, few studies on mixing in liquid-solid fluidized beds have been reported and most of those efforts are focused on bidisperse systems (ESCUDIé *et al.*, 2006; PENG *et al.*, 2016; KHAN *et al.*, 2020; XIE *et al.*, 2021; XIE *et al.*, 2022).

In this work, we assess the mixing behavior of particles in a liquid-solid fluidized bed using the validated unresolved CFD-DEM model (FERREIRA *et al.*, 2023a). To do so, we measured the degree of mixing of three different monodispersed beds at very different regimes. The NNM (FAN *et al.*, 1970) and the method by Doucet *et al.* (2008) were used and compared. We discussed the effect of the regime on the mixing time, and assessed the principal component of mixing. Additionally, the role of particles' collisional properties in the mixing behavior was also investigated.

7.3 Governing equations

Here, we briefly describe the model formulation applied to this work. A full description of the model can be found in previous papers of the group (GEITANI *et al.*, 2023a; GEITANI; BLAIS, 2023b; GEITANI *et al.*, 2023b; FERREIRA *et al.*, 2023a). For a more fundamental description of the Volume-Average Navier-Stokes equations, we refer the reader to Zhou *et al.* (2010), Bérard *et al.* (2020), Gidaspow (1994), and Geitani *et al.* (2023a); and for the DEM formulation, we refer to Zhu *et al.* (2007), Blais *et al.* (2019), and Golshan *et al.* (2022).

7.3.1 Solid phase formulation

For each particle, we apply Newton's second law of motion. The linear and tangential components of the momentum are, respectively:

$$m_i \frac{d\boldsymbol{v}_i}{dt} = \sum_{j=1, j\neq i}^{N_p} (\boldsymbol{f}_{c,ij}) + \sum_{j, j\neq i}^{N_p} (\boldsymbol{f}_{nc,ij}) + \boldsymbol{f}_{pf,i} + \boldsymbol{f}_{g,i} \tag{7.1}$$

$$I_i \frac{d\boldsymbol{\omega}_i}{dt} = \sum_{j, j\neq i}^{N_p} (\boldsymbol{M}_{t,ij} + \boldsymbol{M}_{r,ij}) \tag{7.2}$$

where i is the current particle, j is the interacting particle, N_p is the total number of particles interacting with particle i, $\boldsymbol{v}$ is the particle's velocity, $\boldsymbol{f}_c$ is the set of contact forces, $\boldsymbol{f}_g$ is the gravity force, $\boldsymbol{f}_{pf}$ the summation of fluid-particle interacting forces, and $\boldsymbol{M}_t$ and $\boldsymbol{M}_r$ are the tangential and rolling friction torques.

Particles' collisions are simulated using the soft-sphere model proposed by Cundall and Strack (1979):

$$\boldsymbol{f}_{c,ij} = \boldsymbol{f}_{cn,ij} + \boldsymbol{f}_{ct,ij} = -k_{n,ij}\boldsymbol{\delta}_{n,ij} - \gamma_{t,ij}\dot{\boldsymbol{\delta}}_{n,ij} - k_{t,ij}\boldsymbol{\delta}_{t,ij} - \gamma_{t,ij}\dot{\boldsymbol{\delta}}_{t,ij} \tag{7.3}$$

where the subscripts n and t mean normal and tangential, respectively. k and γ are the stiffness and the damping coefficients of the colliding pair, calculated according to the equations in Table 7.1.

Table 7.1 – DEM equations

Property	**Equation**
Radius of particle i	R_i
Distance between i and contact point	$\boldsymbol{r}_i$
Equivalent mass	$\frac{1}{m^*_{ij}} = \frac{1}{m_i} + \frac{1}{m_j}$
Equivalent radius	$\frac{1}{R^*_{ij}} = \frac{1}{R_i} + \frac{1}{R_j}$
Equivalent Young's modulus	$\frac{1}{Y^*_{ij}} = \frac{1-\nu_i^2}{Y_i} + \frac{1-\nu_j^2}{Y_j}$
Equivalent shear modulus	$\frac{1}{G^*_{ij}} = \frac{2(2+\nu_i^2)(1-\nu_i^2)}{G_i} + \frac{2(2+\nu_j^2)(1-\nu_j^2)}{G_j}$
Normal stiffness	$k_{n,ij} = \frac{4}{3} Y^*_{ij} \sqrt{R^*_{ij} \boldsymbol{\delta_{n,ij}}}$
Tangential stiffness	$k_{t,ij} = 8 G^*_{ij} \sqrt{R^*_{ij} \boldsymbol{\delta_{n,ij}}}$
Normal damping	$\gamma_{n,ij} = -2\sqrt{\frac{5}{6}} \frac{ln(e)}{\sqrt{ln^2(e)+\pi^2}} \sqrt{\frac{2}{3} k_{n,ij} m^*_{ij}}$
Tangential damping	$\gamma_{t,ij} = -2\sqrt{\frac{5}{6}} \frac{ln(e)}{\sqrt{ln^2(e)+\pi^2}} \sqrt{k_{t,ij} m^*_{ij}}$

The tangential and rolling friction torques are calculated, respectively, by:

$$\boldsymbol{M}_{t,ij} = \boldsymbol{r}_i \times (\boldsymbol{f}_{ct,ij}) \tag{7.4}$$

$$\boldsymbol{M}_{r,ij} = -\mu_{r,ij} \left|\boldsymbol{f}_{ct,ij}\right| \frac{\boldsymbol{\omega}_{ij}}{\left|\boldsymbol{\omega}_{ij}\right|} R^*_{ij} \tag{7.5}$$

where the coefficient of rolling friction ($\mu_{r,ij}$) and the equivalent radius (R^*_{ij}) are calculated by the equations in Table 7.1.

The particle-fluid interaction forces are calculated by:

$$\boldsymbol{f}_{pf,i} = \boldsymbol{f}_{d,i} + \boldsymbol{f}_{\nabla p,i} + \boldsymbol{f}_{\nabla \cdot \boldsymbol{\tau},i} + \boldsymbol{f}_{Ar,i} + \boldsymbol{f}_{\mathrm{Saff},i} \tag{7.6}$$

where $\boldsymbol{f}_{d,i}$ is the drag force, $\boldsymbol{f}_{\nabla p,i}$ is the pressure gradient force, $\boldsymbol{f}_{\nabla \cdot \boldsymbol{\tau},i}$ is the particle-fluid shear force, and $\boldsymbol{f}_{\mathrm{Saff},i}$ is the Saffman lift force. Each of these terms are explained in subsection 7.3.3. Other forces, namely Basset, Magnus, and virtual mass, were neglected (RENZO; MAIO, 2007; FERREIRA *et al.*, 2023a).

7.3.2 Liquid phase modeling

The liquid phase in the unresolved CFD-DEM approach is represented by the Volume-Averaged Navier-Stokes equations (VANS). The model type A (GIDASPOW, 1994), or model set II (ZHOU *et al.*, 2010), of the VANS equations was applied to the simulations in the present work (GEITANI *et al.*, 2023a; FERREIRA *et al.*, 2023a).

$$\frac{\partial \varepsilon_f}{\partial t} + \nabla \cdot (\varepsilon_f \boldsymbol{u}) = 0 \tag{7.7}$$

$$\rho_f \left[\frac{\partial \varepsilon_f \boldsymbol{u}}{\partial t} + \nabla \cdot (\varepsilon_f \boldsymbol{u} \otimes \boldsymbol{u}) \right] = -\varepsilon_f \nabla p + \varepsilon_f \nabla \cdot \boldsymbol{\tau} - \boldsymbol{F}_{pf} \tag{7.8}$$

In Eq. (7.8), $\boldsymbol{\tau}$ is the viscous shear stress tensor

$$\boldsymbol{\tau} = \mu \left[(\nabla \cdot \boldsymbol{u}) + (\nabla \cdot \boldsymbol{u})^T - \frac{2}{3} (\nabla \cdot \boldsymbol{u} \boldsymbol{I}) \right] \tag{7.9}$$

where μ is the dynamic viscosity and $\boldsymbol{I}$ is the identity matrix, and $\boldsymbol{F}_{pf}$ is the fluid-particle momentum exchange (source) term

$$\boldsymbol{F}_{pf} = \frac{1}{\Delta V_{\Omega_C}} \sum_{i}^{N_{p,C}} (\boldsymbol{f}_{d,i}) = \frac{1}{\Delta V_{\Omega_C}} \sum_{i}^{N_{p,c}} (\boldsymbol{f}_{pf,i} - \boldsymbol{f}_{\nabla p} - \boldsymbol{f}_{\nabla \cdot \boldsymbol{\tau}} - \boldsymbol{f}_{Ar}) \tag{7.10}$$

where the index $N_{p,C}$ stands for the number of particles inside the cell Ω_C in which the averaging is applied. The force terms in Eq. (7.10) are the same as in Eq. (7.6).

7.3.3 Interphase momentum exchange

In this subsection, the terms in Eqs. (7.10) and (7.6) are individually described. Details about each of the terms can be found on Crowe *et al.* (2011).

7.3.3.1 Pressure and buoyancy forces

In Lethe, ∇p does not account for the hydrostatic pressure, hence, buoyancy forces need to be added separately in Eq. (7.6). The expressions that represent the pressure gradient and buoyancy forces are, respectively:

$$\boldsymbol{f}_{\nabla p,i} = V_{p,i} \nabla p \tag{7.11}$$

$$\boldsymbol{f}_{Ar,i} = V_{p,i} \rho_f \boldsymbol{g} \tag{7.12}$$

where $V_{p,i}$ is the volume of the particle i, ρ_f is the density of the fluid, p stands for dynamic pressure, $\boldsymbol{g}$ is the gravity acceleration vector. Note that Eq. (7.11) accounts for the interchange force due to the undisturbed pressure only.

7.3.3.2 Shear force

The force resulting from the fluid viscous shear stress is written as

$$\boldsymbol{f}_{\nabla \cdot \boldsymbol{\tau},i} = V_{p,i} \boldsymbol{\nabla} \cdot \boldsymbol{\tau} \tag{7.13}$$

7.3.3.3 Drag force

The Drag force is defined as:

$$\boldsymbol{F}_d = \sum_{i}^{N_{p,C}} \boldsymbol{f}_{d,i} = \sum_{i}^{N_{p,C}} \beta_i \left(\boldsymbol{u} - \boldsymbol{v}_i \right) \tag{7.14}$$

For model A, β_i is the interphase momentum transfer coefficient, given by correlations in the literature (GIDASPOW, 1994; FELICE, 1994; BEETSTRA *et al.*, 2007; MAZZEI; LETTIERI, 2007; RONG *et al.*, 2013). The correlation by Rong *et al.* (2013) was applied to all simulations in this work for presenting the best correspondence with the experimental data in the validation campaign (FERREIRA *et al.*, 2023b; FERREIRA *et al.*, 2023a). In the correlation, β_i is:

$$\beta_i = \frac{1}{2} C_{D0} \frac{\pi d_p^2}{4} \rho_f \left| \boldsymbol{u} - \boldsymbol{v}_i \right| G(\varepsilon_f, \mathrm{Re}_{p,i}) \tag{7.15}$$

where C_{D0} is the drag coefficient for a single particle, given by the correlation by DallaValle (1948)

$$C_{D0} = \left(0.63 + \frac{4.8}{\sqrt{\mathrm{Re}_p}} \right)^2 \tag{7.16}$$

and $G(\varepsilon_f, \mathrm{Re}_{p,i})$ is

$$G(\varepsilon_f, \mathrm{Re}_{p,i}) = \frac{C_{D,i}}{C_{D0}} = \varepsilon_f^{\,2 - \left\{ 2.65 \left(\varepsilon_f + 1 \right) - \left(5.3 - 3.5 \varepsilon_f \right) \varepsilon_f^2 exp \left[- \frac{\left(1.5 - log_{10} \mathrm{Re}_{p,i} \right)^2}{2} \right] \right\}} \tag{7.17}$$

7.3.3.4 Saffman lift force

The Saffman lift force is calculated by the Saffman-Mei lift force model, which is a combination of the equations proposed by Saffman (1965), Saffman (1968) and Mei (1992)

$$\boldsymbol{f}_{\mathrm{Saff},i} = 1.161 C_{\mathrm{Saff},i} d_p^2 (\mu \rho_f)^{1/2} \left|\boldsymbol{\omega}_{c,i}\right|^{-1/2} \left[(\boldsymbol{u} - \boldsymbol{v}_i) \times \boldsymbol{\omega}_{c,i}\right] \tag{7.18}$$

where $\boldsymbol{\omega}_{c,i}$ is the fluid vorticity ($\nabla \times \boldsymbol{u}$), $C_{\mathrm{Saff},i}$ is the Saffman lift coefficient

$$C_{\mathrm{Saff},i} = \begin{cases} \left(1 - 0.3314\alpha^{1/2}\right) \exp\left(\frac{-\mathrm{Re}_{p,i}}{10}\right) + 0.3314\alpha^{1/2}, \text{ for } \mathrm{Re}_{p,i} \leq 40 \\ 0.0524(\alpha_l \mathrm{Re}_{p,i})^{1/2}, \text{ for } \mathrm{Re}_{p,i} > 40 \end{cases} \tag{7.19}$$

and α is given by

$$\alpha = \frac{d_p}{2\left|\boldsymbol{u} - \boldsymbol{v}_i\right|} \left|\boldsymbol{\omega}_{c,i}\right| \tag{7.20}$$

We refer the reader to Crowe *et al.* (2011) for further details on the Saffman lift force model. Previous works of the research group present show the importance of Saffman lift on the present case (FERREIRA *et al.*, 2023a).

7.4 Methods

All simulations were carried out using Lethe (BLAIS *et al.*, 2020; GOLSHAN *et al.*, 2022; GEITANI; BLAIS, 2023b; GEITANI *et al.*, 2023a; GEITANI *et al.*, 2023b). The software is an open-source CFD/DEM/CFD-DEM tool based on the robust deal.II Finite Elements Method (FEM) library (ARNDT *et al.*, 2021; ARNDT *et al.*, 2022). The methodology applied to this work was previously validated against pilot-scale experimental data by us (FERREIRA *et al.*, 2023b), as described in Ferreira *et al.* (2023a). Lethe's unresolved CFD-DEM module was also validated for gas-solid fluidized beds (GEITANI; BLAIS, 2023b).

7.4.1 Simulation setup

The simulation setup applied to this work follows what is described in the validation study (FERREIRA *et al.*, 2023a). We used the pilot-scale circulating liquid-solid fluidized bed described in Ferreira *et al.* (2023b) and Ferreira *et al.* (2023a) as a reference to the simulations. The experimental unit is composed of a 1 m height, 10 cm diameter cylindrical vessel, where particles are fluidized in water.

Unless mentioned, simulations follow the parameters described in Table 7.2. Particles' collisional properties, namely Young's modulus (Y), coefficient of restitution (e), coefficient of rolling friction (μ_r), coefficient of sliding friction (μ_f); were the only parameters modified throughout the study (Section 7.4.3).

Table 7.2 – Simulation parameters.

Parameter	**Value**
Time integration method	BDF1
CFD time-step and coupling interval (Δt_{CFD})	0.001 s
DEM time-step (Δt_{DEM})	0.000 01 s
Diameter of the cylinder (D_C)	10 cm
Height of the cylinder (H_C)	1.10 m
Height of the inlet portion	10 cm
Mesh ($n_r \times n_\theta \times n_z$)	6 × 16 × 132
Liquid density (ρ_f)	996.78 kg m^{-3}
Liquid dynamic viscosity (μ)	8.352×10^{-4} Pa s
Young's modulus (Y)	10 MPa
Coefficient of restitution (e)	0.9
Poisson ratio (ν)	0.3
Coefficient of rolling friction (μ_r)	0.2
Coefficient of sliding friction (μ_f)	0.1
VANS model type	A
Void fraction calculation scheme	PCM
Void fraction smoothing length	$2.0 \times d_p$
Boundary conditions at the walls	Free slip
Drag model	Rong *et al.* (2013)
Lift force model	Saffman-Mei[1]
Gravity (g)	9.81 m s^{-2}

7.4.1.1 Mesh and initial packing of particles

We used deal.II's built-in tool to generate the numerical mesh used in all simulations. The mesh is composed of a 1.10 m height, 10 cm diameter cylinder, as shown in Figure 7.1. The number of elements in cylindrical coordinates ($n_r \times n_\theta \times n_z$) is $6 \times 16 \times 132$.

During simulations, particles are held 10 cm above the bottom of the mesh by a floating wall, leaving a free-of-particles inlet portion to the fluid (referred to as inlet portion in Figure 7.1), preventing sharp void fraction gradients which can lead to numerical instability (GEITANI *et al.*, 2023a). In all simulations, the fluid injection only starts after particles are fully settled above the floating wall. This is achieved by running a particle sedimentation pure DEM simulation, carried out on Lethe-DEM (GOLSHAN *et al.*, 2022), prior to the CFD-DEM simulation.

7.4.1.2 Particles

We used 3 different groups of particles, named A, B, and C. Particles A and C are referred to as alginate and alumina in the validation study (FERREIRA *et al.*, 2023a), as shown in Figure 7.2. Since particle C is more than 3 times denser than particle A, we

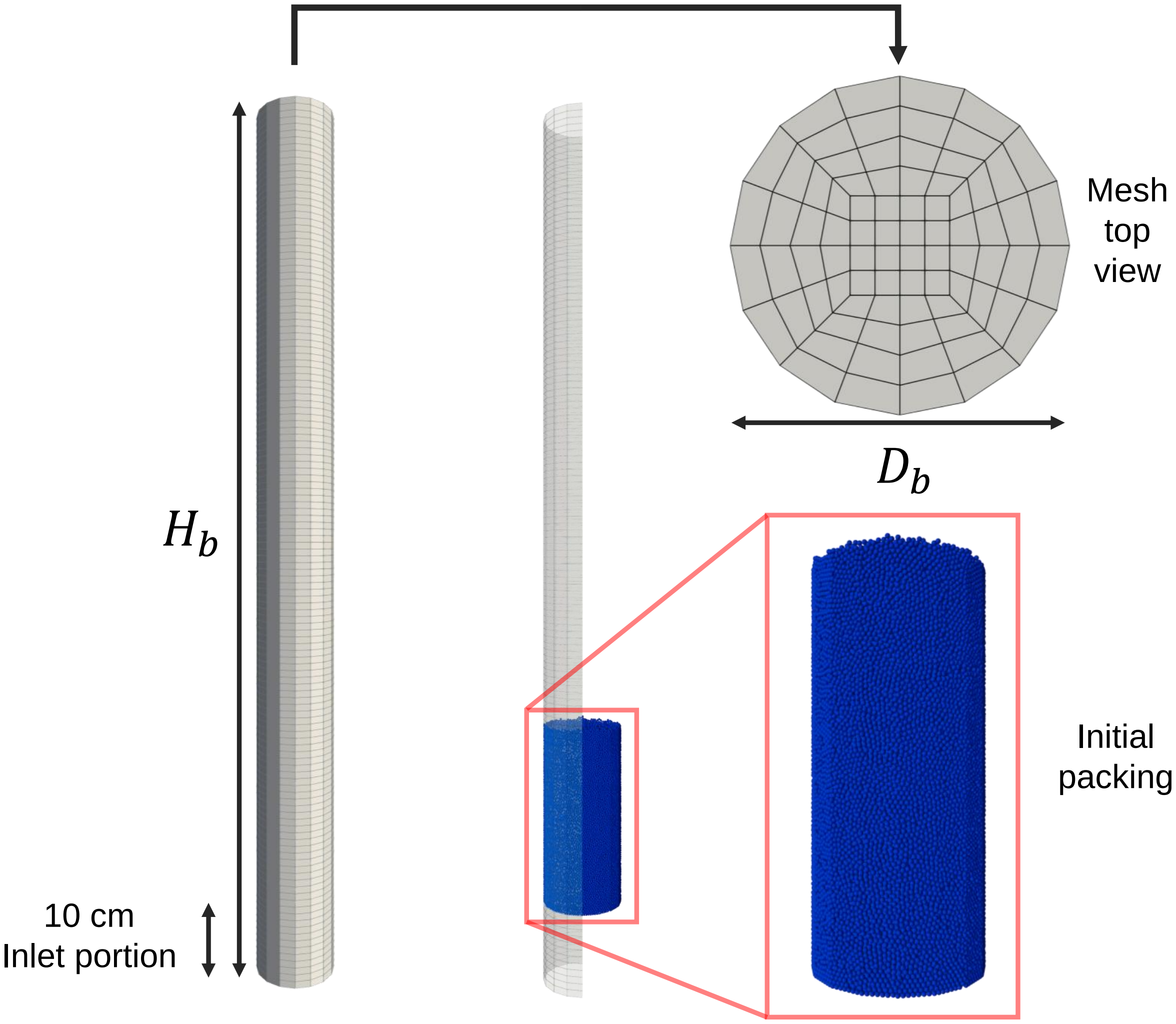


Figure 7.1 – Schematic representation of the mesh and initial packing of particles.

created a fictitious intermediary particle B with the same size as particle A and a density close to the acrylonitrile butadiene styrene (ABS) particle used in Ferreira *et al.* (2023b). The particular parameters associated with each particle are presented in Table 7.3.

Table 7.3 – Particle associated parameters.

Parameter	**Particle A**	**Particle B**	**Particle C**
Total real-time of simulations	35 s	20 s	20 s
Mixing index reference time (t_0)	10 s	5 s	5 s
Output time-step (Δt_{DEM})	0.50 s	0.20 s	0.20 s
Number of particles (N_p)	107960	107960	72400
Particle terminal velocity (U_0)	2.9 cm s^{-1}	20.8 cm s^{-1}	48.2 cm s^{-1}
Lowest and highest inlet velocities (U)	0.44 - 1.09 cm s^{-1}	4.24 - 11.00 cm s^{-1}	8.50 - 15.70 cm s^{-1}
Number of inlet velocities	7	2	8
Diameter of the particles (d_p)	2.66 mm	2.66 mm	3.09 mm
Density of the particles (ρ_p)	1029 kg m^{-3}	1822 kg m^{-3}	3586 kg m^{-3}

7.4.1.3 Validation

The main results of the validation campaign are presented in Figure 7.2. All results of the validation step were obtained using the parameters in Table 7.2. In the figure, the total pressure drop is compared to the analytical pressure drop calculated by (EPSTEIN, 2003b):

$$-\Delta p = \frac{M_p \left(\rho_p - \rho_f\right) g}{\rho_p A_b} \tag{7.21}$$

where M_p is the total mass of particles in the bed, ρ_p is particle's density, ρ_f is fluid's density, g is gravity acceleration constant, and A_b is the bed's cross section area. Bed expansion results (expressed as bed average porosity, $\bar{\varepsilon}_f$) are compared to experiments and estimations using the equation by Richardson and Zaki (1954) (R-Z). In the figures, α corresponds to a double-tail confidence interval. The simulation results correspond to time averages in the pseudo-steady-state regime, and the error bars are equivalent to one temporal standard deviation.

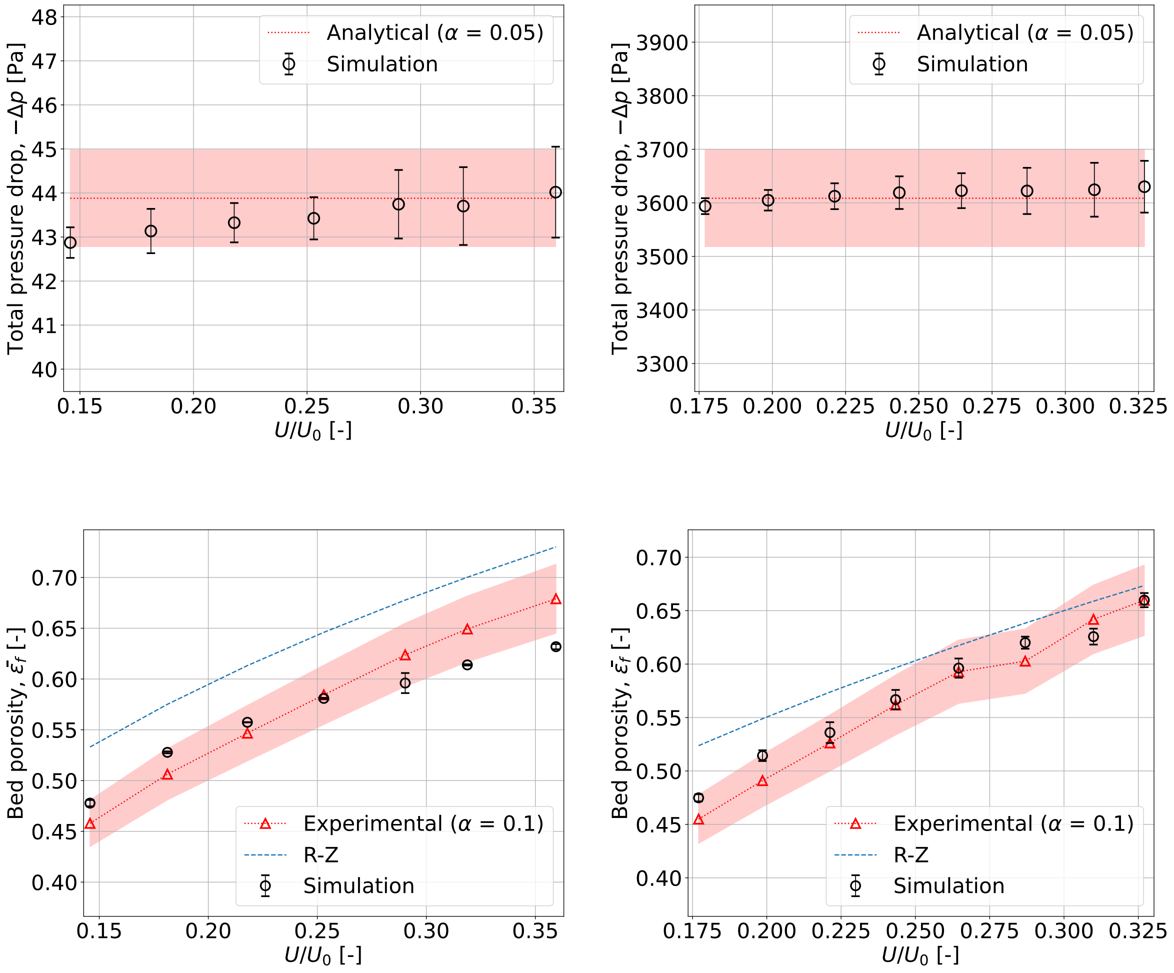
The results in Figure 7.2 were obtained for particles A and C (Table 7.3) using the parameters in Table 7.2 and the mesh in Figure 7.1. The simulations were capable of accurately reproducing the fluidized bed behavior. For further details on the importance of the Saffman lift force, the choice of the drag correlation, and the mesh, we refer the reader to Ferreira *et al.* (2023a).

7.4.2 Mixing index

As said, several methods can be applied to track particles' mixing using DEM information. In this work, we applied two: the nearest neighbors method (NNM) (FAN *et al.*, 1970; GODLIEB *et al.*, 2007) and Doucet's mixing index (DOUCET *et al.*, 2008). Both methods were applied to Lethe-DEM results using a post-processing module written in Python using the PyVista library (SULLIVAN; KASZYNSKI, 2019) (available on Lethe's official GitHub webpage, explained in details on Lethe's official documentation). Different from Lacey's mixing index (LACEY, 1954; GODLIEB *et al.*, 2007), the methods are grid-independent, meaning that only the particle position is required. NNM and Doucet's methods follow different principles, thus, providing different information about the particles mixing, as explained.

7.4.2.1 Nearest neighbors method

The method consists of splitting the particles into two groups according to their position at a given moment and tracking the number of neighbors per particle that are



(a) Alginate (Particle A)

(b) Alumina (Particle C)

Figure 7.2 – Average of the total pressure difference ($-\Delta p$) and bed expansion ($\bar{\varepsilon}_f$) of the validation campaign for the (a) alginate (particle A in this work) and (b) alumina particles (particle C in this work). Adapted from Ferreira *et al.* (2023a).

part of the other group. A schematic representation of the method is presented in Figure 7.3.

The NNM mixing index is calculated by (GODLIEB *et al.*, 2007)

$$\bar{M}_t^{NNM} = 1/N_p \sum_{n=1}^{N_p} M_n^{NNM} = 1/N_p \sum_{n=1}^{N_p} \frac{2N_{n,diff}}{N_{neighbors}} \tag{7.22}$$

where $\bar{M}_t^{NNM}$ is the average mixing index of the system at time t, M_n^{NNM} is the mixing index of the n^th particle, N_p is the total number of particles in the system, $N_{neighbors}$ is the number of nearest neighbors used in the method (user-defined parameter), and $N_{n,diff}$ is the number of neighbor particles that are of the opposite group of the n^{th} particle. In the present work, we chose $N_{neighbors} = 15$. The KD-tree nearest neighbors method implemented in the powerful Scikit-learn Python module (PEDREGOSA *et al.*, 2011) was used to find particles' nearest neighbors.

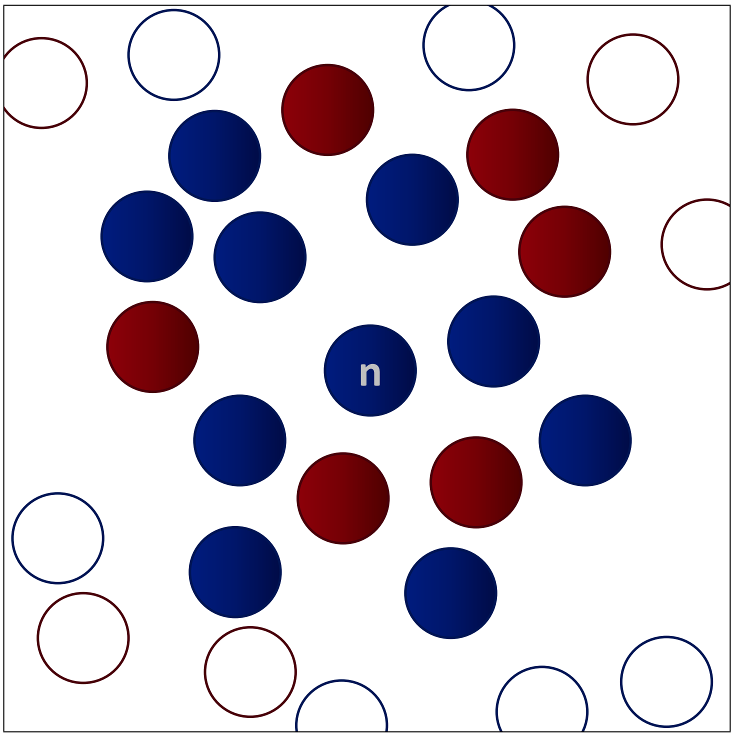
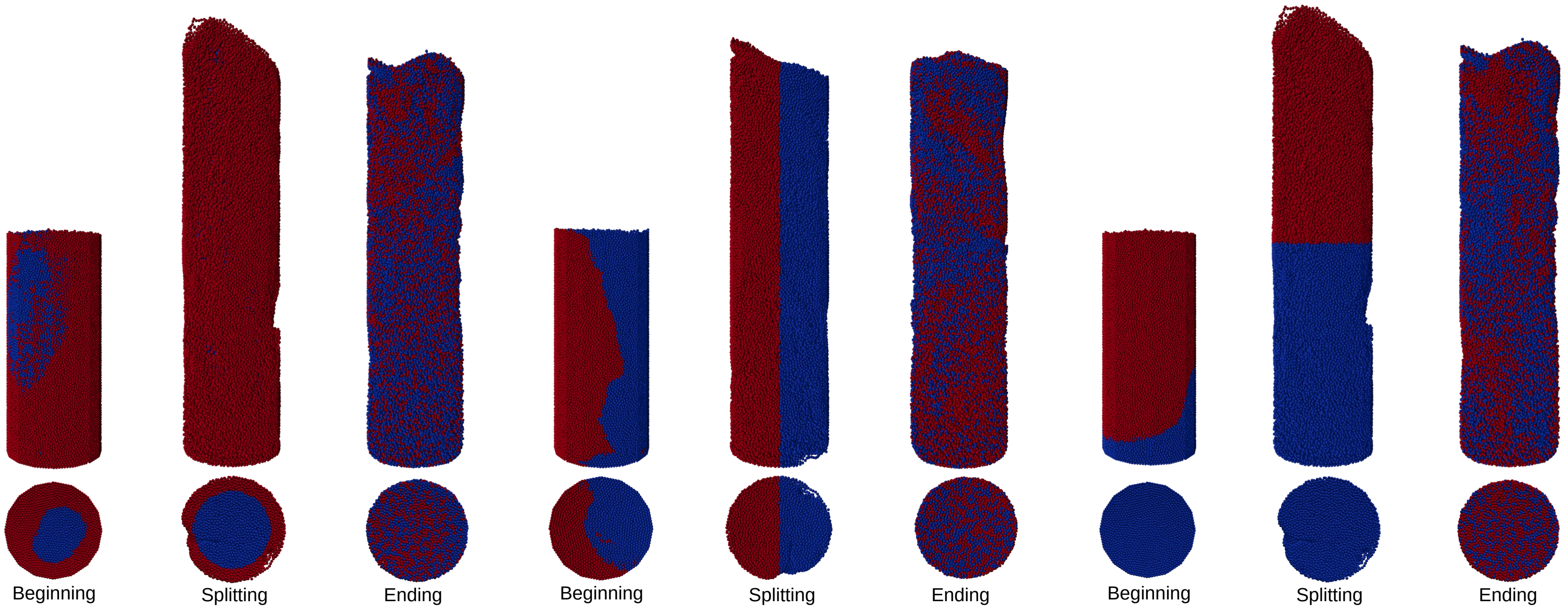


Figure 7.3 – Schematic bi-dimensional representation of the nearest neighbors method for mixing index calculation (GODLIEB *et al.*, 2007). In the figure, blue and red particles are members of two different groups. The n^{th} particle, to which the NNM mixing index is being calculated, is represented with the letter "n" on it. Hollow particles are not part of the 15 nearest neighbors of the particle and, hence, are not used in the mixing index calculation.

In Figure 7.3, blue and red particles are part of different groups. Filled particles are part of the 15 nearest neighbor particles of particle n, while hollow particles are not. $N_{n,diff}$ is 6 (number of filled red particles), according to Eq. (7.22), $M_n^{NNM} = 0.8$. Note that $0 \leq M_n^{NNM} \leq 2$, so that $0 \leq \bar{M}^{NNM} \leq 1$.

In NNM, the particles must be split in half so that the coordinate components can be analyzed individually. Since we are working in a cylindrical geometry, particles are split according to their position in cylindrical coordinates (namely r, θ, and z).

The moment of splitting is chosen so that the initial bed expansion, which presents different bed dynamics from the fully expanded pseudo-stationary bed, is not considered. This reference time (t_0) is 10 s for all particles, which is considered the time needed for Particle A to reach the pseudo-steady state (FERREIRA *et al.*, 2023a). An example of the bed splitting per coordinate is presented in Figure 7.4.



(a) Radial coordinate (r).

(b) Angular coordinate (θ).

(c) Axial coordinate (z).

Figure 7.4 – Example of side and bottom views of the beginning, splitting, and ending time-steps of simulations per coordinate. Particles are split according to their position so that NNM can be applied. Splitting directions follow the cylindrical coordinates system, i.e., radius (r), angle (θ), and height (z). Particles are split at the splitting time step such that half of the particles will be part of each group.

As shown in Figure 7.4, NNM results do not start from zero because, at t_0, the particles close to the splitting boundary will have neighbor particles on the opposite side. As a consequence, the higher the surface area of the splitting layer, the higher the initial mixing index will be. In our particular case, $z < \theta < r$. For this reason, we normalize $\bar{M}_t^{NNM}$ by calculating:

$$\bar{M}_t^{NNM} = \frac{\bar{M}_t^{NNM} - \bar{M}_{t0}^{NNM}}{1 - \bar{M}_{t0}^{NNM}} \tag{7.23}$$

7.4.2.2 Doucet's mixing index

Different from NNM, the method by Doucet *et al.* (2008) does not require domain splitting. Instead, the method applies a Principal Component Analysis (PCA) (SHLENS, 2014) on the correlation between the particles' position ($\boldsymbol{x} \in \mathbb{R}^d$, where d stands for the number of dimensions) at a given time-step ($\boldsymbol{x}_t$) and its position at a reference time-step ($\boldsymbol{x}_{t0}$). The method consists of the following steps:

1. Find the correlation matrix $\boldsymbol{C}_t$ given $C_{t,ij} = \rho(\boldsymbol{x}_{t,i}, \boldsymbol{x}_{t0,j}), \forall i, \forall j$;
2. Compute $\boldsymbol{B}_t = \boldsymbol{C}_t \boldsymbol{C}_t^T$;
3. Diagonalize $\boldsymbol{B}_t$ and find its maximum eigenvalue $\lambda_t = (\boldsymbol{\lambda}_t)_{max}$;
4. Find eigenvector associated to λ_t, $\boldsymbol{w}_t$;
5. Do the same to all t.

In a weak sense of mixing, the more mixed particles are the lower the correlation between the current and its position at the reference time-step and, consequently, the lower λ_t is going to be. The highest component of $\boldsymbol{w}_t$ gives the direction to which the mixing is poorest.

As shown by Blais *et al.* (2017), it is important to respect the natural coordinate system of the problem. Similar to what was done to NNM, the cylindrical coordinate system was used to represent the particles' positions. Also as in NNM, we used 10 s as the reference time (t_0) for the mixing index calculation.

To establish a comparison between NNM and Doucet's mixing index, the latter was normalized so that the index increases with mixing (contrary to Doucet *et al.* (2008), and Blais *et al.* (2017)).

$$\bar{M}_t^{Doucet} = 1 - \frac{\lambda_t}{\lambda_{t0}} \tag{7.24}$$

7.4.3 Interaction characteristics study

In this work, we also investigated the role of particles' interaction properties on the mixing. To do so, we simulated the liquid-solid fluidized bed dynamics varying particles' Young's modulus (Y), coefficient of restitution (e), coefficient of rolling friction (μ_r), and coefficient of sliding friction (μ_f). The properties were varied individually while the others were kept constant (e.g. when μ_f varies from the one presented in Table 7.2, e, Y, and μ_r are equal to those in Table 7.2). The values for each of the properties are shown in Table 7.4.

Table 7.4 – Particle properties applied to the particle interaction study.

Property	**Values**
Young's modulus (Y)	10^5, 10^7, and 10^9 Pa
Coefficient of restitution (e)	0.1, 0.3, 0.6, and 0.9
Coefficient of sliding friction (μ_f)	0.1, 0.2, 0.6, and 0.9
Coefficient of rolling friction (μ_r)	0.1, 0.2, 0.6, and 0.9

7.5 Results and Discussion

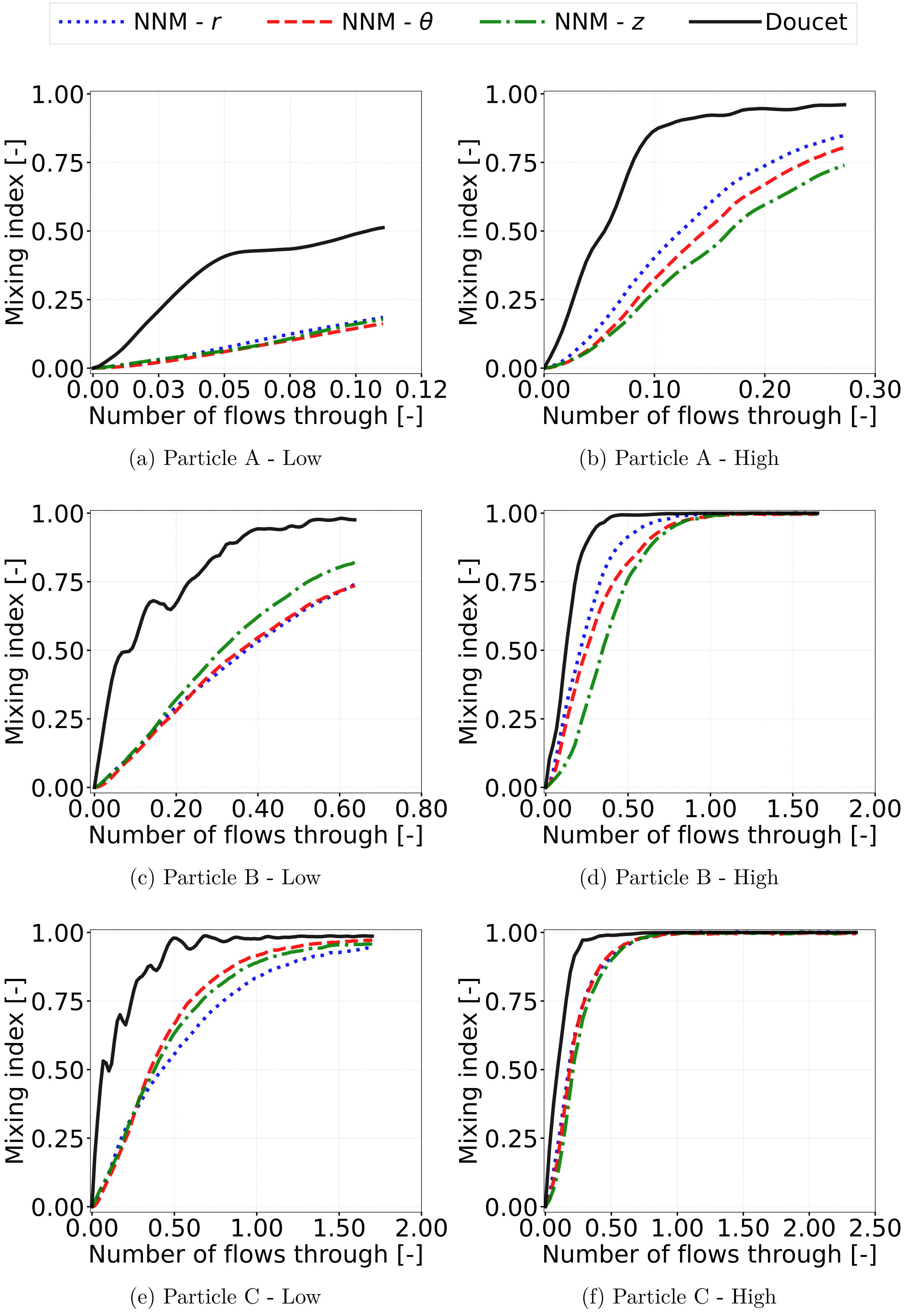
To establish comparisons among regimes, particles, and methods, the time results are presented as the dimensionless number of flows through:

$$N_{flows} = (t - t_0)\frac{U}{H_C} \tag{7.25}$$

where t_0 is the mixing index reference time (Table 7.3), U is the fluid inlet velocity, and H_C is the height of the cylinder. To avoid the influence of the initial bed expansion, all results were obtained in the fully developed pseudo-steady state. Additionally, the results of the study of the particle interaction properties can be found in the tables provided in the Supplementary Material, while in this section only the main features are presented and discussed. The data includes the number of flows through to reach 95, 90, 80, and 70% mixing (N^{95}_{flows}, N^{90}_{flows}, N^{80}_{flows}, and N^{70}_{flows}, respectively), the maximum mixing ($\bar{M}_{max}$), and the maximum mixing divided by the number of flows through to reach it for each mixing index ($\bar{M}_{max}/N^{max}_{flows}$).

7.5.1 Mixing indices comparison

In Figure 7.5 we show how the mixing index evolves with respect to the number of flows through using the different methods, for the three particles, and under the highest (high) and the lowest (low) fluid inlet flow rates.



(a) Particle A - Low

(b) Particle A - High

(c) Particle B - Low

(d) Particle B - High

(e) Particle C - Low

(f) Particle C - High

Figure 7.5 – Mixing index as a function of N_{flows} for the three particles at lowest (low) and highest (high) fluid inlet flow rates.

NNM and Doucet's mixing indices reflect very different aspects of particle mixing. In NNM, we split the domain and count the neighbors originally on the opposite side, which provides a straightforward view of mixing in each direction. The method is very versatile, given that the domain split can be done in any coordinate system. Additionally, the method allows for the evaluation of individual mixing indices to each particle, having more localized responses to each of the components. Such results can point to regions in the domain where the mixing is poorer, and not only returning in a global response.

The main disadvantages of the method are the lack of information about the global mixing without accounting for individual direction components, and the need for a manually defined $N_{neighbors}$. Furthermore, the maximum mixing index of 1 can only be reached if the domain is split such that half of the particles lie on each side, consequently, the need for a precise split is a limitation.

Oppositely, Doucet's method does not require splitting since particles are compared to themselves at a reference time-step only, automatically accounting for mixing in a more general manner. In addition, the poorest mixing component at a time t can be determined by the eigenvector $\boldsymbol{w}_t$, which directly compares the mixing components and highlights time-dependent effects. It is a convenient and efficient approach to estimate mixing using lagrangian information. Nevertheless, the method presents important drawbacks.

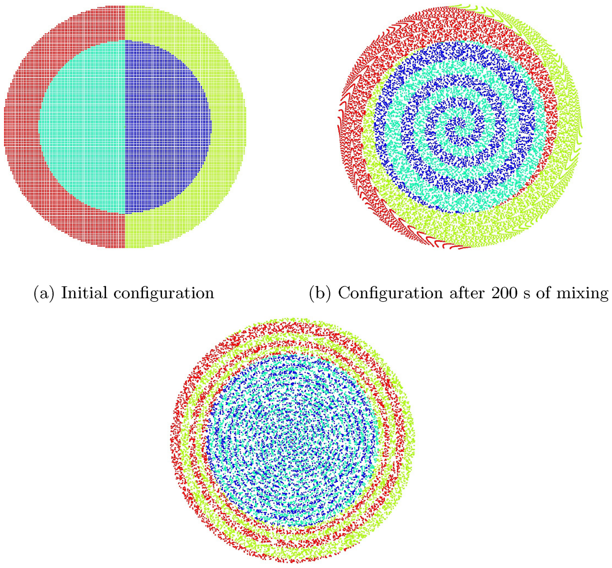
The method implemented by Doucet *et al.* (2008) introduces the weak sense of mixing, which presents limitations previously discussed by Blais *et al.* (2017). Doucet's mixing index applies principal component analysis (PCA) (SHLENS, 2014) on the estimation of mixing, which implies that the maximum eigenvalue at time t (λ_t) will reflect the particle with the maximum correlation between its position at time t ($\boldsymbol{x}_t$) and at reference time t_0 ($\boldsymbol{x}_{t0}$). Thus, it is expected that the correlation will be broken down by mixing, resulting in a vanishing λ_t. However, this is not necessarily true if mixing is non-linear. Blais *et al.* (2017) demonstrated this using the bi-dimensional example of particles mixing in a lamb-like vortex presented in Figure 7.6.

In the example, the particles velocity $\boldsymbol{v}_p$ is given by:

$$\boldsymbol{v}_p = [v_r, v_\theta]^T = \left[0, \frac{1}{r}\left(1 - e^{-r^2}\right)\right]^T \tag{7.26}$$

where v_r and v_θ are radial and azimuthal velocities, respectively. The resulting mixing indices as a function of time along with eigenvectors associated with the index are shown in Figure 7.7.

Although the system shown in Figure 7.6 is clearly unmixed, the results in Figure 7.7a point to a false positive to efficient mixing. The reason for this is the non-linear lamb-like vortex mixing with respect to the cartesian coordinates x and y. On the contrary, when the cylindrical coordinate system was applied (Figure 7.7b), poor mixing was properly captured as a result of the high correlation between the initial and the current position of

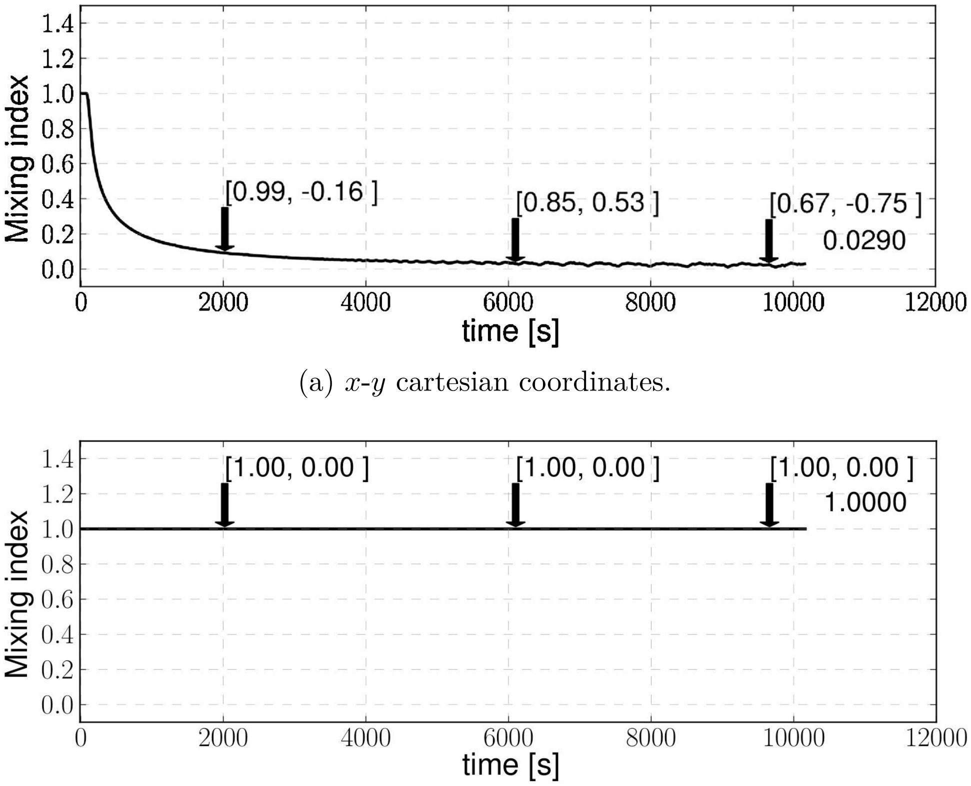


(a) Initial configuration

(b) Configuration after 200 s of mixing

(c) Configuration after 1000 s of mixing

Figure 7.6 – Bi-dimensional mixing of particles in a Lamb-like vortex. Reproduced from Blais *et al.* (2017).



(a) x-y cartesian coordinates.

(b) r-θ cylindrical coordinates.

Figure 7.7 – Mixing index (descending) of particles mixing in lamb-like vortex calculated using the method by Doucet *et al.* (2008) under a) cartesian (x-y) and b) cylindrical (r-θ) systems of coordinates. The values in brackets are the eigenvectors at current times. Reproduced from Blais *et al.* (2017).

the particles in the coordinate system. Additionally, the associated eigenvector $[1.00, 0.00]$ accurately demonstrates that the principal component is r.

The method introduced by Doucet *et al.* (2008) is especially powerful in identifying ill-mixed systems since a decreasing λ_t is a necessary but not sufficient condition for efficient mixing. Blais *et al.* (2017) recommend the use of a system of coordinates that naturally describes the mixing process for a more robust evaluation of mixing. For this reason, we used the cylindrical coordinate system (r-θ-z) to calculate Doucet's mixing index. The results of this work are interpreted as if the Doucet mixing index represented the strong sense of mixing despite the limitations in place.

7.5.2 Inlet flow rate

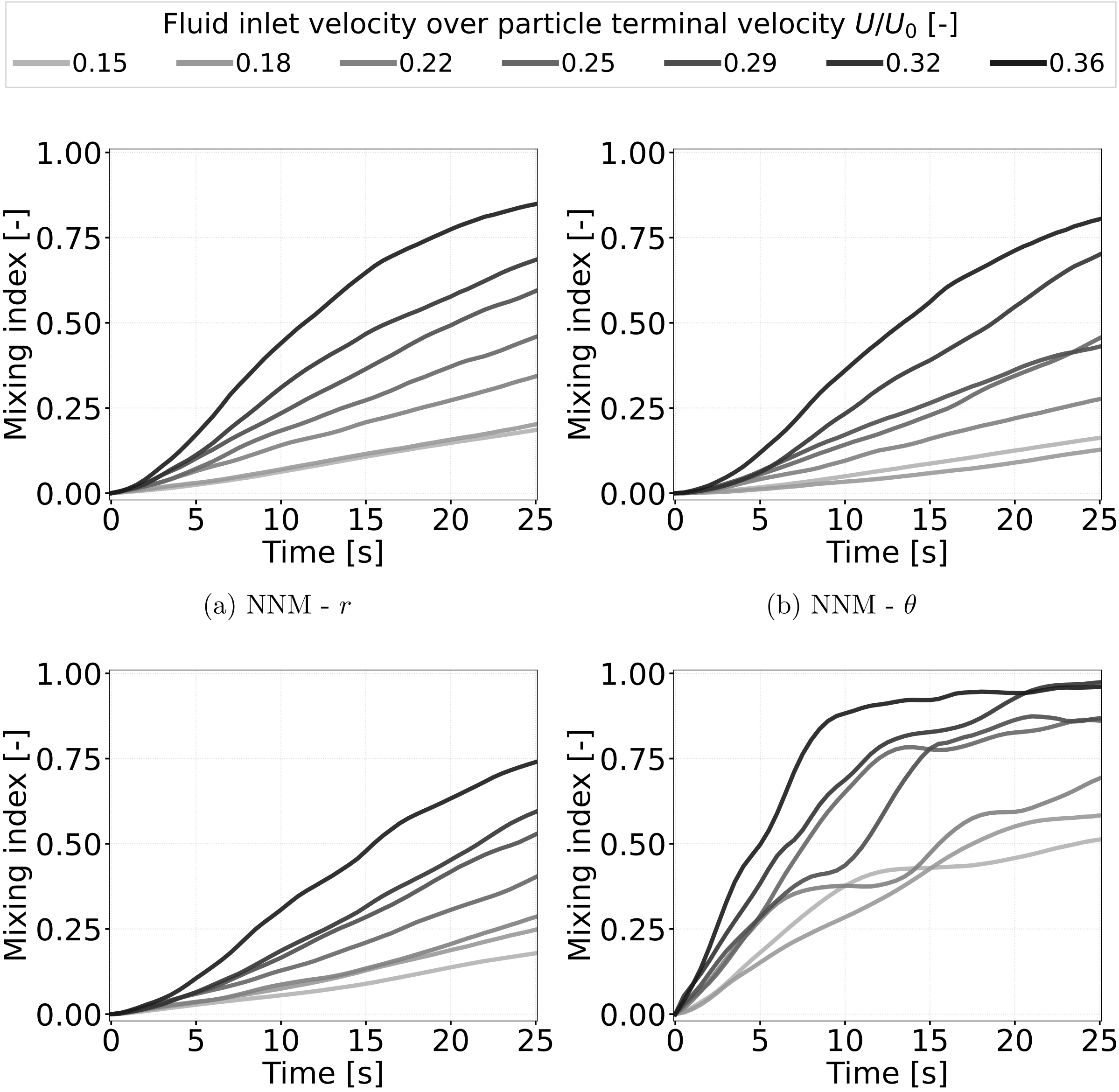
In Figures 7.8 and 7.9, we show the effect of the inlet flow rate on the mixing of particles A and C, respectively.

It is intuitive that the inlet flow rate has a major impact on mixing in a fluidized bed. The higher the inlet flow rate is, the higher the particle-fluid forces are going to be, which increases particle momentum. In addition, increasing the average void fraction $\bar{\varepsilon}_f$ will increase the particles' mean free path. In other words, in theory, the mixing rate is improved because particles move faster with less resistance due to interactions. This behavior is clearly demonstrated in the results in Figures 7.8 and 7.9. In general, cases with U/U_0 closer to 1.00 presented higher mixing speeds and lower mixing times, regardless of the component.

Conversely, observing the evolution of mixing with N_{flows}, this increase seems to reach a *plateau*, as demonstrated in Figure 7.10. In the figure, the number of flows through for the system to reach 90% mixing is shown. It is not clear whether there is a gain in the mixing rate with the increase of U/U_0. Apparently, apart from velocities very close to the minimum fluidization, the effect of the bed expansion on the evolution of mixing with the number of flows through is mild. It is not possible to observe this feature on particle A's data since the mixing barely reaches 0.5 after 25 seconds for most of the inlet conditions. Yet, it is worth mentioning that no significant gains in the N_{flows}^{40} mixing were observed among the highest 4 inlet conditions, regardless of the mixing index adopted. The simulations with particle A at lower inlet flow rates did not reach N_{flows}^{40}.

7.5.3 Principal mixing component

The methods to find the principal (slowest) mixing component are illustrated in Figure 7.11, using particle C's highest inlet flow rate results.



(a) NNM - r

(b) NNM - θ

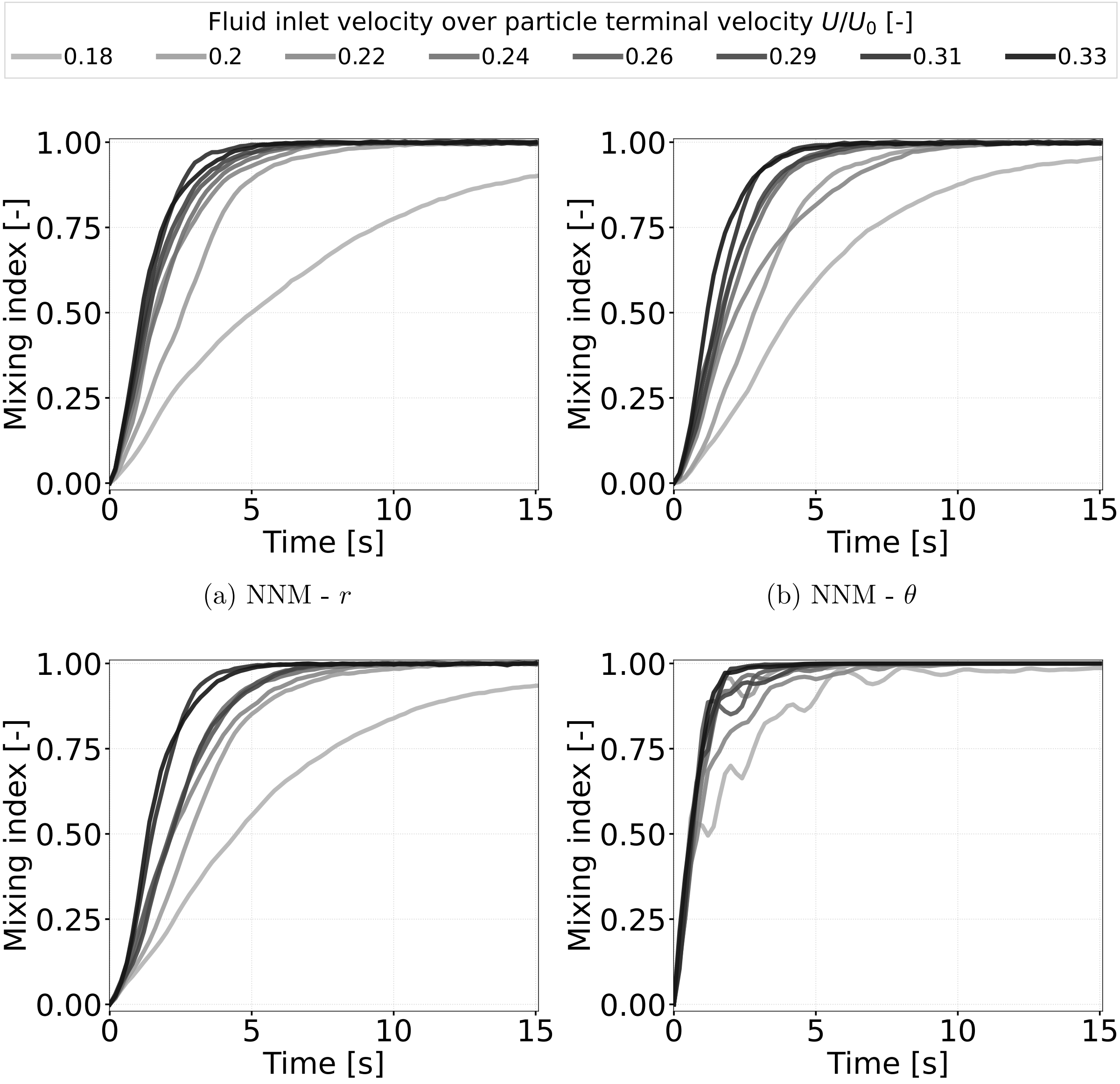
(c) NNM - z

(d) Doucet

Figure 7.8 – Mixing index as a function of time for particle A at several inlet flow rates.

Doucet's mixing index provides a discrete direct answer about the principal component of mixing while the NNM does not. Hence, different criteria were adopted to judge the principal component. For the NNM results, we used the number of flows necessary to reach an index of 0.90 (N_{flows}^{90}). For particle A's results, since N_{flows}^{90} is never reached, we used the maximum mixing index divided by the maximum number of flows through ($\bar{M}_{max}^{NNM}/N_{flows}^{max}$), instead. For the example shown in Figures 7.11a and 7.11b, the principal components are r and z, closely followed by θ.

Oppositely from a stirred tank (BLAIS *et al.*, 2017), apart from the initial bed expansion, the movement of particles in a pseudo-steady state LSFB is far from unidirectional. The resultant force acting over a particle (collisions and interphase momentum exchange) changes direction very frequently. Consequently, the principal mixing component given by



(a) NNM - r

(b) NNM - θ

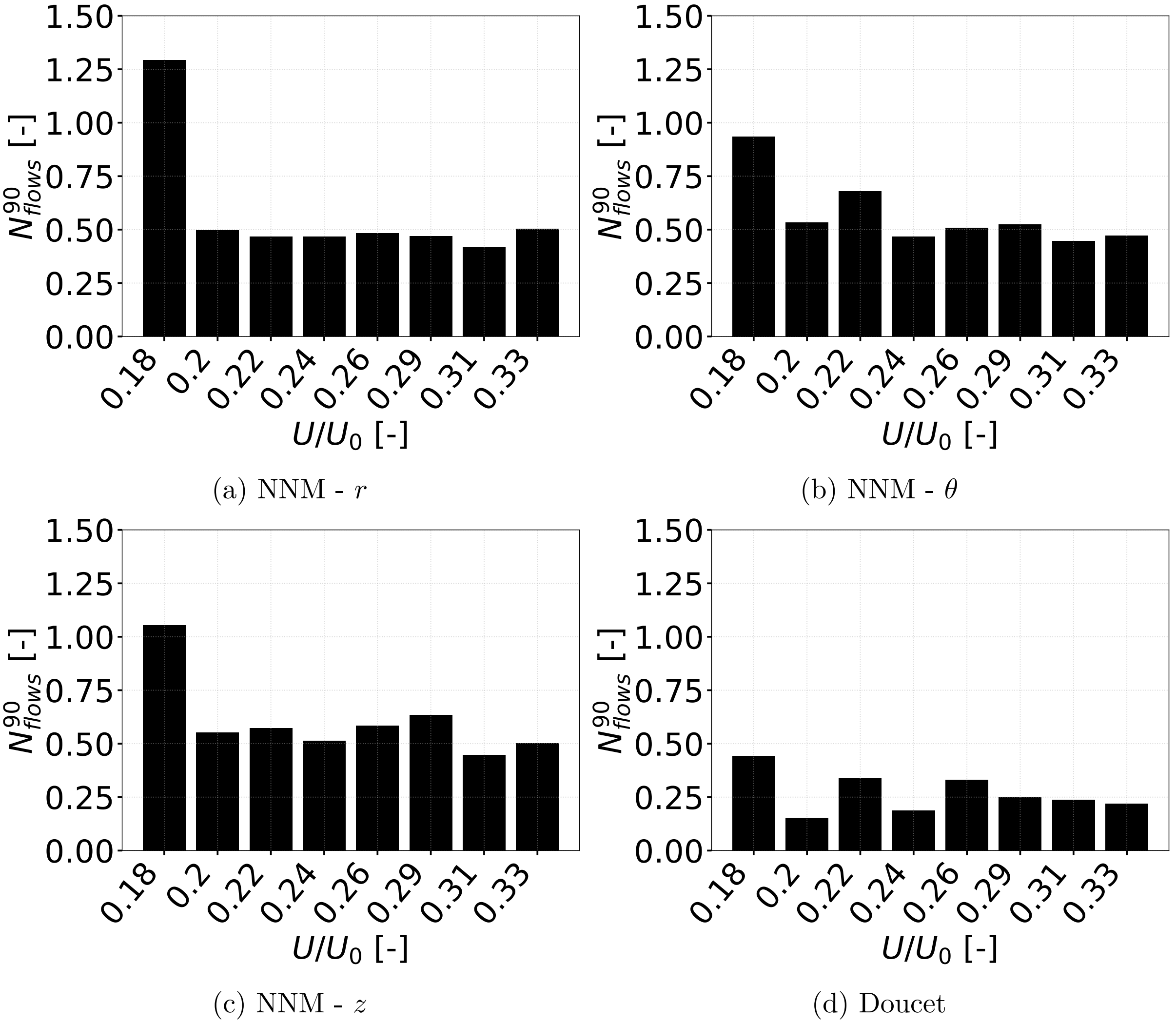
(c) NNM - z

(d) Doucet

Figure 7.9 – Mixing index as a function of time for particle C at several inlet flow rates.

the $\boldsymbol{w}_t$ is not the same throughout a simulation, even at a well-established pseudo-steady state. In Figure 7.11c, we show Doucet's eigenvectors based on particle C's results for the highest inlet flow rate. From the figure, it is possible to observe that the components are not predominantly slower at all times.

To circumvent this, we adopted a frequency-based strategy to find the principal component using Doucet's mixing index. In essence, it consisted of counting the number of timesteps in which each of the components of the $\boldsymbol{w}_t$ had the maximum absolute value. For this analysis, since we are interested in observing the mixing progression before the system gets too close to fully mixed, the simulations that reached 90% mixing were analyzed with the data up to N^{90}_{flows}. In the case of the example (Figure 7.11d), the principal component is z. A summary of the analysis of the principal component is presented in Figure 7.12.



(a) NNM - r

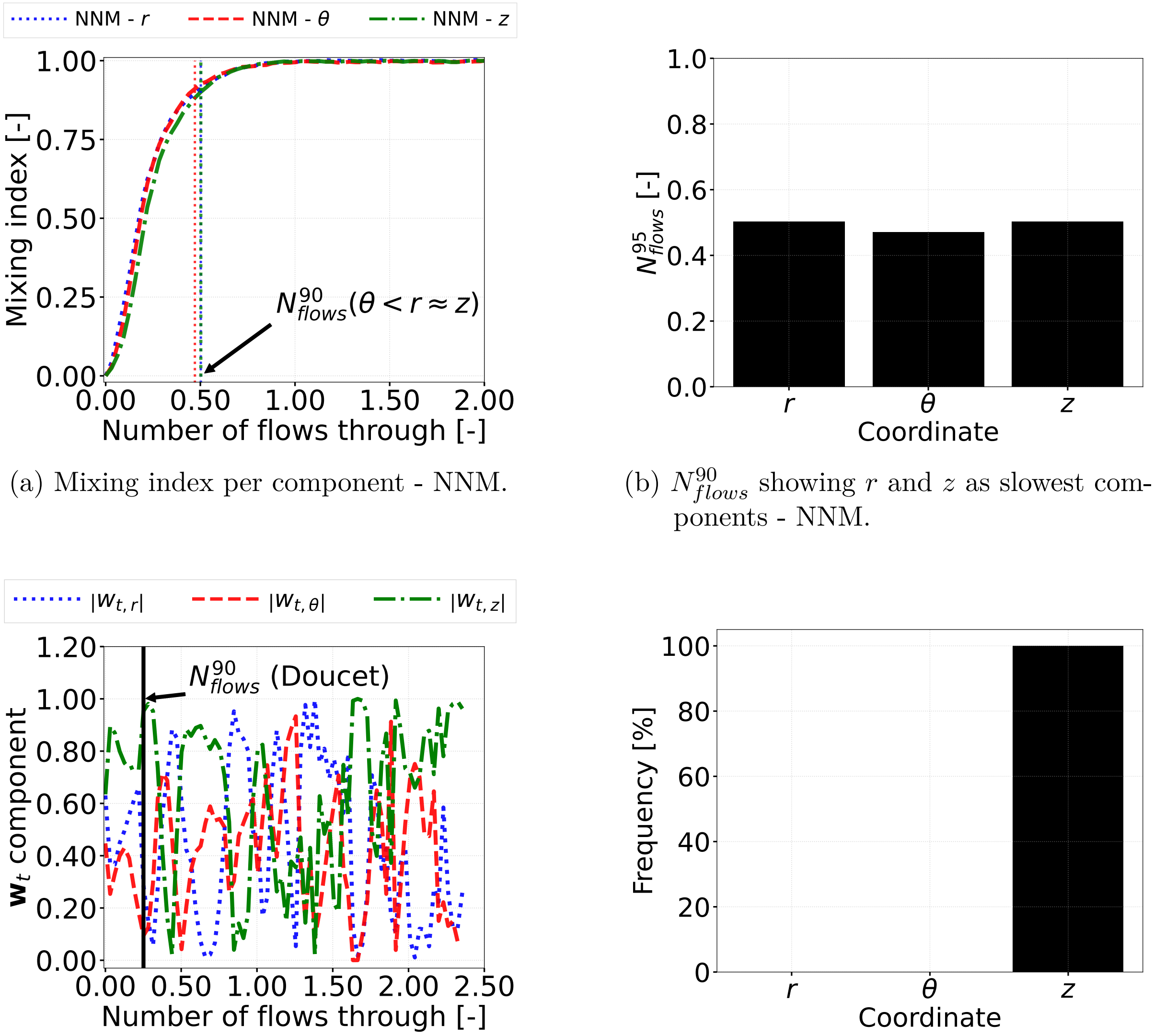
(b) NNM - θ

(c) NNM - z

(d) Doucet

Figure 7.10 – N^{90}_{flows} particle C at several inlet flow rates.

Globally, the principal mixture component was z for the data in figure 7.12. Including the overlapping points, z appears 20 times out of the 30 points (67%) as the slowest component, followed by θ with 6 ($\approx$ 20%), and r with only 5 ($\approx$ 17%). Apparently, the strong counter flux of particles in the vertical direction creates a stronger resistance for the particles' movement compared to the radial and azimuthal components. It is intuitive to think that the collision between round particles is (most probably) not going to result in a force normal to their paths prior to the event. As such, the strong drag and gravity forces in the fluidized bed will facilitate vertically oriented collisions that push particles sideways. Additionally, apart from collisions, no other force counterbalances Saffman lift in the radial direction. As a result, particles are more free to move in r than in z. As for θ, the mixing in this direction can be facilitated by the preferential path through the walls (FERREIRA *et al.*, 2023a). Nonetheless, it is not clear whether these conclusions are reproducible for larger-scale equipment.



(a) Mixing index per component - NNM.

(b) N_{flows}^{90} showing r and z as slowest components - NNM.

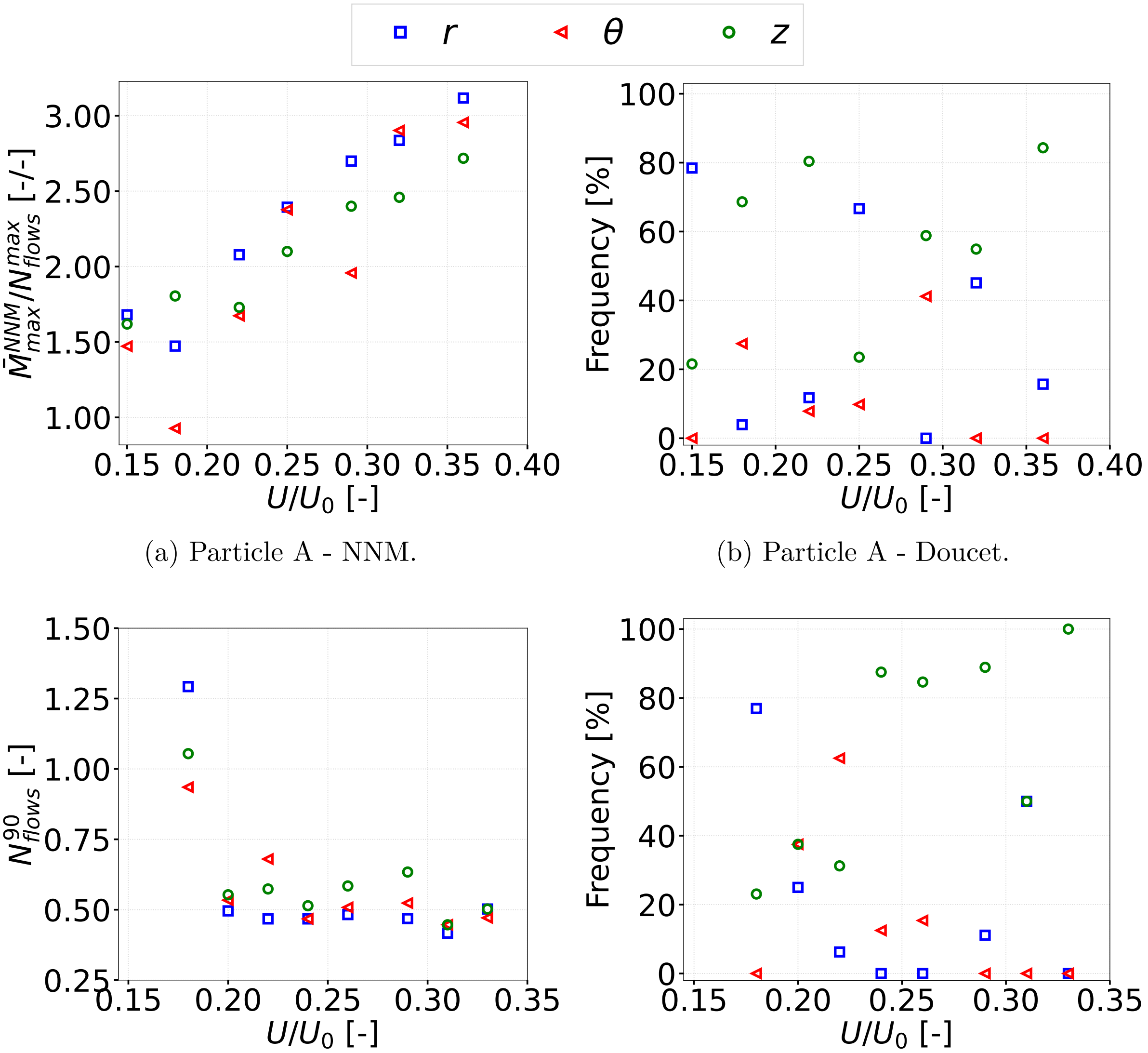
(c) Eigenvector components - Doucet.

(d) Frequency as the principal component before N_{flows}^{90} - Doucet.

Figure 7.11 – Example of principal (slowest) component analysis for with particle C's highest inlet flow rate case.

7.5.4 Regime comparison

Notably, the main difference between simulations with particles A and C is the regime of operation. The free flow Reynolds number for particle A simulations lies in the range of $0.5\times10^3 - 1.3\times10^3$, while simulations with particle C lie between $10\times10^3 - 19\times10^3$. Consequently, particle A simulations result in a maximum number of flows through of around 0.1. This is a very premature stage to evaluate the principal component of mixing using NNM given that the starting points are highly influenced by the mixing index at the moment of splitting. This can be observed in figure 7.12a. From the lowest to the highest inlet flow rate, initially, the components with the highest initial number of nearest neighbors in the opposite group (r and θ) are the ones with the lowest mixing speed ($\bar{M}_{max}^{NNM}/N_{flows}^{max}$), and this gradually changes with the increase on the inlet flow rate (and



(a) Particle A - NNM.

(b) Particle A - Doucet.

(c) Particle C - NNM.

(d) Particle C - Doucet.

Figure 7.12 – Scatter plots of the analysis of the principal mixing component for particles A and C at different inlet flow rates.

consequently N^{max}_{flows}).

For Doucet's method, the discrete response given by the eigenvectors helps to define the component at each individual time step. The frequency analysis presented in figure 7.12b shows that the principal component is z for 5 out of the 7 inlet flow rates ($\approx$ 72%). For very slow mixing processes, earlier responses on the principal mixing component can be obtained with the method by (DOUCET *et al.*, 2008), thus, it may be more recommended.

For particle C, z was the principal component for 11 of the 16 data points ($\approx$ 70%), with a slightly higher consistency among the higher inlet flow rates. The results in Figure 7.12c demonstrate, once more, the consistency between the number of flows through and the NNM mixing in all directions. To the 5 highest inlet flow rates, the NNM average mixing time in terms of N^{90}_{flows} lies between 40 and 70% of a complete flow through the 1 m height equipment. As shown in figure 6.9b, all points are in the region where $\bar{\varepsilon}_f > 0.5$.

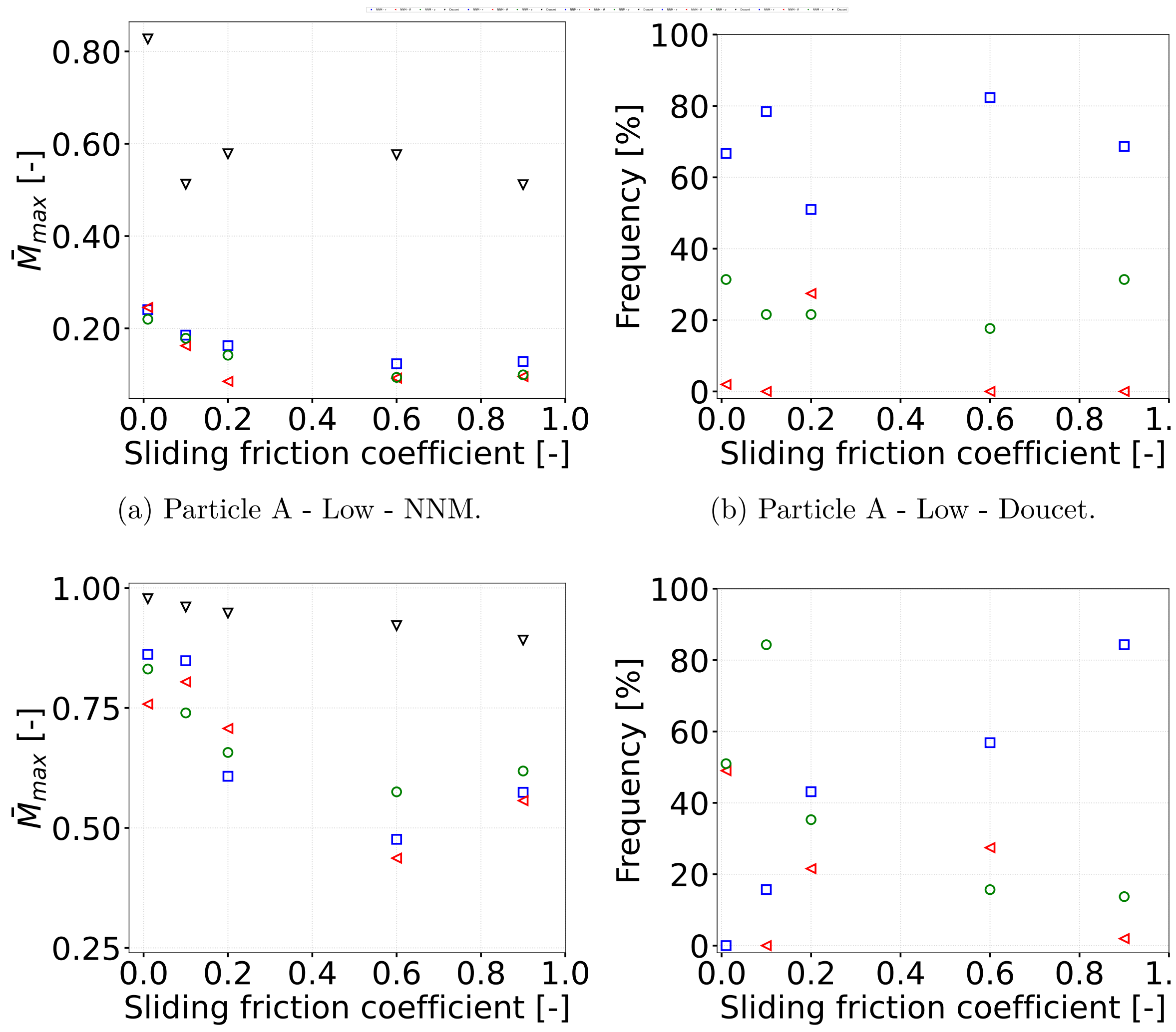
With the data from the supplementary material and considering Doucet's eigenvectors for the standard simulations, for all particles (A, B, and, C), z is the slowest component among the high inlet flow rate and r, the principal for the low one. For a wide range of regimes, using the principal component analysis by (DOUCET *et al.*, 2008), r was the principal component, while more dilute beds presented predominant mixing resistance in z.

7.5.5 Interaction properties

The discussions in this section are based on the data in the Supplementary Material. Overall, changing the particle interaction properties did not imply a significant change in the mixing behavior. We could not observe significant implications when the coefficient of restitution or the coefficient of rolling friction were varied. The highest standard deviation among the mixing dimensionless time for the high inlet flow rate was approximately 9.2% of the average (Doucet's mixing index, N^{95}_{flows}), while $\bar{M}_{max}$ presented a maximum standard deviation of 3.2% (NNM, component z). For the low inlet flow rate, N^{70}_{flows} was achieved by neither of the simulations (except for outliers due to oscillations in Doucet's mixing index). The maximum standard deviation among $\bar{M}_{max}$ was 19% (NNM, component θ). Despite the apparently high standard deviation, the maximum NNM mixing index in this direction was too small to draw any conclusion (0.16). As for particles B and C, the interaction properties were even less influential, especially for the high inlet flow rate. It is an expected result given that fluidized beds are driven by the interphase momentum change, while collisions play a secondary role (GRACE *et al.*, 2020). Yet, one feature can be pointed out.

As shown in Figures 7.13 and 7.14, the results for particle A at the high and low inlet flow rate, and for particle B and C at the low inlet flow rates, there is slight, yet, monotonous decrease in the mixing performance with the increase of the coefficient of sliding friction, prominently in r for most results, and θ for the particle C. For the cases of particles B and C, the concentrated bed with slower particles facilitates longer tangential overlapping. For a very high friction coefficient, the excessive overlapping can cause a "sticking effect", virtually adding an inertial-like effect to the agglomerated particles. In the case of particle A, the very low Stokes number allows for longer overlaps such that this effect can be felt by looser beds.

This result corroborates with previous findings by Blais *et al.* (2017) for a liquid-solid mixer. The authors report that the erosion speed is significantly decreased when lower sliding and rolling friction coefficients are applied, especially at higher impeller velocities.



(a) Particle A - Low - NNM.

(b) Particle A - Low - Doucet.

(c) Particle A - High - NNM.

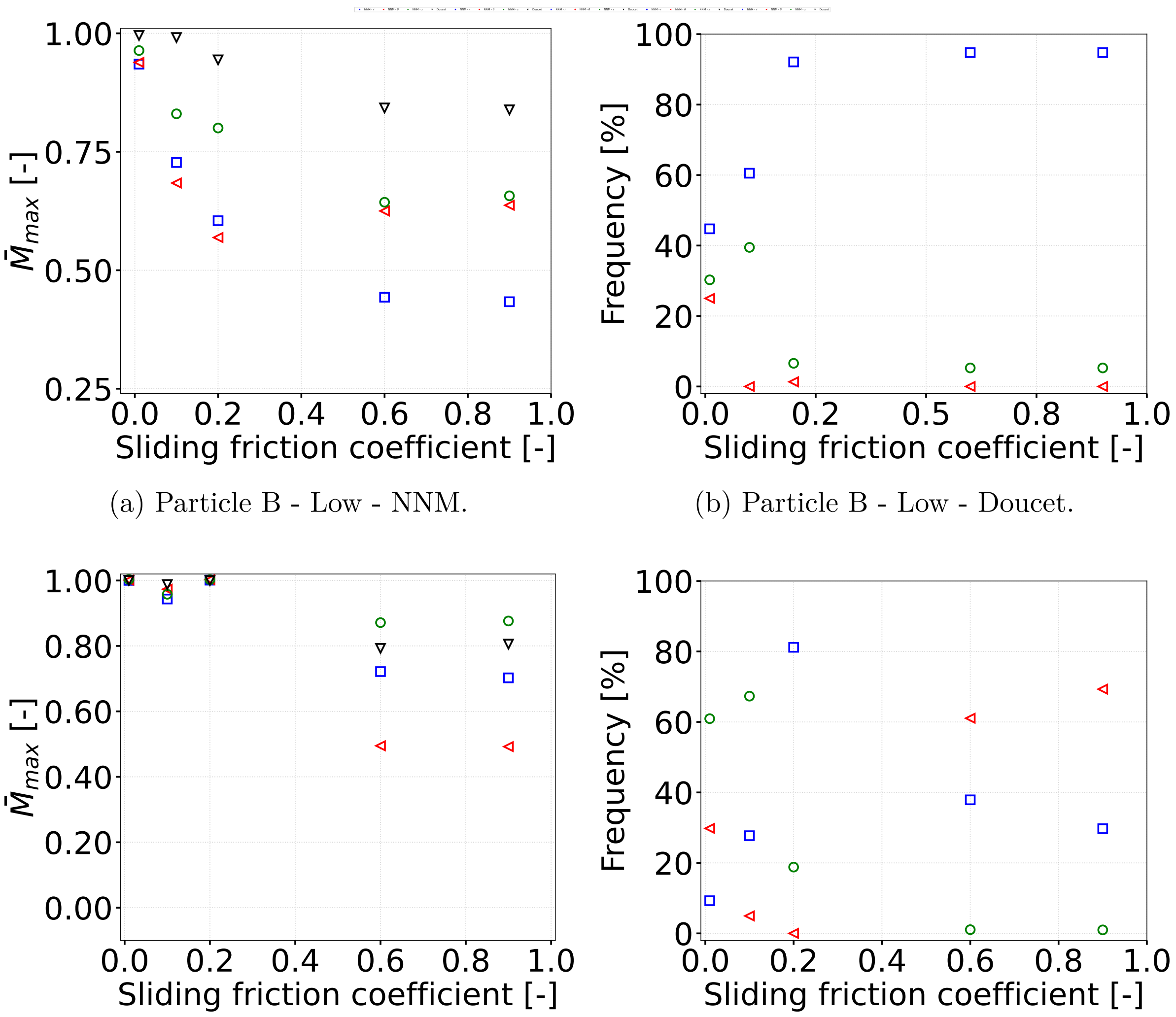
(d) Particle A - High - Doucet.

Figure 7.13 – Scatter plots of the analysis of the principal mixing component for particles A with various sliding friction coefficients.

Still according to the authors, the pseudo-steady state suspension is not affected by the parameters. In the liquid-solid fluidized bed, the particles tend to have lower angular velocities given the multidimensional nature of their dynamics. As such, the rolling friction coefficient plays a less important part in the dynamics. The sliding friction coefficient, oppositely, can affect the bed in a similar fashion as it does for the transitory erosion of particles in the mixing tank.

7.6 Conclusions

In the present work, we extrapolate the CFD-DEM model implemented in Lethe (BLAIS *et al.*, 2020; GOLSHAN *et al.*, 2022; GEITANI *et al.*, 2023a) and validated in a previous work of the research group (FERREIRA *et al.*, 2023a) for the liquid-solid fluidized bed. Simulations with a similar setup as the validation study were used to assess



(a) Particle B - Low - NNM.

(b) Particle B - Low - Doucet.

(c) Particle C - Low - NNM.

(d) Particle C - Low - Doucet.

Figure 7.14 – Scatter plots of the analysis of the principal mixing component for particles B and c with various sliding friction coefficients.

the particle dynamics inside the bed. More specifically, the method by Doucet *et al.* (2008) and the Nearest Neighbors Method (NNM) were applied in the investigation of the bed's mixing performance in the pseudo-steady state. The mixing evolution was expressed in terms of the number of flows through (Eq. (7.25)).

First, we discuss the strengths and weaknesses of each method in the context of liquid fluidization. The global response by the NNM is more coherent with the concept of mixing, given that it is not natural for a dynamic system to reorganize itself. However, the discrete approach by the principal component analysis method by Doucet *et al.* (2008) allows for a faster conclusion of what is the poorest mixing component, even at the very early stages of the mixing. Yet, for a very unsteady system as the fluidized bed, the oscillations on the eigenvectors are very high and the interpretation of the results can be compromised. In this work, we proposed a frequency-based post-processing of the eigenvectors to circumvent this limitation.

Overall, the principal (slowest) mixing component was the axial (z). This behavior is predominant for higher inlet flow rates with looser beds. For the well-developed mixing case of Particle C, this behavior was more easily observed for beds with $\bar{\varepsilon}_f > 0.5$. Additionally, for looser beds, the mixing speed was not significantly enhanced by increasing the inlet flow rate. In fact, the mixing of the bed reached a *plateau* for results expressed in terms of N_{flows}.

Lastly, changing the particles' interaction properties including Young's modulus, coefficient of restitution, coefficient of sliding friction, and coefficient of rolling friction did not have a significant effect on the mixing dynamics compared with the changes in the flow regimes. Nevertheless, the sliding friction coefficient was more influential than the other properties, especially in concentrated beds and beds with slow-moving particles. Our hypothesis is that the more concentrated the bed is, the higher the clustering provoked by the property, which results in a more prominent mixing resistance. Still, this effect vanishes for higher inlet flow rates.

7.7 Acknowledgments

Thanks are due to São Paulo Research Foundation (FAPESP, grant #2019/19173-9, and grant #2020/14567-6); and the Brazilian National Council for Scientific and Technological Development (CNPq, process number 408618/2018-3) for financial support. The authors also thank the Federal University of São Carlos (UFSCar) and Polytechnique Montréal for the infrastructure and all the technical support. This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Finance Code 001.

We 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 the technical support and computing time provided by Compute Canada and Calcul Québec.

7.8 Supplementary material

	$N_{flows}^{\%}$	**Standard**	**Coefficient of restitution**			**Coefficient of rolling friction**				**Coefficient of sliding friction**				**Young's modulus**	
			0.1	**0.3**	**0.6**	**0.01**	**0.1**	**0.6**	**0.9**	**0.01**	**0.2**	**0.6**	**0.9**	**1e5**	1e9
NNM - r	95%	—	—	—	—	—	—	—	—	—	—	—	—	—	—
	90%	—	—	—	—	—	—	—	—	—	—	—	—	—	—
	80%	0.234	0.261	0.250	0.256	—	0.229	0.256	0.261	0.234	—	—	—	—	—
	70%	0.185	0.207	0.207	0.201	0.229	0.185	0.207	0.223	0.196	—	—	—	—	—
	$\bar{M}_{max}$	0.848	0.822	0.831	0.835	0.767	0.868	0.833	0.818	0.862	0.608	0.476	0.574	0.548	—
	$\bar{M}_{max}/N_{flows}^{max}$	3.117	3.019	3.054	3.066	2.820	3.188	3.059	3.006	3.167	2.233	1.750	2.110	2.013	—
NNM - θ	95%	—	—	—	—	—	—	—	—	—	—	—	—	—	—
	90%	—	—	—	—	—	—	—	—	—	—	—	—	—	—
	80%	0.272	—	—	—	—	0.250	0.261	—	—	—	—	—	—	—
	70%	0.218	0.223	0.234	0.223	0.256	0.207	0.212	0.245	0.256	0.272	—	—	—	—
	$\bar{M}_{max}$	0.804	0.793	0.777	0.791	0.735	0.838	0.831	0.755	0.758	0.707	0.437	0.557	0.520	—
	$\bar{M}_{max}/N_{flows}^{max}$	2.955	2.915	2.855	2.906	2.700	3.078	3.054	2.773	2.785	2.599	1.606	2.046	1.909	—
NNM - z	95%	—	—	—	—	—	—	—	—	—	—	—	—	—	—
	90%	—	—	—	—	—	—	—	—	—	—	—	—	—	—
	80%	—	—	—	—	—	—	—	—	0.256	—	—	—	—	—
	70%	0.250	0.245	0.256	0.272	0.261	0.245	0.261	—	0.218	—	—	—	—	—
	$\bar{M}_{max}$	0.740	0.770	0.748	0.704	0.732	0.765	0.728	0.682	0.831	0.657	0.575	0.619	0.674	—
	$\bar{M}_{max}/N_{flows}^{max}$	2.718	2.829	2.749	2.586	2.688	2.810	2.677	2.504	3.054	2.415	2.113	2.273	2.477	—
Doucet	95%	0.239	—	0.191	0.212	—	0.212	0.191	0.223	0.234	—	—	—	0.239	—
	90%	0.125	0.125	0.109	0.131	0.212	0.114	0.174	0.191	0.180	0.158	0.239	—	0.163	—
	80%	0.087	0.082	0.082	0.082	0.147	0.093	0.136	0.136	0.109	0.136	0.191	0.163	0.142	—
	70%	0.076	0.071	0.071	0.065	0.054	0.076	0.098	0.076	0.076	0.120	0.152	0.136	0.131	—
	$\bar{M}_{max}$	0.960	0.934	0.983	0.984	0.946	0.977	0.981	0.966	0.978	0.948	0.922	0.891	0.968	—
	$\bar{M}_{max}/N_{flows}^{max}$	3.529	3.431	3.612	3.617	3.696	3.591	3.606	3.551	3.592	3.482	3.762	3.275	3.630	—
Freq. %	$\lvert w_{t,r}\rvert$	21.739	8.696	10.000	0.000	61.538	0.000	0.000	0.000	0.000	72.414	65.909	84.314	3.333	—
	$\lvert w_{t,\theta}\rvert$	0.000	8.696	15.000	0.000	0.000	0.000	0.000	0.000	54.545	0.000	15.909	1.961	33.333	—
	$\lvert w_{t,z}\rvert$	78.261	82.609	75.000	100.000	38.462	100.000	100.000	100.000	45.455	27.586	18.182	13.725	63.333	—

Table 7.5 – Particle A - High inlet velocity.

	$N^{\%}_{flows}$	**Standard**	**Coefficient of restitution**			**Coefficient of rolling friction**				**Coefficient of sliding friction**				**Young's modulus**	
			0.1	**0.3**	**0.6**	**0.01**	**0.1**	**0.6**	**0.9**	**0.01**	**0.2**	**0.6**	**0.9**	**1e5**	1e9
NNM - r	95%	—	—	—	—	—	—	—	—	—	—	—	—	—	—
	90%	—	—	—	—	—	—	—	—	—	—	—	—	—	—
	80%	—	—	—	—	—	—	—	—	—	—	—	—	—	—
	70%	—	—	—	—	—	—	—	—	—	—	—	—	—	—
	$\bar{M}_{max}$	0.185	0.173	0.184	0.187	0.199	0.187	0.180	0.179	0.241	0.163	0.123	0.128	0.145	—
	$\bar{M}_{max}/N^{max}_{flows}$	1.680	1.571	1.670	1.698	1.801	1.697	1.635	1.688	2.181	1.473	1.118	1.163	1.318	—
NNM - θ	95%	—	—	—	—	—	—	—	—	—	—	—	—	—	—
	90%	—	—	—	—	—	—	—	—	—	—	—	—	—	—
	80%	—	—	—	—	—	—	—	—	—	—	—	—	—	—
	70%	—	—	—	—	—	—	—	—	—	—	—	—	—	—
	$\bar{M}_{max}$	0.162	0.096	0.116	0.125	0.182	0.145	0.138	0.151	0.245	0.085	0.092	0.096	0.050	—
	$\bar{M}_{max}/N^{max}_{flows}$	1.472	0.866	1.053	1.135	1.648	1.310	1.254	1.421	2.224	0.772	0.836	0.867	0.451	—
NNM - z	95%	—	—	—	—	—	—	—	—	—	—	—	—	—	—
	90%	—	—	—	—	—	—	—	—	—	—	—	—	—	—
	80%	—	—	—	—	—	—	—	—	—	—	—	—	—	—
	70%	—	—	—	—	—	—	—	—	—	—	—	—	—	—
	$\bar{M}_{max}$	0.179	0.167	0.167	0.170	0.208	0.178	0.174	0.170	0.220	0.142	0.094	0.099	0.171	—
	$\bar{M}_{max}/N^{max}_{flows}$	1.619	1.516	1.512	1.537	1.889	1.615	1.580	1.607	1.992	1.286	0.851	0.900	1.549	—
Doucet	95%	—	—	—	—	—	—	—	—	—	—	—	—	—	—
	90%	—	—	—	—	—	—	—	—	—	—	—	—	—	—
	80%	—	—	—	—	—	—	—	—	0.090	—	—	—	—	—
	70%	—	—	—	—	—	—	—	—	0.068	—	—	—	—	—
	$\bar{M}_{max}$	0.512	0.578	0.531	0.560	0.638	0.533	0.559	0.556	0.827	0.578	0.576	0.511	0.460	—
	$\bar{M}_{max}/N^{max}_{flows}$	4.643	9.351	4.815	5.071	5.777	4.830	5.066	5.251	7.807	5.241	5.222	4.634	4.165	—
Freq. %	$\lvert w_{t,r}\rvert$	78.431	50.980	84.314	84.314	84.314	82.353	82.353	79.592	66.667	50.980	82.353	68.627	3.922	—
	$\lvert w_{t,\theta}\rvert$	0.000	29.412	0.000	0.000	0.000	0.000	0.000	0.000	1.961	27.451	0.000	0.000	56.863	—
	$\lvert w_{t,z}\rvert$	21.569	19.608	15.686	15.686	15.686	17.647	17.647	20.408	31.373	21.569	17.647	31.373	39.216	—

Table 7.6 – Particle A - Low inlet velocity.

	$N^{\%}_{flows}$	**Standard**	**Coefficient of restitution**			**Coefficient of rolling friction**				**Coefficient of sliding friction**				**Young's modulus**	
			0.1	**0.3**	**0.6**	**0.01**	**0.1**	**0.6**	**0.9**	**0.01**	**0.2**	**0.6**	**0.9**	**1e5**	1e9
NNM - r	95%	0.726	0.682	0.682	0.594	0.704	0.660	0.748	0.660	0.726	0.660	0.616	0.836	0.660	—
	90%	0.550	0.550	0.550	0.484	0.550	0.528	0.616	0.550	0.594	0.528	0.484	0.660	0.572	—
	80%	0.418	0.418	0.418	0.374	0.418	0.418	0.484	0.418	0.440	0.396	0.374	0.484	0.440	—
	70%	0.330	0.352	0.330	0.308	0.330	0.330	0.374	0.330	0.352	0.330	0.308	0.396	0.374	—
	$\bar{M}_{max}$	0.993	1.000	0.997	1.000	1.000	1.000	0.997	1.000	1.000	1.000	1.000	0.994	0.996	—
	$\bar{M}_{max}/N^{max}_{flows}$	0.602	0.623	0.731	0.631	0.688	0.689	0.604	0.689	0.606	0.606	0.640	0.619	0.686	—
NNM - θ	95%	0.968	0.770	0.858	0.748	0.704	0.704	0.946	0.902	0.748	0.814	0.814	0.902	0.748	—
	90%	0.770	0.638	0.682	0.638	0.594	0.550	0.704	0.682	0.594	0.660	0.660	0.748	0.616	—
	80%	0.594	0.484	0.506	0.484	0.462	0.440	0.528	0.506	0.462	0.506	0.506	0.572	0.506	—
	70%	0.484	0.418	0.396	0.374	0.374	0.352	0.418	0.418	0.374	0.418	0.418	0.462	0.418	—
	$\bar{M}_{max}$	0.998	0.999	0.997	0.999	1.000	0.999	0.995	1.000	1.000	1.000	0.999	0.998	1.000	—
	$\bar{M}_{max}/N^{max}_{flows}$	0.622	0.605	0.604	0.606	0.710	0.614	0.603	0.649	0.614	0.622	0.640	0.639	0.699	—
NNM - z	95%	1.298	1.386	1.056	0.792	0.880	1.056	1.474	1.166	1.034	0.968	0.968	1.364	1.276	—
	90%	0.990	1.122	0.924	0.682	0.748	0.814	1.210	0.924	0.858	0.814	0.792	1.122	1.078	—
	80%	0.726	0.858	0.748	0.550	0.594	0.638	0.924	0.660	0.704	0.616	0.594	0.880	0.836	—
	70%	0.616	0.726	0.616	0.462	0.484	0.528	0.770	0.528	0.616	0.506	0.506	0.748	0.682	—
	$\bar{M}_{max}$	0.983	0.983	0.997	1.000	1.000	0.990	0.964	0.994	0.999	1.000	0.997	0.990	0.990	—
	$\bar{M}_{max}/N^{max}_{flows}$	0.596	0.596	0.657	0.770	0.649	0.616	0.592	0.611	0.614	0.622	0.604	0.600	0.600	—
Doucet	95%	1.122	1.056	0.814	0.308	0.396	0.770	1.144	1.056	0.462	0.528	0.528	1.034	0.814	—
	90%	0.616	0.770	0.638	0.264	0.330	0.528	0.924	0.770	0.440	0.440	0.374	0.858	0.726	—
	80%	0.462	0.528	0.396	0.198	0.286	0.352	0.550	0.418	0.396	0.286	0.286	0.528	0.550	—
	70%	0.374	0.440	0.286	0.176	0.242	0.286	0.374	0.220	0.352	0.242	0.220	0.396	0.396	—
	$\bar{M}_{max}$	0.991	0.993	0.999	1.000	1.000	0.998	0.986	0.998	1.000	1.000	0.999	0.999	0.997	—
	$\bar{M}_{max}/N^{max}_{flows}$	0.601	0.602	0.606	0.640	0.710	0.605	0.623	0.605	0.606	0.623	0.605	0.622	0.604	—
Freq. %	$\lvert w_{t,r}\rvert$	0.000	0.000	0.000	0.000	0.000	0.000	2.381	0.000	0.000	0.000	5.882	0.000	0.000	—
	$\lvert w_{t,\theta}\rvert$	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	5.000	0.000	0.000	0.000	3.030	—
	$\lvert w_{t,z}\rvert$	100.000	100.000	100.000	100.000	100.000	100.000	97.619	100.000	95.000	100.000	94.118	100.000	96.970	—

Table 7.7 – Particle B - High inlet velocity.

	$N^{\%}_{flows}$	**Standard**	**Coefficient of restitution**			**Coefficient of rolling friction**				**Coefficient of sliding friction**				**Young's modulus**	
			0.1	**0.3**	**0.6**	**0.01**	**0.1**	**0.6**	**0.9**	**0.01**	**0.2**	**0.6**	**0.9**	**1e5**	1e9
NNM - r	95%	—	—	—	—	—	0.611	—	—	—	—	—	—	—	—
	90%	—	—	—	—	0.492	0.586	—	—	0.560	—	—	—	—	—
	80%	—	—	—	—	0.357	0.577	—	—	0.433	—	—	—	0.603	—
	70%	0.603	—	—	0.586	0.306	0.509	—	0.603	0.357	—	—	—	0.475	—
	$\bar{M}_{max}$	0.727	0.665	0.688	0.741	0.941	0.976	0.675	0.726	0.935	0.605	0.443	0.434	0.830	—
	$\bar{M}_{max}/N^{max}_{flows}$	1.143	1.045	1.081	1.164	1.479	1.534	1.060	1.140	1.469	0.950	0.696	0.681	1.303	—
NNM - θ	95%	—	—	—	—	—	—	—	—	—	—	—	—	—	—
	90%	—	—	—	—	0.509	0.620	—	—	0.552	—	—	—	—	—
	80%	—	—	—	—	0.382	0.577	—	—	0.441	—	—	—	—	—
	70%	—	—	—	0.577	0.340	0.560	—	0.577	0.373	—	—	—	0.560	—
	$\bar{M}_{max}$	0.684	0.644	0.602	0.736	0.944	0.936	0.633	0.745	0.939	0.569	0.625	0.637	0.762	—
	$\bar{M}_{max}/N^{max}_{flows}$	1.075	1.011	0.946	1.156	1.483	1.471	0.995	1.170	1.475	0.894	0.982	1.001	1.196	—
NNM - z	95%	—	—	—	—	—	0.611	—	—	0.603	—	—	—	—	—
	90%	—	—	—	—	—	0.586	—	—	0.509	—	—	—	—	—
	80%	0.594	0.594	0.594	0.594	0.518	0.569	0.577	0.628	0.416	0.637	—	—	0.594	—
	70%	0.484	0.475	0.467	0.475	0.416	0.484	0.467	0.492	0.357	0.560	—	—	0.467	—
	$\bar{M}_{max}$	0.830	0.826	0.833	0.821	0.864	0.972	0.843	0.808	0.964	0.800	0.644	0.657	0.831	—
	$\bar{M}_{max}/N^{max}_{flows}$	1.304	1.298	1.309	1.289	1.357	1.527	1.324	1.269	1.514	1.257	1.011	1.032	1.306	—
Doucet	95%	0.450	0.467	0.475	0.475	0.212	0.577	0.441	0.450	0.289	—	—	—	0.501	—
	90%	0.365	0.424	0.424	0.357	0.195	0.314	0.399	0.340	0.246	0.552	—	—	0.280	—
	80%	0.280	0.289	0.306	0.272	0.153	0.255	0.297	0.263	0.161	0.433	0.603	0.586	0.212	—
	70%	0.136	0.229	0.238	0.212	0.102	0.119	0.229	0.127	0.127	0.331	0.450	0.475	0.170	—
	$\bar{M}_{max}$	0.991	0.979	0.981	0.981	0.997	0.989	0.976	0.989	0.996	0.944	0.843	0.839	0.968	—
	$\bar{M}_{max}/N^{max}_{flows}$	1.743	1.537	1.778	1.628	1.836	1.554	1.576	1.792	1.564	1.483	1.324	1.317	1.521	—
Freq. %	$\lvert w_{t,r}\rvert$	76.744	94.000	92.000	80.952	4.348	51.351	91.489	65.000	41.379	96.923	94.737	94.737	24.242	—
	$\lvert w_{t,\theta}\rvert$	0.000	0.000	0.000	0.000	0.000	21.622	0.000	0.000	41.379	1.538	0.000	0.000	45.455	—
	$\lvert w_{t,z}\rvert$	23.256	6.000	8.000	19.048	95.652	27.027	8.511	35.000	17.241	1.538	5.263	5.263	30.303	—

Table 7.8 – Particle B - Low inlet velocity.

	$N_{flows}^{\%}$	**Standard**	**Coefficient of restitution**			**Coefficient of rolling friction**				**Coefficient of sliding friction**				**Young's modulus**	
			0.1	**0.3**	**0.6**	**0.01**	**0.1**	**0.6**	**0.9**	**0.01**	**0.2**	**0.6**	**0.9**	**1e5**	1e9
NNM - r	95%	0.628	0.534	0.565	0.565	0.534	0.597	0.628	0.534	0.565	0.534	0.785	0.597	0.597	0.534
	90%	0.503	0.440	0.471	0.503	0.471	0.503	0.503	0.408	0.471	0.440	0.628	0.471	0.503	0.408
	80%	0.345	0.377	0.377	0.408	0.377	0.377	0.408	0.314	0.345	0.345	0.471	0.377	0.377	0.345
	70%	0.283	0.314	0.314	0.314	0.314	0.314	0.345	0.251	0.283	0.283	0.377	0.314	0.314	0.283
	$\bar{M}_{max}$	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
	$\bar{M}_{max}/N_{flows}^{max}$	0.468	0.601	0.816	0.482	0.430	0.777	0.442	0.612	0.637	0.692	0.522	0.624	0.650	0.692
NNM - θ	95%	0.597	0.503	0.503	0.565	0.565	0.628	0.660	0.597	0.503	0.534	0.911	0.628	0.534	0.628
	90%	0.471	0.440	0.440	0.503	0.471	0.534	0.565	0.503	0.408	0.440	0.754	0.503	0.440	0.534
	80%	0.345	0.345	0.345	0.377	0.377	0.440	0.471	0.377	0.314	0.314	0.597	0.377	0.345	0.408
	70%	0.283	0.283	0.283	0.314	0.345	0.377	0.408	0.283	0.251	0.283	0.503	0.314	0.283	0.345
	$\bar{M}_{max}$	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
	$\bar{M}_{max}/N_{flows}^{max}$	0.498	0.514	0.430	0.724	0.579	0.724	0.468	0.475	0.816	0.579	0.461	0.498	0.590	0.861
NNM - z	95%	0.628	0.565	0.565	0.597	0.628	0.660	0.628	0.722	0.597	0.628	0.848	0.660	0.565	0.628
	90%	0.503	0.471	0.471	0.534	0.534	0.534	0.534	0.597	0.503	0.503	0.691	0.565	0.471	0.534
	80%	0.377	0.408	0.408	0.440	0.440	0.440	0.440	0.471	0.408	0.440	0.534	0.440	0.377	0.440
	70%	0.314	0.345	0.345	0.377	0.377	0.345	0.377	0.377	0.345	0.377	0.440	0.377	0.314	0.377
	$\bar{M}_{max}$	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
	$\bar{M}_{max}/N_{flows}^{max}$	0.425	0.663	0.677	0.758	0.482	0.601	0.724	0.677	0.796	0.637	0.590	0.540	0.430	0.777
Doucet	95%	0.283	0.251	0.314	0.377	0.377	0.471	0.440	0.408	0.251	0.283	0.534	0.377	0.283	0.345
	90%	0.220	0.220	0.283	0.220	0.283	0.314	0.377	0.251	0.220	0.251	0.440	0.314	0.251	0.283
	80%	0.188	0.188	0.220	0.157	0.220	0.188	0.188	0.188	0.188	0.220	0.345	0.220	0.188	0.220
	70%	0.157	0.157	0.188	0.126	0.220	0.157	0.157	0.157	0.157	0.188	0.188	0.188	0.188	0.188
	$\bar{M}_{max}$	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
	$\bar{M}_{max}/N_{flows}^{max}$	0.692	0.740	0.624	0.531	0.677	0.590	0.637	0.590	0.442	0.475	0.514	0.663	0.448	0.637
Freq. %	$\lvert w_{t,r}\rvert$	0.000	14.286	33.333	28.571	0.000	0.000	0.000	0.000	14.286	0.000	0.000	0.000	25.000	0.000
	$\lvert w_{t,\theta}\rvert$	0.000	0.000	0.000	14.286	11.111	0.000	25.000	0.000	0.000	0.000	28.571	0.000	0.000	55.556
	$\lvert w_{t,z}\rvert$	100.000	85.714	66.667	57.143	88.889	100.000	75.000	100.000	85.714	100.000	71.429	100.000	75.000	44.444

Table 7.9 – Particle C - High inlet velocity.

	$N^{\%}_{flows}$	**Standard**	**Coefficient of restitution**			**Coefficient of rolling friction**				**Coefficient of sliding friction**				**Young's modulus**	
			0.1	**0.3**	**0.6**	**0.01**	**0.1**	**0.6**	**0.9**	**0.01**	**0.2**	**0.6**	**0.9**	**1e5**	1e9
NNM - r	95%	—	1.292	—	—	1.071	1.343	—	1.632	0.646	0.969	—	—	1.173	1.530
	90%	1.292	1.258	1.360	1.275	0.833	1.020	1.360	1.241	0.527	0.935	—	—	0.850	1.224
	80%	0.918	0.969	0.986	0.935	0.595	0.748	0.969	0.901	0.391	0.918	—	—	0.629	0.918
	70%	0.714	0.748	0.782	0.765	0.493	0.629	0.782	0.714	0.306	0.867	1.513	1.700	0.544	0.731
	$\bar{M}_{max}$	0.943	0.998	0.934	0.945	0.982	0.976	0.943	0.956	1.000	1.000	0.722	0.702	0.984	1.000
	$\bar{M}_{max}/N^{max}_{flows}$	0.566	0.631	0.549	0.556	0.590	0.574	0.555	0.562	0.520	0.668	0.451	0.413	0.597	0.429
NNM - θ	95%	1.258	1.190	1.683	1.326	1.173	1.598	1.122	1.343	0.731	0.935	—	—	1.122	1.258
	90%	0.935	0.935	1.156	0.969	0.901	1.292	0.901	1.054	0.595	0.901	—	—	0.884	0.986
	80%	0.697	0.697	0.799	0.714	0.697	0.799	0.697	0.765	0.442	0.714	—	—	0.629	0.731
	70%	0.544	0.561	0.612	0.578	0.544	0.595	0.561	0.595	0.340	0.578	—	—	0.527	0.578
	$\bar{M}_{max}$	0.973	1.000	0.950	0.968	0.978	0.961	0.975	0.971	1.000	1.000	0.495	0.492	0.989	1.000
	$\bar{M}_{max}/N^{max}_{flows}$	0.572	0.653	0.564	0.575	0.587	0.565	0.573	0.571	0.442	0.653	0.310	0.290	0.582	0.463
NNM - z	95%	1.428	1.292	1.615	1.700	1.394	1.496	1.632	1.700	0.782	0.952	—	—	1.241	1.632
	90%	1.054	1.156	1.088	1.122	1.105	1.071	1.054	1.224	0.663	0.935	—	—	0.969	1.241
	80%	0.765	0.799	0.782	0.799	0.748	0.748	0.731	0.816	0.544	0.833	1.054	1.037	0.680	0.816
	70%	0.595	0.612	0.629	0.612	0.544	0.595	0.578	0.612	0.425	0.629	0.765	0.765	0.544	0.612
	$\bar{M}_{max}$	0.958	1.000	0.956	0.951	0.974	0.967	0.952	0.951	1.000	1.000	0.871	0.876	0.984	1.000
	$\bar{M}_{max}/N^{max}_{flows}$	0.569	0.588	0.568	0.559	0.579	0.569	0.560	0.559	0.507	0.700	0.545	0.515	0.590	0.470
Doucet	95%	0.459	0.629	0.629	0.629	0.323	0.459	0.510	0.476	0.357	0.833	—	—	0.442	0.476
	90%	0.442	0.442	0.425	0.442	0.289	0.340	0.459	0.340	0.170	0.595	—	—	0.323	0.442
	80%	0.272	0.272	0.323	0.323	0.170	0.255	0.340	0.255	0.136	0.476	—	1.632	0.238	0.255
	70%	0.238	0.238	0.255	0.272	0.136	0.221	0.255	0.153	0.102	0.374	0.850	1.088	0.221	0.170
	$\bar{M}_{max}$	0.988	1.000	0.987	0.986	0.999	0.997	0.987	0.994	1.000	1.000	0.793	0.806	0.999	1.000
	$\bar{M}_{max}/N^{max}_{flows}$	1.452	0.588	0.611	1.137	0.587	0.604	0.611	0.596	0.452	0.606	0.496	0.483	0.816	0.414
Freq. %	$\lvert w_{t,r}\rvert$	76.923	84.615	92.000	92.308	47.059	65.000	88.889	70.000	0.000	94.286	37.895	29.703	63.158	69.231
	$\lvert w_{t,\theta}\rvert$	0.000	3.846	4.000	3.846	0.000	0.000	0.000	0.000	0.000	0.000	61.053	69.307	10.526	3.846
	$\lvert w_{t,z}\rvert$	23.077	11.538	4.000	3.846	52.941	35.000	11.111	30.000	100.000	5.714	1.053	0.990	26.316	26.923

Table 7.10 – Particle C - Low inlet velocity.

CHAPTER 8

GENERAL DISCUSSION

As pointed out, this research project's main objective was to come up with a robust methodology to predict the internal dynamics of liquid-solid fluidized beds. The results obtained in the experimental and computational sights were pursued with this objective as a guide. In order to achieve this broader (yet complex) goal, specific objectives were drawn and addressed. In this section, each of these objectives is confronted with the findings of the entire research.

8.1 Objective 1: Experimental setup

Judging the precision of the prediction methods requires reliable data. Having an experimental setup was key to obtaining it. At the beginning of the project, the experimental setup was composed of the fluidization column (including the calming section, distributor, and outlet portion), the metal panel, the reservoir, tubes, valves, connections, and a pump. Practical adaptations to the original setup needed to be done to work properly. This includes changing the original pump to one with a higher flow rate capacity, drilling holes in the fluidization column to add the pressure plugs, drilling holes in the panel to fix the microcontrollers and sensors, changing the direction of the outlet water to prevent air bubbles, and others. The current setup is fully capable of fluidizing particles with very low and very high Archimedes numbers.

The experimental measurements were done using the sensors presented in Section 4.1.3. Robust commercial pressure gauges and monitors, thermocouples, and flow rate sensors can be very expensive and limiting. For instance, at the moment when the pressure sensor was assembled, a pressure sensor (gauge and monitor) had a market price of around R$ 10000.00 price in 2020. The full Arduino pressure sensor setup used in this work costs around R$ 300.00 (price in 2023). This was an important motivation for the use of Arduino-based sensors.

All sensors were developed and continuously improved throughout the project. The use of the simple Arduino technology in the measurement of actual physical quantities can be very challenging, given the limitations on its robustness and computing capabilities. Most sensors, especially the pressure gauge, required adaptations on their original circuits and a careful post-treatment of the data. The assembly of the experimental setup was successfully concluded, allowing for its use in this and future projects.

8.2 Objective 2: Particles

Particles with very different properties were key to investigating the liquid-solid fluidized bed dynamics in a wide range of regimes. It is very difficult to find particles with very low densities in the local market. Hence, the alginate particles needed to be synthesized. The synthesizing procedure followed an extensively investigated and refined method developed during previous research in the group by Lopes *et al.* (2018) and Melo *et al.* (2021a), Melo *et al.* (2021b). The small contributions of this work to the improvement of the method include scaling to a more productive and efficient synthesis and the use of different dripping tips to obtain particles of different sizes.

As for the particle characterization, the main challenge was the measurement of the particle's diameter. This challenge was circumvented with the procedure described in Section 4.1.1.2. Between handcrafting steps and image post-treatment, the diameter of the particles was successfully measured. The remaining characterization procedures followed standard methods.

8.3 Objective 3: Experiments

The calibration consisted of small adaptations to the original setup. For the pressure sensor, an adaptation of the transfer function sensor factor was incorporated to account for the influence of the analog-digital converter. The procedure for the height measurement with the laser setup was developed throughout this work and reported in Section 4.1.3 and Chapter 5.

The experimental determination of the bed expansion behavior was done for all particles at several inlet flow rates. The results of these measurements are presented in Chapters 5 and 6 (Articles 1 and 2). The full validation campaign of the prediction methods used throughout this work was done with this data.

8.4 Objective 4: Validation of the bed expansion predictions

In Section 5 (Article 1), an alternative procedure to predict the bed expansion was developed and validated. This method consists of applying a force balance on the bed at the pseudo-steady state. Drag is represented by typical correlations in the literature. In special, when the drag correlation by Rong *et al.* (2013) was applied, the method presented more accurate results than the traditional Richardson-Zaki equation (RICHARDSON; ZAKI, 1954).

8.5 Objective 5: Validation of the CFD-DEM simulations

At first, a software other than Lethe would be applied to the unresolved CFD-DEM simulations of the liquid-solid fluidized beds. The software, based on the finite volume method, presented limitations in terms of scalability and precision. During an internship in Montréal, preliminary tests with Lethe pointed to a promising direction. At the time, the unresolved CFD-DEM module of the software was not fully capable of simulating the liquid-solid fluidized bed case. Faster than first anticipated, the module was fully developed, presenting very accurate results for the liquid-solid fluidized bed case.

An in-depth validation of the unresolved CFD-DEM model and computation using Lethe is presented in Chapter 6 (Article 2). The model was applied to a wide range of regimes and has proved to accurately resemble the behavior of the experiments, both quantitatively (bed expansion and pressure drop) and qualitatively (fluidization pattern and flow structures). Apart from the validation itself, the main findings of the validation campaign include the importance of the Saffman lift force to the bed's dynamics, the determination of the most accurate drag correlation, and the consistency of the method with the mesh topology.

8.6 Objective 6: Extrapolation of the CFD-DEM results

The validated case has been extrapolated for the investigation of the particles' behavior. This was done using mixing indices and for several scenarios, including different interaction particle properties and fluidization regimes. The Chapter 7 (Article 3) presents several of these results. So far, the main contributions include a discussion of the differences between the two mixing indices (NNM and Doucet's) applied and the information they bring, a plateauing behavior of the mixing with the dimensionless mixing time for looser beds, a frequency-based adaptation to the principal component analysis by (DOUCET *et al.*, 2008) to judge the principal mixing component, the determination of the axial component as the slowest component in most cases, and the negligible quantitative influence of most particle interaction properties on the process. These findings are to be submitted for publication.

CHAPTER 9

CONCLUSIONS

The previous chapters demonstrate the achievements of this work. In this chapter, we present a brief summary of these achievements, the challenges and limitations of the project, perspectives, and a foreword.

9.1 Summary

In this thesis, an experimental pilot-scale liquid-solid fluidized bed system was assembled using Arduino-based sensors developed throughout the research. This setup was extensively used to acquire experimental bed expansion data for a wide range of regimes. This data was employed in the verification and validation of predictions using the classic Richardson-Zaki equation. It was also used to validate a proposed method to estimate the bed expansion. The method applies drag correlations that are valid for various particle-fluid systems. Applying the method is easy and can be further used for other liquid-solid systems. From the results of this work, the correlation by Rong *et al.* (2013) is recommended for this approach.

The experimental results were also applied to the validation of the unresolved CFD-DEM capabilities of Lethe. The simulations accurately reproduced the experimental behavior, even for the very coarse mesh. The Saffman lift force has proven to be an important component in the reproduction of the flow structure. The choice of the drag correlation also played an important role. Again, the correlation by Rong *et al.* (2013) presented higher quantitative accuracy. Additionally, the validation was done for two very different systems.

The model and methods applied in the CFD-DEM validation campaign were used to investigate the bed's mixing behavior in the liquid-solid fluidized bed. The principal mixing component was the axial for the majority of the studied cases. The mixing rate can be influenced by the inlet flow rate, however, this influence is not monotonous and a *plateau* can be observed after beds with porosity higher than 50%. The importance of particle interaction properties is discussed, however, overall, the system was not sensitive to the variations. More concentrated systems can be influenced by the sliding friction coefficient but further investigation is still necessary.

9.2 Challenges and limitations

As discussed in Chapter 8, using Arduino-based sensors imposes resolution and range limitations on measurements. For the cases in this work, this was circumvented using specific measures for each experiment requirement. Yet the Arduino framework is highly adaptable, these measures are case-sensitive. It is worth mentioning that we failed to develop a collision sensor based on piezoelectric pastilles. The main reason for this is the difficulty in coming up with a target probe that was, at the same time, waterproof, mechanically resistant, sensitive enough to register the very weak collisions with particles, and with enough computational performance to register successive events.

The limitations of the proposed approach for the bed expansion estimation using drag correlations still need further validation. This includes comparing the predictions with several other particle-fluid pairs and equipment scales. Our setup is currently limited to the use of water as liquid. Changing scales would require another experimental setup, but the findings and technologies implemented in the one used in this work can be applied to a new one.

The unresolved CFD-DEM method is still limited in terms of scale and precision. Besides the recent enhancements in computational power and the high efficiency of algorithms, the method is still computationally intensive. Simulating the behavior of a fluid-particle system with hundreds of millions of particles still requires a robust high-performance computing infrastructure. In terms of precision, as shown, most interphase forces are represented by correlations that still need further validation. Additionally, the volume-averaging approach represents spatial averages among regions at least one scale larger than particles. For very coarse particles in limited geometries, the volume averaging assumption breaks down.

9.3 Perspectives and future works

This research is already being continued by the research group. The pilot-scale fluidized bed can be used for further investigation of the liquid-solid fluidized bed behavior. The equipment is very versatile, and it is possible to attach different columns with different shapes and sizes. A more powerful pump would also allow for the use of denser particles, such as stainless steel beads. Furthermore, other sensors can be developed and easily attached using the same structure. Furthermore, a three-phase equipment is being developed partially based on the findings of this research.

Further development on the unresolved CFD-DEM capabilities of Lethe can be done. For instance, the implementation of non-contact forces would allow for an investigation of the lubrication effect on other solid-liquid systems. Semi-resolved approaches could also be applied to increase the precision of the fluid behavior at smaller scales. Methods such as coarse-graining could be applied to allow for simulations with hundreds of millions of particles. Lastly, virtual mass, Basset, and Magnus interphase force models can be implemented and tested.

9.4 Forewords

The experimental and computational methods applied allowed for a better understanding of the internal dynamics of liquid-solid fluidized beds. The use of Arduino, Lethe, and other open-source technologies allowed for a very immersive and collaborative research journey while saving important resources with licenses and experimental apparatuses. The main findings were published or will be submitted for publishing in international journals, allowing for ampler discussion on liquid fluidization fluid and particle dynamics. The contributions of this work help enhance the predictability of the particle and fluid dynamics behavior of liquid-solid fluidized beds. As a result of the validation campaign, the methods adopted in this work, especially the unresolved CFD-DEM model, are more safely applicable to the design, fluid dynamics analysis, and optimization of this promising intensification technology.

Lastly, this author would like to express his immense gratitude for the people who actively participated in this work. Thanks are due to Daniel Silva Junior, a brilliant student who worked very hard on the experimental setup and shares authorship of the articles based on these results; Toni El Geitani Nehme, Ph.D., a great working partner who developed the unresolved CFD-DEM module of Lethe, faced together the validation campaign, and shares the authorship of the article based on the findings; Professor Bruno Blais, Ph.D., who embraced the project as if it was conceived by himself and very actively co-directed this thesis; and Professor Gabriela Cantarelli Lopes, Ph.D., for conceiving this research project, for the opportunity, for all the incentive, the lessons, and for everything else this author have experienced since the starting of this journey. Thank you very much for dedicating your precious time and energy to this project.

BIBLIOGRAPHY

AKGIRAY, O.; SOYER, E. An evaluation of expansion equations for fluidized solid–liquid systems. **Journal of Water Supply: Research and Technology-Aqua**, v. 55, n. 7-8, p. 517–526, nov. 2006. Publisher: IWA Publishing.

ALDACO, R.; GAREA, A.; IRABIEN, A. Modeling of particle growth: Application to water treatment in a fluidized bed reactor. **Chemical Engineering Journal**, v. 134, n. 1-3, p. 66–71, nov. 2007. Publisher: Elsevier BV.

ALMENDROS-IBáñEZ, J. *et al.* A review of solar thermal energy storage in beds of particles: Packed and fluidized beds. **Solar Energy**, v. 192, p. 193–237, nov. 2019. ISSN 0038092X. Available on: https://linkinghub.elsevier.com/retrieve/pii/S0038092X18304808.

AMIRTHARAJAH, A. Optimum Backwashing of Sand Filters. **Journal of the Environmental Engineering Division**, v. 104, n. 5, p. 917–932, maio 1978. ISSN 0090-3914, 2690-2435. Available on: https://ascelibrary.org/doi/10.1061/JEEGAV.0000827.

ANDERSON, T. B.; JACKSON, R. Fluid Mechanical Description of Fluidized Beds. Equations of Motion. **Industrial & Engineering Chemistry Fundamentals**, v. 6, n. 4, p. 527–539, nov. 1967. Publisher: American Chemical Society (ACS).

ANGELAKIS, A.; SNYDER, S. Wastewater Treatment and Reuse: Past, Present, and Future. **Water**, v. 7, n. 12, p. 4887–4895, set. 2015. ISSN 2073-4441. Available on: http://www.mdpi.com/2073-4441/7/9/4887.

ANSYS. **ANSYS Fluent Theory Guide**. 2013.

AOUF, M.; DOUNIT, S. Remediation of aged hydrocarbon contaminated soil by washing in fluidized bed column. **Archives of Environmental Protection**, ago. 2023. ISSN 2083-4772, 2083-4810. Available on: https://journals.pan.pl/dlibra/publication/140762/edition/123563/content.

ARNDT, D. *et al.* The deal.II Library, Version 9.5. **Journal of Numerical Mathematics**, v. 31, n. 3, p. 231–246, set. 2023. ISSN 1570-2820, 1569-3953. Available on: https://www.degruyter.com/document/doi/10.1515/jnma-2023-0089/html.

ARNDT, D. *et al.* The deal.II library, Version 9.3. **Journal of Numerical Mathematics**, v. 29, n. 3, p. 171–186, set. 2021. Publisher: Walter de Gruyter GmbH.

ARNDT, D. *et al.* The deal.II library, Version 9.4. **Journal of Numerical Mathematics**, v. 30, n. 3, p. 231–246, jul. 2022. Publisher: Walter de Gruyter GmbH.

ARNTZ, M. M. H. D. *et al.* Granular mixing and segregation in a horizontal rotating drum: A simulation study on the impact of rotational speed and fill level. **AIChE Journal**, v. 54, n. 12, p. 3133–3146, dez. 2008. Publisher: Wiley.

BARBEAU, L. *et al.* Development of a high-order continuous Galerkin sharp-interface immersed boundary method and its application to incompressible flow problems. **Computers &amp\mathsemicolon Fluids**, v. 239, p. 105415, maio 2022. Publisher: Elsevier BV.

BATISTA, J.; BRITO, R.; BéTTEGA, R. Influence of inlet air distributor geometry on the fluid dynamics of conical spouted beds: A CFD study. **Chemical Industry and Chemical Engineering Quarterly**, v. 24, n. 4, p. 369–378, 2018. ISSN 1451-9372, 2217-7434. Available on: https://doiserbia.nb.rs/Article.aspx?ID=1451-93721800009B.

BAXTER, G. W.; BEHRINGER, R. P. Cellular automata models of granular flow. **Physical Review A**, v. 42, n. 2, p. 1017–1020, jul. 1990. ISSN 1050-2947, 1094-1622. Available on: https://link.aps.org/doi/10.1103/PhysRevA.42.1017.

BEETSTRA, R.; HOEF, M. A. v. d.; KUIPERS, J. A. M. Drag force of intermediate Reynolds number flow past mono- and bidisperse arrays of spheres. **AIChE Journal**, v. 53, n. 2, p. 489–501, 2007. Publisher: Wiley.

BELLO, M. M.; RAMAN, A. A. A.; PURUSHOTHAMAN, M. Applications of fluidized bed reactors in wastewater treatment – A review of the major design and operational parameters. **Journal of Cleaner Production**, v. 141, p. 1492–1514, jan. 2017. Publisher: Elsevier BV.

BENYAHIA, S.; SYAMLAL, M.; O'BRIEN, T. J. Extension of Hill-Koch-Ladd drag correlation over all ranges of Reynolds number and solids volume fraction. **Powder Technology**, v. 162, n. 2, p. 166–174, mar. 2006. Publisher: Elsevier BV.

BHALODE, P.; IERAPETRITOU, M. A review of existing mixing indices in solid-based continuous blending operations. **Powder Technology**, v. 373, p. 195–209, ago. 2020. Publisher: Elsevier BV.

BIRD, R. B.; STEWART, W. E.; LIGHTFOOT, E. N. **Transport phenomena**. Revised ed. New York: Wiley, 2007. ISBN 978-0-470-11539-8.

BLAIS, B. **Développement d'un modèle Euler-Lagrange robuste pour la simulation des écoulements solide-liquide dans les opérations de mélange**. Tese (Doutorado) — Polytechnique Montréal, Montreal, Quebec, Canada, 2016. Available on: https://publications.polymtl.ca/2219/.

BLAIS, B. *et al.* Lethe: An open-source parallel high-order adaptative CFD solver for incompressible flows. **SoftwareX**, v. 12, p. 100579, jul. 2020. Publisher: Elsevier BV.

BLAIS, B.; BERTRAND, F. CFD-DEM investigation of viscous solid–liquid mixing: Impact of particle properties and mixer characteristics. **Chemical Engineering Research and Design**, v. 118, p. 270–285, fev. 2017. Publisher: Elsevier BV.

BLAIS, B. *et al.* CFD-DEM simulations of early turbulent solid–liquid mixing: Prediction of suspension curve and just-suspended speed. **Chemical Engineering Research and Design**, v. 123, p. 388–406, jul. 2017. Publisher: Elsevier BV.

BLAIS, B. *et al.* Development of an unresolved CFD–DEM model for the flow of viscous suspensions and its application to solid–liquid mixing. **Journal of Computational Physics**, v. 318, p. 201–221, ago. 2016. Publisher: Elsevier BV.

BLAIS, B. *et al.* Experimental Methods in Chemical Engineering: Discrete Element Method—DEM. **The Canadian Journal of Chemical Engineering**, v. 97, n. 7, p. 1964–1973, jun. 2019. Publisher: Wiley.

BRADSKI, G. The OpenCV Library. **Dr. Dobb's Journal of Software Tools**, 2000.

BREY, J.; RUIZ-MONTERO, M. Direct Monte Carlo simulation of dilute granular flow. **Computer Physics Communications**, v. 121-122, p. 278–283, set. 1999. ISSN 00104655. Available on: https://linkinghub.elsevier.com/retrieve/pii/S0010465599003318.

BROWN, D.; HANSON, R.; CHRISTIAN, W. Tracker Video Analysis and Modeling Tool. 2022.

BURSTEDDE, C.; WILCOX, L. C.; GHATTAS, O. p4est : Scalable Algorithms for Parallel Adaptive Mesh Refinement on Forests of Octrees. **SIAM Journal on Scientific Computing**, v. 33, n. 3, p. 1103–1133, jan. 2011. ISSN 1064-8275, 1095-7197. Available on: http://epubs.siam.org/doi/10.1137/100791634.

BUYEVICH, Y. A. Statistical hydromechanics of disperse systems Part 1. Physical background and general equations. **Journal of Fluid Mechanics**, v. 49, n. 03, p. 489, out. 1971. ISSN 0022-1120, 1469-7645. Available on: http://www.journals.cambridge.org/abstract_S0022112071002222.

BYERLY, W. E. **Introduction to the Calculus of Variations**. [S.l.: s.n.], 1917. (Mathematical Tracts for Physiscs).

BéRARD, A.; PATIENCE, G. S.; BLAIS, B. Experimental methods in chemical engineering: Unresolved CFD-DEM. **The Canadian Journal of Chemical Engineering**, v. 98, n. 2, p. 424–440, jan. 2020. Publisher: Wiley.

BéTTEGA, R. *et al.* CFD modelling of a semi-cylindrical spouted bed: Numerical simulation and experimental verification. **The Canadian Journal of Chemical Engineering**, v. 87, n. 2, p. 177–184, abr. 2009. ISSN 00084034, 1939019X. Available on: https://onlinelibrary.wiley.com/doi/10.1002/cjce.20153.

CALLISTER, W. D.; RETHWISCH, D. G. **Materials science and engineering: an introduction: SI Version**. 10th edition, global edition. ed. Hoboken, NJ: Wiley, 2020. ISBN 978-1-119-45391-8.

CAMPBELL, C. S. Rapid Granular Flows. **Annual Review of Fluid Mechanics**, v. 22, n. 1, p. 57–90, jan. 1990. ISSN 0066-4189, 1545-4479. Available on: https://www.annualreviews.org/doi/10.1146/annurev.fl.22.010190.000421.

CAMPBELL, C. S.; BRENNEN, C. E. Computer simulation of granular shear flows. **Journal of Fluid Mechanics**, v. 151, n. -1, p. 167, fev. 1985. ISSN 0022-1120, 1469-7645. Available on: http://www.journals.cambridge.org/abstract_S002211208500091X.

CAMPOS, J. P. F.; MELO, K. R. B.; LOPES, G. C. Implementation, validation and application of a lubrication force model in CFD-DEM simulations. **Brazilian Journal of Chemical Engineering**, v. 39, n. 2, p. 429–440, jun. 2022. ISSN 0104-6632, 1678-4383. Available on: https://link.springer.com/10.1007/s43153-021-00134-1.

CAPECELATRO, J.; DESJARDINS, O. An Euler–Lagrange strategy for simulating particle-laden flows. **Journal of Computational Physics**, v. 238, p. 1–31, abr. 2013. Publisher: Elsevier BV.

CAYETANO, R. D. A. *et al.* Biofilm formation as a method of improved treatment during anaerobic digestion of organic matter for biogas recovery. **Bioresource Technology**, v. 344, p. 126309, jan. 2022. ISSN 09608524. Available on: https://linkinghub.elsevier.com/retrieve/pii/S0960852421016515.

CELLO, F.; DI RENZO, A.; DI MAIO, F. P. A semi-empirical model for the drag force and fluid–particle interaction in polydisperse suspensions. **Chemical Engineering Science**, v. 65, n. 10, p. 3128–3139, maio 2010. Publisher: Elsevier BV.

CHEKNANE, B. *et al.* Fluidized bed flocculation-coagulation of seawater from the Algiers area. **Desalination**, v. 179, n. 1-3, p. 273–280, jul. 2005. ISSN 00119164. Available on: https://linkinghub.elsevier.com/retrieve/pii/S0011916405002912.

CHEN, J. J. J. "Improved equation for the calculation of minimum fluidization velocity." Comments. **Industrial & Engineering Chemistry Research**, v. 26, n. 3, p. 633–634, mar. 1987. ISSN 0888-5885, 1520-5045. Available on: https://pubs.acs.org/doi/abs/10.1021/ie00063a039.

CHEN, R. C.; FAN, L.-S. Particle image velocimetry for characterizing the flow structure in three-dimensional gas-liquid-solid fluidized beds. **Chemical Engineering Science**, v. 47, n. 13-14, p. 3615–3622, set. 1992. Publisher: Elsevier BV.

CHEN, X.-g. *et al.* Bed expansion behavior and sensitivity analysis for super-high-rate anaerobic bioreactor. **Journal of Zhejiang University SCIENCE B**, v. 11, n. 2, p. 79–86, fev. 2010. ISSN 1673-1581, 1862-1783. Available on: http://link.springer.com/10.1631/jzus.B0900256.

CHENG, J. *et al.* Review on the applications and development of fluidized bed electrodes. **Journal of Solid State Electrochemistry**, v. 24, n. 10, p. 2199–2217, out. 2020. ISSN 1432-8488, 1433-0768. Available on: https://link.springer.com/10.1007/s10008-020-04786-w.

CLARKE, D. A. *et al.* Investigation of Void Fraction Schemes for Use with CFD-DEM Simulations of Fluidized Beds. **Industrial & Engineering Chemistry Research**, v. 57, n. 8, p. 3002–3013, fev. 2018. Publisher: American Chemical Society (ACS).

CLEASBY, J. *et al.* Backwashing of Granular Filters. **Journal AWWA**, v. 69, n. 2, p. 115–126, fev. 1977. ISSN 0003-150X, 1551-8833. Available on: https://awwa.onlinelibrary.wiley.com/doi/10.1002/j.1551-8833.1977.tb06668.x.

CLIFT, R.; GRACE, J. R.; WEBER, M. E. **Bubbles, drops, and particles**. [S.l.]: Academic Press, 1978. ISBN 0-12-176950-X.

CORNELISSEN, J. T. *et al.* CFD modelling of a liquid–solid fluidized bed. **Chemical Engineering Science**, v. 62, n. 22, p. 6334–6348, nov. 2007. Publisher: Elsevier BV.

COULSON, J. M. **Particulate systems and particle technology**. Sixth edition. Oxford Cambridge, MA: Elsevier, Butterworth-Heinemann, 2019. (Coulson and Richardson's chemical engineering, Volume 2A). ISBN 978-0-08-101098-3.

COURANT, R.; FRIEDRICHS, K.; LEWY, H. On the Partial Difference Equations of Mathematical Physics. **IBM Journal of Research and Development**, v. 11, n. 2, p. 215–234, mar. 1967. ISSN 0018-8646, 0018-8646. Available on: http://ieeexplore.ieee.org/document/5391985/.

CROWE, C. T. *et al.* **Multiphase Flows with Droplets and Particles**. [S.l.]: CRC Press, 2011.

CUNDALL, P. A.; STRACK, O. D. L. A discrete numerical model for granular assemblies. **Géotechnique**, v. 29, n. 1, p. 47–65, mar. 1979. Publisher: Thomas Telford Ltd.

DALLAVALLE, J. M. **Micromeritics: The Technology of Fine Particles**. [S.l.]: Pitman Pub. Corp., 1948.

DANCKWERTS, P. V. Continuous flow systems. **Chemical Engineering Science**, v. 2, n. 1, p. 1–13, fev. 1953. Publisher: Elsevier BV.

DAVIDSON, P. A. **Turbulence: an introduction for scientists and engineers**. Second edition. Oxford, United Kingdom ; New York, NY, United States of America: Oxford University Press, 2015. OCLC: ocn911091654. ISBN 978-0-19-872258-8 978-0-19-872259-5.

DDBST. **Dortmund Data Bank. Online Calculation**. 2021. Available on: http://www.ddbst.com/calculation.html.

DEAL.II. **Reference documentation for deal.II version 9.5.0**. 2023.

DEB, S.; TAFTI, D. K. A novel two-grid formulation for fluid–particle systems using the discrete element method. **Powder Technology**, v. 246, p. 601–616, set. 2013. Publisher: Elsevier BV.

DENG, Z. *et al.* Design of anaerobic fluidized bed bioreactor – Dyeing effluents. **Chemical Engineering Science**, v. 139, p. 273–284, jan. 2016. ISSN 00092509. Available on: https://linkinghub.elsevier.com/retrieve/pii/S000925091500651X.

DHARMARAJAH, A. H.; CLEASBY, J. L. Predicting the Expansion Behavior of Filter Media. **Journal - American Water Works Association**, v. 78, n. 12, p. 66–76, dez. 1986. Publisher: Wiley.

DIAS, T. R. *et al.* Fluidized particles in flow analysis: potentialities, limitations and applications. **Talanta**, v. 184, p. 325–331, jul. 2018. Publisher: Elsevier BV.

DOUCET, J.; BERTRAND, F.; CHAOUKI, J. A measure of mixing from Lagrangian tracking and its application to granular and fluid flow systems. **Chemical Engineering Research and Design**, v. 86, n. 12, p. 1313–1321, dez. 2008. Publisher: Elsevier BV.

DREW, D. A. Mathematical Modeling of Two-Phase Flow. **Annual Review of Fluid Mechanics**, v. 15, n. 1, p. 261–291, jan. 1983. ISSN 0066-4189, 1545-4479. Available on: https://www.annualreviews.org/doi/10.1146/annurev.fl.15.010183.001401.

DURAN, J. **Sands, powders, and grains: an introduction to the physics of granular materials**. Place of publication not identified: Springer, 2012. OCLC: 1136082869. ISBN 978-1-4612-6790-4.

EITZLMAYR, A.; KHINAST, J. Co-rotating twin-screw extruders: Detailed analysis of conveying elements based on smoothed particle hydrodynamics. Part 2: Mixing. **Chemical Engineering Science**, v. 134, p. 880–886, set. 2015. Publisher: Elsevier BV.

ELMAN, H. C.; SILVESTER, D. J.; WATHEN, A. J. **Finite elements and fast iterative solvers: with applications in incompressible fluid dynamics**. Second edition. Oxford, United Kingdom: Oxford University Press, 2014. (Numerical mathematics and scientific computation). ISBN 978-0-19-967879-2 978-0-19-967880-8.

ELREEDY, A. *et al.* Pathways of 3-biofules (hydrogen, ethanol and methane) production from petrochemical industry wastewater via anaerobic packed bed baffled reactor inoculated with mixed culture bacteria. **Energy Conversion and Management**, v. 122, p. 119–130, ago. 2016. Publisher: Elsevier BV.

EPSTEIN, N. Applications of Liquid-Solid Fluidization. **International Journal of Chemical Reactor Engineering**, v. 1, n. 1, nov. 2003. Publisher: Walter de Gruyter GmbH.

EPSTEIN, N. Liquid-solids fluidization. In: YANG, W.-C. (Ed.). **Handbook of fluidization and fluid-particle systems**. [S.l.]: Marcel Dekker, New York, NY, USA, 2003. ISBN 0-8247-0259-X. Section: 26.

ERGUN, S. Fluid flow through packed columns. **Chemical Engineering Progress**, v. 48, n. 2, p. 89–94, 1952.

ESCUDIé, R. *et al.* Effect of particle shape on liquid-fluidized beds of binary (and ternary) solids mixtures: segregation vs. mixing. **Chemical Engineering Science**, v. 61, n. 5, p. 1528–1539, mar. 2006. Publisher: Elsevier BV.

FAN, L. T.; CHEN, S. J.; WATSON, C. A. ANNUAL REVIEW Solids Mixing. **Industrial & Engineering Chemistry**, v. 62, n. 7, p. 53–69, jul. 1970. Publisher: American Chemical Society (ACS).

FEFFERMAN, C. L. Existence and smoothness of the Navier-Stokes equation. 2000. Available on: https://www.claymath.org/wp-content/uploads/2022/06/navierstokes.pdf.

DI FELICE, R. The voidage function for fluid-particle interaction systems. **International Journal of Multiphase Flow**, v. 20, n. 1, p. 153–159, fev. 1994. Publisher: Elsevier BV.

FERELLEC, J.; MCDOWELL, G. Modelling realistic shape and particle inertia in DEM. **Géotechnique**, v. 60, n. 3, p. 227–232, mar. 2010. ISSN 0016-8505, 1751-7656. Available on: https://www.icevirtuallibrary.com/doi/10.1680/geot.9.T.015.

FERREIRA, V. O. *et al.* In-depth validation of unresolved CFD-DEM simulations of liquid fluidized beds. **Powder Technology**, v. 426, p. 118652, ago. 2023. ISSN 00325910. Available on: https://linkinghub.elsevier.com/retrieve/pii/S0032591023004369.

FERREIRA, V. O. *et al.* Prediction of the bed expansion of a liquid fluidized bed bioreactor applied to wastewater treatment and biogas production. **Energy Conversion and Management**, v. 290, p. 117224, 2023. ISSN 01968904. Available on: https://linkinghub.elsevier.com/retrieve/pii/S0196890423005708.

FERZIGER, J. H.; PERIć, M.; STREET, R. L. **Computational methods for fluid dynamics**. Fourth edition. Cham: Springer, 2020. ISBN 978-3-319-99691-2.

FIDLERIS, V.; WHITMORE, R. L. Experimental determination of the wall effect for spheres falling axially in cylindrical vessels. **British Journal of Applied Physics**, v. 12, n. 9, p. 490–494, set. 1961. Publisher: IOP Publishing.

FRAIA, S. D.; MASSAROTTI, N.; VANOLI, L. A novel energy assessment of urban wastewater treatment plants. **Energy Conversion and Management**, v. 163, p. 304–313, maio 2018. Publisher: Elsevier BV.

FRAIGE, F.; LANGSTON, P. Integration schemes and damping algorithms in distinct element models. **Advanced Powder Technology**, v. 15, n. 2, p. 227–245, 2004. ISSN 09218831. Available on: https://linkinghub.elsevier.com/retrieve/pii/S0921883108601314.

FRANKOPAN, P. **The silk roads: a new history of the world**. First vintage books edition. New York: Vintage Books, 2017. OCLC: 951948048. ISBN 978-1-101-91237-9.

FREIREICH, B.; KODAM, M.; WASSGREN, C. An exact method for determining local solid fractions in discrete element method simulations. **AIChE Journal**, v. 56, n. 12, p. 3036–3048, nov. 2010. Publisher: Wiley.

GAO, R. *et al.* A new method to simulate irregular particles by discrete element method. **Journal of Rock Mechanics and Geotechnical Engineering**, v. 4, n. 3, p. 276–281, set. 2012. ISSN 16747755. Available on: https://linkinghub.elsevier.com/retrieve/pii/S167477551530189X.

GARCIA, X. *et al.* A clustered overlapping sphere algorithm to represent real particles in discrete element modelling. **Géotechnique**, v. 59, n. 9, p. 779–784, nov. 2009. ISSN 0016-8505, 1751-7656. Available on: https://www.icevirtuallibrary.com/doi/10.1680/geot.8.T.037.

GARCíA-CALDERóN, D. *et al.* Influence of biomass accumulation on bed expansion characteristics of a down-flow anaerobic fluidized-bed reactor. **Biotechnology and Bioengineering**, v. 57, n. 2, p. 136–144, jan. 1998. Publisher: Wiley Online Library.

GARSIDE, J.; AL-DIBOUNI, M. R. Velocity-Voidage Relationships for Fluidization and Sedimentation in Solid-Liquid Systems. **Industrial & Engineering Chemistry Process Design and Development**, v. 16, n. 2, p. 206–214, abr. 1977. Publisher: American Chemical Society (ACS).

GEITANI, T.; BLAIS, B. Solid-liquid rotary kilns: An experimental and CFD-DEM study. **Powder Technology**, p. 119008, set. 2023. ISSN 00325910. Available on: https://linkinghub.elsevier.com/retrieve/pii/S003259102300791X.

GEITANI, T. E.; BLAIS, B. Quadrature-Centered Averaging Scheme for Accurate and Continuous Void Fraction Calculation in Computational Fluid Dynamics–Discrete Element Method Simulations. **Industrial & Engineering Chemistry Research**, mar. 2023. Publisher: American Chemical Society (ACS).

GEITANI, T. E.; GOLSHAN, S.; BLAIS, B. A high-order stabilized solver for the volume averaged Navier-Stokes equations. **International Journal for Numerical Methods in Fluids**, fev. 2023. Publisher: Wiley.

GEITANI, T. E.; GOLSHAN, S.; BLAIS, B. Towards High-Order CFD-DEM: Development and Validation. **Industrial & Engineering Chemistry Research**, jan. 2023. Publisher: American Chemical Society (ACS).

DE GENNES, P. Reflections on the mechanics of granular matter. **Physica A: Statistical Mechanics and its Applications**, v. 261, n. 3-4, p. 267–293, dez. 1998. ISSN 03784371. Available on: https://linkinghub.elsevier.com/retrieve/pii/S0378437198004385.

GEORGE, W. K. **Lectures in Turbulence for the 21st Century**. 2013. Available on: http://www.turbulence-online.com/Publications/Lecture_Notes/Turbulence_Lille/TB_16January2013.pdf.

GIDASPOW, D. **Multiphase Flow and Fluidization: Continuum and Kinetic Theory Descriptions**. Boston: Academic Press, 1994. ISBN 978-0-12-282470-8.

GIDASPOW, D.; BEZBURUAH, R.; DING, J. Hydrodynamics of circulating fluidized beds: Kinetic theory approach. In: . Illinois Inst. of Tech., Chicago, IL (United States). Dept. of Chemical Engineering, 1991. Available on: https://www.osti.gov/biblio/5896246.

GODLIEB, W.; DEEN, N. G.; KUIPERS, J. A. M. Characterizing solids mixing in DEM simulations. In: **6th International Conference on Multiphase Flow, ICMF 2007, Leipzig, Germany**. [s.n.], 2007. Available on: https://www.researchgate.net/profile/Niels-Deen/publication/228722534_Characterizing_solids_mixing_in_DEM_simulations/links/00b495289f429c5b39000000/Characterizing-solids-mixing-in-DEM-simulations.pdf.

GOLSHAN, S. *et al.* Lethe-DEM: an open-source parallel discrete element solver with load balancing. **Computational Particle Mechanics**, v. 10, n. 1, p. 77–96, maio 2022. Publisher: Springer Science and Business Media LLC.

GRACE, J.; BI, X.; ELLIS, N. (Ed.). **Essentials of Fluidization Technology**. 1. ed. Wiley, 2020. ISBN 978-3-527-34064-4 978-3-527-69948-3. Available on: https://onlinelibrary.wiley.com/doi/book/10.1002/9783527699483.

GUI, L. *et al.* Liquid solid fluidized bed crystallization granulation technology: Development, applications, properties, and prospects. **Journal of Water Process Engineering**, v. 45, p. 102513, fev. 2022. ISSN 22147144. Available on: https://linkinghub.elsevier.com/retrieve/pii/S2214714421006000.

GUSTAVII, B. **How to Prepare a Scientific Doctoral Dissertation Based on Research Articles**. 1. ed. Cambridge University Press, 2012. ISBN 978-1-107-66904-8 978-1-139-15125-2 978-1-107-02349-9. Available on: https://www.cambridge.org/core/product/identifier/9781139151252/type/book.

HAGER, A. *et al.* Parallel Resolved Open Source CFD-DEM: Method, Validation and Application. **The Journal of Computational Multiphase Flows**, v. 6, n. 1, p. 13–27, mar. 2014. Publisher: SAGE Publications.

HAY, A. *et al.* hp-Adaptive time integration based on the BDF for viscous flows. **Journal of Computational Physics**, v. 291, p. 151–176, jun. 2015. ISSN 00219991. Available on: https://linkinghub.elsevier.com/retrieve/pii/S0021999115001692.

HEISTER, T.; RAPIN, G. Efficient augmented Lagrangian-type preconditioning for the Oseen problem using Grad-Div stabilization: AUGMENTED LAGRANGIAN-TYPE PRECONDITIONING USING GRAD-DIV STABILIZATION. **International Journal for Numerical Methods in Fluids**, v. 71, n. 1, p. 118–134, jan. 2013. ISSN 02712091. Available on: https://onlinelibrary.wiley.com/doi/10.1002/fld.3654.

HENSLER, T. *et al.* Positron emission particle tracking in fluidized beds with secondary gas injection. **Powder Technology**, v. 279, p. 113–122, jul. 2015. Publisher: Elsevier BV.

VAN DER HOEF, M. A. *et al.* Multiscale Modeling of Gas-Fluidized Beds. In: **Advances in Chemical Engineering**. Elsevier, 2006. v. 31, p. 65–149. ISBN 978-0-12-008531-6. Available on: https://linkinghub.elsevier.com/retrieve/pii/S0065237706310022.

HOPKINS, M. A.; SHEN, H. H. A Monte Carlo Simulation of a Simple Shear Flow of Granular Materials. In: **Studies in Applied Mechanics**. Elsevier, 1988. v. 20, p. 349–358. ISBN 978-0-444-70523-5. Available on: https://linkinghub.elsevier.com/retrieve/pii/B9780444705235500473.

ISLAM, M. T.; NGUYEN, A. V. Effect of particle size and shape on liquid–solid fluidization in a HydroFloat cell. **Powder Technology**, v. 379, p. 560–575, fev. 2021. Publisher: Elsevier BV.

JAIN, V. *et al.* Experimental and numerical investigation of liquid–solid binary fluidized beds: Radioactive particle tracking technique and dense discrete phase model simulations. **Particuology**, v. 33, p. 112–122, ago. 2017. Publisher: Elsevier BV.

JAJCEVIC, D. *et al.* Large-scale CFD–DEM simulations of fluidized granular systems. **Chemical Engineering Science**, v. 98, p. 298–310, jul. 2013. ISSN 00092509. Available on: https://linkinghub.elsevier.com/retrieve/pii/S000925091300345X.

JAMALI, N. S. *et al.* Particle size variations of activated carbon on biofilm formation in thermophilic biohydrogen production from palm oil mill effluent. **Energy Conversion and Management**, v. 141, p. 354–366, jun. 2017. Publisher: Elsevier BV.

JENKINS, J. T.; SAVAGE, S. B. A theory for the rapid flow of identical, smooth, nearly elastic, spherical particles. **Journal of Fluid Mechanics**, v. 130, n. -1, p. 187, maio 1983. ISSN 0022-1120, 1469-7645. Available on: http://www.journals.cambridge.org/abstract_S0022112083001044.

KARNIADAKIS, G.; BEşKöK, A.; ALURU, N. R. **Microflows and nanoflows: fundamentals and simulation**. New York, NY: Springer, 2005. (Interdisciplinary applied mathematics, v. 29). ISBN 978-0-387-22197-7.

KELLY, D. W. *et al.* A posteriori error analysis and adaptive processes in the finite element method: Part I—error analysis. **International Journal for Numerical Methods in Engineering**, v. 19, n. 11, p. 1593–1619, nov. 1983. ISSN 0029-5981, 1097-0207. Available on: https://onlinelibrary.wiley.com/doi/10.1002/nme.1620191103.

KHAN, A. R.; RICHARDSON, J. F. Fluid-particle interactions and flow characteristics of fluidized beds and settling suspensions of spherical particles. **Chemical Engineering Communications**, v. 78, n. 1, p. 111–130, abr. 1989. Publisher: Informa UK Limited.

KHAN, M. S. *et al.* Analysis of particle dispersion coefficient in solid-liquid fluidised beds. **Powder Technology**, v. 365, p. 60–73, abr. 2020. Publisher: Elsevier BV.

KHAWAJA, H. A. *et al.* Quantitative Analysis of Accuracy of Voidage Computations in CFD-DEM Simulations. **The Journal of Computational Multiphase Flows**, v. 4, n. 2, p. 183–192, jun. 2012. Publisher: SAGE Publications.

KLOSS, C.; GONIVA, C. LIGGGHTS - Open Source Discrete Element Simulations of Granular Materials Based on LAMMMPS. **Supplemental Proceedings**, p. 781–788, abr. 2011. Publisher: John Wiley & Sons, Inc.

KOERICH, D. M.; LOPES, G. C.; ROSA, L. M. Investigation of phases interactions and modification of drag models for liquid-solid fluidized bed tapered bioreactors. **Powder Technology**, v. 339, p. 90–101, nov. 2018. Publisher: Elsevier BV.

KOHN, W. An essay on condensed matter physics in the twentieth century. **Reviews of Modern Physics**, v. 71, n. 2, p. S59–S77, mar. 1999. ISSN 0034-6861, 1539-0756. Available on: https://link.aps.org/doi/10.1103/RevModPhys.71.S59.

KOPPEJAN, V. *et al.* Mathematical modelling of expanded bed adsorption – a perspective on *in silico* process design. **Journal of Chemical Technology & Biotechnology**, v. 93, n. 7, p. 1815–1826, jul. 2018. ISSN 0268-2575, 1097-4660. Available on: https://onlinelibrary.wiley.com/doi/10.1002/jctb.5595.

KRAMER, O. J. I. *et al.* Accurate voidage prediction in fluidisation systems for full-scale drinking water pellet softening reactors using data driven models. **Journal of Water Process Engineering**, v. 37, p. 101481, out. 2020. Publisher: Elsevier BV.

KREMER, G. M. **An introduction to the Boltzmann equation and transport processes in gases**. Berlin ; New York: Springer, 2010. (Interaction of mechanics and mathematics). OCLC: ocn502034118. ISBN 978-3-642-11695-7 978-3-642-11696-4.

KUWABARA, G.; KONO, K. Restitution Coefficient in a Collision between Two Spheres. **Japanese Journal of Applied Physics**, v. 26, n. 8R, p. 1230, ago. 1987. ISSN 0021-4922, 1347-4065. Available on: https://iopscience.iop.org/article/10.1143/JJAP.26.1230.

LACEY, P. M. C. Developments in the theory of particle mixing. **Journal of Applied Chemistry**, v. 4, n. 5, p. 257–268, maio 1954. Publisher: Wiley.

LARSON, M. G.; BENGZON, F. **The finite element method: theory, implementation, and applications**. [S.l.]: Springer, 2013. ISBN 978-3-642-33286-9.

LETHE. **Lethe's documentation and theory guide**. 2023. Available on: https://lethe-cfd.github.io/lethe/index.html.

LEVEQUE, R. J. **Finite difference methods for ordinary and partial differential equations: steady-state and time-dependent problems**. Philadelphia: Society for industrial and applied mathematics, 2007. ISBN 978-0-89871-629-0.

LI, B. Q. **Discontinuous finite elements in fluid dynamics and heat transfer**. London: Springer, 2006. (Computational fluid and solid mechanics). ISBN 978-1-85233-988-3 978-1-84628-205-8.

LI, X. *et al.* IBM-LBM-DEM Study of Two-Particle Sedimentation: Drafting-Kissing-Tumbling and Effects of Particle Reynolds Number and Initial Positions of Particles. **Energies**, v. 15, n. 9, p. 3297, abr. 2022. ISSN 1996-1073. Available on: https://www.mdpi.com/1996-1073/15/9/3297.

LI, Y. *et al.* LDV measurements of particle velocity distribution and annular film thickness in a turbulent fluidized bed. **Powder Technology**, v. 305, p. 578–590, jan. 2017. Publisher: Elsevier BV.

LI, Y.; XU, Y.; THORNTON, C. A comparison of discrete element simulations and experiments for 'sandpiles' composed of spherical particles. **Powder Technology**, v. 160, n. 3, p. 219–228, dez. 2005. ISSN 00325910. Available on: https://linkinghub.elsevier.com/retrieve/pii/S0032591005004079.

LIANG, L.; MICHAELIDES, E. E. The Magnitude of Basset Forces in Unsteady Multiphase Flow Computations. **Journal of Fluids Engineering**, v. 114, n. 3, p. 417–419, set. 1992. Publisher: ASME International.

LIGGGHTS. **LIGGGHTS(R)-PUBLIC Documentation, Version 3.X**. 2016. Available on: http://www.liggghts.com.

LIN, L.; YANG, H.; XU, X. Effects of Water Pollution on Human Health and Disease Heterogeneity: A Review. **Frontiers in Environmental Science**, v. 10, p. 880246, jun. 2022. ISSN 2296-665X. Available on: https://www.frontiersin.org/articles/10.3389/fenvs.2022.880246/full.

LOPES, G. C. *et al.* Hydrodynamic characteristics of particles with different roughness and deformability in a liquid fluidized bed. **Chemical Engineering Science**, v. 185, p. 50–63, ago. 2018. Publisher: Elsevier BV.

LV, B. *et al.* Hydrodynamics and adsorption performance of liquid–solid fluidized bed with granular activated carbon for removal of copper ions from wastewater. **Journal of Cleaner Production**, v. 328, p. 129627, dez. 2021. ISSN 09596526. Available on: https://linkinghub.elsevier.com/retrieve/pii/S0959652621038051.

MAESTRI, M. *et al.* CFD-DEM modeling of solid motion in a water-calcium alginate fluidized column and its comparison with results from radioactive particle tracking. **Chemical Engineering Journal**, v. 377, p. 120339, dez. 2019. Publisher: Elsevier BV.

MALAN, A. G.; LEWIS, R. W.; NITHIARASU, P. An improved unsteady, unstructured, artificial compressibility, finite volume scheme for viscous incompressible flows: Part I. Theory and implementation. **International Journal for Numerical Methods in Engineering**, v. 54, n. 5, p. 695–714, jun. 2002. ISSN 0029-5981, 1097-0207. Available on: https://onlinelibrary.wiley.com/doi/10.1002/nme.447.

MARCHELLI, F. *et al.* Comparison of different drag models in CFD-DEM simulations of spouted beds. **Powder Technology**, v. 360, p. 1253–1270, jan. 2020. Publisher: Elsevier BV.

MARINACK, M. C.; HIGGS, C. F. The Inclusion of Friction in Lattice-Based Cellular Automata Modeling of Granular Flows. **Journal of Tribology**, v. 133, n. 3, p. 031302, jul. 2011. ISSN 0742-4787, 1528-8897. Available on: https://asmedigitalcollection.asme.org/tribology/article/doi/10.1115/1.4004103/475862/The-Inclusion-of-Friction-in-LatticeBased-Cellular.

MATSUSHIMA, T. *et al.* 3D Shape Characterization and Image-Based DEM Simulation of the Lunar Soil Simulant FJS-1. **Journal of Aerospace Engineering**, v. 22, n. 1, p. 15–23, 2009. ISSN 0893-1321, 1943-5525. Available on: https://ascelibrary.org/doi/10.1061/%28ASCE%290893-1321%282009%2922%3A1%2815%29.

MAXEY, M. R.; RILEY, J. J. Equation of motion for a small rigid sphere in a nonuniform flow. **The Physics of Fluids**, v. 26, n. 4, p. 883–889, abr. 1983. ISSN 0031-9171. Available on: https://pubs.aip.org/pfl/article/26/4/883/816404/Equation-of-motion-for-a-small-rigid-sphere-in-a.

MAZZEI, L.; LETTIERI, P. A drag force closure for uniformly dispersed fluidized suspensions. **Chemical Engineering Science**, v. 62, n. 22, p. 6129–6142, nov. 2007. Publisher: Elsevier BV.

MBUNGU, N. T. *et al.* An overview of renewable energy resources and grid integration for commercial building applications. **Journal of Energy Storage**, v. 29, p. 101385, jun. 2020. ISSN 2352152X. Available on: https://linkinghub.elsevier.com/retrieve/pii/S2352152X19316962.

MEI, R. An approximate expression for the shear lift force on a spherical particle at finite reynolds number. **International Journal of Multiphase Flow**, v. 18, n. 1, p. 145–147, jan. 1992. Publisher: Elsevier BV.

MELO, K. R. B.; GRACE, J. R.; LOPES, G. C. A new correlation for the coefficient of restitution of particles with low mechanical resistance for modeling wastewater treatment biofilm reactors. **Clean Technologies and Environmental Policy**, v. 24, n. 2, p. 623–632, abr. 2021. Publisher: Springer Science and Business Media LLC.

MELO, K. R. B.; PáDUA, T. F. d.; LOPES, G. C. A coefficient of restitution model for particle–surface collision of particles with a wide range of mechanical characteristics. **Advanced Powder Technology**, v. 32, n. 12, p. 4723–4733, dez. 2021. Publisher: Elsevier BV.

MELO, K. R. B.; DE PáDUA, T. F.; LOPES, G. C. Investigation of carbon steel particle collisions in dry and wet environments. **Brazilian Journal of Chemical Engineering**, maio 2023. ISSN 0104-6632, 1678-4383. Available on: https://link.springer.com/10.1007/s43153-023-00353-8.

METOLINA, P.; LOPES, G. C. Numerical analysis of liquid-solid flow in tapered and cylindrical fluidized beds for wastewater treatment and biogas production. **Energy Conversion and Management**, v. 187, p. 447–458, maio 2019. Publisher: Elsevier BV.

MINDLIN, R. D. Compliance of Elastic Bodies in Contact. **Journal of Applied Mechanics**, v. 16, n. 3, p. 259–268, set. 1949. ISSN 0021-8936, 1528-9036. Available on: https://asmedigitalcollection.asme.org/appliedmechanics/article/16/3/259/1106367/Compliance-of-Elastic-Bodies-in-Contact.

MINDLIN, R. D.; DERESIEWICZ, H. Elastic Spheres in Contact Under Varying Oblique Forces. **Journal of Applied Mechanics**, v. 20, n. 3, p. 327–344, set. 1953. ISSN 0021-8936, 1528-9036. Available on: https://asmedigitalcollection.asme.org/appliedmechanics/article/20/3/327/1110114/Elastic-Spheres-in-Contact-Under-Varying-Oblique.

MIURA, H. *et al.* Bed expansion in liquid-solid two-phase fluidized beds with Newtonian and non-Newtonian fluids over the wide range of Reynolds numbers. **Powder Technology**, v. 117, n. 3, p. 239–246, jun. 2001. Publisher: Elsevier BV.

MUKHERJEE, A. K.; MISHRA, B. K.; KUMAR, R. V. Application of liquid/solid fluidization technique in beneficiation of fines. **International Journal of Mineral Processing**, v. 92, n. 1-2, p. 67–73, jul. 2009. Publisher: Elsevier BV.

NADAIS, H. *et al.* Enhancing wastewater degradation and biogas production by intermittent operation of UASB reactors. **Energy**, v. 36, n. 4, p. 2164–2168, abr. 2011. Publisher: Elsevier BV.

NAGHIB, S. D. *et al.* Expansion properties of alginate beads as cell carrier in the fluidized bed bioartificial liver. **Powder Technology**, v. 316, p. 711–717, jul. 2017. Publisher: Elsevier BV.

NELSON, M. J.; NAKHLA, G.; ZHU, J. Fluidized-Bed Bioreactor Applications for Biological Wastewater Treatment: A Review of Research and Developments. **Engineering**, v. 3, n. 3, p. 330–342, jun. 2017. ISSN 20958099. Available on: https://linkinghub.elsevier.com/retrieve/pii/S2095809917304253.

NIGMATULIN, R. Spatial averaging in the mechanics of heterogeneous and dispersed systems. **International Journal of Multiphase Flow**, v. 5, n. 5, p. 353–385, out. 1979. ISSN 03019322. Available on: https://linkinghub.elsevier.com/retrieve/pii/0301932279900132.

NIJSSEN, T. M. J. *et al.* Complete liquid-solid momentum coupling for unresolved CFD-DEM simulations. **International Journal of Multiphase Flow**, v. 132, p. 103425, nov. 2020. Publisher: Elsevier BV.

NOROUZI, H. R. *et al.* **Coupled CFD-DEM modeling: formulation, implementation and application to multiphase flows**. Chichester, West Sussex, United Kingdom: Wiley, 2016. ISBN 978-1-119-00513-1.

OESTERLé, B.; DINH, T. B. Experiments on the lift of a spinning sphere in a range of intermediate Reynolds numbers. **Experiments in Fluids**, v. 25, n. 1, p. 16–22, jun. 1998. ISSN 0723-4864, 1432-1114. Available on: http://link.springer.com/10.1007/s003480050203.

OLSHANSKII, M. *et al.* Grad–div stabilization and subgrid pressure models for the incompressible Navier–Stokes equations. **Computer Methods in Applied Mechanics and Engineering**, v. 198, n. 49-52, p. 3975–3988, nov. 2009. ISSN 00457825. Available on: https://linkinghub.elsevier.com/retrieve/pii/S004578250900303X.

OPENFOAM. **OpenCFD, OpenFOAM - the open source CFD toolbox**. 2021.

ORSZAG, S. A. Analytical theories of turbulence. **Journal of Fluid Mechanics**, v. 41, n. 2, p. 363–386, abr. 1970. ISSN 0022-1120, 1469-7645. Available on: https://www.cambridge.org/core/product/identifier/S0022112070000642/type/journal_article.

OVERMEEN, A. **What is the difference between laminar flow and turbulent flow?** 2021. Available on: https://www.bronkhorst.com/int/blog-1/what-is-the-difference-between-laminar-flow-and-turbulent-flow/.

PAN, S. *et al.* Theoretical and experimental insight into the homogeneous expansion of wet particles in a fluidized bed. **Powder Technology**, v. 397, p. 117016, jan. 2022. ISSN 00325910. Available on: https://linkinghub.elsevier.com/retrieve/pii/S0032591021010147.

PEDREGOSA, F. *et al.* Scikit-learn: Machine Learning in Python. **Journal of Machine Learning Research**, v. 12, p. 2825–2830, 2011.

PENG, Z. *et al.* Influence of void fraction calculation on fidelity of CFD-DEM simulation of gas-solid bubbling fluidized beds. **AIChE Journal**, v. 60, n. 6, p. 2000–2018, mar. 2014. Publisher: Wiley.

PENG, Z. *et al.* Segregation and dispersion of binary solids in liquid fluidised beds: A CFD-DEM study. **Chemical Engineering Science**, v. 152, p. 65–83, out. 2016. Publisher: Elsevier BV.

PEPIOT, P.; DESJARDINS, O. Numerical analysis of the dynamics of two- and three-dimensional fluidized bed reactors using an Euler-Lagrange approach. **Powder Technology**, v. 220, p. 104–121, abr. 2012. Publisher: Elsevier BV.

PERRY, R. H.; GREEN, D. W. (Ed.). **Perry's chemical engineers' handbook**. 8th ed. ed. New York: McGraw-Hill, 2008. OCLC: ocm72470708. ISBN 978-0-07-142294-9.

PICABEA, J. *et al.* Validation of CFD-DEM simulation of a liquid–solid fluidized bed by dynamic analysis of time series. **Particuology**, v. 68, p. 75–87, set. 2022. Publisher: Elsevier BV.

PIOVANO, S. *et al.* Bed Expansion and Particle Classification in Liquid Fluidized Beds with Structured Internals. **Chemical Engineering & Technology**, v. 38, n. 3, p. 423–430, mar. 2015. ISSN 09307516. Available on: https://onlinelibrary.wiley.com/doi/10.1002/ceat.201400463.

POPE, S. B. **Turbulent flows**. 1. publ., 12. print. ed. Cambridge: Cambridge Univ. Press, 2015. ISBN 978-0-521-59886-6 978-0-521-59125-6.

PORCU, R. *et al.* MFIX-Exa: CFD-DEM simulations of thermodynamics and chemical reactions in multiphase flows. **Chemical Engineering Science**, v. 273, p. 118614, jun. 2023. ISSN 00092509. Available on: https://linkinghub.elsevier.com/retrieve/pii/S0009250923001707.

RAJCHENBACH, J.; CLEMENT, E.; DURAN, J. Experiments in Model Granular Media: A Study of Gravity Flows. **MRS Proceedings**, v. 367, p. 525, 1994. ISSN 0272-9172, 1946-4274. Available on: http://link.springer.com/10.1557/PROC-367-525.

RAPAGNà, S. *et al.* Steady-state expansion characteristics of monosize spheres fluidised by liquids. **Chemical Engineering Communications**, v. 79, n. 1, p. 131–140, maio 1989. Publisher: Informa UK Limited.

RAZZAK, S. A.; BARGHI, S.; ZHU, J.-X. Electrical resistance tomography for flow characterization of a gas–liquid–solid three-phase circulating fluidized bed. **Chemical Engineering Science**, v. 62, n. 24, p. 7253–7263, dez. 2007. Publisher: Elsevier BV.

REDDY, J. N. **An introduction to nonlinear finite element analysis: with applications to heat transfer, fluid mechanics, and solid mechanics**. Second edition. Oxford: Oxford University Press, 2015. ISBN 978-0-19-964175-8.

REDDY, J. N. **Introduction to the finite element method**. Fourth edition. New York, NY: McGraw Hill Education, 2018. OCLC: 1081298311. ISBN 978-1-259-86191-8.

REDDY, J. N.; GARTLING, D. K. **The Finite Element Method in Heat Transfer and Fluid Dynamics, Third Edition**. 3rd ed. ed. Hoboken: CRC Press, 2010. OCLC: 908077524. ISBN 978-1-4398-8257-3.

REDDY, R. K.; JOSHI, J. B. CFD modeling of solid–liquid fluidized beds of mono and binary particle mixtures. **Chemical Engineering Science**, v. 64, n. 16, p. 3641–3658, ago. 2009. Publisher: Elsevier BV.

REEKS, M. W.; MCKEE, S. The dispersive effects of Basset history forces on particle motion in a turbulent flow. **The Physics of Fluids**, v. 27, n. 7, p. 1573–1582, jul. 1984. ISSN 0031-9171. Available on: https://pubs.aip.org/pfl/article/27/7/1573/838666/The-dispersive-effects-of-Basset-history-forces-on.

DI RENZO, A.; CELLO, F.; DI MAIO, F. P. Simulation of the layer inversion phenomenon in binary liquid–fluidized beds by DEM–CFD with a drag law for polydisperse systems. **Chemical Engineering Science**, v. 66, n. 13, p. 2945–2958, jul. 2011. Publisher: Elsevier BV.

DI RENZO, A.; DI MAIO, F. P. Comparison of contact-force models for the simulation of collisions in DEM-based granular flow codes. **Chemical Engineering Science**, v. 59, n. 3, p. 525–541, fev. 2004. ISSN 00092509. Available on: https://linkinghub.elsevier.com/retrieve/pii/S0009250903005414.

DI RENZO, A.; DI MAIO, F. P. Homogeneous and bubbling fluidization regimes in DEM–CFD simulations: Hydrodynamic stability of gas and liquid fluidized beds. **Chemical Engineering Science**, v. 62, n. 1-2, p. 116–130, jan. 2007. Publisher: Elsevier BV.

REYNOLDS, O. XXIX. An experimental investigation of the circumstances which determine whether the motion of water shall be direct or sinuous, and of the law of resistance in parallel channels. **Philosophical Transactions of the Royal Society of London**, v. 174, p. 935–982, dez. 1883. ISSN 0261-0523, 2053-9223. Available on: https://royalsocietypublishing.org/doi/10.1098/rstl.1883.0029.

REYNOLDS, O. IV. On the dynamical theory of incompressible viscous fluids and the determination of the criterion. **Philosophical Transactions of the Royal Society of London. (A.)**, v. 186, p. 123–164, 1895. ISSN 0264-3820, 2053-9231. Available on: https://royalsocietypublishing.org/doi/10.1098/rsta.1895.0004.

RICHARD, P. *et al.* Slow relaxation and compaction of granular systems. **Nature Materials**, v. 4, n. 2, p. 121–128, fev. 2005. Publisher: Springer Science and Business Media LLC.

RICHARDSON, J. F.; ZAKI, W. N. Sedimentation and fluidisation: Part I. **Chemical Engineering Research and Design**, v. 75, p. S82–S100, dez. 1954. Publisher: Elsevier BV.

RICHARDSON, L. F.; LYNCH, P. **Weather Prediction by Numerical Process**. 2. ed. Cambridge University Press, 2007. ISBN 978-0-521-68044-8 978-0-511-61829-1. Available on: https://www.cambridge.org/core/product/identifier/9780511618291/type/book.

RODRIGUEZ, S. **Applied Computational Fluid Dynamics and Turbulence Modeling: Practical Tools, Tips and Techniques**. Cham: Springer International Publishing, 2019. ISBN 978-3-030-28690-3 978-3-030-28691-0. Available on: http://link.springer.com/10.1007/978-3-030-28691-0.

RODUNER, E. Size matters: why nanomaterials are different. **Chemical Society Reviews**, v. 35, n. 7, p. 583, 2006. ISSN 0306-0012, 1460-4744. Available on: http://xlink.rsc.org/?DOI=b502142c.

RONG, L. W.; DONG, K. J.; YU, A. B. Lattice-Boltzmann simulation of fluid flow through packed beds of uniform spheres: Effect of porosity. **Chemical Engineering Science**, v. 99, p. 44–58, ago. 2013. Publisher: Elsevier BV.

ROOS, H.-G.; STYNES, M.; TOBISKA, L. **Numerical Methods for Singularly Perturbed Differential Equations**. Berlin, Heidelberg: Springer Berlin Heidelberg, 1996. v. 24. (Springer Series in Computational Mathematics, v. 24). ISBN 978-3-662-03208-4 978-3-662-03206-0. Available on: http://link.springer.com/10.1007/978-3-662-03206-0.

ROTT, N. Note on the History of the Reynolds Number. **Annual Review of Fluid Mechanics**, v. 22, n. 1, p. 1–12, jan. 1990. ISSN 0066-4189, 1545-4479. Available on: https://www.annualreviews.org/doi/10.1146/annurev.fl.22.010190.000245.

ROWE, P. N. A convenient empirical equation for estimation of the Richardson-Zaki exponent. **Chemical Engineering Science**, v. 42, n. 11, p. 2795–2796, 1987. Publisher: Elsevier BV.

SAAD, Y. **Iterative methods for sparse linear systems**. 2nd ed. ed. Philadelphia: SIAM, 2003. ISBN 978-0-89871-534-7.

SAAD, Y.; SCHULTZ, M. H. GMRES: A Generalized Minimal Residual Algorithm for Solving Nonsymmetric Linear Systems. **SIAM Journal on Scientific and Statistical Computing**, v. 7, n. 3, p. 856–869, jul. 1986. ISSN 0196-5204, 2168-3417. Available on: http://epubs.siam.org/doi/10.1137/0907058.

SAAVEDRA, L. P. *et al.* An implicit large-eddy simulation perspective on the flow over periodic hills. **arXiv preprint arXiv:2206.08145v1**, 2022. Publisher: arXiv.

SAFFMAN, P. G. The lift on a small sphere in a slow shear flow. **Journal of Fluid Mechanics**, v. 22, n. 2, p. 385–400, jun. 1965. Publisher: Cambridge University Press (CUP).

SAFFMAN, P. G. The lift on a small sphere in a slow shear flow - Corrigendum. **Journal of Fluid Mechanics**, v. 31, n. 3, p. 624–624, fev. 1968. Publisher: Cambridge University Press (CUP).

SAGAUT, P. **Large Eddy Simulation for Incompressible Flows**. Berlin/Heidelberg: Springer-Verlag, 2006. (Scientific Computation). ISBN 978-3-540-26344-9. Available on: http://link.springer.com/10.1007/b137536.

SARAVANAN, V.; SREEKRISHNAN, T. Modelling anaerobic biofilm reactors—A review. **Journal of Environmental Management**, v. 81, n. 1, p. 1–18, out. 2006. ISSN 03014797. Available on: https://linkinghub.elsevier.com/retrieve/pii/S0301479705003439.

SHI, H.; KOMRAKOVA, A.; NIKRITYUK, P. Fluidized beds modeling: Validation of 2D and 3D simulations against experiments. **Powder Technology**, v. 343, p. 479–494, fev. 2019. ISSN 00325910. Available on: https://linkinghub.elsevier.com/retrieve/pii/S0032591018309471.

SHLENS, J. A Tutorial on Principal Component Analysis. **arXiv**, 2014. Publisher: arXiv Version Number: 1. Available on: https://arxiv.org/abs/1404.1100.

SOFTWARE, S. D. I. **Simcenter \uppercase{STAR-CCM+} {U}ser {G}uide v. 2021.1**. 2021.

SULLIVAN, C.; KASZYNSKI, A. PyVista: 3D plotting and mesh analysis through a streamlined interface for the Visualization Toolkit (VTK). **Journal of Open Source Software**, v. 4, n. 37, p. 1450, maio 2019. Publisher: The Open Journal.

TAGHIPOUR, F.; ELLIS, N.; WONG, C. Experimental and computational study of gas–solid fluidized bed hydrodynamics. **Chemical Engineering Science**, v. 60, n. 24, p. 6857–6867, dez. 2005. ISSN 00092509. Available on: https://linkinghub.elsevier.com/retrieve/pii/S0009250905004653.

TANG, C.; LIU, M.; LI, Y. Experimental investigation of hydrodynamics of liquid–solid mini-fluidized beds. **Particuology**, v. 27, p. 102–109, ago. 2016. Publisher: Elsevier BV.

TEAM, T. T. P. **The Trilinos Project Website**. [S.l.], 2020. Available on: https://trilinos.github.io/index.html.

TEZDUYAR, T. Stabilized Finite Element Formulations for Incompressible Flow Computations. In: **Advances in Applied Mechanics**. Elsevier, 1991. v. 28, p. 1–44. ISBN 978-0-12-002028-7. Available on: https://linkinghub.elsevier.com/retrieve/pii/S0065215608701534.

TEZDUYAR, T.; SATHE, S. Stabilization Parameters in SUPG and PSPG Formulations. **Journal of Computational and Applied Mechanics**, n. 4, p. 71–88, 2003. Available on: https://www.tafsm.org/PUB_PRE/jALL/j106-JCAM-SP.pdf.

TRAINA, K. *et al.* Flow abilities of powders and granular materials evidenced from dynamical tap density measurement. **Powder Technology**, v. 235, p. 842–852, fev. 2013. ISSN 00325910. Available on: https://linkinghub.elsevier.com/retrieve/pii/S0032591012007887.

TSUJI, Y.; TANAKA, T.; ISHIDA, T. Lagrangian numerical simulation of plug flow of cohesionless particles in a horizontal pipe. **Powder Technology**, v. 71, n. 3, p. 239–250, set. 1992. Publisher: Elsevier BV.

TURTON, R.; CLARK, N. An explicit relationship to predict spherical particle terminal velocity. **Powder Technology**, v. 53, n. 2, p. 127–129, dez. 1987. ISSN 00325910. Available on: https://linkinghub.elsevier.com/retrieve/pii/0032591087850076.

VERLET, L. Computer "Experiments" on Classical Fluids. I. Thermodynamical Properties of Lennard-Jones Molecules. **Physical Review**, v. 159, n. 1, p. 98–103, jul. 1967. ISSN 0031-899X. Available on: https://link.aps.org/doi/10.1103/PhysRev.159.98.

WAGNER, W.; PRUß, A. The IAPWS Formulation 1995 for the Thermodynamic Properties of Ordinary Water Substance for General and Scientific Use. **Journal of Physical and Chemical Reference Data**, v. 31, n. 2, p. 387–535, jun. 2002. ISSN 0047-2689, 1529-7845. Available on: https://pubs.aip.org/jpr/article/31/2/387/241937/The-IAPWS-Formulation-1995-for-the-Thermodynamic.

WALLIS, G. B. **One-Dimensional Two-Phase Flow**. [S.l.]: Dover Publications, Incorporated, 2020. ISBN 978-0-486-84282-0.

WANG, W. *et al.* A review of multiscale CFD for gas–solid CFB modeling. **International Journal of Multiphase Flow**, v. 36, n. 2, p. 109–118, fev. 2010. ISSN 03019322. Available on: https://linkinghub.elsevier.com/retrieve/pii/S0301932209000184.

WANG, Z.; TENG, Y.; LIU, M. A semi-resolved CFD–DEM approach for particulate flows with kernel based approximation and Hilbert curve based searching strategy. **Journal of Computational Physics**, v. 384, p. 151–169, maio 2019. Publisher: Elsevier BV.

WEN, C.; YU, Y. Mechanics of fluidization. In: **Chemical Engineering Progress Symposium Series**. [S.l.: s.n.], 1966.

WILCOX, D. C. **Turbulence modeling for CFD**. 3. ed., 2. print. ed. La Cañada, Calif.: DCW Industries, 2010. ISBN 978-1-928729-08-2.

WU, C. L. *et al.* Accurate void fraction calculation for three-dimensional discrete particle model on unstructured mesh. **Chemical Engineering Science**, v. 64, n. 6, p. 1260–1266, mar. 2009. Publisher: Elsevier BV.

XIE, Z.; GU, X.; SHEN, Y. A Machine Learning Study of Predicting Mixing and Segregation Behaviors in a Bidisperse Solid–Liquid Fluidized Bed. **Industrial & Engineering Chemistry Research**, v. 61, n. 24, p. 8551–8565, mar. 2022. Publisher: American Chemical Society (ACS).

XIE, Z.; WANG, S.; SHEN, Y. CFD-DEM study of segregation and mixing characteristics under a bi-disperse solid-liquid fluidised bed. **Advanced Powder Technology**, v. 32, n. 11, p. 4078–4095, nov. 2021. Publisher: Elsevier BV.

YANG, W.-C. Flow Through Fixed Beds. In: YANG, W.-C. (Ed.). **Handbook of fluidization and fluid-particle systems**. [S.l.]: Marcel Dekker, New York, NY, USA, 2003. ISBN 0-8247-0259-X. Section: 1.

YANG, W.-C. Particle Characterization and Dynamics. In: YANG, W.-C. (Ed.). **Handbook of fluidization and fluid-particle systems**. [S.l.]: Marcel Dekker, New York, NY, USA, 2003. ISBN 0-8247-0259-X. Section: 1.

YEOH, G. H.; TU, J. Basic Theory and Conceptual Framework of Multiphase Flows. In: YEOH, G. H. (Ed.). **Handbook of Multiphase Flow Science and Technology**. Singapore: Springer Singapore, 2016. p. 1–47. ISBN 978-981-4585-86-6. Available on: http://link.springer.com/10.1007/978-981-4585-86-6_1-1.

ZANOTTO, E. D.; MAURO, J. C. The glassy state of matter: Its definition and ultimate fate. **Journal of Non-Crystalline Solids**, v. 471, p. 490–495, set. 2017. ISSN 00223093. Available on: https://linkinghub.elsevier.com/retrieve/pii/S0022309317302685.

ZBIB, H. *et al.* Comprehensive analysis of fluid-particle and particle-particle interactions in a liquid-solid fluidized bed via CFD-DEM coupling and tomography. **Powder Technology**, v. 340, p. 116–130, dez. 2018. Publisher: Elsevier BV.

ZHONG, W. *et al.* CFD simulation of dense particulate reaction system: Approaches, recent advances and applications. **Chemical Engineering Science**, v. 140, p. 16–43, fev. 2016. ISSN 00092509. Available on: https://linkinghub.elsevier.com/retrieve/pii/S0009250915006661.

ZHOU, Y. *et al.* Rolling friction in the dynamic simulation of sandpile formation. **Physica A: Statistical Mechanics and its Applications**, v. 269, n. 2-4, p. 536–553, jul. 1999. ISSN 03784371. Available on: https://linkinghub.elsevier.com/retrieve/pii/S0378437199001831.

ZHOU, Z. Y. *et al.* Discrete particle simulation of particle–fluid flow: model formulations and their applicability. **Journal of Fluid Mechanics**, v. 661, p. 482–510, ago. 2010. Publisher: Cambridge University Press (CUP).

ZHU, H. P. *et al.* Discrete particle simulation of particulate systems: Theoretical developments. **Chemical Engineering Science**, v. 62, n. 13, p. 3378–3396, jul. 2007. Publisher: Elsevier BV.

ZOU, Q. GMRES algorithms over 35 years. **Applied Mathematics and Computation**, v. 445, p. 127869, maio 2023. ISSN 00963003. Available on: https://linkinghub.elsevier.com/retrieve/pii/S0096300323000383.

APPENDIX A

DERIVATION OF WEAK FORM OF POISSON EQUATION

Given T represented by the Poisson equation:

$$-\nabla \cdot (c\nabla T) = Q \text{ in } \Omega \tag{A.1}$$

with a constant c and boundaries:

$$\text{Dirichlet (essential): } T\left(\boldsymbol{x} = \boldsymbol{x}_{\Gamma_T}\right) = f_T\left(S\right) \text{ on } \Gamma_T \tag{A.2a}$$

$$\text{Neumann (natural): } q = c_b \nabla T\left(\boldsymbol{x} = \boldsymbol{x}_{\Gamma_q}\right) + h\left(T\left(\boldsymbol{x} = \boldsymbol{x}_{\Gamma_q}\right) - T_\infty\right) = f_q\left(S\right) \text{ on } \Gamma_q \tag{A.2b}$$

we want the finite element approximation to T in the domain Ω with boundaries Γ using finite elements Ω_e with boundaries Γ_e. In FEM, the nodes are defined on the vertices of the elements. For the unknown T, a finite element approximation is defined by (REDDY; GARTLING, 2010):

$$T\left(\boldsymbol{x}\right) \approx T_e\left(\boldsymbol{x}\right) = \sum_{j=1}^{N_{nodes}} T_{e,j}\psi_{e,j} \tag{A.3}$$

where N_{nodes} is the number of j nodes and ψ_e are approximation functions.

The next step is to derive the weak form. To do so, the Eq. (A.1) is multiplied by a weight function w, integrated by parts in Ω_e and the divergence theorem is applied. The result is:

$$0 = \int_{\Omega_e} \left[\nabla w \cdot (c\nabla T) - wQ\right] d\boldsymbol{x} - \oint_{\Gamma_e} wc \left.\nabla T\right|_{\Gamma_e} dS \tag{A.4}$$

For brevity, assuming that inside the domain the transport of T is diffusive, at the boundaries Γ_e, $c \left.\nabla T\right|_{\Gamma_e}$ can be represented by $q_{\Gamma e}$. The result is the weak form:

$$0 = \int_{\Omega_e} \left[\nabla w \cdot (c\nabla T) - wQ\right] d\boldsymbol{x} - \oint_{\Gamma_e} wq_{\Gamma e} dS \tag{A.5}$$

Note that the formulation “weakens” the continuity requirement by moving part of the differentiation from T to w (REDDY; GARTLING, 2010). The weak form is also represented in a bilinear form, as in:

$$0 = \underbrace{\int_{\Omega_e} \nabla w \cdot (c\nabla T)\, d\boldsymbol{x}}_{B(w,T)} - \underbrace{\int_{\Omega_e} wQd\boldsymbol{x} - \oint_{\Gamma_e} wq_{\Gamma e}dS}_{l(w)} \tag{A.6}$$

where $B(\cdot,\cdot)$ and $l(\cdot)$ stand for bilinear form and linear form, respectively:

$$B(w,t) = l(w) \tag{A.7a}$$

$$B(w,t) = \int_{\Omega_e} \nabla w \cdot (c\nabla T)\; d\boldsymbol{x} \tag{A.7b}$$

$$l(w) = \int_{\Omega_e} wQ\, d\boldsymbol{x} + \oint_{\Gamma_e} wq_{\Gamma e}\, dS \tag{A.7c}$$

The following step is the assembly of the linear system. In this example, applying Eq. (A.3) on Eq. (A.5), we obtain:

$$\int_{\Omega_e} \nabla w \cdot \left(c \sum_{j=1}^{N_{nodes}} T_{e,j} \nabla \psi_{e,j} \right) d\boldsymbol{x} = \int_{\Omega_e} wQd\boldsymbol{x} + \oint_{\Gamma_e} wq_{\Gamma e}dS \tag{A.8}$$

The terms containing $T_{e,j}$ are on the left-hand side of Eq. (A.8), while the remaining terms depending on w only are on the right-hand side. The weak form Galerkin formulation is obtained assuming the weight functions w to be equal to $\psi_{e,i}$, where $i = 1, 2, ..., N_T$ and N_T is the number of unknowns in the vector $\mathrm{T_e}$:

$$\underbrace{\sum_{j=1}^{N_{nodes}} \left[\int_{\Omega_e} \left[\nabla \psi_{e,i} \cdot (c\nabla \psi_{e,j}) \right] d\boldsymbol{x} \right]}_{A_{ij}} T_{e,j} = \underbrace{\int_{\Omega_e} Q\psi_{e,i}d\boldsymbol{x} + \oint_{\Gamma_e} q_{\Gamma e}\psi_{e,i}dS}_{Q_i} \tag{A.9}$$

Doing the same for the entire domain, the Eq. (A.9) can be generalized to:

$$\mathbf{A}\mathrm{T_e} = \mathrm{Q} \tag{A.10}$$

which is solved using linear algebra. The matrix $\mathbf{A}$ is called the coefficient matrix, $\mathrm{T_e}$ is the solution, and Q is the right-hand side (RHS). The Dirichlet boundary conditions are applied to the nodes at Γ_T, while the Neumann boundary condition requires the derivation of a weak form to be applied to nodes at Γ_q.

The phenomenon modeled by Eq. (A.1) is not time-dependent. A transient equivalent is:

$$C\frac{\partial T}{\partial t} - \nabla \cdot (c\nabla T) = Q \text{ in } \Omega \tag{A.11}$$

The finite element approximation expressed by Eq. (A.3) becomes:

$$T(\boldsymbol{x}, t) \approx \sum_{j=1}^{N_{nodes}} T_{e,j}(t)\, \psi_{e,j}(\boldsymbol{x}) \tag{A.12}$$

The weight function w is not time-dependent, so the i^{th} weak form Galerkin model $(w = \psi_{e,i}(\boldsymbol{x}))$ is:

$$0 = \sum_{j=1}^{N_{nodes}} \left[M_{e,ij} \frac{\mathrm{d}T_{e,j}}{\mathrm{d}t} + A_{ij} T_{e,j} \right] - Q_i \tag{A.13}$$

The linear system (assembly) can be represented by $(\dot{\mathrm{T}}_\mathrm{e} = \partial \mathrm{T}_\mathrm{e} / \partial t)$ (REDDY; GARTLING, 2010):

$$\mathbf{M}\dot{\mathrm{T}}_\mathrm{e} + \mathbf{A}\mathrm{T}_\mathrm{e} = \mathrm{Q} \tag{A.14}$$

APPENDIX B

LETHE PARAMETER FILES

B.1 Example of DEM particles packing settings file

```
# Listing of Parameters
# ---------------------

set dimension = 3

# --------------------------------------------------
# Simulation and IO Control
#---------------------------------------------------

subsection simulation control
  set time step        = 0.000005
  set time end         = 2.5
  set log frequency    = 20000
  set output frequency = 20000
  set output path      = ./output_dem/
end

#---------------------------------------------------
# Timer
#---------------------------------------------------

subsection timer
  set type = none
end

#---------------------------------------------------
# Restart
#---------------------------------------------------
```

```

subsection restart
  set checkpoint = true
  set frequency  = 20000
  set restart    = false
  set filename   = dem
end

# --------------------------------------------------
# Model parameters
#---------------------------------------------------

subsection model parameters
  subsection contact detection
    set contact detection method = dynamic
    set neighborhood threshold   = 1.5
  end
  subsection load balancing
    set load balance method     = dynamic
    set threshold               = 0.5
    set dynamic check frequency = 10000
  end
  set particle particle contact force method = hertz_mindlin_limit_overlap
  set particle wall contact force method     = nonlinear
  set integration method                     = velocity_verlet
end

#---------------------------------------------------
# Physical Properties
#---------------------------------------------------

subsection lagrangian physical properties
  set gx                       = -9.81
  set gy                       = 0
  set gz                       = 0
  set number of particle types = 1
  subsection particle type 0
    set size distribution type            = uniform
    set diameter                          = 0.003087
    set number                            = 72400
    set density particles                 = 3585.9
    set young modulus particles           = 1e7
    set poisson ratio particles           = 0.3
    set restitution coefficient particles = 0.9
    set friction coefficient particles    = 0.1
    set rolling friction particles        = 0.2
  end
  set young modulus wall           = 1e7
  set poisson ratio wall           = 0.3
  set restitution coefficient wall = 0.2
  set friction coefficient wall    = 0.1
  set rolling friction wall        = 0.3
end

#---------------------------------------------------
# Insertion Info
#---------------------------------------------------

subsection insertion info
  set insertion method                               = non_uniform
  set inserted number of particles at each time step = 48841 # for alginate, we recommend 79600
  set insertion frequency                            = 200000
  set insertion box minimum x                        = -0.15
  set insertion box minimum y                        = -0.035
  set insertion box minimum z                        = -0.035
  set insertion box maximum x                        = 0.53
  set insertion box maximum y                        = 0.035
  set insertion box maximum z                        = 0.035
  set insertion distance threshold                   = 1.3
  set insertion random number range                  = 0.3
  set insertion random number seed                   = 19
end

#---------------------------------------------------
# Floating walls
#---------------------------------------------------

subsection floating walls
  set number of floating walls = 1
  subsection wall 0
    subsection point on wall
      set x = -0.45
      set y = 0
      set z = 0
    end
    subsection normal vector
      set nx = 1
      set ny = 0
      set nz = 0
    end
    set start time = 0
    set end time   = 50
  end
end

#---------------------------------------------------
# Mesh
#---------------------------------------------------

subsection mesh
  set type               = dealii
  set grid type          = subdivided_cylinder
  set grid arguments     = 33:0.05:0.55
  set initial refinement = 2
  set expand particle-wall contact search = true
end
```

Listing B.1 – Example of DEM particles packing settings file.

B.2 Example of liquid-solid fluidized bed unresolved CFD-DEM settings file

```
# Listing of Parameters
# ---------------------

set dimension = 3

# --------------------------------------------------
# Simulation and IO Control
#---------------------------------------------------

subsection simulation control
  set method            = bdf1
  set number mesh adapt = 0
  set output name       = cfd_dem
  set output frequency  = 100
  set time end          = 20
  set time step         = 0.001
  set output path       = ./output/
end

#---------------------------------------------------
# FEM
#---------------------------------------------------

subsection FEM
```

```
  set velocity order = 1
  set pressure order = 1
end

#---------------------------------------------------
# Physical Properties
#---------------------------------------------------

subsection physical properties
  subsection fluid 0
    set kinematic viscosity = 0.0000008379
    set density             = 996.7775
  end
end

#---------------------------------------------------
# Mesh
#---------------------------------------------------

subsection mesh
  set type               = dealii
  set grid type          = subdivided_cylinder
  set grid arguments     = 33:0.05:0.55
  set initial refinement = 2
  set expand particle-wall contact search = true
end

#---------------------------------------------------
# Void Fraction
#---------------------------------------------------

subsection void fraction
  set mode                = pcm
  set read dem            = true
  set dem file name       = dem
  set l2 smoothing factor = 0.000028387584
end

#---------------------------------------------------
# CFD-DEM
#---------------------------------------------------

subsection cfd-dem
  set vans model         = modelA
  set grad div           = true
  set drag model         = rong
  set buoyancy force     = true
  set shear force        = true
  set pressure force     = true
  set saffman lift force = true
  set coupling frequency = 100
  set void fraction time derivative = false
end

#---------------------------------------------------
# Initial condition
#---------------------------------------------------

subsection initial conditions
  subsection uvwp
    set Function expression = 0; 0; 0; 0
  end
end

# --------------------------------------------------
# Boundary Conditions
#---------------------------------------------------

subsection boundary conditions
  set number = 2
  subsection bc 0
    set id   = 0
    set type = slip
  end
  subsection bc 1
    set id   = 1
    set type = function
    subsection u
      set Function expression = 0.157033
    end
    subsection v
      set Function expression = 0
    end
    subsection w
      set Function expression = 0
    end
  end
end

# --------------------------------------------------
# Source term
#---------------------------------------------------

subsection source term
```

```
  set enable = false
  subsection xyz
    set Function expression = 0; 0; 0; 0
  end
end

# ---------------------------------------------------
# Mesh Adaptation Control
#----------------------------------------------------

subsection mesh adaptation
  set type = none
end

#----------------------------------------------------
# Timer
#----------------------------------------------------

subsection timer
  set type = iteration
end

# ---------------------------------------------------
# Model parameters
#----------------------------------------------------

subsection model parameters
  subsection contact detection
    set contact detection method = dynamic
    set neighborhood threshold   = 1.5
  end
  set particle particle contact force method =
   hertz_mindlin_limit_overlap
  set particle wall contact force method     = nonlinear
  set integration method                     = velocity_verlet
end

#----------------------------------------------------
# Physical Properties
#----------------------------------------------------

subsection lagrangian physical properties
  set gx                       = -9.81
  set gy                       = 0
  set gz                       = 0
  set number of particle types = 1
  subsection particle type 0
```

```
    set size distribution type            = uniform
    set diameter                          = 0.003087
    set number                            = 72400
    set density particles                 = 3585.9
    set young modulus particles           = 1e7
    set poisson ratio particles           = 0.3
    set restitution coefficient particles = 0.9
    set friction coefficient particles    = 0.1
    set rolling friction particles        = 0.2
  end
  set young modulus wall             = 1e7
  set poisson ratio wall             = 0.3
  set restitution coefficient wall = 0.2
  set friction coefficient wall    = 0.1
  set rolling friction wall        = 0.3
end

#---------------------------------------------------
# Floating walls
#---------------------------------------------------

subsection floating walls
  set number of floating walls = 1
  subsection wall 0
    subsection point on wall
      set x = -0.45
      set y = 0
      set z = 0
    end
    subsection normal vector
      set nx = 1
      set ny = 0
      set nz = 0
    end
    set start time = 0
    set end time   = 50
  end
end

# --------------------------------------------------
# Non-Linear Solver Control
#---------------------------------------------------

subsection non-linear solver
  set solver           = inexact_newton
  set tolerance        = 1e-10
  set max iterations   = 10
  set verbosity        = verbose
end

# --------------------------------------------------
# Linear Solver Control
#---------------------------------------------------

subsection linear solver
  set method                                 = gmres
  set max iters                              = 5000
  set relative residual                      = 1e-3
  set minimum residual                       = 1e-11
  set ilu preconditioner fill                = 1
  set ilu preconditioner absolute tolerance = 1e-14
  set ilu preconditioner relative tolerance = 1.00
  set verbosity                              = verbose
end
```

Listing B.2 – Example of liquid-solid fluidized bed unresolved CFD-DEM settings file.

APPENDIX C

LETHE POST-PROCESSING PYTHON MODULE

C.1 Module source code.

```python
"""Class of methods to post-process lethe results with pyvista"""


# Import modules
import os
import shutil
import numpy as np
import pyvista as pv
from tqdm import tqdm
from operator import itemgetter

# Define class:
class lethe_pyvista_tools():

    def __init__(self, case_path = ".", prm_file_name = "", pvd_name = "", prefix = "mod_", first = 0, last = None,
                 step = 1, read_to_df = False, ignore_data = []):
        '''
        Contructor of post-processing object.

        The data follows the data models provided by PyVista.
        For further information, consult:
        https://docs.pyvista.org/user-guide/data_model.html

        The constructor parameters are:

        case_path = "."         -> Path to the case, that is, the folder with
        the prm file. By default, the present folder.

        prm_file_name = ""        -> Name of the .prm file (including ".prm")
        with simulation setup.

        The resulting objects own the following attributes:

        self.case_path          -> Returns the path to the case.

        self.prm_file           -> Returns the name of the prm_file.

        self.prm_dict.get($PARAMETER)-> Returns the parameter value given a
        string with the parameter's name.

        self.path_output        -> Returns the path to the output folder.

        pvd_name                -> Name of the .pvd file containing the
        reference to Lethe data.

        prefix                  -> Prefix of the modified vtu and pvd files. By default, "mod_". IMPORTANT!!!! If this parameter is empty, that is, "", data will be written over the original vtu and pvd files.

        first = 0               -> First time-step to be read into PyVista
        dataset.

        last  = None            -> Last time-step to be read into PyVista
        dataset.

        step  = 1               -> Step between datasets.

        read_to_df = False      -> Choose whether dataframes will be stored on
        RAM, that is, will be available on self.df list.

        self.ignore_data        -> List of data to be ignored when reading

        This method assigns the following attributes to the object:

        self.pvd_name           -> Returns the name of the .pvd file.
```

```

        self.time_list          -> Returns the list of times corresponding to
        datasets.

        self.list_vtu           -> Returns the list of names of .vtu files.

        '''

        self.path_case = case_path
        self.prm_file = prm_file_name
        self.ignore_data = ignore_data

        if not ".prm" in self.prm_file:
            self.prm_file = self.prm_file + ".prm"

        # Read .prm file to dictionary
        # Create dictionary
        self.prm_dict = {}

        # Use .prm path as argument
        with open(self.path_case + '/' + self.prm_file) as file:

            # Loop through lines in .prm
            for line in file:

                # If the line has 'subsection' in it (and it is not commented)
                if 'subsection' in line and not '#' in line:

                    # Remove "subsetction"
                    subsection_clean_line = line.replace('subsection', ' ')
                    # Clean line from spaces and assign key-value
                    subsection_clean_line = subsection_clean_line.strip()

                # Else, if the line has 'set' in it (and it is not commented)
                elif 'set' in line and not '#' in line:

                    # Remove "set" from string "line"
                    clean_line = line.replace('set', '')

                    # Split the string in [variable, value]
                    clean_line = clean_line.split('=')
```

```python

                # Clean line from spaces
                for element in range(len(clean_line)):

                    clean_line[element] = clean_line[element].strip()

                # Convert values to float when possible
                try:

                    clean_line[1] = float(clean_line[1])
                except:

                    pass

                # Define [variable, value] as key and value in the
                # dictionary
                # If 'set' is a 'Function expression' or 'type'
                if clean_line[0] == 'Function expression' or clean_line[0] == 'type':

                    # If attribute already exists, create a list
                    # Otherwise, create key-value
                    if subsection_clean_line in self.prm_dict.keys():
                        if type(self.prm_dict[subsection_clean_line]) is list:
                            self.prm_dict[subsection_clean_line].append(clean_line[1])

                        else:
                            self.prm_dict[subsection_clean_line] = [self.prm_dict[subsection_clean_line]]
                            self.prm_dict[subsection_clean_line].append(clean_line[1])

                    else:
                        self.prm_dict[subsection_clean_line] = clean_line[1]

                else:

                    # If attribute already exists, create a list
                    # Otherwise, create key-value
                    if clean_line[0] in self.prm_dict.keys():
                        if type(self.prm_dict[clean_line[0]]) is list:
                                self.prm_dict[clean_line[0]].append(clean_line[1])

                            else:
                                self.prm_dict[clean_line[0]] = [self.prm_dict[clean_line[0]]]
                                self.prm_dict[clean_line[0]].append(clean_line[1])

                        else:
                            self.prm_dict[clean_line[0]] = clean_line[1]

            print(f'Successfully constructed. To see the .prm dictionary, print($NAME.prm_dict)')

        # Define path where vtu files are
        self.path_output = self.path_case + self.prm_dict['output path'].replace('.', '')

        # Read name of files in .pvd file
        self.reader = pv.get_reader(f"{self.path_output}/{pvd_name}")

        # Create list of pvd datasets
        pvd_datasets = self.reader.datasets

        # Create a list of time-steps
        self.time_list = self.reader.time_values

        # Create a list of all files' names
        list_vtu = [pvd_datasets[x].path for x in range(len(pvd_datasets))]
        list_vtu = [x.replace(".pvtu", ".0000.vtu") for x in list_vtu]

        # Remove duplicates
        list_vtu = list(dict.fromkeys(list_vtu))

        # Select data
        if last == None:
            list_vtu = list_vtu[first::step]
            self.time_list = self.time_list[first::step]
            self.first = first
            self.step = step
            self.last = len(self.time_list) - 1
        else:
            list_vtu = list_vtu[first:last:step]
            self.time_list = self.time_list[first:last:step]
            self.first = first
            self.step = step
            self.last = last

        # List of paths among read data
        read_files_path_list = [pvd_datasets[x].path for x in range(len(pvd_datasets))]

        # Write new vtu and pvd files to store modified data.
        # IMPORTANT!!!! If this parameter is empty, that is,  "", data will be written over the original vtu and pvd files.
        with open(f'{self.path_output}/{pvd_name}') as pvd_in:
            with open(f'{self.path_output}/{prefix}{pvd_name}', 'w') as pvd_out:
                for line in pvd_in:

                    # If line refers to a dataset
                    if "vtu" in line:

                        # For all read files
                        for path in read_files_path_list:

                            # If line matches one of the files
                            if path in line:
                                line = line.replace('.pvtu', '.0000.vtu')
                                line = line.replace('file="', f'file="{prefix}')
                                pvd_out.write(line)
                                read_files_path_list.remove(path)
                                pass

                    # Write config lines
                    else:
                        pvd_out.write(line)

        # Make a copy of VTU files
        N_vtu = len(list_vtu)
        pbar = tqdm(total = N_vtu, desc="Writting modified VTU and PVD files")
        self.list_vtu = []
        for i in range(len(list_vtu)):
            # Copy file
            shutil.copy2(f'{self.path_output}/{list_vtu[i]}', f'{self.path_output}/{prefix}{list_vtu[i]}')

            # Append to list of names of VTU files
            self.list_vtu.append(f'{prefix}{list_vtu[i]}')
```

```python
            pbar.update(1)

        # Fix name of PVD file
        self.pvd_name = prefix + pvd_name

        # Create pyvista reader for files in the new .pvd file
        self.reader = pv.get_reader(f"{self.path_output}/{self.pvd_name}")

        # Create list of PVD datasets with new files
        self.pvd_datasets = self.reader.datasets

        # Boolean indicating that the dataframes are not stored in the
        # self.df object. If read_to_df = True is called, all data
        # will be stored in self.df, thus, consuming a lot of RAM.
        # Reading data into df can make the post-processing steps faster, since
        # each step will be already available in self.df. However, this
        # consumes a lot of RAM and for large simulations the tool will crash.
        # Alternatively, if read_to_df = False, all functions
        # will loop through the vtu files and flush.
        self.df_available = False

        if read_to_df:
            # Create empty array to store results
            self.df = []

            # Read VTU data
            N_vtu = len(self.list_vtu)
            pbar = tqdm(total = N_vtu, desc="Reading VTU files")
            for i in range(len(self.list_vtu)):

                # Read dataframes from VTU files into df
                self.df.append(self.get_df)
                pbar.update(1)

            self.df_available = True

            print(f'Written .df[timestep] from timestep = 0 to timestep = {len(self.list_vtu)-1}')


    # Return single pyvista dataset from list
    def get_df(self, time_step):
        # Get reader for the VTU file
        vtu_reader = pv.get_reader(f"{self.path_output}/{self.list_vtu[time_step]}")

        # Ignore selected data in order to reduce RAM usage
        for data in self.ignore_data:
            vtu_reader.disable_point_array(data)

        return vtu_reader.read()


    # Write modifications on each df to VTU files
    def write_df_to_vtu(self, prefix = "mod_"):
        ’’’
        Writes .pvd and .vtu files from data stored in self.df.
        The files are written in self.output_path.

        Parameter:

        prefix = "mod_"            -> String with prefix of the written files.
        By default, "mod_" is added in front of the regular files.
        ’’’

        # List of paths among read data
        read_files_path_list = [self.pvd_datasets[x].path for x in range(len(self.pvd_datasets))]

        # Write modified PVD to match new VTU files
        with open(f’{self.path_output}/{self.pvd_name}’) as pvd_in:
            with open(f’{self.path_output}/{prefix}{self.pvd_name}’, ’w’) as pvd_out:
                for line in pvd_in:

                    # If line refers to a dataset
                    if "vtu" in line:

                        # For all read files
                        for path in read_files_path_list:

                            # If line matches one of the files
                            if path in line:
                                line = line.replace(’.pvtu’, ’.0000.vtu’)
                                line = line.replace(’file="’, f’file="{prefix}’)
                                pvd_out.write(line)
                                read_files_path_list.remove(path)
                                pass

                    # Write config lines
                    else:
                        pvd_out.write(line)


        if self.df_available:
            # Write modified VTU file
            N_vtu = len(self.df)
            pbar = tqdm(total = N_vtu, desc="Writting new VTU and PVD files")
            for i in range(len(self.df)):
                self.df[i].save(f'{self.path_output}/{prefix}{self.list_vtu[i]}')
                pbar.update(1)

            print(f"Modified .vtu and .pvd files with prefix {prefix} successfully written")

        else:
            print(f"No df available for writing. Try to use read_lethe_to_pyvista first")

    # Sort all data given reference array
    def sort_by_array(self, reference_array_name = "ID"):
        '''
        Sorts all self.df according to a reference array:

        Parameter:

        reference_array_name = "ID" -> String with name of reference array.
        "ID" is used as default for particles, but any other 1D array can be
        used.
        '''

        pbar = tqdm(total = len(self.time_list), desc = f"Sorting dataframe by {reference_array_name}")

        if self.df_available:
            for i in range(len(self.time_list)):
                self.df[i].points = self.df[i].points[self.df[i][reference_array_name].argsort()]
                for name in self.df[0].array_names:
                    self.df[i][name] = self.df[i][name][self.df[i][reference_array_name].argsort()]
                pbar.update(1)

        else:
            for i in range(len(self.list_vtu)):
                df = self.get_df(i)
                df.points = df.points[df[reference_array_name].argsort()]
                for name in df.array_names:
                    df[name] = df[name][df[reference_array_name].argsort()]

                df.save(f'{self.path_output}/{self.list_vtu[i]}')
                pbar.update(1)

    # Creates or modifies array
    def modify_array(self, reference_array_name = "ID", array_name = "new_array", restart_array = False,  condition = "", array_values = 0, standard_value = 0, reference_time_step = 0, time_dependent = False):
        '''
        Creates or modifies array

        Parameters are:

        reference_array_name = "ID"        -> array to be used as reference to
        create or modify the other. All arrays will be sorted and written according to this one.

        array_name           = "new_array" -> name of the new array. If there is an
        array with the same name, it will be rewritten according to the other
        arguments.

        restart_array        = False       -> if True, zeroes the entire array before
        modifying it. If you want to modify part of the array keeping the rest
        intact, set it to False

        condition            = ""          -> takes a string and uses it in an if
        condition to modify the array. Variables accepted include x, y, z, u, v, w,
        t, and any other array.It also accepts a combination of them, such as:
        "x*w**2 + t > 2"

        array_values         = 0           -> new values to the array. This argument
        accepts a single value (which will be repeated to all data respecting the
        given condition), an numpy array or python list (with the same len of all
        other arrays), or a string such as "2*x + t" (working just like the condition
        argument)

        standard_value       = 0           -> if restart array is True, the
        standard_value will be the one plugged to the entire array before modifying
        it.

        reference_time_step  = 0           -> reference time step to which the
        modification will be applied. The others will follow this given one.

        time_dependent       = False       -> the modifier can be time dependent or
        not. If set True, the condition will be tested to each of the time-steps,
        while if False, it will be applied using the reference_time_step instead, and
        the modification will be just replicated to the other time steps
        '''

        print("Generating array based on condition and array_value")

        # Sort all data by reference_array_name
        print(f"Sort array by {reference_array_name}")
        self.sort_by_array(reference_array_name)

        # Time array
        t = self.time_list

        # Create list of array names
        # This step is necessary to allow the usage of
        # the variables x, y, z, u, v, w, t, f_x, f_y, and f_z
        # in the condition argument
        if self.df_available:
            df = self.df[0]
        else:
            df = self.get_df(0)
        array_names = df.array_names
        array_names.append("x")
        array_names.append("y")
        array_names.append("z")
        array_names.append("u")
        array_names.append("v")
        array_names.append("w")
        array_names.append("f_x")
        array_names.append("f_y")
        array_names.append("f_z")
        array_names.append("t")

        # Restart array if asked or if array does not exist
        # If the array is restarted or created, the standard_value will be
        # assigned to the entire array.
        # If restart_array is set to False and the array exists,
        # The previous values in it will be preserved.
        # This can be used to apply multiple conditions without affecting
        # Previous modifications, for example.
        if restart_array == True or array_name not in df.array_names:
            # Create array if does not exist
            new_array = np.repeat(standard_value, len(df[reference_array_name]))
            print(f"Creating array ’{array_name}’ with standard_value {standard_value}")

            # Push array to all pyvista arrays
            pbar = tqdm(total = len(self.list_vtu), desc = f"Creating array: {array_name}")
            for i in range(len(self.list_vtu)):
                if self.df_available:
                    self.df[i][array_name] = np.repeat(standard_value, len(self.df[i][reference_array_name]))

                else:
                    df = self.get_df(i)
                    df[array_name] = np.repeat(standard_value, len(df[reference_array_name]))
                    df.save(f’{self.path_output}/{self.list_vtu[i]}’)

                pbar.update(1)
```

```python

        else:
            if self.df_available:
                # Reading array from reference timestep
                print("Reading previous array")
                new_array = self.df[reference_time_step][array_name]

            else:
                print("Reading previous array")
                df_reference = self.get_df(reference_time_step)
                new_array = df_reference[array_name]


        # Create a list of array names that are used either in
        # "conditions" or in "array_values"
        new_variables = set([])

        # Prepare "condition" and "array_value" for elementwise loop
        # Note that "k" is used here because it is the specific counter
        # that will be used for testing the "condition" further
        for name in array_names:
            if name in condition:
                condition = condition.replace(name, name + "[k]")

                # If one of the variables used in "condition"
                # is a pyvista array, create a list with the
                # name of the variable for further manipulation
                if name in df.array_names:
                    if self.df_available:
                        exec(f"global {name}; {name} = self.df[reference_time_step][name]")
                    else:
                        df_reference = self.get_df(reference_time_step)
                        exec(f"global {name}; {name} = df_reference[name]")
                    new_variables.add(name)

        if type(array_values) == type(str()):
            for name in array_names:
                if name in array_values:
                    array_values = array_values.replace(name, name + "[k]")

                    # If one of the variable used in "array_value"
                    # is a pyvista array, create a list with the
                    # name of the variable for further manipulation
                    if name in df.array_names:
                        if self.df_available:
                            exec(f"global {name}; {name} = self.df[reference_time_step][name]")
                        else:
                            df_reference = self.get_df(reference_time_step)
                            exec(f"global {name}; {name} = df_reference[name]")
                        new_variables.add(name)

        # If results vary with time,
        # the condition and array_values will be applied
        # to all time steps
        if time_dependent:
            ("Creating time-dependent array:")
            pbar = tqdm(total = len(self.time_list), desc = f"Looping through time-steps")
            for i in range(len(self.list_vtu)):
                # Assign velocities and positions to variables using the ith
                # time step
                if self.df_available:
                    df = self.df[i]

                else:
                    df = self.get_df(i)

                exec(f"global x; x = df.points[:, 0]")
                exec(f'global y; y = df.points[:, 1]')
                exec(f'global z; z = df.points[:, 2]')


                # In case velocity is written with caps V or v
                if "velocity" in array_names:
                    exec(f'global u; u = df["velocity"][:, 0]')
                    exec(f'global v; v = df["velocity"][:, 1]')
                    exec(f'global w; w = df["velocity"][:, 2]')


                elif "Velocity" in array_names:
                    exec(f'global u; u = df["Velocity"][:, 0]')
                    exec(f'global v; v = df["Velocity"][:, 1]')
                    exec(f'global w; w = df["Velocity"][:, 2]')

                # In case of FemForce or fem_force
                if "FemForce" in array_names:
                    exec(f'global f_x; f_x = df["FemForce"][:, 0]')
                    exec(f'global f_y; f_y = df["FemForce"][:, 1]')
                    exec(f'global f_z; f_z = df["FemForce"][:, 2]')

                elif "fem_force" in array_names:
                    exec(f'global f_x; f_x = df["fem_force"][:, 0]')
                    exec(f'global f_y; f_y = df["fem_force"][:, 1]')
                    exec(f'global f_z; f_z = df["fem_force"][:, 2]')

                # In case of fem_torque
                if "fem_torque" in array_names:
                    exec(f'global t_x; t_x = df["fem_torque"][:, 0]')
                    exec(f'global t_y; t_y = df["fem_torque"][:, 1]')
                    exec(f'global t_z; t_z = df["fem_torque"][:, 2]')

                # Update lists used either in "condition" or "array_value":
                for variable in new_variables:
                    exec(f"{variable} = df[variable]")

                # Reading array from reference timestep
                new_array = df[array_name]

                # Fill new_array with array_value
                for k in range(len(new_array)):
                    if eval(condition):
                        if type (array_values) == type(int(1)):
                            new_array[k] = array_values
                        elif type(array_values) == type(np.array([])) or type(array_values) == type([]):
                            new_array[k] = array_values[k]
                        else:
                            new_array[k] = eval(array_values)

                # Assign new_array to pyvista dataframe
                df[array_name] = new_array

                if self.df_available:
                    self.df[i] = df
                else:
                    df.save(f'{self.path_output}/{self.list_vtu[i]}')

                pbar.update(1)

        # If not time dependent, the condition and array_values will be applied
        # at the reference_time_step.
        # This is very useful if you want to track a group of particles in a
        # DEM simulation. The value of 1 can be assigned to this group of
        # particles
        # according to a given condition (position, for example) while 0 can be
        # given to the others,
        # This way, the groups of particles will be colored in the
        # reference_time_step and will keep this color regardless of time.
        else:
            print(f"Creating array based on time-step number: {reference_time_step}")
            print(f"Corresponding time: {self.time_list[reference_time_step]}")

            if self.df_available:
                df_reference = self.df[reference_time_step]

            else:
                df_reference = self.get_df(reference_time_step)

            exec(f'global x; x = df_reference.points[:, 0]')
            exec(f'global y; y = df_reference.points[:, 1]')
            exec(f'global z; z = df_reference.points[:, 2]')


            # In case velocity is written with caps V or v
            if "velocity" in array_names:
                exec(f'global u; u = df_reference["velocity"][:, 0]')
                exec(f'global v; v = df_reference["velocity"][:, 1]')
                exec(f'global w; w = df_reference["velocity"][:, 2]')


            elif "Velocity" in array_names:
                exec(f'global u; u = df_reference["Velocity"][:, 0]')
                exec(f'global v; v = df_reference["Velocity"][:, 1]')
                exec(f'global w; w = df_reference["Velocity"][:, 2]')

            # In case of FemForce or fem_force
            if "FemForce" in array_names:
                exec(f'global f_x; f_x = df_reference["FemForce"][:, 0]')
                exec(f'global f_y; f_y = df_reference["FemForce"][:, 1]')
                exec(f'global f_z; f_z = df_reference["FemForce"][:, 2]'
   )

            elif "fem_force" in array_names:
                exec(f'global f_x; f_x = df_reference["fem_force"][:, 0]
   ')
                exec(f'global f_y; f_y = df_reference["fem_force"][:, 1]
   ')
                exec(f'global f_z; f_z = df_reference["fem_force"][:, 2]
   ')

            # In case of fem_torque
            if "fem_torque" in array_names:
                exec(f'global t_x; t_x = df_reference["fem_torque"][:,
   0]')
                exec(f'global t_y; t_y = df_reference["fem_torque"][:,
   1]')
                exec(f'global t_z; t_z = df_reference["fem_torque"][:,
   2]')

            # Fill new_array with array_value
            print(f"Creating new array named: {array_name}")
            for k in range(len(new_array)):
                if eval(condition):
                    if type (array_values) == type(int(1)):
                        new_array[k] = array_values
                    elif type(array_values) == type(np.array([])) or
   type(array_values) == type([]):
                        new_array[k] = array_values[k]
                    else:
                        new_array[k] = eval(array_values)

            # Assign new_array to pyvista dataframe
            if self.df_available:
                self.df[reference_time_step][array_name] = new_array
            else:
                df_reference[array_name] = new_array
                df_reference.save(f'{self.path_output}/{self.list_vtu[
   reference_time_step]}')

            # Create dictionary (map) based on reference_array
            reference_time_step_dict = dict(zip(df_reference[
   reference_array_name], df_reference[array_name]))

            key_list = df_reference[reference_array_name]

            # Use the same values for all time steps
```

```
            # Note that "reference_array_name" is used as criterium here
            # for sorting purposes, and that it can be changed
            # according to the user by changin the parameter
            # "reference_array_name" to any other array name in the original
            # pyvista arrays
            pbar = tqdm(total = len(self.list_vtu), desc = f"Assigning {array_name} to dataframes")
            for i in range(len(self.list_vtu)):
                # Find elements in common in current and reference arrays
                if self.df_available:
                    df = self.df[i]
                else:
                    df = self.get_df(i)

                keys, indices, _ = np.intersect1d(df[reference_array_name], key_list, assume_unique = True, return_indices = True)

                if self.df_available:
                    self.df[i][array_name][indices] = itemgetter(*keys)(reference_time_step_dict)
                else:
                    df[array_name][indices] = itemgetter(*keys)(reference_time_step_dict)
                    df.save(f'{self.path_output}/{self.list_vtu[i]}')


                pbar.update(1)

    # Get cylindrical coordinates of each point of all dataframes
    def get_cylindrical_coords(self, radial_components = "yz"):
        '''
        Get cylindrical coordinates of points in self.df datasets

        Parameter:
        radial_components = "yz"         -> Cartesian directions of radial
        component.

        This method assigns the following attribute to the object:

        self.df[$TIME-STEP]['points_cyl'] -> Returns a .points like array with all
        points in cylindrical [radius, theta, height].
        '''
```

```python

        # List of indices of radial components
        radial_indices = []

        # Add indices according to parameter radial_components
        if "x" in radial_components:
            radial_indices.append(0)
        if "y" in radial_components:
            radial_indices.append(1)
        if "z" in radial_components:
            radial_indices.append(2)

        # Kill process if radial_components have more or less than 2 coords
        if len(radial_components) != 2:
            print(f"radial_components has {len(radial_components)} axis")
            exit()

        # Find index other than the radial components
        z_index = [x for x in [0, 1, 2] if x not in radial_indices]

        # Loop through data
        pbar = tqdm(total = len(self.list_vtu), desc = "Getting cylindrical coords")
        for i in range(len(self.list_vtu)):

            if self.df_available:
                df = self.df[i]
            else:
                df = self.get_df(i)

            # Get cartesian position
            cartesian = df.points

            # Calculate radial coord
            radius = np.sqrt(cartesian[:, radial_indices[0]]**2 + cartesian[:, radial_indices[1]]**2)

            # Calculate theta
            theta = np.arctan2(cartesian[:, radial_indices[1]], cartesian[:, radial_indices[0]])

            # Get z
            z = cartesian[:, z_index].flatten()

            # Store coordinates into points_cyl (same shape as .points)
            if self.df_available:
                self.df[i]['points_cyl'] = np.empty(df.points.shape)
                self.df[i]['points_cyl'][:, 0] = radius.tolist()
                self.df[i]['points_cyl'][:, 1] = theta
                self.df[i]['points_cyl'][:, 2] = z
            else:
                df['points_cyl'] = np.empty(df.points.shape)
                df['points_cyl'][:, 0] = radius.tolist()
                df['points_cyl'][:, 1] = theta
                df['points_cyl'][:, 2] = z
                df.save(f'{self.path_output}/{self.list_vtu[i]}')

            pbar.update(1)


    # Get neighbors of points
    def get_nearest_neighbors(self, return_id = True, n_neighbors = 15):
        '''
        Get indices, distances, and "ID" (if requested) of nearest neighbors of
        each point in self.df.

        Parameters:
        return_id = False          -> Decide whether ID is returned or not. If
        True, but self.df does not have "ID", attribute neighbors_id is not
        assigned.

        This method assigns the following attributes to the object:

        self.df[$TIME-STEP].neighbors        -> Returns a lists with
        indices of neighbors per point in dataset.

        self.df[$TIME-STEP].neighbors_dist   -> Returns a list with distances
        between neighbor points per point in dataset.

        self.df[$TIME-STEP].neighbors_id     -> Returns a list with "ID"
        of neighbor points per point in dataset.

        !!!IMPORTANT!!!

        This method uses KDTree to find neighbors.
        Details:
        https://scikit-learn.org/stable/modules/generated/sklearn.neighbors.KDTree.html

        If library sklearn is missing, the method will not work.
        To install sklearn, run the following line in your terminal:
        $ pip install scikit-learn
        or
        $ pip3 install scikit-learn
        At the moment of the implementation, sklearn version was: 1.2.1
        '''

        # Import KDTree library
        from sklearn.neighbors import KDTree

        # Loop through dataframes to search for neighbors
        pbar = tqdm(total = len(self.list_vtu), desc = "Finding neighbors")
        for i in range(len(self.list_vtu)):

            if self.df_available:
                df = self.df[i]
            else:
                df = self.get_df(i)

            # Create a tree from points
            tree = KDTree(df.points)

            # Get the distance and the indices of the n_neighbors neighbors
            # It is important to note that the closest neighbor is going to
            # be the point itself, so we ask for n_neighbors + 1
            dist, indices = tree.query(df.points, k = n_neighbors+1)

            # Remove itself from indices and dist for all points
            indices = indices[:, 1:]
            dist = dist[:, 1:]

            # Add neighbors_id, neighbors indices, and neighbors distances
            # to each dataframe
            if self.df_available:
                if return_id and hasattr(df, "ID"):
                    self.df[i]["neighbors_id"] = self.df[i]["ID"][indices]
                self.df[i]["neighbors"] = indices
                self.df[i]["neighbors_dist"] = dist
            else:
                if return_id and hasattr(df, "ID"):
                    df["neighbors_id"] = df["ID"][indices]
                df["neighbors"] = indices
                df["neighbors_dist"] = dist
                df.save(f'{self.path_output}/{self.list_vtu[i]}')

            pbar.update(1)


    def mixing_index_nearest_neighbors(self, n_neighbors = 15, reference_array = "particle_color", mixing_index_array_name = "mixing_index"):
        '''
        Calculates mixing index per time-step using the Nearest
        Neighbors Method (NNM) by Godlieb et al. (2007).
        # Godlieb, W., N. G. Deen, and J. A. M. Kuipers. "Characterizing solids
        mixing in DEM simulations." cell 1 (2007).

        Parameters:

        n_neighbors = 15                        -> Number of neighbors to
        account for in the calculation of the mixing index.

        reference_array = "particle_color"      -> Name of the array containing
        the particle's type, that is, which group the particle is part of. For
        a better understanding, check the documentation of the modify_array
        method.

        mixing_index_array_name = "mixing_index"-> Name of the array assigned
        to each self.df. This array can be used to see the mixing index per
        point in the dataset (particle).

        This method assigns the following attributes to the object:

        self.mixing_index       -> Average mixing index per time-step.

        self.mixing_index_std   -> Standard deviation of the mixing index per
        time-step.

        self.df[$TIME-STEP]     -> Assign array to dataset named according to
        mixing_index_array_name. This array can be used in visual postprocessing
        softwares, such as ParaView. Check the write_vtu method of this module.
        '''

        # Apply NNM by Godlieb et al. (2007)
        # Godlieb, W., N. G. Deen, and J. A. M. Kuipers.
        # "Characterizing solids mixing in DEM simulations." cell 1 (2007).

        # If neighbors is not an attribute of the dataframe

        if self.df_available:
            df = self.df[0]
        else:
            df = self.get_df(0)

        if hasattr(df, "neighbors") == False or len(df['neighbors'][0]) != n_neighbors:
            self.get_nearest_neighbors(n_neighbors = n_neighbors)

        # Create empty list to store mixing_index per time-step
        self.mixing_index = []
        self.mixing_index_std = []

        # Loop through dataframes and find its mixing index
        pbar = tqdm(total = len(self.list_vtu), desc = "Calculating mixing index")
        for i in range(len(self.list_vtu)):

            if self.df_available:
                df = self.df[i]
            else:
                df = self.get_df(i)

            # Find particles with different values for the reference array per
            # particle
            list_neighbor_reference_array = df[reference_array][df['neighbors']]
            n_equal_neighbors_per_particle = np.sum(np.equal(df[reference_array][:, None], list_neighbor_reference_array), axis = 1)

            # Calculate mixing index per particle
            mixing_index_per_particle = 2*(1-(1/n_neighbors) * n_equal_neighbors_per_particle)

            # Create array of mixing index per particle
            if self.df_available:
                self.df[i][mixing_index_array_name] = mixing_index_per_particle
            else:
                df[mixing_index_array_name] = mixing_index_per_particle
                df.save(f'{self.path_output}/{self.list_vtu[i]}')

            mixing_index = np.mean(mixing_index_per_particle)
            mixing_index_std = np.std(mixing_index_per_particle)

            # Store mixing index
            self.mixing_index.append(mixing_index)
            self.mixing_index_std.append(mixing_index_std)
            pbar.update(1)

    def mixing_index_doucet(self, reference_time_step = 0, use_cyl = False, increasing_index = False, normalize = True):
        '''
        Calculates mixing index per time-step using the method by
        Doucet et al. (2008).
        J. Doucet, F. Bertrand, J. Chaouki. "A measure of mixing from
        Lagrangian tracking and its application to granular and fluid flow
        systems." Chemical Engineering Research and Design 86.12 (2008):
        1313-1321.

        Parameters:

        reference_time_step = 0      -> Time-step used as reference to
        calculate mixing index.

        use_cyl = False              -> Choose whether to use cylindrical or
        cartesian coordinates. If use_cyl = True, .point_cyl will be used
        (check get_cylindrical_coords method). Otherwise cartesian . points are
        used.

        increasing_index = False     -> Choose whether the mixing index is
        increasing or decreasing with mixing. Doucet et al. (2008) uses a
        decreasing mixing index, however, most mixing indices increase with
        mixing.

        normalize = False           -> Choose whether the mixing index is
        normalized according to the mixing index of the reference_time_step.

        This method assigns the following attributes to the object:

        self.mixing_index           -> Normalized Doucet mixing index per
        time-step. The normalization is done using the mixing index at
        reference_time_step.

        self.mixing_eigenvector     -> Eigenvector associated to the
        mixing index.
        '''

        from scipy.linalg import eigh
        # Apply method by  J. Doucet, F. Bertrand, J. Chaouki.
        # "A measure of mixing from Lagrangian tracking and its application to
        # granular and fluid flow systems." Chemical Engineering Research and
        # Design 86.12 (2008): 1313-1321.

        # Get cylindrical coordinates if requested and not previously
        # calculated

        if self.df_available:
            df = self.df[reference_time_step]
        else:
            df = self.get_df(reference_time_step)

        if use_cyl and hasattr(df, "points_cyl") == False:
            self.get_cylindrical_coords()

            if self.df_available == False:
                df = self.get_df(reference_time_step)

        # If cylindrical coordinates requested, assign points_cyl to reference
        # position, otherwise use cartesian
        if use_cyl:
            reference_position = df['points_cyl']
```

```python
        else:
            reference_position = df.points

        # Get position of particles corresponding IDs
        id_keys = df["ID"]


        # Create list of mixing indices per time-step and array of eigenvectors
        self.mixing_index = []
        self.mixing_eigenvector = np.empty((len(self.list_vtu), 3))

        # Loop through dataframes and find its mixing index
        pbar = tqdm(total = len(self.list_vtu), desc = "Calculating mixing index")
        for i in range(len(self.list_vtu)):

            if self.df_available:
                df = self.df[i]
            else:
                df = self.get_df(i)

            # If cylindrical coordinates requested, assign points_cyl to current
            # position, otherwise use cartesian
            if use_cyl:
                i_position = df[’points_cyl’]
            else:
                i_position = df.points

            # Find indices of particles in different time-steps
            _, indices_i, indices_ref = np.intersect1d(df["ID"], id_keys, assume_unique = True, return_indices = True)

            # Calculate correlation matrix
            correlation_matrix = np.corrcoef(i_position[indices_i], reference_position[indices_ref], rowvar=False)[3:, :3]

            # Transpose matrix
            correlation_matrix_transpose = correlation_matrix.T

            # Multiply correlation and transposed correlation matrices
            M = np.matmul(correlation_matrix, correlation_matrix_transpose)

            # Find maximum eigenvalue and the eigenvector associated to it
            max_eigenvalue, assoc_eigenvectors = eigh(M, subset_by_index=[2, 2])
            max_eigenvalue = max_eigenvalue[0]

            # Store reference eigenvalue for further normalization
            if i == reference_time_step:
                max_eigenvalue_reference = max_eigenvalue

            # Store mixing index and associated eigenvector
            self.mixing_index.append(max_eigenvalue)
            self.mixing_eigenvector[i] = assoc_eigenvectors.flatten()
            pbar.update(1)

        # Normalize index
        if normalize:
            self.mixing_index = np.divide(self.mixing_index, max_eigenvalue_reference)

        # Use increasing instead of decreasing index
        if increasing_index:
            self.mixing_index = 1 - self.mixing_index
```

Listing C.1 – Module source code.

C.2 Example of use of the module on fluid results.

```
# To use the lethe_pyvista_tools, you need to have python 3
# installed in your computer

# The modules necessary to run lethe pyvista tools are:
# Pandas: pip install pandas
# PyVista: pip install pyvista

# To use lethe_pyvista_tools, add /contrib/postprocessing to your PythonPATH
# adding
# ‘‘export PYTHONPATH="${PYTHONPATH}:$LETHE_PATH/contrib/postprocessing"‘‘
# to your ~/.bashrc, or append the path to the /contrib/postprocessing
# folder in Lethe to sys

import sys
# Path to the module
path_to_module = '$LETHE_PATH/contrib/postprocessing/'
```

```
sys.path.append(path_to_module)

# or even put the "lethe_pyvista_tools.py" file inside
# the same directory as your python script and proceed as follows

# This line imports all lethe_pyvista_tools functionalities
from lethe_pyvista_tools import *

'''Parameters'''

figure = False #True if plot
plot_P_t = False
write_to_excel = True

'''Importing Libraries'''
from math import pi
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from sklearn import linear_model
from tqdm import tqdm

import os

'''Simulation properties'''

#Take case path as argument and store at case_path
case_path = sys.argv[1]
saveFigDir = case_path


fluid = lethe_pyvista_tools(case_path = case_path, prm_file_name = "liquid-solid-fluidized-bed.prm", pvd_name = "cfd_dem.pvd", step = 100)

particle = lethe_pyvista_tools(case_path = case_path, prm_file_name = "liquid-solid-fluidized-bed.prm", pvd_name = "cfd_dem_particles.pvd", step = 100)

dp = particle.prm_dict['diameter']
rp = dp/2
Vp = 4/3*pi*rp**3
rhop = particle.prm_dict['density particles']
Np = particle.prm_dict['number'][1]
inlet_velocity = fluid.prm_dict['u']


```

```
g = abs(particle.prm_dict[’gx’])       #m/s^2
Vnp = Np * 4/3 * pi * rp**3

rhol = fluid.prm_dict[’density’]  #kg/m^3
nu = fluid.prm_dict[’kinematic viscosity’] #m^2/s
mu = rhol*nu    #Pa
Db = 0.1        #m
Rb = Db/2       #m
Area = pi*Rb**2 #m^2
Hb = 1.1

’’’Functions’’’
def Analytical(Np, rp, rhol, rhop, g, Area):
    Mp = Np*4/3*pi*rp**3*rhop
    delta_p = Mp*(rhop - rhol)*g/(rhop * Area)
    return delta_p

#Define eps_list and voidfraction_list to append value for each time-step
eps_list = []
voidfraction_list = []
total_deltaP = []

#Create a list with all x values for the pressure takes
x_pressure_takes = np.arange(-0.45, Hb/2 + 0.01, 0.01)

deltaP_analytical = Analytical(Np, rp, rhol, rhop, g, Area)

if os.path.isdir(case_path + ’/P_x’) == False:
    os.mkdir(case_path + ’/P_x’)

pbar = tqdm(total = len(fluid.time_list)-1, desc="Processing data")
for i in range(len(fluid.time_list)):
    #Define "df" as last time step df
    df = fluid.get_df(i)

    #Take the first slice of the domain at pos0
    pos0 = [-0.45, 0, 0]
    slice0 = df.slice(normal=[1, 0, 0], origin = pos0)
    p0 = slice0.integrate_data()[’pressure’]/slice0.area

    #Create empty lists to fill with x values and pressure as function of x values
    p_x = []
    x_list = []

    #Loop through z values above z0
```

```python
    for j in range(len(x_pressure_takes)):
        x_list.append(x_pressure_takes[j]-pos0[0])
        slice_x = df.slice(normal=[1, 0, 0], origin = [x_pressure_takes[j], 0, 0])
        p_x.append((p0 - slice_x.integrate_data()['pressure']/slice_x.area)[0]*rhol)

    #Store the total pressure drop
    total_deltaP.append(p_x[-1])

    #Apply least-squares to find the porosity
    regr = linear_model.LinearRegression() #fit_intercept = 0
    x_list = np.array(x_list).reshape(-1, 1)
    p_x = np.array(p_x).reshape(-1, 1)
    model = regr.fit(x_list, p_x)
    r_sq = model.score(x_list, p_x)

    #Find linear portion of the graph and adjust a linear function to it
    x = x_list
    y = p_x

    for j in range(len(y)-1):
        if y[j] >= deltaP_analytical*0.9:
                x = x[:j]
                y = y[:j]
                break

    model = regr.fit(x, y)

    #Calculate bed voidage by the slope of the pressure drop curve
    eps = 1-(model.coef_[0][0]/((rhop-rhol)*g))
    eps_list.append(eps)


    fig1 = plt.figure()
    ax1 = fig1.add_subplot(111)

    fig1.suptitle(f'Time = {fluid.time_list[i]} s')
    ax1.plot(x_list, p_x, 'ok', ms = 5, label ='Simulation')
    ax1.plot(x_list, np.repeat(deltaP_analytical, len(x_list)), '.r', ms = 2, label = 'Analytical')
    ax1.plot(x, y, '.g')
    ax1.plot(x, model.predict(x), '-b')
    ax1.set_ylabel(r'$-\Delta p \/\ [Pa]$')
    ax1.set_xlabel(r'$Height \/\ [m]$')
    ax1.set_xlim(0, 1.02)
    ax1.set_ylim(0, deltaP_analytical*1.30)
    ax1.legend()
    ax1.annotate(r'$\varepsilon (-dp/dz) = {:1.2}$'.format(eps), (x[round(len(x)/2)], y[round(len(y)/2)]), xytext=(0.65, 0.4), textcoords='axes fraction', arrowprops=dict(facecolor='black', shrink=0.04), fontsize=14, horizontalalignment='right', verticalalignment='top')
    fig1.savefig(f'{saveFigDir}/P_x/P_x-{i}.png')
    plt.close(fig1)

    pbar.update(1)

#Export the total pressure results as a function of time
csv = pd.DataFrame([fluid.time_list, total_deltaP], index=['time', 'deltaP']).transpose()
csv.to_csv(f'{saveFigDir}/deltaP_t.csv')

print(f'Average - delta p = {np.mean(total_deltaP)} Pa')


#Plot void fraction with time
fig0, ax0 = plt.subplots(1, 1)

ax0.plot(fluid.time_list, eps_list, 'ok', ms = 5)
ax0.legend()
ax0.grid()
#ax0.set_xlim(0, 35)
ax0.set_ylim(0, 1)
ax0.set_ylabel(r'$Bed \/\ voidage \/\ [-]$')
ax0.set_xlabel(r'$Time \/\ [s]$')
fig0.savefig(f'{saveFigDir}/eps_t.png')
plt.close(fig0)

eps_ave = np.average(eps_list[-10:])
eps_std = np.std(eps_list[-10:])

print('Average void fraction among the last 10 time steps: ' + str(eps_ave))
```

Listing C.2 – Example of use of the module on fluid results.

C.3 Example of use of the module on particle results

```
#Post-processing tool to track mixing in small rotating drum

# import sys
```

```python
import sys

# Path to the module
path_to_module = '../../../contrib/postprocessing/'
sys.path.append(path_to_module)

# Import tools from module
from lethe_pyvista_tools import *
import matplotlib.pyplot as plt


# Create object named "particles"
particles = lethe_pyvista_tools(".", "small-rotating-drum-dem.prm", "out.pvd")


# State condition for particle_color array creation
condition = "(y**2 + z**2)**(1/2) > 0.025"

# Create array named particle_color and give the value of 1 to all particles
# within the given condition
# The packing of particles is finished after 40th output time step
# So, it is important to set reference_time_step to 40
particles.modify_array(array_name = "particle_color", condition = condition, array_values = 1, restart_array = True, reference_time_step = 40)


# State a second condition to modify particle_color
condition = "(y**2 + z**2)**(1/2) > 0.04"

# Modify the array to give the value of 2 to all particles within the new
# condition
particles.modify_array(array_name = "particle_color", condition = condition, array_values = 2, restart_array = False, reference_time_step = 40)


# VISUALIZE DATA USING PYVISTA:

# Create a sphere with diameter 1
sphere = pv.Sphere(theta_resolution=50, phi_resolution=50)

# Use sphere as basis to create sheric representation of particles
```

```
particle_glyph = particles.get_df(40).glyph(scale=’diameter’, geom=sphere)

# Create a plotter object
plotter = pv.Plotter()
# Add particles to object and color them by particle_color array
plotter.add_mesh(particle_glyph, scalars = "particle_color")
# Open plot window
plotter.show()


# MIXING INDEX:

# To measure the mixing index using the nearest neighbors technique by
# Godlieb et al. (2007), the domain must be split in half. Thus:

# Find center of mass radial position
# To do so, first we will need the cylindric coods of each particle
# Note that the radial components of our cylinder are y and z.
particles.get_cylindrical_coords(radial_components = "yz")

# Since all particles are of the same size (radius = coord 0)
r_center_mass = np.mean(particles.get_df(40)["points_cyl"][:, 0])

# Split domain in half (restarting array)
condition = f"(y**2 + z**2)**(1/2) > {r_center_mass}"
particles.modify_array(array_name = "particle_color", condition = condition, array_values = 1, restart_array = True, reference_time_step = 40)

# Get list of nearest neighbors of each particle
particles.get_nearest_neighbors(return_id = True, n_neighbors = 15)

# Print position of nearest neighbors:
print(f"\nPosition of nearest neighbor of particle 2 at time-step 5 = {particles.get_df(5).points[particles.get_df(5)[’neighbors’][2][0]]}\n")

# Calculate mixing index using nearest neighbors technique by
# Godlieb et al. (2007)
particles.mixing_index_nearest_neighbors(reference_array = "particle_color", n_neighbors = 15, mixing_index_array_name = "mixing_index_NNM")

# Store mixing index in a variable to compare with Doucet et al. (2008)
particles.mixing_index_nnm = particles.mixing_index

```

```python
# Calculate mixing index using Doucet et al. (2008)

# Since the problem is cylindrical, we will set use_cyl = True
# Also, since Doucet does not use the color of particles, we need
# to set reference_time_step = 40
# Additionally, by default, Doucet mixing index decreases with mixing,
# but, to compare with NNM, we need to use and increasing index.
# Finally, to avoid values above 1, we normalize it using the mixing index
# of the 40th time-step seting normalize = True.
particles.mixing_index_doucet(reference_time_step = 40, use_cyl = True, increasing_index = True, normalize = True)

# Store the second index:
particles.mixing_index_doucet = particles.mixing_index

# Save plot with mixing index as a function of time
# Note that the mixing index will only make sense for
# a time-steps greater than the one colored.
# In this case, we will plot only from the 40th time-step on.
plt.plot(particles.time_list[40:], particles.mixing_index_nnm[40:], ’-b’, label = "NNM")
plt.plot(particles.time_list[40:], particles.mixing_index_doucet[40:], ’--k’, label = "Doucet")
plt.plot(particles.time_list[40:], np.repeat(1, len(particles.time_list[40:])), ’:r’)
plt.xlabel("Time [s]")
plt.ylabel("Mixing index [-]")
plt.xlim(particles.time_list[40], particles.time_list[-1])
plt.ylim(0, 1.1)
plt.legend()
plt.savefig("./mixing_index.png")
plt.close()
```

Listing C.3 – Example of use of the module on particle results.

APPENDIX D

SENSIBILIDADE PARAMÉTRICA NO CÁLCULO DA EXPANSÃO DE UM LEITO FLUIDIZADO LÍQUIDO

Daniel Silva Junior, Victor Oliveira Ferreira, Gabriela Cantarelli Lopes.

Presented by the author of this thesis at the conference entitled "XL Congresso Brasileiro de Sistemas Particulados" (ENEMP), in Uberlândia-MG, Brazil. October, 2020.

D.1 Resumo

Neste trabalho, analisou-se a sensibilidade do cálculo da porosidade (ε) de um leito fluidizado sólido-líquido a imprecisões nas características do líquido e das partículas. Quatro grupos de partículas, de três materiais diferentes (ABS, Alumina e Porcelana), foram caracterizados quanto a: diâmetro (d_p), densidade (ρ_p) e velocidade terminal (U_0). Essas partículas foram submetidas à fluidização em diferentes vazões, com temperatura do líquido (T) monitorada, e foi obtido o coeficiente angular da pressão em função da altura ($\frac{\mathrm{d}P}{\mathrm{d}z}$). A porosidade foi calculada por dois métodos: utilizando a equação de Richardson-Zaki, e utilizando $\frac{\mathrm{d}P}{\mathrm{d}z}$. Foram obtidos os valores de porosidade com os valores médios das variáveis de entrada (ρ_p, T, d_p, U_0), e em seguida foram repetidos os cálculos utilizando valores-limite dados por média ± incerteza ou variação. A porosidade avaliada nestes limites foi comparada à porosidade obtida com os valores médios, para quantificar a influência de cada um deles no cálculo. Os desvios foram baixos, mostrando que as incertezas ficaram em um nível aceitável. O aumento de temperatura não comprometeu os resultados, apesar da grande amplitude. Em quase todas as relações analisadas, o desvio relativo foi menor que 1, mostrando que os dois métodos analisados possuem boa resistência a imprecisões nas entradas.

D.2 Introdução

Leitos fluidizados sólido-líquido possuem diversas aplicações na indústria, como classificação de partículas (PIOVANO *et al.*, 2015), adsorção (KOPPEJAN *et al.*, 2018), biorreações (DENG *et al.*, 2016), entre outras. Esses equipamentos consistem em colunas preenchidas com partículas, que são percoladas por um líquido em movimento ascendente com velocidade entre a de mínima fluidização (U_{mf}) e a velocidade terminal (U_0). Esse movimento é responsável pela agitação das partículas, promovendo uma melhora na transferência de calor e massa entre fases se comparado ao leito fixo (BELLO *et al.*, 2017).

Apesar de sua grande aplicabilidade, a utilização de leitos fluidizados é limitada pela robustez das equações disponíveis para predição de sua porosidade (ε), importante para estimar, por exemplo, a conversão de reações e taxas de transferência de calor e massa. A porosidade corresponde à fração de fluido do leito, definida pela (D.1).

$$\varepsilon = \frac{V_l}{V_T} \tag{D.1}$$

em que V_l é o volume de líquido, e V_T é o volume total do leito. A porosidade é maior quanto maior for a velocidade superficial do fluido (U). A equação proposta por Richardson and Zaki (1954) (Equação (D.2)) é amplamente utilizada para determinar essa relação.

$$\frac{U}{U_0} = k\varepsilon_{R-Z}^{n} \tag{D.2}$$

em que n e k são parâmetros adimensionais, e ε_{R-Z} é a porosidade do leito. Esses dois parâmetros (n e k) podem ser calculados pelas correlações propostas por Khan and Richardson (1989), descritas pelas Equações (D.3) e (D.4).

$$\frac{4,8 - n}{n - 2,4} = 0,043\mathrm{Ar}^{0,57} \tag{D.3}$$

$$k = 1 - 1,15\left(\frac{d_p}{D}\right)^{0,6} \tag{D.4}$$

em que Ar é o número de Arquimedes, dado pela Equação (D.5).

$$\mathrm{Ar} = \frac{d_p^3\rho_l\left(\rho_p - \rho_l\right)g}{\mu^2} \tag{D.5}$$

em que d_p é o diâmetro da partícula, ρ_l é a densidade do líquido, ρ_p é a densidade da partícula, g é a gravidade e μ é a viscosidade do líquido.

As correlações mostram que a relação entre porosidade e velocidade superficial depende de características do fluido e das partículas. Essas características podem variar com o tempo durante a fluidização, ou entre partículas individuais. Como essas variações não são previstas pelas correlações, é importante avaliar a sensibilidade da porosidade a elas.

A avaliação pode ser feita calculando-se a porosidade, empregando valores médios para as variáveis de entrada, e em seguida repetindo os cálculos com valores maiores ou menores para cada variável. A sensibilidade paramétrica de leitos fluidizados gás-sólido tem sido investigada (TAGHIPOUR *et al.*, 2005; BRADSKI, 2000; SHI *et al.*, 2019; PAN *et al.*, 2022), avaliando majoritariamente o efeito da variação dos valores de parâmetros de simulações na fluidodinâmica do leito.

Em se tratando de fluidização líquida, são poucos os trabalhos que analisam a sensibilidade de propriedades a variáveis de entrada. Chen *et al.* (2010), por exemplo, estudaram a expansão de um biorreator de leito fluidizado sólido-líquido, com produção de gases. A expansão do leito se mostrou igualmente sensível à velocidade superficial do líquido e do gás, enquanto o tempo de contato foi mais sensível à velocidade superficial do líquido. Lopes *et al.* (2018) analisaram o efeito da rugosidade superficial e propriedades mecânicas das partículas na expansão de um leito fluidizado sólido-líquido. Foi verificado que, para partículas macias, o número de Stokes possui uma influência maior no desvio da porosidade, do que para partículas rígidas.

Apesar disso, nenhum dos trabalhos encontrados avaliou a sensibilidade do cálculo da expansão do leito em relação às imprecisões nas medidas de parâmetros relativos às características do líquido e das partículas. Nesse sentido, o presente trabalho realizou uma análise de sensibilidade paramétrica na fluidização de quatro grupos diferentes de partículas em água, com o objetivo de avaliar a sensibilidade da equação de Richardson-Zaki a estas imprecisões. Também foi estudada a sensibilidade do método do coeficiente angular da pressão em função da altura, para determinação experimental da porosidade. Para isso, foram analisados os efeitos de alterações nos valores de: densidade e diâmetro da partícula; temperatura da água e velocidade terminal.

D.3 Metodologia

D.3.1 Caracterização das partículas

Neste trabalho, foram utilizados quatro grupos de partículas esféricas comerciais, de três materiais diferentes: Porcelana, Acrilonitrila-Butadieno-Estireno (ABS) e Alumina (em dois diâmetros: $6,37$ e $3,09$ mm). Os grupos de partículas foram caracterizados quanto aos seus diâmetros, densidades e velocidades terminais.

A densidade foi obtida por picnometria. Para isso, utilizou-se um picnômetro de 25 mL, uma balança analítica (marca Bel, modelo L3102i) e água destilada como fluido de referência. Pesou-se uma amostra de partículas de cada grupo, que foi inserida no picnômetro, e ele foi preenchido com água. A densidade foi calculada por:

$$\rho_p = \frac{m_p \rho_l}{V_{pic}\rho_l + m_p + m_{pic} - m} \tag{D.6}$$

em que m_p é a massa das partículas, m_{pic} é a massa do picnômetro vazio, V_{pic} é o volume do picnômetro, ρ_l é a densidade da água e m é a massa do picnômetro preenchido com a amostra de partículas e água. As medidas foram repetidas 5 vezes para cada grupo de partículas. As densidades obtidas, assim como o desvio padrão, estão apresentados na Tabela D1.

Partícula	ρ_p (kg m^{-3})	d_p (mm)	U_0 (m s^{-1})
ABS	1822 ± 5	5,95 ± 0,07	0,39 ± 0,01
Alumina maior	3573 ± 10	6,37 ± 0,21	0,71 ± 0,03
Alumina menor	3586 ± 10	3,09 ± 0,16	0,48 ± 0,02
Porcelana	2407 ± 3	6,13 ± 0,19	0,51 ± 0,02

Table D1 – Propriedades das Partículas.

D.3.2 Aparato experimental

O equipamento utilizado nos experimentos de fluidização e suas partes constituintes são mostrados na Figura D1.

O equipamento consiste em uma coluna de acrílico transparente de 10 cm de diâmetro interno e 1 m de altura (A); um reservatório de água (B); uma bomba centrífuga de 1/2 HP (C); um leito fixo de partículas de vidro utilizado para uniformizar o perfil de velocidades na entrada (D), sobre o qual há uma placa perfurada que funciona como distribuidor; um sensor de pressão diferencial (modelo MPX5010-DP) (E); e um sensor de vazão de efeito Hall (modelo YF-G1) (F).

Ao longo da altura da coluna, há 15 tomadas de pressão. A primeira fica 1,5 cm acima do distribuidor, a segunda fica a 4,5 cm da primeira, e as outras estão espaçadas 6 cm entre si. Em cada uma delas, há tubos flexíveis que podem ser fechados ou abertos por presilhas. Esses tubos são conectados ao sensor de pressão diferencial, que permite medir a queda de pressão entre dois pontos da coluna, correspondentes aos dois tubos que estiverem abertos no momento da medição.

Em cada corrida experimental, uma quantidade de partículas foi pesada e introduzida na coluna. A água do reservatório foi então bombeada, passando pela coluna em movimento ascendente e provocando a fluidização das partículas. Ao atravessar a coluna, a água era reconduzida ao reservatório. Foi realizada a fluidização de cada grupo

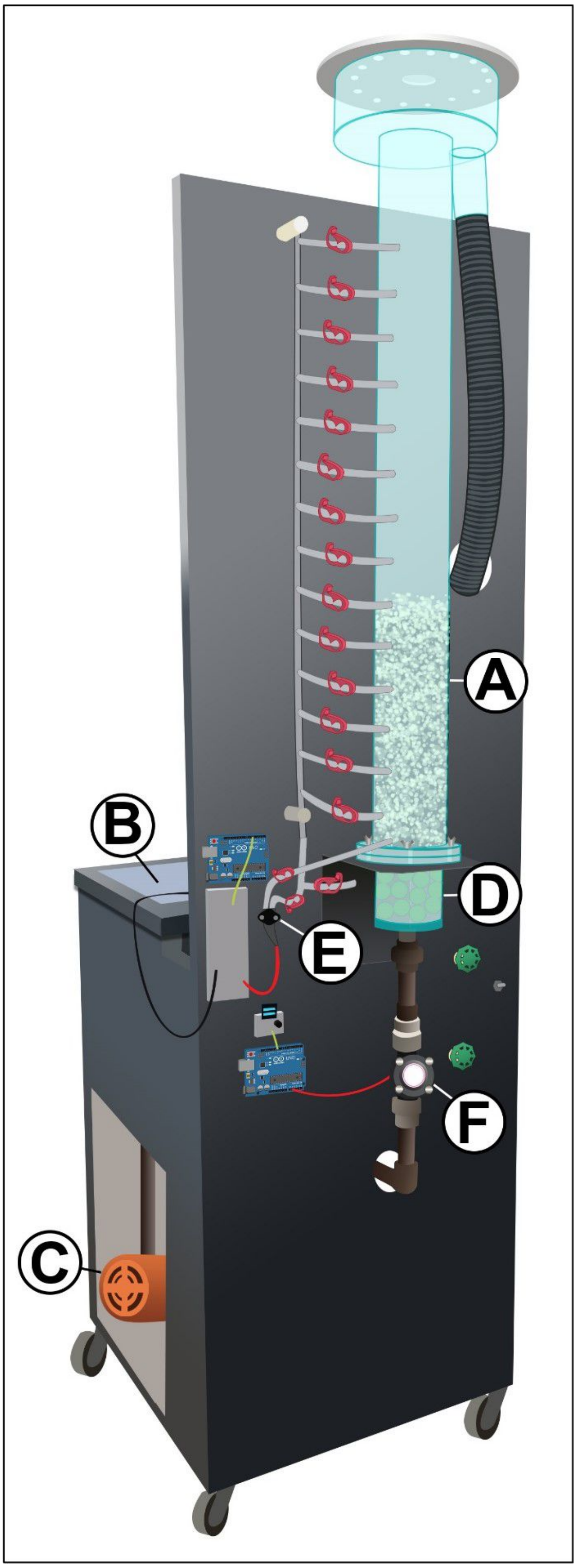


Figure D1 – Equipamento de Fluidização.

de partículas em diferentes vazões de líquido, que podem ser alteradas por válvulas na entrada da coluna e monitoradas pelo sensor de vazão.

A temperatura da água foi monitorada com um sensor NTC (*Negative Temperature Coefficient*) posicionado no reservatório. Foram obtidos valores de temperatura a cada 50 ms, durante todos os experimentos. Durante a fluidização, a temperatura da água subiu em decorrência do bombeamento. As temperaturas máxima, mínima e média observadas durante a fluidização de cada grupo de partículas foram registradas. Os valores de porosidade calculados com a temperatura máxima e mínima foram comparados aos valores obtidos com o valor médio, para avaliar a sensibilidade em relação à temperatura.

Com o sensor de pressão, foram obtidos os valores de pressão diferencial mantendo aberta a primeira tomada de pressão (1,5 cm acima do distribuidor) e utilizando sucessivamente cada uma das outras tomadas como segundo ponto de medida. A pressão diferencial foi medida a cada 50 ms, durante 30 s, para cada dupla de tomadas de pressão, e foi registrado o valor médio durante este período. Isso permitiu obter a queda de pressão em várias alturas, e determinar o coeficiente angular da pressão em função da altura ($\frac{\mathrm{d}P}{\mathrm{d}z}$). Cada medida foi tomada em triplicata. Estes valores foram utilizados no cálculo da porosidade experimental por meio da Equação (D.7). As condições experimentais empregadas são mostradas na Tabela D2.

Partícula	**Massa** (kg)	**N° vazões**	**Faixa vazões** ($\mathrm{L\,min^{-1}}$)
ABS	2,0	11	25 - 75
Alumina maior	4,0	8	40 - 75
Alumina menor	4,0	8	40 - 75
Porcelana	2,7	9	25 - 75

Table D2 – Massas e Vazões utilizadas nos Experimentos.

D.3.3 Determinação da Porosidade

A porosidade do leito fluidizado sólido-líquido foi determinada experimentalmente utilizando o coeficiente angular da pressão em função da altura ($\frac{\mathrm{d}P}{\mathrm{d}z}$), conforme Equação (D.7).

$$\varepsilon_{exp} = 1 - \frac{\frac{\mathrm{d}P}{\mathrm{d}z}}{(\rho_p - \rho_l)\,g} \tag{D.7}$$

A densidade e a viscosidade da água foram calculadas a partir da temperatura (T) com as correlações fornecidas por DDBST (2021), a seguir:

$$\rho_l = 0,14395 \cdot 0,0112^{\left[-1+(1-\frac{T+273}{649,727})^{0,05107}\right]} \tag{D.8}$$

$$\rho_l = 0,001 \cdot \exp^{\left[-3,7188+\frac{578,919}{(T+273)-137,546}\right]} \tag{D.9}$$

Para cada vazão de líquido, foram realizados os cálculos da porosidade experimental (Equação (D.7)) e da porosidade obtida por Richardson-Zaki (Equação (D.2)), utilizando os valores médios das variáveis de entrada (ρ_p, T, d_p, U_0). Em seguida, foram repetidos os cálculos com os limites superior e inferior, dados pela incerteza (caso de ρ_p, d_p, U_0) ou variação no tempo (caso de T) destas variáveis. A magnitude dos desvios correspondentes foi analisada para determinar a sensibilidade dos cálculos a cada uma destas variáveis de entrada, e para avaliar se neste experimento a precisão dos valores de porosidade pode ter sido afetada significativamente.

As relações entre as entradas (Medidas Experimentais) e saídas (Resultados) do equacionamento utilizado são mostradas na Figura D2. A porosidade calculada pelas correlações é afetada pelas quatro variáveis analisadas, enquanto a porosidade obtida experimentalmente é afetada, dentre estas variáveis, apenas pela densidade da partícula e temperatura do líquido. A sensibilidade de cada uma destas seis relações foi analisada individualmente.

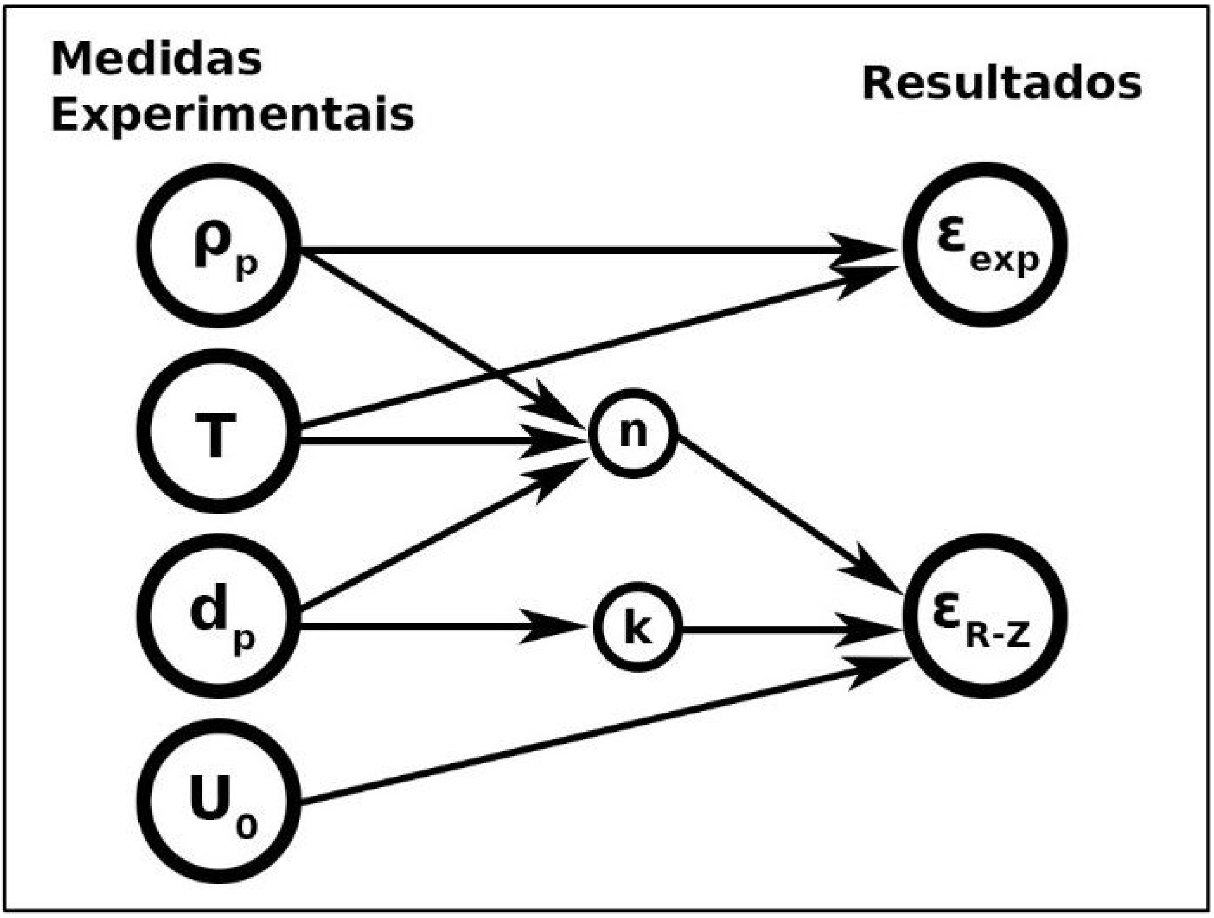


Figure D2 – Relações entre Medidas Experimentais e Resultados.

D.4 Resultados e discussões

Em virtude do calor gerado pela bomba durante o bombeamento, e da recirculação do líquido, a temperatura da água subiu durante os experimentos. As distribuições de temperatura para cada grupo de partículas estão representadas nas Figuras D3 à D6.

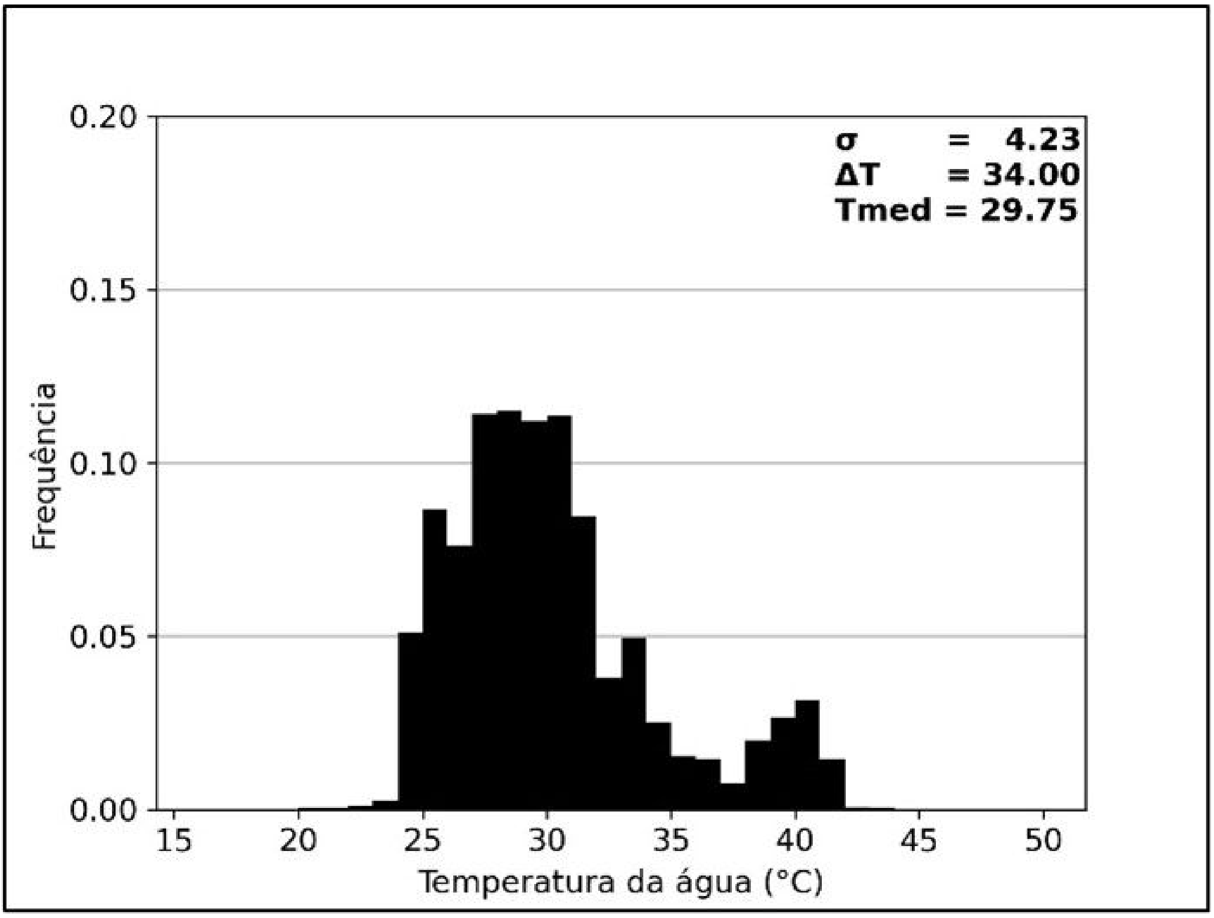


Figure D3 – Distribuição de Temperaturas – ABS.

Como observado nas Figuras D3 à D6, a distribuição de temperaturas assumiu formas diferentes entre os experimentos. Isso mostra que a temperatura não subiu de forma uniforme. No caso dos grupos ABS e Alumina maior, podem ser observados dois picos de frequência, que correspondem a dados de experimentos com temperaturas iniciais da água diferentes. Isso se deve à variação da temperatura ambiente em dias e horários diferentes, e a experimentos realizados em sequência.

Os valores máximo, mínimo e médio de temperatura, assim como a amplitude observada para cada tipo de partícula estão listados na Tabela D3. As temperaturas médias ficaram na mesma faixa. Os valores de amplitude foram consideráveis (31 a 40 °C). Isso pode ser relacionado ao fato de os experimentos de fluidização utilizarem um tempo relativamente grande (cerca de 3 horas), o que causou um grande aumento de temperatura.

Os desvios percentuais máximos observados na porosidade, em relação a cada

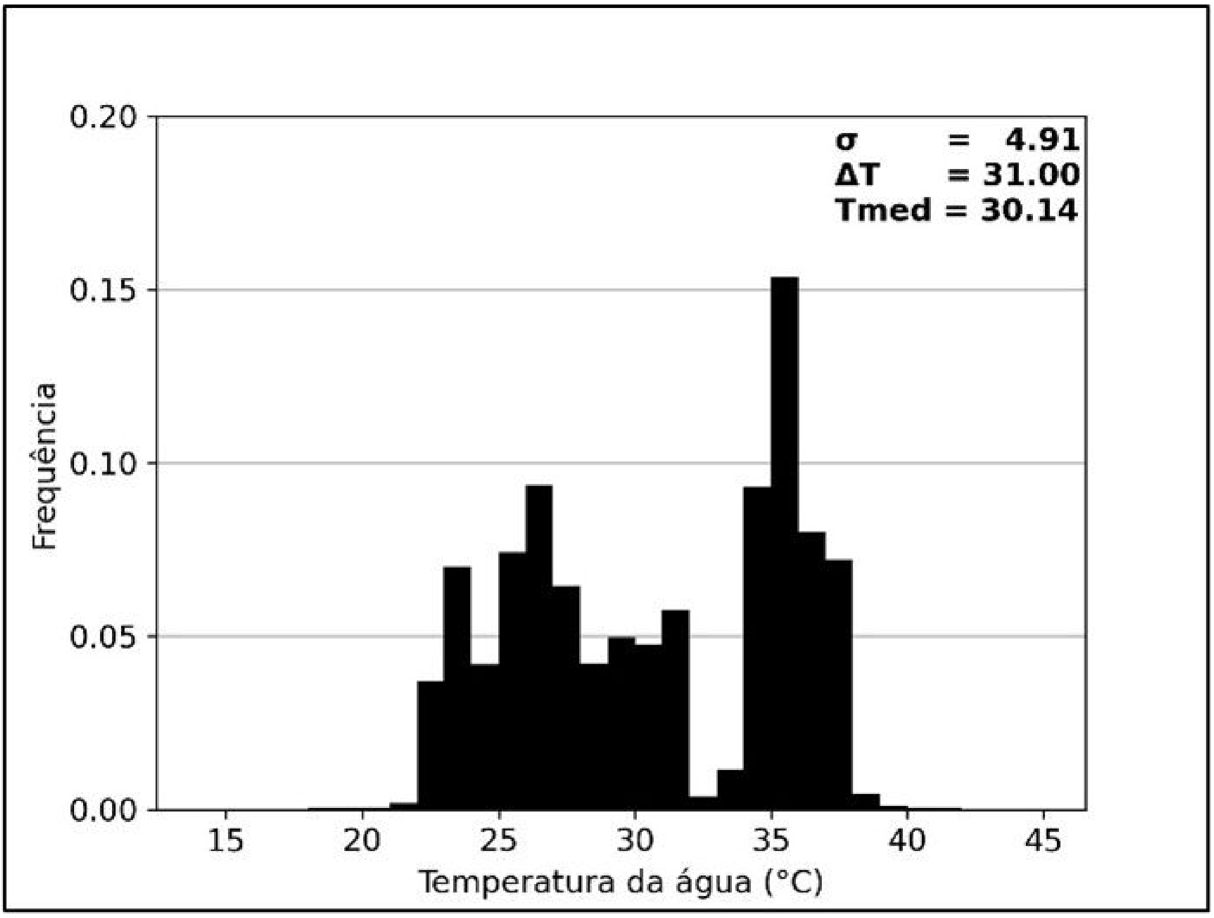


Figure D4 – Distribuição de Temperaturas – Alumina maior.

Partícula	T_{max}	T_{med}	T_{min}	**Amplitude**
ABS	50 °C	16 °C	29,8 °C	34 °C
Alumina maior	45°C	14 °C	30,1 °C	31 °C
Alumina menor	47°C	11 °C	28,0 °C	36 °C
Porcelana	48°C	08 °C	26,7 °C	40 °C

Table D3 – Dados de Temperatura.

medida experimental analisada, são mostrados na Tabela D4. Para cada relação entre medida experimental e porosidade, foi listado o maior desvio observado entre os quatro grupos. Os maiores desvios foram da porosidade experimental em relação à temperatura, e da porosidade calculada em relação à velocidade terminal. Porém, o maior de todos os desvios foi de 2,03%, o que pode ser considerado baixo. Estes valores atestam a robustez do método experimental e das correlações utilizadas para cálculo da porosidade.

Em quatro das seis relações analisadas, os maiores desvios foram observados nas partículas de alumina. Este é o material com a maior densidade entre as partículas utilizadas. Isso pode indicar que o cálculo da porosidade é mais sensível a incertezas com partículas de materiais mais densos.

As incertezas percentuais nas medidas experimentais ficaram, de uma maneira

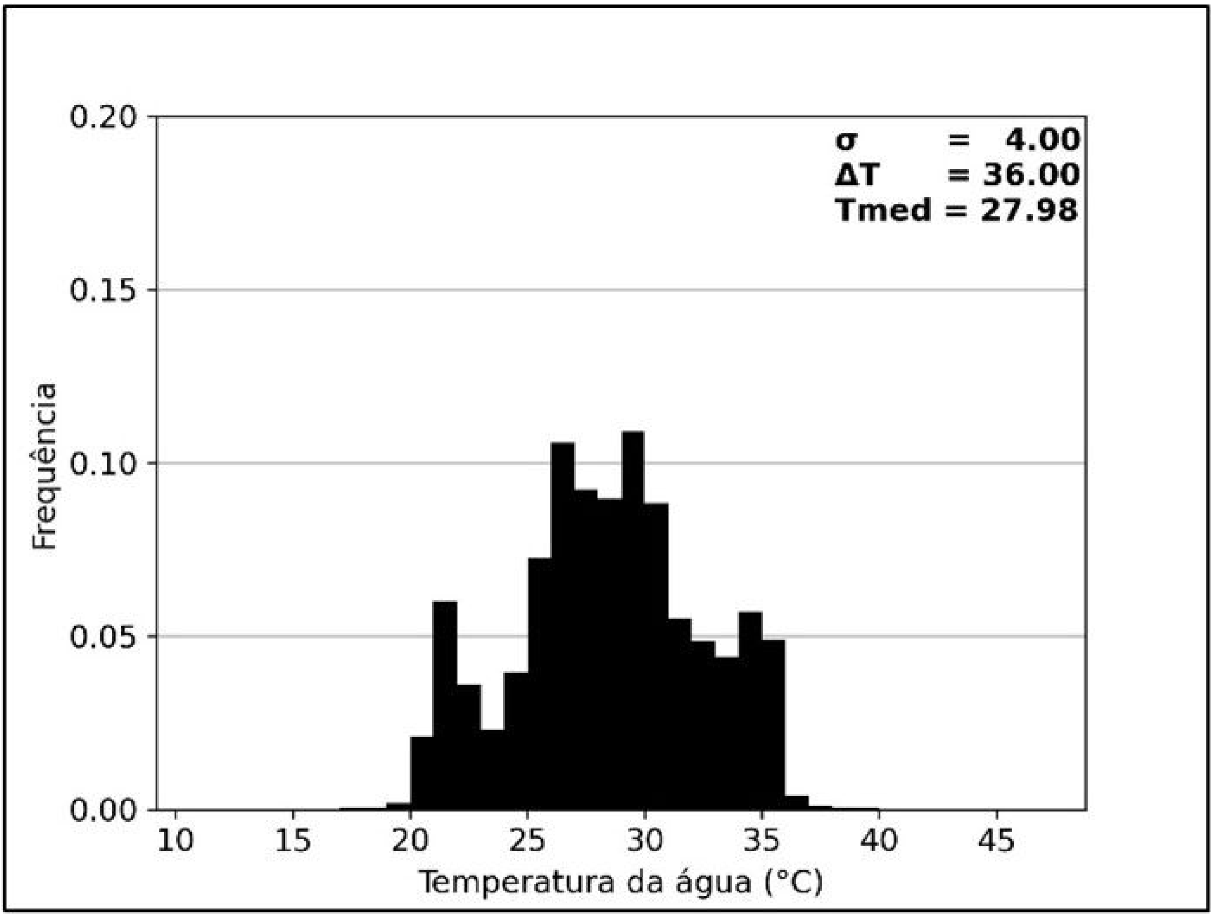


Figure D5 – Distribuição de Temperaturas – Alumina menor.

geral, dentro da faixa reportada na literatura (LOPES *et al.*, 2018). Entre as alterações percentuais que causaram os maiores desvios, as maiores foram observadas no valor de temperatura. Isso causou um efeito maior na porosidade experimental (por meio da densidade do líquido) do que na porosidade calculada (por meio do parâmetro n). Isso mostra a relativa insensibilidade do parâmetro n ao número de Arquimedes (Ar), quando este assume valores maiores que 106 (KHAN; RICHARDSON, 1989).

O desvio relativo foi calculado pela razão entre o desvio e a alteração que o causou. Ele pode ser entendido como a derivada média do desvio percentual em relação à alteração percentual na entrada, na faixa de valores analisada. Por exemplo, a partícula de Porcelana apresentou em média 0,28% de desvio em ε_{exp} para cada 1% de alteração em T. Em quase todos os casos, o desvio relativo foi menor que 1, o que indica que o desvio nos resultados é menor que a alteração nas entradas. Isso mostra que, nas relações analisadas, os cálculos atenuam o efeito destas alterações. A única exceção foi da porosidade experimental em relação à densidade da partícula. O desvio correspondeu à alteração da densidade multiplicada por 2,54. Isso indica que esta é a relação mais sensível entre as analisadas. Contudo, devido à alteração na densidade ser pequena, o desvio permaneceu menor que 1%.

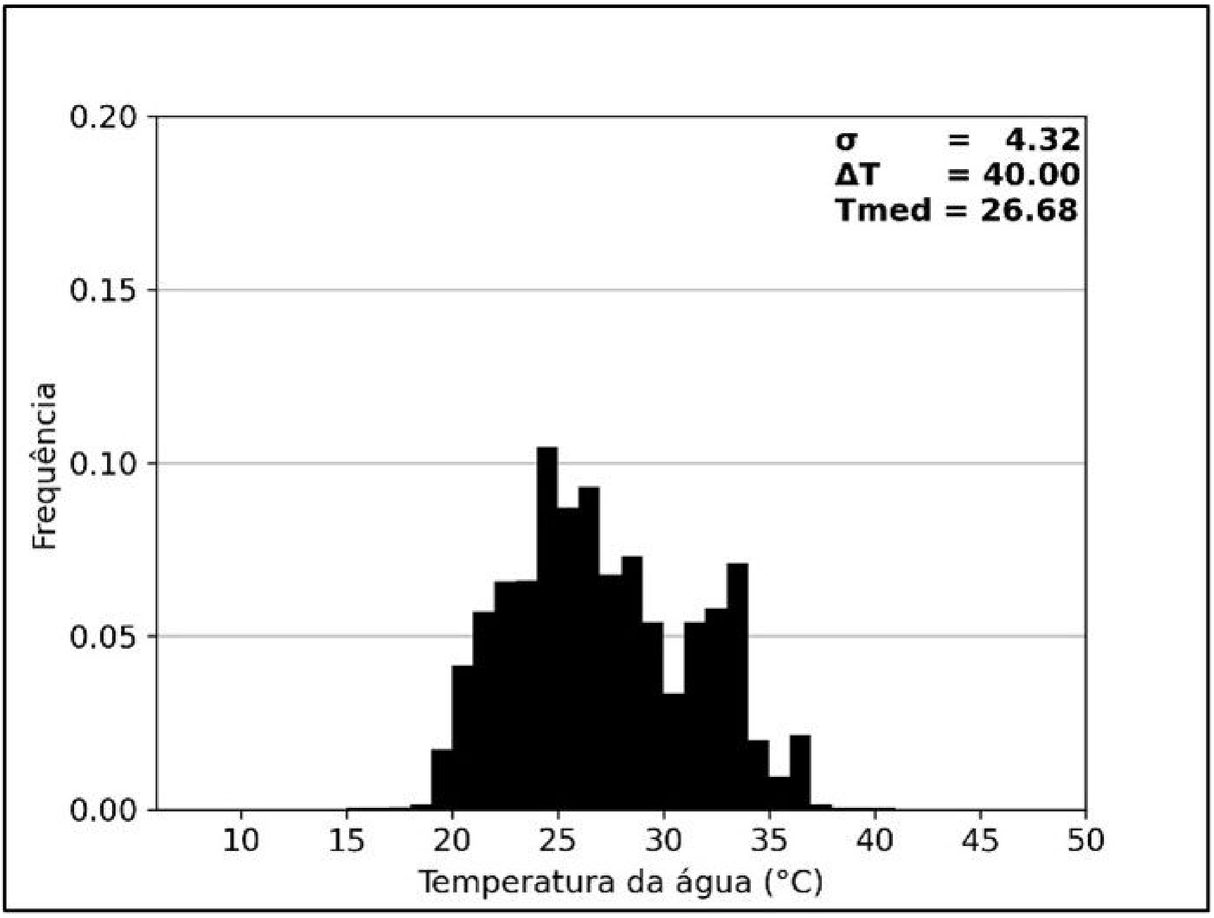


Figure D6 – Distribuição de Temperaturas – Porcelana.

Medida	Partícula	Alteração	Desvio máximo	Desvio relativo
ρ_p	Alumina	0,28 %	ε_{exp}: 0,71 %	2,54 %/%
	ABS	0,27 %	ε_{R-Z}: 0,001 %	0,005 %/%
T	Porcelana	7,11 %	ε_{exp}: 1,96 %	0,28 %/%
	Alumina$_{min}$	6,32 %	ε_{R-Z}: 0,32 %	0,05 %/%
d_p	Alumina	3,30 %	ε_{R-Z}: 0,23 %	0,07 %/%
U_0	Alumina	4,23 %	ε_{R-Z}: 2,03 %	0,48 %/%

Table D4 – Desvios Máximos nos Resultados.

D.5 Conclusões

A sensibilidade na determinação da porosidade de um leito fluidizado sólido-líquido foi analisada. Quatro grupos de partículas com diferentes propriedades físicas foram caracterizados quanto à densidade, o diâmetro e a velocidade terminal. Estas partículas foram submetidas à fluidização em água, em diferentes vazões, e foi obtido o coeficiente angular da pressão em função da altura, utilizando um sensor de pressão diferencial. A temperatura durante os experimentos foi registrada. A porosidade do leito foi calculada utilizando a equação de Richardson and Zaki (1954) em conjunto com as correlações de Khan and Richardson (1989). A porosidade também foi calculada com os valores de gradiente vertical de pressão obtidos experimentalmente. Estes cálculos foram primeiramente feitos utilizando valores médios das variáveis de entrada (ρ_p, d_p, T, U_0). Os cálculos foram então repetidos, usando valores maiores e menores que os médios, determinados pela incerteza nas propriedades das partículas e pela variação da temperatura durante o experimento. Os desvios correspondentes em ε_{exp} e ε_{R-Z} foram analisados.

O maior desvio observado foi de 2,03%, o que pode ser considerado baixo. Isso mostra que, neste experimento, as incertezas das medidas experimentais ficaram em um nível aceitável, pois não produziram desvios muito grandes nos valores de porosidade. Os valores de porosidade podem ser obtidos utilizando os valores médios das variáveis de entrada com boa confiabilidade.

A variação na temperatura durante os experimentos, apesar de ter sido relativamente grande (amplitude de 31 a 40 °C), não produziu alterações significativas nos valores de porosidade. Para todos os grupos de partículas, o desvio da porosidade em relação à temperatura ficou em menos de 2,0%. A utilização direta da temperatura média nos cálculos não comprometeu a precisão dos resultados.

O maior desvio relativo observado (2,54 %/%) foi de ε_{exp} em relação a ρ_p. Isso indica a maior sensibilidade do método experimental ao valor da densidade da partícula. Isso mostra que é necessário um cuidado na avaliação da densidade das partículas em experimentos deste tipo. Se a sua incerteza for muito grande, a imprecisão da porosidade experimental pode se tornar significativa em decorrência de sua maior sensibilidade.

Os demais desvios relativos ficaram em menos de 1 %/%. Isso mostra que, na maioria das relações analisadas, uma alteração de 1% em uma variável de entrada causa menos de 1% de alteração na saída. O método experimental e o equacionamento proposto para obter valores de porosidade mostraram, de forma geral, uma boa resistência a imprecisões nas entradas. A utilização de valores médios de temperatura, densidade, diâmetro e velocidade terminal das partículas se mostrou válida para calcular a porosidade.

D.6 Agradecimentos

O presente trabalho foi realizado com apoio da Agência Nacional do Petróleo, Gás Natural e Biocombustíveis (ANP), da Financiadora de Estudos e Projetos (FINEP), da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES – Código de Financiamento 001), do Conselho Nacional de Desenvolvimento Científico e Tecnológico – processo nº 408618/2018-3 e da Fundação de Amparo à Pesquisa do Estado de São Paulo - processo nº 2019/19173-9, Fundação de Amparo à Pesquisa do Estado de São Paulo (FAPESP).