Federico Municchi, MSc

Coarse Grained models for momentum, heat and mass transfer in dense gas-particle suspensions from Particle-Resolved Direct Numerical Simulation

DOCTORAL THESIS

to achieve the university degree of

Doktor der technischen Wissenschaften

submitted to

Graz University of Technology

Supervisor

Univ.-Prof. Dipl.-Ing. Dr.techn. Johannes Khinast

Institute of Process and Particle Engineering

Second supervisor
Ass. Prof. Dipl.-Ing. Dr. techn. Stefan Radl

External examinator
Univ.-Prof. Dipl.-Ing. Dr.techn. Alfredo Soldat

Graz, october 2017

AFFIDAVIT

I declare that I have authored this thesis independently, that I have not used other than the declared sources/resources, and that I have explicitly indicated all material which has been quoted either literally or by content from the sources used. The text document uploaded to TUGRAZonline is identical to the present doctoral thesis.

Date

Signature

Abstract

Due to the dramatic increase in available computational resources, Particle-Unresolved Euler-Lagrange (PU-EL) models for particulate flow simulations are becoming more and more appealing for solving problems of industrial interest. In fact, particulate flows are widely encountered in industry where the suspended and suspending phases exchange momentum, heat and mass. Often, even a variety of (heterogeneous) chemical reactions take place in parallel to flow processes, typically involving a plurality (100 or more!) of chemical species. Also, physical processes like particle breakage or agglomeration, as well as droplet coalescence or coating may be relevant. The main advantage of Euler-Lagrange models with respect to others is that each particle (or parcel that represents a set of particles) is tracked. Thus, the above mentioned particle-based chemical-physical processes can be studied in their most natural form - statistical models that rely on a pre-averaging of particle flow quantities are not needed. In other words, only the fluid phase is coarse-grained, while particle-related phenomena can be simulated directly. This makes the addition of new models describing chemical and physical processes on the scale of the particles straightforward.

PU-EL formulations require closure models for all relevant interphase exchange processes. While several closures are available for Euler-Euler simulations (in which some form of pre-averaging of particle phase quantities is performed), it is not clear if they can be straightforwardly applied to PU-EL models. Furthermore, for dense flows, it is unclear how one shall formulate the interphase exchange terms near walls. Therefore, the present thesis studies per-particle momentum, heat and mass transfer in gas-solid suspensions. The ultimate goal is deriving closure models for the interphase transfer coefficients that are suitable for PU-EL. The main idea is that such models can be obtained by analysing data from highly resolved simulations. In order to do so, we first develop the basic tools which are: (i) a parallel data processing library which allows extracting relevant information, and (ii) a method to perform highly resolved simulations of momentum, heat and mass transfer for Reynolds numbers up to $O(1,000)$.

Secondly, we focus on bi-disperse gas-particle systems where we provide a refined statistical description of per-particle interphase exchange rates. Existing correlations found in literature are compared against our data, and we discuss the validity of the widely used correlation of Beetstra et al. [11]. Additionally, we provide new insight with respect to particle-based heat (or mass) transfer coefficients. Specifically, we conclude that the scaled standard deviation of all transfer coefficients in a dense suspension is, to a first approximation, a universal constant and in the order of 0.4.

Finally, we focus on wall effects with the aim to develop a description of wall normal profiles of flow and scalar fields. In addition, our results indicate that interphase transfer coefficients in the proximity of walls systematically deviate from their corresponding bulk values.

Kurzfassung

Aufgrund der dramatischen Zunahme der verfügbaren Rechenressourcen können Euler-Lagrange (EL) Modelle für die Simulationen von industriellen Partikelströmungen zunehmend angewendet werden. Derartige Partikelströmungen sind in der Verfahrenstechnik weit verbreitet. In diesen Strömungen muss typischer Weise der Austausch von Impuls, Wärme, sowie die Stoffübertragung berücksichtigt werden. Zusätzlich können (heterogene) chemische Reaktionen auftreten, die die Beschreibung einer Vielzahl (oftmals 100 und mehr!) von chemischen Spezies bedürfen. Ebenso können physikalische Prozesse wie Partikelbruch oder Agglomeration, sowie Tropfenkoaleszenz oder Tropfendeposition relevant sein. Der Hauptvorteil von Euler-Lagrange Modellen gegenüber anderen Modellen ist das jedes Partikel (bzw. so-genannte 'Parcels' die ein Ensemble von Partikeln repräsentieren) separat verfolgt wird. Deshalb können die oben erwähnten partikelbasierten chemisch-physikalischen Prozesse in ihrer natürlichsten Form verfolgt werden - statistische Modelle die auf gemittelten Partikelinformationen basieren werden nicht benötigt. Mit anderen Worten, nur Fluidphaseneigenschaften werden gemittelt, während partikelbezogene Phänomene direkt simuliert werden können. Dies macht die Kombination von Partikelströmungsmodellen mit Modellen die chemische und physikalische Prozesse auf der Skala der Partikel betrachten einfach.

EL-Formulierungen erfordern Schließbedingungen für alle relevanten Austauschprozesse zwischen Fluid und Partikelphase. Während mehrere dieser Schließbedingungen für Euler-Euler-Modelle existieren (bei diesen Modellen wird eine Form der Mittelung von Fluid- und Partikelphasen vorgenommen), ist es nicht klar, ob diese Bedingungen auch direkt auf EL Modelle angewendet werden können. Darüber hinaus ist es für dichte Partikelsysteme unklar, wie man die Schließbedingungen für Austauschprozesse in der Nähe von Wänden formulieren soll. Die vorliegende Dissertation untersucht deshalb den partikel-individuellen Transfer von Impuls, Wärme und Stoff in Gas-Partikel-Suspensionen. Das Ziel ist die Ableitung von Schließbedingungen für Transferkoeffizienten die für EL Modelle geeignet sind. Diese Schließbedingungen werden durch die Analyse von Daten aus hochaufgelösten Simulationen erhalten. Für die praktische Umsetzung dieses Ansatzes wurden zunächst die grundlegenden Werkzeuge entwickelt, diese sind (i) eine parallele Datenverarbeitungsbibliothek, die es ermöglicht, relevante Informationen aus hochaufgelösten Simulationen zu extrahieren, sowie (ii) eine Methode, um hochaufgelöste Simulationen von Impuls-, Wärme- und Stoffübertragung für Reynoldszahlen bis $O(1.000)$ zu realisieren.

In weiterer Folge konzentriert sich die vorliegende Dissertation auf bi-disperse Gas-Partikel-Systeme, wobei auf eine verfeinerte statistische Beschreibung der Partikel-individuellen Transferraten Wert gelegt wird. Bestehende Korrelationen aus der Literatur werden mit eigenen Daten verglichen. Weiters wird die Gültigkeit der weit verbreiteten Korrelation von Beetstra et al. [11] diskutiert. Darüber hinaus bietet die vorliegende Arbeit neue Einblicke in Bezug auf Partikel-individuelle Wärme- (oder

Stoff-) Transferkoeffizienten. Im Speziellen zeigen wir, dass die skalierte Standardabweichung aller Transferkoeffizienten in einer dichten Suspension in erster Näherung eine universelle Konstante ist, und einen approximativen Wert von 0.4 annimmt.

Schließlich konzentrieren wir uns auf Wandeffekte, mit dem Ziel die Profile von wandnahen Strömungs- und Skalarfeldern zu beschreiben. Unsere Ergebnisse zeigen weiters, dass Transferkoeffizienten in der Nähe von Wänden systematisch von ihren Werten im Bulk abweichen.

Acknowledgements

I would like to express my sincere gratitude to Prof. Johannes Khinast who made this work possible thanks to his support and his trust on me. I would like to thank my external examiner Prof. Alfredo Soldati for his valuable time reading this thesis and for the interesting discussion we had at the International Conference on Numerical Methods in Multiphase flows in Japan. I am truly honoured that he accepted to participate to my dissertation.

I am very thankful to Dr. Stefan Radl who supervised me during these three years and who focused a large amount of time and resources into my work. He helped me in every aspect of my research, following my progresses from the first research outline to journal submission and provided helpful and cunning solutions to many of the issues we encountered during our work together. Indeed, he supported me more than any PhD student could hope, allowing me to participate to many conferences and to attend the course on MPI in Vienna. During these years, he gave me access to powerful computing facilities like the Vienna Scientific Cluster, putting me in charge of a huge amount of CPU hours. I am happy to say that the faith he put on me was worth after all.

Special gratitude is given to the 'SimuNerd@TUGraz' group: Thomas, Christian, Benedict, Sadegh, Maryam, Mingqiu, Jakob and Theresa. They were all good friends and they helped me a lot during these three years.

Furthermore, I would like to thank Adela, Michaela and Silvia for their valuable help with administrative stuff which are a kind of nightmare for me!

Most of all, I would like to thank my family. This work would have not been possible without their continuous support and encouragement through many years, spanning from my early undergraduate studies to the present thesis which marks the end of my career as a student and the beginning of my career as a scholar. I am grateful to my wife, who would follow me everywhere, even to Antarctica or Jupiter.

Finally, I want to thank my beloved grandfather Nello Municchi who inspired me to become a scientist. I really miss him. This work is dedicated to his memory.

Contents

List of Figures

List of Tables

Disclaimer

This thesis is based on three works published in, or submitted to peer-reviewed journals. The author's role in each publication is briefly outlined

Title	Authors	Contribution
Highly efficient spatial data filtering in parallel using the opensource library CPPPO	F. Municchi, S. Radl, C. Goniva	F. Municchi coded most of the library, designed and coded the parallel filtering algorithms, run the simulations, performed the accuracy and scalability checks and wrote the manuscript.
Consistent closures for Euler-Lagrange models of bi-disperse gas-particle suspensions derived from particle-resolved direct numerical simulations	F. Municchi, S. Radl	F. Municchi designed, coded and tested the HFD-IB algorithm, run the simulations and analysed the data. F. Municchi had also the original idea for the saturation study and wrote the manuscript.
Momentum, heat and mass transfer simulations of bounded dense monodispersed gas-particle systems	F. Municchi, S. Radl	F. Municchi run the simulations, analysed the results and wrote the manuscript.

The authors acknowledge support by the European Commission through FP7 Grant agreement 604656 (“NanoSim”), and the NAWI Graz project by providing access to dcluster.tugraz.at. CFDEM®is a registered trademark of DCS Computing GmbH. The computational results presented have been achieved (in part) using the Vienna Scientific Cluster (VSC-3).

1. Introduction

Disperse multiphase flows can be found in a wide variety of engineering applications ranging from combustion to pharmaceutical processes and the oil & gas industry. They also find applications in other branches of science like geology (for example the study of river beds or vulcanoes) or medicine (e.g., modelling of blood flow).

A central theme in the study of disperse multiphase flows is the need for predicting (and modelling) the detailed behaviour of the flow field and the phenomena induced by the presence of the dispersed phase within the (continuous) fluid phase. In fact, unlike what happens in separated flows where two or more phases are separated by connected *macro-scale* interfaces, in disperse multiphase flows there is a large number of *micro scale* interfaces where the phenomena of interest (e.g., momentum, heat and mass transfer) take place. Especially *dense, poly-disperse* disperse multiphase flows are challenging due to their humongous number of interfaces.

In gas-particle systems (for example, in fluidized beds), the *micro scale* is related to the particle diameter, which can range between 10 and $1,000$ μm. Therefore, a detailed *direct* solution strategy aiming at interphase transfer at the particle scale is not a viable tool to design devices on the scale of $O(1)$ m. This is due to the complexity of the flow field in the interstices between the particles.

As a result, a number of approaches to reduce the number of degrees of freedom, and to develop theories that allow the process to be described on a larger scale, have been proposed [51, 101, 107]. Similarly to coarse grained theories for turbulence, all of these theories contain unclosed terms that require further modelling.

The accurate formulation of such theories together with correct closure models is of pivotal importance for the design of industrial scale equipment involving disperse multiphase flows.

1.1. The *NanoSim* project

One way to attack the above problem is developing a succession of such theories for different spatio-temporal scales from a molecular level to the full scale device. Results from "small" scale formulations are then used to develop appropriate closures to be used in models developed for the successive "larger" scale. This is the approach followed by the *NanoSim* project: *"A Multi-scale Simulation-Based Design Platform for Cost-Effective CO_2 Capture Processes using Nano-Structured Materials"*. Even

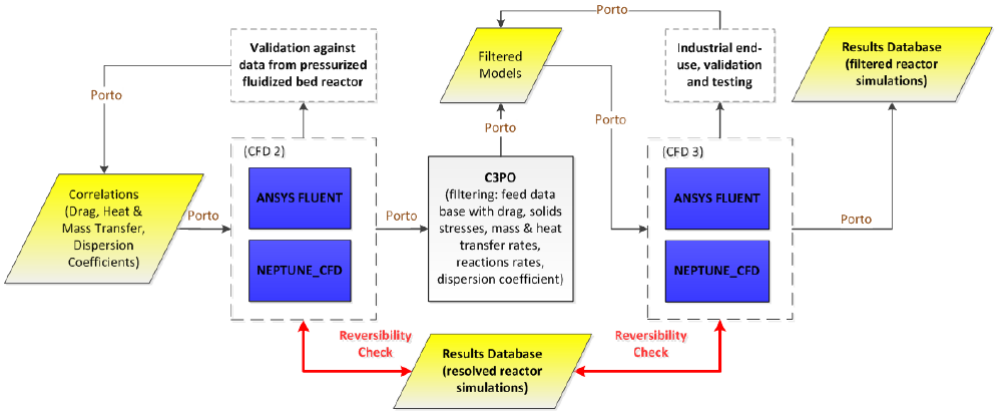


Figure 1.1.: Typical flow of information from a *resolved* simulation model (here the "CFD 2" level) to a *unresolved* simulation model (i.e., the "CFD 3" level) as applied in the *NanoSim* project.

though the main objective of *NanoSim* is to provide a simulation platform for the accurate design of gas-particle CO_2 capture technologies, the resulting simulation platform (formed around the tool *Porto*) is much wider applicable.

A key tool on the *NanoSim* simulation platform is CPPPO (Compilation of fluid/particle post processing routines) [68], which was developed within the present thesis (see Chapter 2 for details). As depicted in Figure 1.1, the role of CPPPO is to act as scale bridging tool by extracting relevant information from *resolved* simulations to derive appropriate closures for *unresolved* simulations. Here we use the terms *resolved* and *unresolved* in the most general context: a quantity that is modelled (i.e., that requires a closure) in an *unresolved* simulation is computed from the solution of a *resolved* simulation. Therefore, the two terms generally refer to some specific quantity of interest, e.g., momentum or heat transfer.

In this sense, CPPPO can be applied as a bridging tool regardless the details of the resolved or unresolved formulations, and, thus, can be seen as a universal tool.

1.2. Euler-Lagrange and Euler-Euler formulations

Coarse grained theories for disperse multiphase flows are generally formulated using a Particle-Unresolved Euler-Lagrange approach (PU-EL), or an Euler-Euler (EE) approach. In the first approach, only the fluid phase is coarse grained, so that a grid with a spacing on the order of several particle diameters can be used (smaller grid sizes are not useful, since only *average* particle information is considered when solving the fluid equations). The interphase transfer has to be modelled, but the particle cloud is still a discrete entity where each discrete element correspond to one particle.

On the contrary, in Euler-Euler formulations both phases are coarse grained, and therefore there is no definition of a single independent particle. Particles are treated as a continuum, and their interactions are accounted statistically.

In this work, we focus on developing closure models for PU-EL formulations from Particle-Resolved Direct Numerical Simulations (PR-DNS). In PR-DNS all relevant flow scales are taken into account, and modeling errors may only originate from the algorithms employed to obtain a numerical solution but results are consistent with the fundamental assumptions of the model (e.g., the fluid's rheology, particle shape, etc.). There are two main reasons why approaching first PU-EL models (via PR-DNS) is preferable compared to directly approaching EE:

i EE models are based on a continuum, i.e., a statistical description of the discrete phase. This requires considering a "statistically significant" number of particles. On the contrary, in PU-EL formulations the number of discrete particles within a grid cell can vary, but in general is much lower than the equivalent in EE models. Therefore, PU-EL can be seen as an intermediate step between PR-DNS and EE, which leads to the second point:

ii In PU-EL the thermodynamical history of each particle is tracked, together with the particles' internal state (e.g., the particle's chemical composition) and per-particle transfer coefficients. Therefore, PU-EL models provide a much more powerful tool to study multi-physics and reactive systems compared to EE models. For the latter a continuum description of such processes becomes necessary, significantly complicating the analysis. For example, PU-EL models allow to study the effect of thermal radiation in gas-particle systems based on detailed particle-to-particle radiative heat transfer models [36]. This example illustrates that the relative arrangement of particles may be of significant importance, a factor that cannot be accounted for in current EE models.

1.3. The closure problem

As mentioned in the previous section, coarse grained theories are generally unclosed in the sense that one or more terms are a function of the microscopic fields (i.e., a solution of the resolved equations). The unclosed terms in case of PU-EL include (i) the stress arising from volume averaging the flow field (as in turbulence theory), (ii) scalar dispersion terms, as well as (iii) interphase transfer terms originating from the transformation of the conditions at the particles' boundaries into source terms for the governing equations in PU-EL. Term (i) is typically of secondary importance for dense flows - it will not be discussed further. Also, term (ii) is typically of subordinate importance, and is only briefly touched in the present thesis. The last terms are, however,

significant. Generally, these terms are proportional to the integral over the particles' interface, considering the gradients of microscopic variables:

$$Q \propto \frac{1}{S} \int_S \frac{\partial \varphi}{\partial n} \mathrm{d}S \tag{1.1}$$

Where S is the particles' surface, φ is a microscopic variable, and $\partial/\partial n$ is the surface normal derivative. Generally, one introduces the following ansatz to arrive at expressions that connect microscopic and coarse-grained quantities:

$$\varphi = \varphi' \left(\varphi_S - \langle \varphi \rangle \right) \tag{1.2}$$

Where $\langle \varphi \rangle$ is a (smooth and slowly varying in space) coarse grained variable, φ_S is a constant surface value, and φ' is a rescaled microscopic variable. Such scaling allows to rewrite the interphase transfer term as:

$$Q \propto \frac{\left(\varphi_S - \langle \varphi \rangle \right)}{S} \int_S \frac{\partial \varphi'}{\partial n} \mathrm{d}S = \Lambda \left(\varphi_S - \langle \varphi \rangle \right) \tag{1.3}$$

Where Λ is an interphase transfer coefficient which is generally a function of the Reynolds number, the concentration of neighbouring particles, and some parameters of the coarse graining operator (for example the characteristic length Δ of the filtering kernel). Such approach requires finding a proper functional form for Λ, which is the ultimate goal of most of the work done in the present thesis.

However, above we assumed that $\langle \varphi \rangle$ does not vary over S in order to take it outside the integration operator. This is justified when the characteristic length of the dispersed phase ℓ (e.g., the particle diameter in case of spherical particles) is much smaller than the characteristic length of the filtering kernel Δ, i.e., $\ell << \Delta$ (see for example the book of Whitaker [107]). Therefore, the ansatz shown in Eqn. 1.2 is valid, for example, in the case of point particles. Unfortunately, this is not always the case in PU-EL models that consider particles with a finite diameter, and for which one can have $\ell \approx \Delta$.

1.4. Objectives of this thesis

The main objective of this thesis is to provide insight with respect to closure models for PU-EL simulations by means of coarse graining data from PR-DNS. Specifically, we are particularly interested in heat and mass transfer phenomena and their modelling by means of an interphase transfer coefficient. Most important, we consider volume averaging with a characteristic filter length on the order of a few particle diameters (i.e., one could call this 'fine-grained' PU-EL).

The present thesis aims on accomplishing five major objectives, each one aiming on significantly advancing the current state of the art, or on making new open-source software tools available to the community for future investigations.

1.4.1. Objective I: Developing and testing the scale bridging tool CPPPO

Generally, a resolved simulation results in a huge amount of data, often scattered among several dozens or hundreds of processors. Furthermore, only statistical information is generally required to develop closure models, which makes the storage of full simulation data useless and expensive. Therefore, a data processing library that can be interfaced with virtually any solver and that generates the desired statistical information from a data stream (during a parallel simulation run) is of primary importance. CPPPO (Compilation of fluid/particle post processing routines) is developed and integrated with the CFDEMCoupling®[cfd] library by means of a general OpenFOAM®[ope] interface. The library is able to perform "on-the-fly" filtering operations (i.e., alongside with the execution of a solver), and to output data at runtime with high parallel efficiency and low RAM usage. Also, test cases to show the accuracy of the computation are provided.

1.4.2. Objective II: Establishing a novel algorithm for PR-DNS that allows heat and mass transfer simulations

The Eulerian-Lagrangian library CFDEMCoupling®is extended with a new algorithm for PR-DNS which makes use of elements from the Immersed Boundary (IB) [73] and the Fictitious Domain (FD) method [86] to simulate momentum, heat and mass transfer in gas-particle flows. The new algorithm is named Hybrid Fictitious Domain-Immersed Boundary method (HFD-IB). The algorithm features (i) a Taylor expansion-based boundary layer reconstruction to compute cell values that satisfy the boundary conditions to a desired order, and (ii) an internal compensation of boundary fluxes to correctly evaluate the interphase fluxes. The algorithm is coded for massively parallel applications and it is verified against several test cases showing outstanding accuracy on coarse grids. Furthermore, the HFD-IB method can be easily extended to impose general boundary conditions, e.g., a fixed flux condition.

1.4.3. Objective III: Assessing scale separation for heat and mass transfer in homogeneous suspensions

We aim on providing more insight on the phenomena of heat and mass transfer in homogeneous (but random) gas-particle suspensions. Specifically, the problem of scale

separation [91] is studied using analytical techniques to identify the regimes under which it still makes sense to use a formulation in terms of transfer coefficients (i.e., equation 1.1) to close the interphase source term. In fact, below certain values of the Peclet number and voidage, the interphase transfer happens so quickly that the fluid phase gets saturated (i.e., the fluid assumes almost the same temperature or concentration as that present near the particle surface) within one or two particle diameters. This means that the coarse grained temperature (or concentration) field is far from being constant over the particle surface. Furthermore, the heat transfer coefficient in the saturated regime would be so large that it would pose stability problems in coarse-grained simulations in case explicit flux coupling is performed. In such situations, one could just set the coarse grained fluid temperature (or concentration) to the particle surface value and distribute the transferred heat among the particles based on the flow direction.

1.4.4. Objective IV: Closure models for momentum, heat and mass transfer in bi-disperse systems

We make use of PR-DNS to develop and test the validity of closure models for momentum, heat and mass transfer in bi-disperse gas-particle suspensions. Specifically we explore the possibility of exploiting an analogy between momentum and heat/mass transfer to implement computationally more efficient closures. Those closures rely on the idea that heat (or mass) transfer coefficients are calculated from the drag coefficient (which has to be calculated anyway). Also, we perform a statistical analysis of the deviation from the values predicted by the correlation and we provide, by means of a proper scaling of the variables, universal distribution functions that can be used in stochastic models. Furthermore, we asses the validity of existing correlations, specifically that of Beetstra et al. [11], with respect to predicting the drag in bi- or polydisperse suspensions. Thereby, a focus is on the correct treatment of the mean pressure gradient contribution that must be subtracted from PR-DNS data in order to compute a meaningful drag force.

1.4.5. Objective V: Quantifying wall effects in gas-particle suspensions

The description of momentum, heat and mass transfer between particles and a gas in the proximity of walls still lacks fundamental understanding. By means of an array of PR-DNS for such systems, we find that the main cause of such disturbances is the anisotropic particle distribution induced by the wall. Therefore, we aim on characterizing the wall-near "particle-induced boundary layer", in which flow and temperature (or concentration) fields exhibit peculiar profiles. Also, by means of a set of Discrete

Element Method-based (DEM) simulations, we aim on obtaining an accurate expression for the wall normal particle volume concentration profile as a function of the bulk concentration. Furthermore, we provide correlations for the flow and temperature (or concentration) fields within the particle-induced boundary layer, and we study the interphase transfer coefficient.

1.5. Thesis content

1.5.1. Highly efficient spatial data filtering in parallel using the library CPPPO

In Chapter 2 we introduce the main tool used throughout the rest of the present study: CPPPO. CPPPO allows volume averaging (i.e., filtering) operations to be performed on-the-fly (i.e., while a simulation is running) by acting directly on data pointers. After a brief introduction that motivates the need and use for such a library, we illustrate the heart of the library: the FSB loop (Filtering-Sampling-Binning) which allows a straight pipeline of operations that limits the amount of required RAM. Subsequently, we detail how CPPPO interfaces with OpenFOAM®and how it can be linked to any other simulator. We also introduce the basics of spatial filtering, as well as the novel convergent and divergent filtering algorithms featured in CPPPO. Parallel algorithms are described in detail with particular emphasis on the number of required operations. Finally, we present test calculations and the parallel scalability analysis together with the application of the library to PR-DNS.

This work was published in *Computer Physics Communications* [68].

1.5.2. Consistent closures for Euler-Lagrange models of bi-disperse gas-particle suspensions derived from Particle-Resolved Direct Numerical Simulations

Chapter 3 deals with the study of momentum, heat and mass transfer in bi-disperse gas-particle suspensions. After briefly introducing the governing equations, we introduce the concept of fluid saturation which is akin to the problem of scale separation. We develop an analytical model to predict the range of parameters under which saturation occurs and we verify the model against PR-DNS.

We also introduce the HFD-IB method, and describe the newly implemented algorithm in detail. The algorithm is benchmarked in a number of test cases involving one or more particles.

Finally, we perform several simulations of bi-disperse gas-particle suspensions considering an array of different particle concentrations, Reynolds numbers, and small-to-

large diameter ratios. We first focus on the particle-based drag force, and we then move on to the particle-based Nusselt number. We conclude Chapter 3 with considerations regarding the role of the pressure gradient contribution in the average drag force for poly-disperse systems.

This work was published in the *International Journal of Heat and Mass Transfer* [69].

1.5.3. Momentum, heat and mass transfer simulations of bounded dense mono-disperse gas-particle systems

Moving to Chapter 4 we investigate the effects of adiabatic walls on momentum, heat and mass transfer in mono-disperse suspensions. First, we introduce the relevant equations, we discuss the post-processing, and we describe the numerical settings. Secondly, we present results from a large number of simulations involving different values of the Reynolds number and the particle volume concentration. We focus on the modelling of wall induced disturbances by means of expressions for the wall normal profiles, starting with the particle concentration. Then, we model (i) the temperature (or concentration), and (ii) the flow field in the proximity of the wall. Finally, we investigate the effect of wall disturbances on the interphase transfer coefficients for momentum and heat (or mass).

This work has been recently submitted to the *International Journal of Heat and Mass Transfer* and is currently under review.

1.5.4. Conclusions and Appendix

In Chapter 5 we draw overall conclusions from the results of the present thesis, and provide an outlook for further investigations. Finally, in A we show how the HFD-IB algorithm can be extended to impose general boundary conditions.

2. Highly efficient spatial data filtering in parallel using the opensource library CPPPO

CPPPO is a compilation of parallel data processing routines developed with the aim to create a library for "scale bridging" (i.e. connecting different scales by mean of closure models) in a multi-scale approach. CPPPO features a number of parallel filtering algorithms designed for use with structured and unstructured Eulerian meshes, as well as Lagrangian datasets. In addition, data can be processed on the fly, allowing the collection of relevant statistics without saving individual snapshots of the simulation state. Our library is provided with an interface to the widely-used CFD solver "OpenFOAM", and can be easily connected to any other software package via interface modules. Also, we introduce a novel, extremely efficient approach to parallel data filtering, and show that our algorithms scale super-linearly on multi-core clusters. Furthermore, we provide a guideline for choosing the optimal Eulerian cell selection algorithm depending on the number of CPU cores used. Finally, we demonstrate the accuracy and the parallel scalability of CPPPO in a showcase focusing on heat and mass transfer from a dense bed of particles.

2.1. Introduction

Many relevant physical systems involve a wide spectrum of length scales that interact in a non-linear way. Hence, an accurate prediction of all relevant phenomena in these physical systems in engineering-scale equipment is challenging due to the inability to directly simulate certain small-scale phenomena. One example are dense fluid-particle flows, which are usually encountered in many industrial processes: details of the flow around each individual particle cannot be directly predicted, but are modeled instead, e.g., by a drag coefficient. In addition, the simulation of flows in engineering-scale equipment often necessitates the use of Eulerian models on comparably coarse computational grids, i.e., the continuum hypothesis has to be adopted, and small-scale information is lost. Consequently, closures have to be derived to account for a variety of phenomena, e.g., fluid-particle and particle-particle interactions, or unresolved turbulent motion. In order to accurately model such flow problems, a so-called

multi-scale approach is often used [101, 102]. The multi-scale approach consists in decomposing the original problem into various levels of description, each one involving a typical range of length scales. Then, simulations on the most detailed level (typically on the smallest length scales) are performed to extract quantities which can be used in coarse grained models. In a coarse-grained model, only coarse flow structures are resolved (where "coarse" means on the same order of the mesh size or larger). Transport processes occurring at smaller scales are considered by closures, e.g., filter-size dependent closures for scalar dispersion rates, inter-phase exchange rates, or effective stresses. Nowadays, these closures are often derived from simulations on a more detailed level, and not from experimental data. This process is normally referred to as "coarse-graining", and has become a major trend in a variety of scientific disciplines [50, 56, 60, 83].

CPPPO (i.e., the "Compilation of fluid/Particle PostPrOcessing routines") has been developed as a flexible library that provides a collection of efficient algorithms to perform these coarse-graining operations. The main purpose of CPPPO is to act as a tool for "scale-bridging", regardless of the effective scale range, the model formulation, or the simulator used. CPPPO is designed to interact with any purely Eulerian, or mixed Eulerian-Lagrangian data set. This allows one to apply CPPPO for a number of different scientific and engineering applications. For example, this includes the verification of Large Eddies Simulation (LES) models based on differential filtering [39], anisotropic filtering of flow data [12], or the development of sub-grid stress tensors for LES. In what follows, however, we focus on a multi-scale scenario applied to study dense fluid-particle flows in order to outline how CPPPO can be used for scale bridging.

At the most fundamental level, Direct Numerical Simulations (DNS) are used to derive coefficients for heat, momentum and mass transfer in dense particulate systems [22–24, 35, 88, 110]. Typically, a certain number of realizations for the case studied are needed [92, 93] in order to derive statistically meaningful correlations. This approach requires to process data from large datasets in order to compute averaged (mean) quantities (which are needed to evaluate transfer coefficient), standard deviations, or orther statistics like the distribution of the angle between two vector fields [54]. Also, time-averaged quantities are often used to evaluate transport coefficients in fluid-particle systems [27]. In case of non-equilibrium systems (like fluidized beds, in which instabilities are system-inherent [51]), the modelling of drag and stresses may require higher-order closures. Unfortunately, these models are difficult to develop [28]. Another example can be found in the field of granular materials: here the calculation of effective transport properties requires the evaluation of filtered fields and fluxes [110, 112]. The same approach, i.e., considering statistical data of, e.g., the velocity fluctuations, can be used on intermediate length scales when deriving models for engineering applications. Typically, this results in an Eulerian "grid coarsening" approach, e.g., by deriving models for the sub-grid-scale fluid-particle agitation [50]. Favre averaging of relevant fluid variables (e.g., the fluid velocity), and fluid-particle interactions

(e.g., the coupling force) is generally adopted to derive these closures [74, 81] . In case an Euler-Lagrange approach is followed, the effect of "particle coarsening" (i.e., each simulated particle is a proxy for a prescribed number of particles named parcel) has to be taken into account as well [81]. All these examples demonstrate that spatial averaging operation on Eulerian and Lagrangian data sets of of key importance for multi-scale model development nowadays.

In principle, the application of an appropriate filtering strategy is straight forward once the fluid-particle flow simulator is available. However, filtering of scientific data and "coarse-graining" poses several challenges from the software point of view. For example, spatio-temporal averages have to be computed across different processors for the (typically large) filter sizes. Typically, filter sizes to be used when filtering DNS data of fluid-particle flows have a size of two to five particle diameters. Thus, filtering is typically performed over 20^3 to 50^3 Eulerian grid cells, often located on different processors. This requires an algorithm that can deal with parallel communication, and that does not require mirroring the full field information on every processor. The latter is of course a feasible approach, however, when aiming on large-scale simulations this would require an excessive use of RAM. At the same time, parallel communication of local field values requires significant network resources due to the large amount of data to communicate. Another issue is the amount of data generated during the simulation run: the implemented algorithms should be able to work "on the fly" in order to process (i.e., time-average) data from different time steps. Also, there should be a clear separation (in terms of namespaces and classes) between the simulator and the post-processing utility such that the latter can be linked to different simulators. Finally, the library should be modular in order to make the addition of new features as easy as possible.

In the present work we present the library CPPPO that addresses the above challenges. While most of the existing filtering algorithms documented in literature were developed for image processing applications [71, 97], CPPPO is able to handle three-dimensional data sets in parallel. Specifically, CPPPO can process data sets from complex geometries and unstructured meshes. Another important difference with respect to image filtering is the type of data: images just deal with a limited set of scalar quantities represented by integers (i.e., the color intensity). In contrast, relevant simulation results are vectorial data that is represented by floating point values. Furthermore, CPPPO handles cell/particle selectors separately from filtering routines to allow an easier implementation of custom filtering kernels, and a higher flexibility in the choice of the algorithm. In summary CPPPO features a flexible code architecture tailored for scientific computations on high performance clusters. The library is designed to perform three kind of operations on the data set:

- Filtering: field volume averaging that can be performed on every cell (for the Eulerian filter option) or at specific user-defined locations (for the Lagrangian

filter option). The user can customize the kernel function (see section 2.3) by adding an arbitrary amount of weights (which have to be scalar fields) or by modifying the kernel's functional form.

- Sampling: This operation allows to take samples from the domain (results from the filtering operations may be sampled as well) and relate each sampled value with one or more markers. For example, CPPPO can sample a spatially-filtered fluid velocity field at every cell using the fluid phase fraction as marker.

- Binning: Data collected from sampling operations can be collapsed using binning operations. The marker field values are discretized according to the user input, and a conditional averaging calculation is performed on the sampled field. This data collapsing allows to reduce the amount of data that needs to be written to disk in case the user is only interested in correlations between the means of the sampled quantities and one (or more) markers.

For every user-specified filter (i.e., kernel function), the library performs these three operations in sequence (see Figure 2.1). We will refer to this loop as the FSB loop (Filtering, Sampling and Binning).

Our paper is structured as follows. In section 2.2, the basic data structure and the range of applicability of the library are discussed. In Section 2.3, the basics of filtering and Favre-averaging are introduced, as well as the novel *divergent* approach to parallel filtering implemented in CPPPO. Available routines for statistics calculation are described in Section 2.4, and the *sampling/binning* process is outlined. The implementation of the algorithms for cell selection and filtering is presented in Section 2.5, with emphasis on the parallelization strategy. In Section 2.6 several simple test cases for code verification are described, and in Section 2.7 we present a parallel scalability analysis for CPPPO. We also present a typical application of CPPPO in section 2.8 for the evaluation of heat and mass transfer coefficients. Finally, conclusions are summarized in Section 2.9.

2.2. Library interface to simulators

Before detailing the algorithms available in CPPPO, it is worth to describe the conditions that a simulator needs to meet in order to be linked to CPPPO. In the following, we will also describe the general set of data to which the algorithms can be applied.

2.2.1. Basic data structure

CPPPO can be coupled to simulators using finite volume, finite difference, finite elements, spectral, lattice Boltzmann or smoothed particle hydrodynamics methods. These

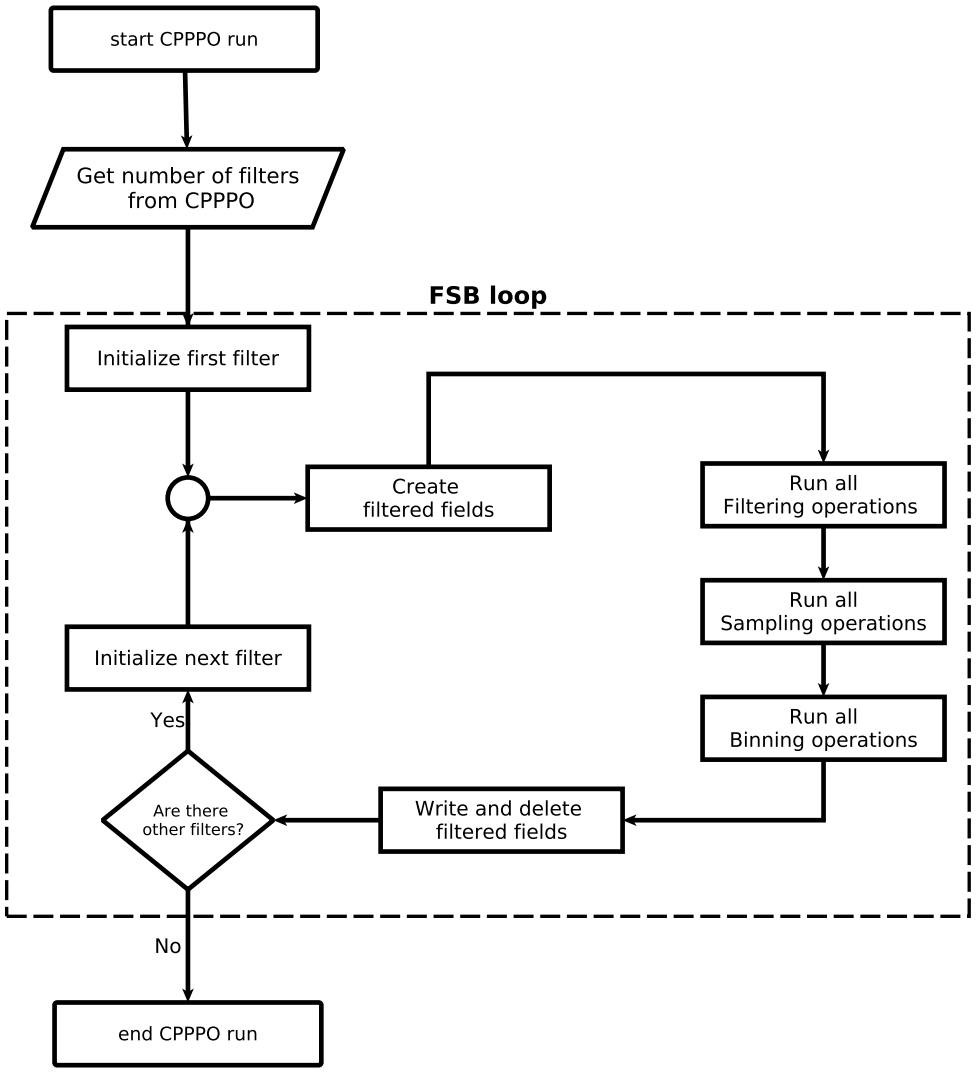


Figure 2.1.: Filtering-Sampling-Binning loop: the structure of a typical CPPPO run.

methods normally consist in solving partial differential equations within a computational domain Ω_c of volume V_{Ω_c}. The library requires the simulator to provide the following data:

- A set of nodes (points) lying within Ω_c, each one identified with a set of three spatial coordinates.
- A set of scalars representing the measure of the spatial volume surrounding each node. Notice that, in order to correctly calculate spatial filtered quantities, the volumes must not overlap and their sum must be equal to the total computational volume V_{Ω_c}.

- A set of scalars representing field values (e.g., pressure, temperature, velocity components, species concentration, *et cetera*) at each node.

In the following, we will refer at the entity composed of a node and the associated volume as *cell*. The union of all cells is termed as *mesh* and field values associated with each node are named *cell values*. Mesh and cell values form the *Eulerian* data in CPPPO.

Notice that CPPPO does not require any information regarding cell shape or surfaces. The topological details of Ω_c or the original mesh are not considered and a cell is considered to lie within a certain region if its node is included in that region. Thus, filtering operations are affected by errors due to: (i) cell shape (or cell quality) and (ii) the ratio between cell size and filter size. However, this is not really an issue since (i) is generally controlled in the simulator in order to reduce numerical errors in the computation (before running CPPPO) and (ii) should always be low due to cell shape regularity required in (i) and the fact that filtering volumes are often much larger than smallest field structures (which normally require lumped nodes and thus, small cells, to be sufficiently resolved).

Additionally, the user can provide a set of *Lagrangian* (particle) data which may represent particle clouds or sampling probes. While probes are just defined by their position in Ω_c, particle clouds can be defined with several more properties (like particle diameter, velocity, torque, forcing terms, and scalars) which can be passed to CPPPO directly from the simulator. These properties can be used, for example, to calculate inter-phase transfer coefficients "on-the-fly".

Further information on data structure can be found in the CPPPO documentation.

2.2.2. General linking architecture

CPPPO has his own way of handling field, mesh and particle data which, in general, does not have to conform to any particular simulation software. In order to exchange data between CPPPO and a simulator, an *interface library* is required. The interface library is specific for every simulator, and it will typically rely on the simulator's classes and namespaces. CPPPO comes with an interface library for OpenFOAM®.

The role of the interface library is to get pointers to memory locations of all relevant data fields (e.g., holding mesh and particle information), and pass them to the core library in an appropriate format. For example,vector fields require pointers to the array of doubles containing each component. Similarly, for the mesh data pointers to coordinates and volumes of every cell are transferred. Some simulators store all the components of a vector or all the mesh point coordinates in just one array, one example of which is OpenFOAM®. CPPPO allows to specify a displacement between the component values (for example a displacement of 3 for three dimensional data) in order to automatically take into account this data structure. In addition, the interface

library allocates space for the filtered fields (i.e., those fields that store the result of filtering operations) and register them (i.e., pass the required pointers) into the CPPPO core library. Thus, the interface library performs additional storage operations using the simulator namespace. In this way, the resulting filtered fields can be saved and used in the simulator format, which positively contributes to the usability of CPPPO.

The CPPPO core library performs filtering, sampling and binning operations using the memory allocated by the interface library and the simulator. The core library allocates heap memory for the filtering operations. For example every Eulerian filtering operation (i.e., where a filter is centred at every cell's center) requires an array of doubles with size equal to the number of cells to store intermediate values (see Section 2.3 for details). This data flow is summarized in Figure 2.2.

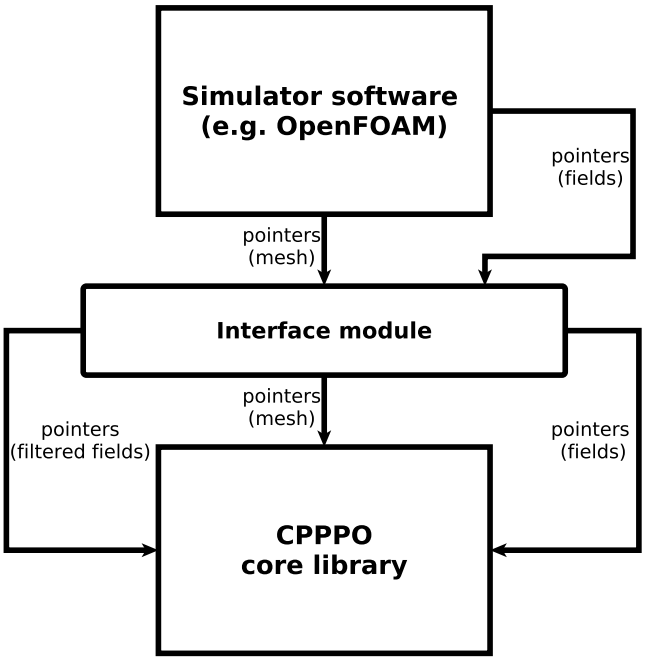


Figure 2.2.: Flow of information from the simulator to the interface and, finally, to the CPPPO core library.

2.2.3. Parallel data handling

The CPPPO core library represents the domain as a set of nodes which, in the case of the OpenFOAM® interface, correspond to the cell centers. CPPPO ignores the cell shape, but requires the interface to provide cell volumes. When a filtering or searching operation is performed, the cells with the closest cell center are selected.

CPPPO is designed for applications holding parallel-decomposed data (i.e., a physical domain is subdivided into smaller subdomains) where the domain subdivision is made of boxes whose faces are perpendicular to the corresponding Cartesian axis. This is an important requirement since it ensures that CPPPO exactly knows the position of every processor boundary. Every box can contain a different number of cells, or have different size in any direction. However, in order to keep an high parallel efficiency, it is recommended to keep the same number of cells for every processor.

CPPPO is parallelized using MPI [66], and can be run in parallel with any simulator that splits the computational domain in several box-shaped subdomains. Since CPPPO is designed for spatial filtering, data should be decomposed according to their position in space. This is true in almost the totality of currently available simulators for CFD either using a finite volume approach (e.g., OpenFOAM®,ANSYS FLUENT®, *Code_Saturne*®, STAR-CCM+®, AVL FIRE®, ecc.) or not (e.g., Palabos®, Nektar++®, Nek5000®, ecc.).

CPPPO requires each process to have its memory address space. Thus, simulators which rely on a GPU hardware architecture, or OpenMP may not be suitable for linking with CPPPO at the current stage. Also, RMA (remote memory access), and MPI one-sided communications in general, will most likely create problems in case of passive synchronization. This is because a processor would have to call `MPI_Lock` to himself in order to access local data inside an MPI window. In summary, we recommend careful testing of the interface routines when using CPPPO in connection with simulators that rely on RMA or one-sided communication. For the standard interface (to OpenFOAM®) we only require that:

- The global domain is decomposed in box-shaped subdomains,
- Each process holds only one subdomain, and that
- Accessibility of local data is ensured.

In case a user wants to link CPPPO to a software that does not meet the above requirements, a more careful design of the interface library is necessary. For example, we have implemented an interface to CSV data files that performs the domain decomposition, and does not require the input file to be already decomposed in parallel. Also, the above mentioned issue with not accessible local data can be circumvented by copying the shared data in separate arrays, thus creating a compact addresses set for the whole subdomain. We next detail on some practical aspects when implementing such a new interface library.

2.2.4. Coding an interface library

In case the user wants to link CPPPO with a software, he/she will have to use the existing OpenFOAM® or CSV interface module, or code a new interface library. For

the latter, the OpenFOAM® interface library is a good template. In the following, we will refer to this library to illustrate the main steps needed to code an interface library.

Remember that all the functions that are called in an interface module are summarized in *core/c3po.h*.

- An interface should create an instance of the *c3po* class.
- An interface should be able to access pointers to mesh data and pass them to CPPPO, which then use them to calculate and communicate other required quantities. An example can be found in the file *interface_OF/mesh_check.C* . Notice that the OpenFOAM® interface provides a public function that can be called in the simulator. This is done to track the evolution of dynamic meshes and ensure that pointers handed over to CPPPO are always valid.
- An interface should possess a *run* function which (i) registers (i.e., handles relevant pointers to) the required fields in CPPPO according to their data format, and (ii) starts the FSB loop of CPPPO. The *interface_OF/c3po_OF_interface.C* file provides an example therefore.
- During the FSB loop, the interface should be able to allocate heap memory for the required fields (e.g., filtered fields), and delete them when necessary. Note that CPPPO will already provide suitable names to label these new fields.

In general, the amount of time required to code a new interface can vary significantly with the architecture of the simulation software and the programmer's skills. Thus, it is useful to first study the architecture of the simulator and CPPPO, e.g., by using the training material available at `http://www.tugraz.at/en/institute/ippt/downloads-software/`.

2.3. Spatial filtering

Spatial filtering can be considered as a subset of the general operation [79]:

$$\overline{\phi}(\mathbf{x},t) = \int K(\mathbf{x}-\mathbf{x}',t-t')\,\phi(\mathbf{x}',t')\,d\mathbf{x}'dt' \tag{2.1}$$

Where ϕ is a generic field, K is the kernel function, and the integration is performed over the whole space and time domain. An important property of the kernel function is normalization, thus:

$$\int K(\mathbf{x}-\mathbf{x}',t-t')\,d\mathbf{x}'dt' = 1 \tag{2.2}$$

CPPPO will automatically normalize your kernel function. In the case of spatial filtering, the kernel function is expressed as:

$$K\left(\mathbf{x}-\mathbf{x}',t-t'\right)=K\left(\mathbf{x}-\mathbf{x}'\right)\delta\left(t-t'\right) \tag{2.3}$$

Thus, the argument is integrated over space only:

$$\overline{\phi}\left(\mathbf{x},t\right)=\int K\left(\mathbf{x}-\mathbf{x}'\right)\phi\left(\mathbf{x}',t\right)d\mathbf{x}' \tag{2.4}$$

The corresponding fluctuating field ϕ'' is defined as:

$$\phi''\left(\mathbf{x},t\right)=\phi\left(\mathbf{x},t\right)-\overline{\phi}\left(\mathbf{x},t\right) \tag{2.5}$$

CPPPO solves equation 2.4 at every cell centre $\mathbf{x}$, or alternatively at predefined positions $\mathbf{r}$. The kernel used in this study is the top-hat kernel:

$$K\left(\mathbf{x}-\mathbf{x}'\right)=\prod_i\frac{\mathcal{H}\left(\frac{\Delta_i}{2}-|x_i-x_i'|\right)}{\Delta_i} \tag{2.6}$$

Where Δ_i is the filter size in the i-th direction of a Cartesian coordinate system and $\mathcal{H}$ is the Heaviside function. CPPPO also features a top-hat kernel in a spherical coordinate system. It has to be noticed that this kernel, while acting as a sharp cut-off in the physical space, features a smooth cut-off in the spectral space [21], resulting in a wave number overlap between $\overline{\phi}$ and ϕ''.

2.3.1. Favre filtering

The Favre averaging technique [31, 32] consists in a decomposition of the flow field variables in terms of density-weighted variables:

$$\widetilde{\phi}=\frac{\overline{\rho\phi}}{\overline{\rho}} \tag{2.7}$$

Favre averaging is often used for multiphase flows to derive filtered transport equations, and to decouple the phase fraction from the flow variables. In addition, variance and covariance calculations are of major importance to evaluate the components of the SGS (Sub Grid Scale) stress tensor, or SGS fluxes. CPPPO is able to perform Favre averaging and Favre variance and covariance calculation for every cell inside the domain, or at specific user-defined positions. To illustrate the equivalence between the

variance (or covariance) and the components of the SGS stress tensor, we consider the definition of the latter as:

$$\tau_{ij}^{sgs} = \widetilde{u_i u_j} - \widetilde{u}_i \widetilde{u}_j \tag{2.8}$$

Here u_i is the velocity of in the spatial direction i. For example, the diagonal elements of the tensor shown in Eqn. 2.8 can be obtained from the Favre variance as follows:

$$\begin{aligned} Var(u_i(\mathbf{x})) &= \int G(\mathbf{x}' - \mathbf{x}) \left(u_i(\mathbf{x}') - \widetilde{u}_i(\mathbf{x})\right)^2 d\mathbf{x}' \\ &= \int G(\mathbf{x}' - \mathbf{x}) u_i^2(\mathbf{x}') d\mathbf{x}' + \widetilde{u}_i{}^2(\mathbf{x}) \\ &- 2\widetilde{u}_i(\mathbf{x}) \int G(\mathbf{x}' - \mathbf{x}) u_i(\mathbf{x}') d\mathbf{x}' \\ &= \widetilde{u_i u_i} - \widetilde{u}_i \widetilde{u}_i = \tau_{ii}^{sgs} \end{aligned} \tag{2.9}$$

Where $G(\mathbf{x}' - \mathbf{x})$ is a function representing the top-Hat kernel and the Favre averaging operation. The other components of τ_{ij}^{sgs} can be calculated in a similar manner using the Favre covariances. The same approach applies to evaluate SGS fluxes. Notice that CPPPO allows the user to define an arbitrary number of weighting fields for the kernel function, and hence offers the freedom to compute filtered quantities for virtually any application.

2.3.2. Convergent and divergent filtering algorithm

Since the calculation of filtered quantities implies long range interactions, processor communication has to be taken into account when designing an algorithm to numerically solve Eqn. 2.4. In case Eqn. (2.4) is projected into the discrete space (and when considering a top-Hat kernel), it can be written as:

$$\overline{\phi_i} = \frac{\sum_{j=0}^{j=N_f} v_j \phi_j}{V_f} \tag{2.10}$$

Where the sum is over all N_f cells inside the filter region, v_i is the volume of the i-th cell, and V_f is the total filter volume. The extension of the above equation to Favre averaging or arbitrary weighted averaging is obvious. The above calculation has to be performed for every cell i in order to compute a complete field of the filtered quantity. Considering Eqn. 2.10 it is clear that, before the calculation can start, it is necessary to evaluate which cells are inside the filter.

The approach described by equation 2.10 is what we call the *convergent approach* for filtering. This is because, after the list of cells inside the filter is assembled, data

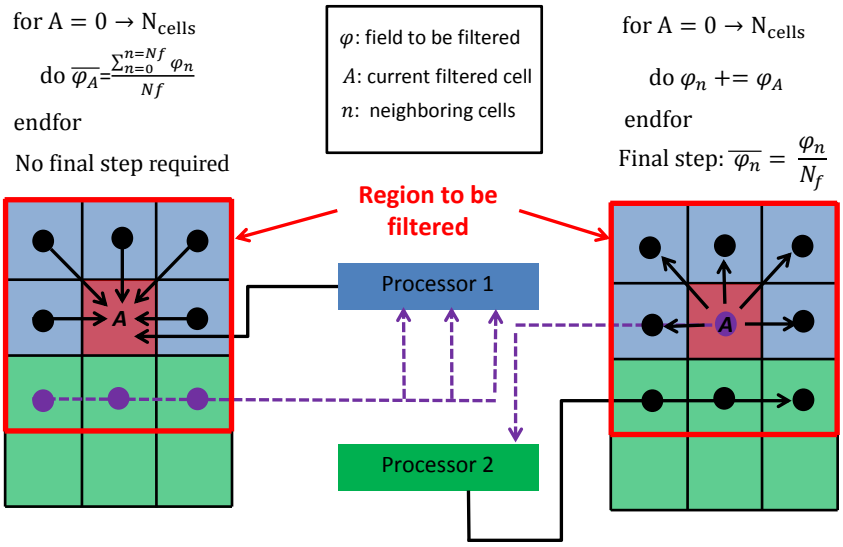


Figure 2.3.: Convergent approach (left) and divergent approach (right) for filtering. Continuous arrows represent intra-processor operations while dashed arrows indicate data exchange between processors. Processor domains are identified with the owner color, red cells represent the current cell to be filtered. Dots represent cell centres involved in local data operations (black) and parallel data operations (purple). In the picture N_{cells} is the total number of cells per processor (for simplicity we consider the same number N_{cells} in each processor) and N_f is the number of cells within the region to be filtered. In the divergent approach, field values at cell A are spread to the neighbouring cells while the opposite occurs in the convergent algorithm. The divergent algorithm also requires the communication of less data.

from the neighbouring cells is passed to the location where the filter is centred. This approach requires (for every location to filter) $N_f - 1$ summation and multiplication operations, and one division operation. The amount of multiplication operations could be reduced in case the multiplied field values are stored and then communicated. However, this would require additional memory. Also, communication of the cell list and values with other processors has to be performed. As shown in figure 2.3 (left panel), the convergent approach requires the communication of every required cell data owned by another processor. In our case three values need to be transferred from processor 2 to processor 1. The convergent approach is the most basic approach for spatial filtering, and most of the available filtering algorithms used for image processing are based on it.

In order to reduce the computational load and enhance parallel efficiency, we developed a novel approach named the *divergent approach*. The divergent algorithm does not evaluate the filtered value at any position sequentially, but updates the filtered fields at every step, and ends with a final division step. Specifically, every step consists of:

(i) selecting a cell from the computational domain.

(ii) creating a list of cells located in the region to be filtered around the selected cell.

(iii) multiplying the field value at the selected cell with the required weight (i.e., cell volume or mass density). Note, when the filter size tends to the domain size only one cell value needs to be stored and communicated instead of having to allocate and communicate the whole field.

(iv) communicating and adding the multiplied field values to all cells inside the cell list

The loop has to be repeated for every cell inside the domain. At the end, one last step is needed to divide the values of filtered fields by the filter volume (or by the summed weights in case of Favre averaging). This approach requires (for every cell) $N_f - 1$ additions, but only one multiplication. The number of divisions in the final step equals the total number of cells. Overall, less multiplication operations are required in the divergent approach compared to a convergent approach (without allocating memory for the multiplied fields as explained above). Most important, the key advantage of the divergent algorithm over the convergent algorithm is the amount of data that needs to be communicated. As shown in figure 2.3 (right panel), in the divergent algorithm the direction of communication is reversed, and just the field value at the current cell needs to be communicated once. The communicated value is then processed locally on the relevant processor (in our case processor 2), which does not involve any communication overhead any more.

It should be clear that the computational bottleneck for these kind of algorithms is not the number of standard operations, but the number of MPI operations. While image

filtering algorithms tend toward a reduction on the number of standard operations, the algorithms implemented in CPPPO have the reduction of the number of MPI operations as the main goal. Since field and mesh data can be very large in terms of the consumed memory, it is often not feasible to rely on massive data copying and thus, processor communications are rather frequent. The number of MPI communications in CPPPO can be of the same order of magnitude as the mesh size. More details on the parallel implementation will be given in section 2.5.

There are several differences between CPPPO, and other tools for averaging like those provided in OpenFOAM® (or sub-modules such as swak4Foam [swa]). These modules can just calculate averages over lines, faces and volumes using predefined lists (so-called "sets"). They cannot average at every cell, and cannot average around several moving Lagrangian objects (even if could be possible to program the required utility). Also, OpenFOAM® does not feature a divergent algorithm to compute averages and variances. In general, other filtering utilities are based on the convergent algorithm, or on the improved convergent algorithm we describe in section 2.5.

2.4. CPPPO statistics routines

CPPPO features a collection of sampling routines which allow to relate fields (named *sampled fields* in CPPPO) with other fields (named *markers* in CPPPO). The sampling utility will draw samples of the specified quantities of interest (accordingly to the functions described in subsection 2.4.1) at user-defined locations, or alternatively over the whole domain. Every sample will contain values of *sampled fields* and *markers*. *Sampled fields* are then binned accordingly to the related *markers* following user-defined settings for discretization of the binning process. Every time a value is added to a bin, CPPPO will automatically update the mean value and variance related to that bin using a running statistics approach [105]. Therefore, CPPPO will also keep track of the number of values added to every bin. Following this procedure, a large data-set is reduced to a multidimensional array, in which each element contains a (conditional) average and variance with respect to the *markers* of the *sampled fields*. This multidimensional array is then written to disk in the form of one dimensional arrays. CPPPO allows to create new files (and thus, new statistics) at every time step, or to update the current files and statistics in order to collect a single (time-averaged) data set. This *sampling/binning* procedure has been developed to perform automatic correlation of quantities of interest during, or after a simulation run. In such a way, a user can quickly assess whether a simulation needs to be run longer, or can be aborted to save computational resources.

2.4.1. Available sampling operations

At the current state, the available sampling routines are:

- *General sampling*: This routine draws samples over the whole domain, or just a portion of it. *sampled fields* and *markers* are defined by the user. Also, *General sampling* allows the use of a formula parser implemented in CPPPO to draw samples of quantities which are not explicitly calculated in the simulator.
- *Angle vector-vector*: This routine can sample vector fields using the angle between the original and a second vector field as *marker*.
- *Two point correlation*: This routine will sample the value of the trace of the two point velocity correlation.

2.5. Parallel implementation

CPPPO has been designed to (i) maximize the speed of data averaging calculations, and (ii) to provide a flexible architecture for the future addition of new models and algorithms. For this reason, a separation between cell selectors and filters was needed. Despite performance could possibly be affected by this approach in a negative way, the philosophy behind CPPPO is to allow the user to code new filters without implementing a new cell selector. As we will discuss in the following, however, new algorithms have been developed in order to increase performance and minimize the number of parallel communications and the amount of communicated data.

A CPPPO run is initialized via the interface class which allocates memory for filtered fields. For every user-defined filter operation, the interface class can trigger an FSB loop. All these operations are encapsulated in the CPPPO core library. Note that filtered fields, while always available in the interface class, are mapped in the core library only once (and not for each filter, or filter size). Thus, fields created for a certain filter are not available when running CPPPO for another filter. This requires the interface and the core library to run at two different levels: while the interface class has pointers to quantities used over the whole run, CPPPO's core library has pointers to relevant quantities only for the current filter (with the exception of mesh data and source fields data). This allows an easier and more intuitive use of pointers in the core library when accessing filtered field data.

Parallel communication in CPPPO mainly relies on collective MPI operations, since most of the time all processors have to synchronize during the calculations. These MPI routines have shown excellent performance in many applications [57, 59] on HPC hardware. The load partitioning is mainly a function of the domain decomposition, and the distribution of sampling locations, so that it is mostly user dependent. This is

particularly true when using Lagrangian filtering operations (i.e., filtering is performed at pre-defined probing positions). This is because the user affects directly the processor load in such a situation. For example, in case all probes are positioned in a sub-domain belonging to the same processor, the calculation would be slow. Thus, all operations would be focused on just one processor.

In the following we will discuss the implementation of parallel selectors and filters in CPPPO.

2.5.1. Parallel selectors

At the current state CPPPO features two parallel cell selectors: general unstructured and IJK structured. The former can deal with any unstructured mesh, while the latter is designed for structured meshes whose cells are equal of size. Both selectors have a similar structure that can be summarized as follows:

(i) Evaluate the position of the current cell (or probe location) to filter.

(ii) Communicate this position to all the other processors.

(iii) For every position, calculate the filter size (according to the boundary conditions) and create a list of cells that reside within the filter. For every cell added, update the total filter volume (this volume calculation allows to deal with complex cell shapes).

(iv) Communicate the filter volume to all the other processors (optional, since the total volume is generally not necessary).

This algorithm does not calculate the complete cell list for a single cell or probe location, but every processor calculates the cell lists corresponding to the fraction of every filter residing within its boundary. The above algorithm has been used to optimize the run time of divergent and convergent filtering operations. A workflow which illustrates the main steps in the selecting operation is shown Figure 2.4.

CPPPO selectors also take periodic boundary conditions into account. In addition, and in case the processor sub domain is entirely inside the filter, all its cells are automatically added to a list with no further operations. Both parallel selectors that are currently implemented require one collective MPI operation, during which every processor communicates the coordinates of its currently filtered cell (i.e., in total $3n_p$ doubles where n_p is the number of processors).

The structured IJK selector takes advantage of the possibility to define a coordinate system using the grid axis. Thus, a one-to-one correspondence between the cell id and

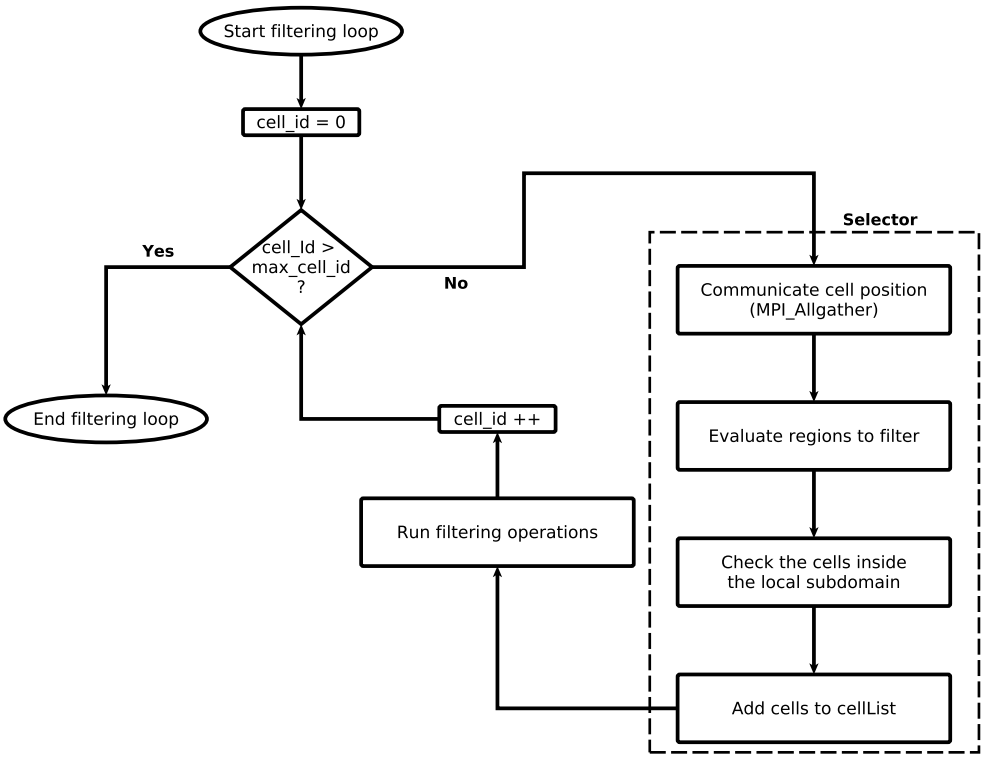


Figure 2.4.: Workflow illustrating the selector algorithm within the filtering loop. `max_cell_id` is the total number of cells on the processor. In case the filtering is carried over a set of probes, `max_cell_id` represents the number of probes on the processor.

a location in the Cartesian reference frame can be obtained. In order to do that, we express the new cell centre coordinates ζ_i in the form:

$$\zeta_i = \frac{c_i - \delta_i/2}{\delta_i} \tag{2.11}$$

Where c_i is the non-ijk cell centre coordinate and δ_i is the cell size in the i-th spatial direction. This coordinate transformation allows us to immediately evaluate the cells inside a region and their id, consequently speeding up the calculation.

In contrast, the more general unstructured selector loops over all the cells in the processor subdomain and checks, for every cell, if its centre lies within the filter region. Despite the fact that this algorithm is expensive in terms of computational time, it can deal with arbitrarily-shaped computational meshes. The latter are often used in engineering applications, and are also considered in the showcase detailed in section 2.8.

2.5.2. Parallel filters

At the current state CPPPO features a top-Hat kernel filtering operation that can be run in Eulerian or Lagrangian mode depending if the filtering has to occur for every cell or at specific Lagrangian points. Since filtering operations are repeated for each cell/probe (see figure 2.4), in the following we will consider the parallel communications required to filter at just one location (Lagrangian mode) or one cell (Eulerian mode).

The Eulerian mode uses the divergent algorithm to update the filtered fields, and can be summarized as follows:

(i) Calculate weighted fields for the cell at the current step. Weights are defined by the user.

(ii) Communicate the values of weighted fields to the neighbouring processors.

(iii) Update the filtered field.

Using this algorithm, just one `MPI_Allgather` operation is needed, and every processor exchanges a number of values equal to the total number of fields to filter. Clearly, in case vector fields are filtered, each spatial component has to be considered as a separate field when calculating the size of communicated data.

The Lagrangian mode uses an improved convergent algorithm, which can be summarized as follows:

(i) Calculate the locally-filtered value for the selected cell.

(ii) Communicate these filtered values, and calculate the final filtered value accounting for the locally-filtered values from all neighbouring processors.

In the improved convergent algorithm every processor, instead of communicating the whole list of cell values, performs a local filtering calculation (i.e., performs the averaging using only the cells it owns). Thus, only locally-filtered values need to be communicated (see left panel in Figure 2.5). This greatly reduces the number of communicated data. However, still this algorithm is less efficient than the divergent algorithm (see right panel in Figure 2.5). In fact, the divergent algorithm requires the communication of just $n_f + 1$ values per processor (where n_f is the number of fields to filter and the additional one is the weight). In contrast, the improved convergent algorithm, requires the communication of $n_p(n_f + 1)$ values per processor (where n_p is the number of processors). This is due to the direction of the data flow which, in the convergent algorithm, points from the neighboring processors to the *central* one as shown in figure 2.3, and not *viceversa*. In this context, the term *central* refers to the processor owning the cell to be filtered. However, since the algorithm is running in parallel, n_p cells are filtered at the same time and thus, every processor represents the *central* processor

with respect to the cell it owns. This means that the convergent approach requires the communication of, at least, $2n_p$ values so that the parallel efficiency will inevitably decrease with increasing number of cores. The variance calculation is another weak point of the improved convergent approach (even over the classic convergent algorithm). In fact, since the variance calculation requires the information on the filtered value (and not the partially-filtered value), additional communication is required to make filtered values available to every processor. In principle, the variance calculation follows an approach similar to the averaging step (i.e., partial variances are computed). This results in $n_p(2n_f + 1)$ data to be communicated (this calculation includes communication of the locally-computed variances). Clearly, the original convergent algorithm does not have this issue, since all the required values become available at the *central* processor after the first (and only) communication step. However, for the improved convergent algorithm, the number of communicated data scales linearly with n_p, while for the original convergent algorithm, this quantity is difficult to evaluate since each processor would need to communicate a different number of elements. That would require the use of less efficient collective operations like `MPI_Allgatherv`. Anyhow, the number of cells in the mesh is typically several orders of magnitude larger than n_p. Also, the filter size is, for the majority of applications, of the order of 10^{-1} times the domain length. Hence, we can conclude that, for almost any application, the number of exchanged data in the original convergent approach is much larger than in the improved convergent approach.

In contrast to the improved convergent algorithm, the divergent algorithm only requires the communication of $n_f + 1$ doubles, regardless of the fact whether variance calculation is performed or not. In terms of MPI collective operations, both algorithms require one operation per filtered cell for averaging. In case the variance is also computed, the convergent algorithm requires two additional MPI operations per cell.

2.6. Test calculations

The accuracy of CPPPO was tested by considering two well-known problems of fluid dynamics: Stokes flow and irrotational (i.e., potential) flow around a sphere. The main objective of these tests is to evaluate the accuracy of the parallel filter routines, and to illustrate the dependency of the results on the grid size. Therefore, we compared CPPPO results with analytical solutions of filtered quantities at the particle center. Recalling the analytical solution for Stokes flow (i.e., zero Reynolds number) around a sphere [10], and when considering the velocity component in the stream-wise (i.e., x-) direction, the flow field is described by:

$$u_x = U_\infty \left[cos^2\theta \left(1 + \frac{R^3}{2r^3} - \frac{3R}{2r} \right) + sin^2\theta \left(1 - \frac{R^3}{4r^3} - \frac{3R}{4r} \right) \right] \tag{2.12}$$

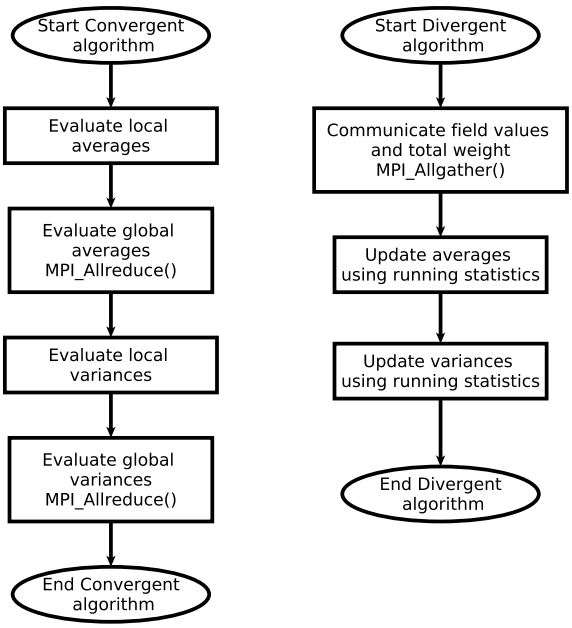


Figure 2.5.: Workflow for the convergent (left panel) and the newly proposed divergent algorithm (right panel). The Convergent algorithm requires an additional `MPI_Allreduce` operation to calculate the variance. In addition, every `MPI_Allreduce` require the exchange of more data than the `MPI_Allgather`.

Here θ and r describe the radial and polar position in a spherical coordinate system (the solution is symmetric with respect to the azimuthal coordinate). U_∞ is the flow velocity far from the particle, and R is the particle radius. The corresponding solution for irrotational flow (i.e., a flow characterized by an infinitely large Reynolds number) past a sphere is:

$$u_x = U_\infty \left[cos^2\theta \left(1 - \frac{R^3}{r^3}\right) + sin^2\theta \left(1 + \frac{R^3}{2r^3}\right)\right] \tag{2.13}$$

We now consider a spherical filter, and define a dimensionless filter size as:

$$\rho = \frac{R_f}{R} \tag{2.14}$$

Where R_f is the filter radius. Integration of Eqn. 2.12 and Eqn. 2.13 to obtain the mean and variance of the stream-wise velocity component leads to:

$$\widetilde{u_x}|_{Stokes} = \frac{(2\,\rho^2 - \rho - 1)\ U_\infty}{2\ (\rho^2 + \rho + 1)} \tag{2.15}$$

$$\tau_{xx}^{sgs}|_{Stokes} = \frac{(18\,\rho^5 - 32\rho^4 + 14\,\rho^3 - 3\rho^2 + 2\,\rho + 1)\ U_\infty^2}{(4\,\rho^4 + 8\,\rho^3 + 12\,\rho^2 + 8\,\rho + 4)\ 5\,\rho^3} \tag{2.16}$$

$$\widetilde{u_x}|_{Irr} = U_\infty \tag{2.17}$$

$$\tau_{xx}^{sgs}|_{Irr} = \frac{U_\infty^2}{5\rho^3} \tag{2.18}$$

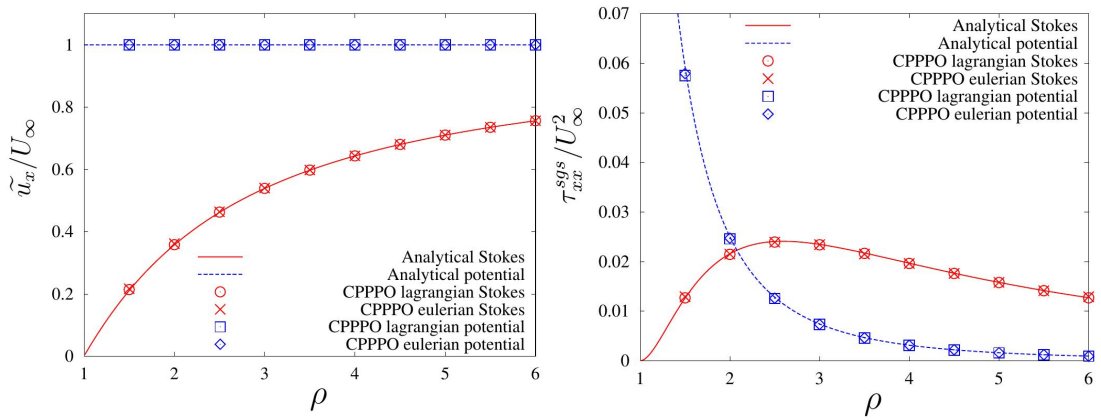
The solutions in Eqns. 2.15, 2.16, 2.17 and 2.18 provide a set of cases to verify the filtering routines of CPPPO for the situation of a top-Hat filter kernel in spherical coordinates. However, Stokes and potential flows are are not easily reproduced with standard CFD solvers unless the convective (or the viscous) term is removed from the equation. Even the use of specific solvers like *potentialFOAM* could induce some errors due to the discrete representation of the particle by an Eulerian mesh, or the finite size of the bounding walls. As a consequence, we preferred to impose the flow field rather than solving the governing equations with a CFD solver. The two newly implemented applications that impose the Stokes and the irrotational flow field are *stokesFilter* and *irrotationalFilter*, respectively. Two computational grids of 100x100x100 and 160x160x160 cells were used to evaluate the flow field. These resolutions resulted in a negligible effect of the mesh resolution on both the results for the mean and the variance. Test cases were run using 128 processes in order to asses the accuracy and speed of the parallel computation.

Results displayed in figure 2.6 show that CPPPO is able to correctly calculate the Favre average of a field both with Lagrangian and Eulerian filtering routines. Table 2.1 shows that the average relative error remains smaller than 1.7%, and that the variance may experience larger errors compared to the average.

The deviations between Lagrangian and Eulerian results can be explained considering that the two algorithms perform different algebraical operations and, thus, are subjected to different round-off errors.

In order to assess the runtime filtering routines, a low Reynolds number flow pas a sphere was simulated using a computational grid of $120 \times 120 \times 120$ cells, and a computational domain of $10d_p \times 10d_p \times 10d_p$. CPPPO was linked to OpenFOAM®'s *pisoFoam* solver, and the simulation was run until a steady-state was obtained.

The results show that there is a difference in the values of the Favre averaged velocity field calculated by CPPPO with respect to the analytical results. This discrepancy is due to the incorrect velocity field computed by the solver as shown in picture 2.7. Specifically, this is caused by the finite domain size, as can be seen by the stronger deviations with increasing distance from the particle surface.



(a) (b)

Figure 2.6.: Comparison between CPPPO's Lagrangian and Eulerian filtering tools with analytical results for the filtered quantities at the particle centre. Results for the Favre average are shown in panel a, while in panel b the Favre variance was calculated. The average velocity is normalized with the 'far field' velocity U_∞, while the variance with U_∞^2.

2.7. Parallel scalability and performance

In this section, we analyse the parallel scalability and performance of CPPPO. In particular, we compare the performance of the divergent and convergent algorithm, as well as the performance of the unstructured and IJK cell selector. Different metrics where used in order to quantitatively establish the performance of every algorithm, and to asses their preferred field of use.

Since CPPPO makes extensive use of MPI collective operations, individual processes are forced to synchronize often. This will result in acceptable parallel performance in case the load balance is uniform. However, the total time each processor takes to complete a certain task is subject to some fluctuations that are different for every run. For this reason, the average time τ_p (where p is the number of processes) and the time variance σ_p are used as main performance metrics in the following. Specifically, these metrics are defined as:

$$\tau_p = \frac{\sum_{n=1}^{p} t_{p,n}}{p} \tag{2.19}$$

case	operation	average relative error
Stokes Eulerian	average	0.004877
Stokes Lagrangian	average	0.002028
potential Eulerian	average	below machine precision
potential Lagrangian	average	below machine precision
Stokes Eulerian	variance	0.006865
Stokes Lagrangian	variance	0.003445
potential Eulerian	variance	0.011906
potential Lagrangian	variance	0.017841

Table 2.1.: Computed relative error for Stokes and potential flow test cases.

$$\sigma_p = \sqrt{\frac{\sum_{n=1}^{p} \left(t_{p,n} - \tau_p\right)^2}{p}} \tag{2.20}$$

Where $t_{p,n}$ is the time needed by process n to complete a certain task in case a total of p processes are used for the computation. In the following we refer to $\tau_p|_k$ as the average time taken by subroutine k when p processors are used. It should be noted that the wall time is $max(t_{p,n})$. Furthermore, the standard deviation was, in general, observed to be small compared to the average time due to the frequent MPI barriers used for synchronization. Thus, σ_p is not discussed in further detail below.

The strong parallel efficiency is defined as:

$$\eta_s = \frac{\tau_1}{p\tau_p} \tag{2.21}$$

In this study, we also define an advantage factor α_n^k to quantify the advantage, in terms of computational time, of using the subroutine n instead of the subroutine k. The advantage factor is then defined as:

$$\alpha_n^k = \frac{\tau_p|_k}{\tau_p|_n} \tag{2.22}$$

In particular, we will evaluate the advantage factor of the divergent filtering over the convergent filtering (α_d^c) and the advantage factor of the ijk selector over the unstructured selector (α_i^u).

In order to assess the parallel performance of the implemented algorithms, we run the *stokesFilter* test case introduced in section 2.6 in order to evaluate the Favre averaged velocity field for every cell in the domain using a box filter. The filter size was approximately one quarter of the domain length in every direction. Nine test case were run in total, using different routines and mesh size as reported in Table 2.2.

Figure 2.8 shows average time and strong efficiency from the studied cases. It can be seen that the unstructured selector requires significantly more time compared to the IJK selector. However, the unstructured selector shows a far better parallel efficiency. Thus, the IJK selector shows significant advantages with respect to the unstructured selector in case the number of cores is small (e.g, when using a local workstation).

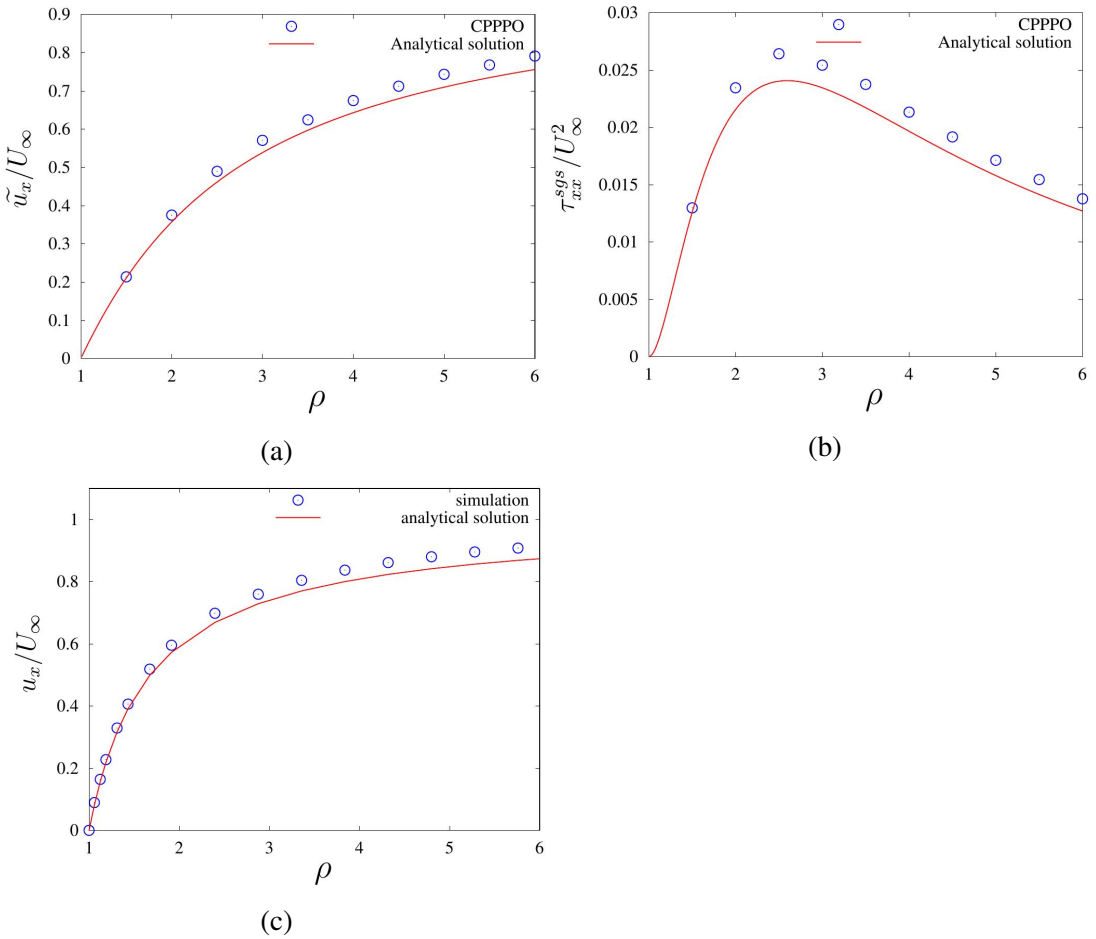


Figure 2.7.: Normalized Favre average (panel a) and variance (panel b) of the velocity field based on a simulation using OpenFOAM®'s *pisoFoam* solver. Subfigure 2.7c shows the calculated velocity profile along the span-wise direction, i.e., $\theta = \pi$, and the corresponding analytical solution.

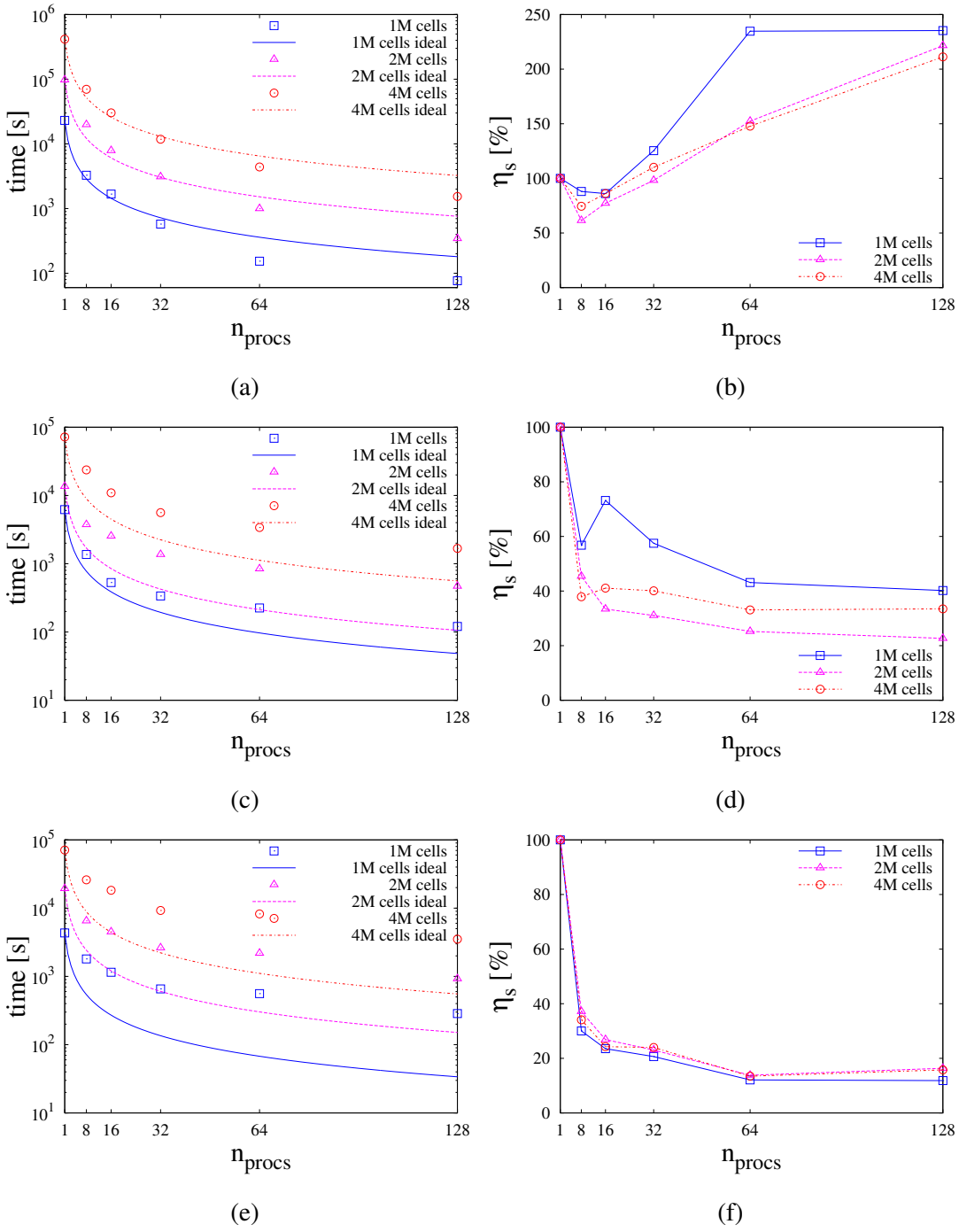
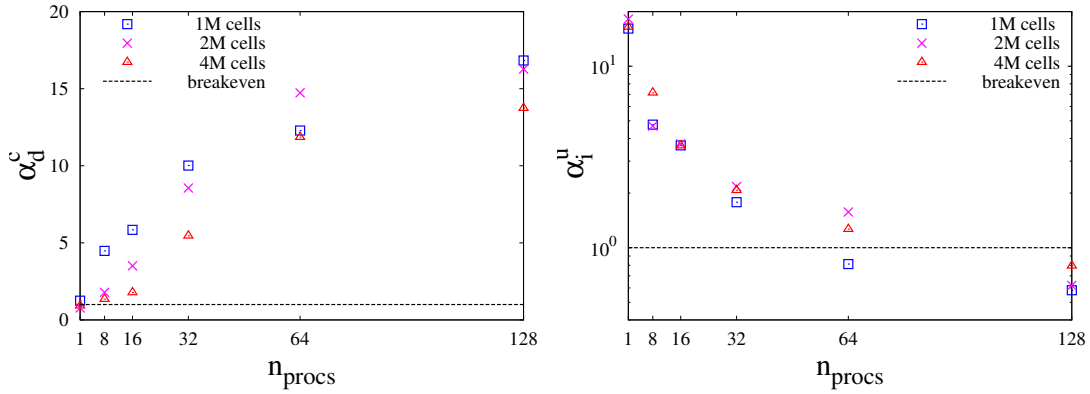


Figure 2.8.: Average computation time and strong parallel efficiency for the divergent filtering approach with unstructured selector (panel 2.8a and 2.8b), as well as the IJK selector (panel 2.8c and 2.8d). Panel 2.8e and 2.8f show average time and strong parallel efficiency for the convergent filtering approach with the IJK selector.

Filtering algorithm	Selector	Mesh size (cells)
Divergent	Unstructured	1×10^6
Divergent	Unstructured	2×10^6
Divergent	Unstructured	4×10^6
Divergent	IJK	1×10^6
Divergent	IJK	2×10^6
Divergent	IJK	4×10^6
Convergent	IJK	1×10^6
Convergent	IJK	2×10^6
Convergent	IJK	4×10^6

Table 2.2.: Test cases for the parallel scalability analysis.



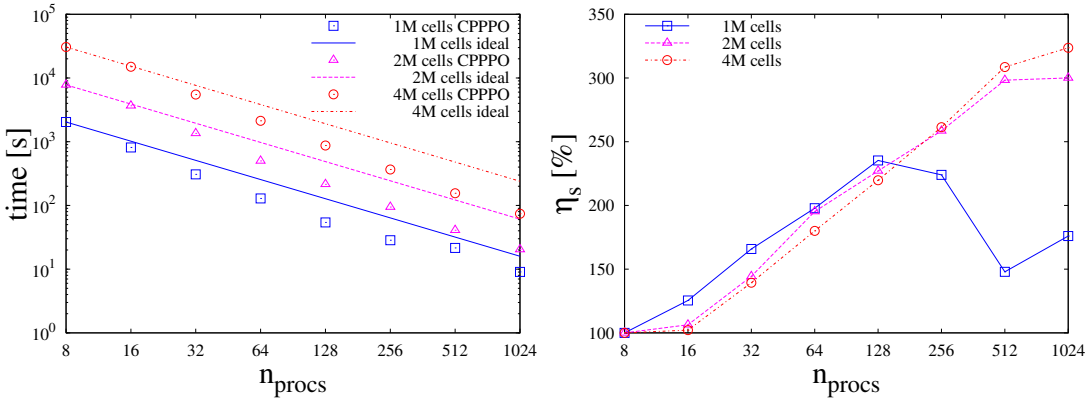
(a) (b)

Figure 2.9.: Advantage factor of the divergent filtering approach over the convergent filtering approach (panel 2.9a), as well as advantage factor of the IJK selector over the unstructured selector (panel 2.9b).

This fact is well represented by the advantage factors displayed in figure 2.9. When 128 cores are used, the IJK selector provides no more significant advantages in terms of computational time and α_i^u drops below unity. Figure 2.9a shows that α_c^d is very close to unity when one processor is used, but increases rapidly with the number of cores. However, the effect on the total time is only moderate (see Figure 2.8), since filtering operations are generally faster than selector operations. In summary, a divergent filtering approach and (surprisingly) an unstructured selector seems to be the optimal combination for a large number of cores.

Finally, the Vienna Scientific Cluster VSC-3 was used to test the library up to 1024 cores. The results generally showed a higher performance of VSC-3 with respect to

TU Graz' dcluster (see Figure 2.10). However, the parallel scalability was very similar to dcluster, showing the expected drop in performance for the smaller case involving 10^6 grid cells and when using less than approximately $4\,10^3$ grid cells per core. In summary, our benchmark calculations on VSC-3 confirmed the previously described excellent scalability of CPPPO.



(a) (b)

Figure 2.10.: Average time for the divergent filtering algorithm utilizing an unstructured grid selector on the VSC-3 cluster (panel 2.10a), as well as strong parallel efficiency (2.10b).

2.8. Heat transfer in a dense particle bed

2.8.1. Transport in dense particle beds

Particle-resolved direct numerical simulations (PR-DNS) of flow through dense particle beds have become a key instrument to develop closures for predicting momentum, heat and mass transfer rates in these systems [89, 93]. Typically, these simulations require extremely large computational grids (with $O(10^7)$ cells) to resolve regions with large velocity, concentration, or temperature gradients. Furthermore, a large number of realizations (e.g., particle configurations in a channel) is needed to represent reality reasonably well. This naturally leads to large data sets, asking for on-the-fly data filtering and an automation of the post-processing workflow. In the following, we will show that such a workflow can be carried out efficiently and in a fully-automated fashion using CPPPO. Specifically, we study a situation similar to the one considered by [104] and [72], and limit our attention to the prediction of flow and a single inert scalar.

2.8.2. Governing equations and numerical solution

The open source library OpenFOAM® is used in order to solve the incompressible momentum transport equations, the continuity equation and the transport equation of an inert scalar. These equations can be re-written in their dimensionless form using the Einstein notation to arrive at:

$$\frac{\partial u_i}{\partial x_i} = 0 \tag{2.23a}$$

$$\frac{\partial u_i}{\partial t} + \frac{\partial \left(u_j u_i\right)}{\partial x_j} = -\frac{\partial p}{\partial x_i} + \frac{1}{Re_p}\frac{\partial^2 u_i}{\partial x_j \partial x_j} \tag{2.23b}$$

$$\frac{\partial \phi}{\partial t} + \frac{\partial \left(u_i \phi\right)}{\partial x_i} = \frac{1}{Pe}\frac{\partial^2 \phi}{\partial x_i \partial x_i} \tag{2.23c}$$

Where u is the velocity field, p is the pressure and ϕ is the scalar field. The scalar transport equation can model heat or mass transfer without additional source terms, e.g., due to chemical reactions. The relevant dimensionless group is therefore represented by the Peclet number Pe and the particle Reynolds number Re_p:

$$Re_p = \frac{U d_p}{\nu} \tag{2.24a}$$

$$Pe = \frac{U d_p}{\Gamma} \tag{2.24b}$$

Where U is a typical flow speed (i.e., the superficial fluid velocity), d_p is the particle diameter, ν is the fluid kinematic viscosity, and Γ is the scalar's diffusion coefficient. For the present simulation we choose $Re_p = 10$ and $Pe = 20$. The equations were discretized using second order discretization schemes, and the PISO algorithm was adopted to solve the pressure equation.

The computational domain is a cylinder of radius $3d_p$ and a length of $16d_p$. A fixed particle bed having a void fraction of approximately 0.2 (involving 130 particles) was generated using the soft-sphere particle motion simulator LIGGGHTS® [53]. A body-fitted unstructured mesh was generated using the snappyHexMesh tool available in OpenFOAM®. Since constant velocity and zero gradient boundary conditions were used at the inlet and outlet surface, the particle bed was positioned between $x = 2d_p$ and $x = 14d_p$ in the cylinder's axial (i.e., x-) direction to reduce the effect of the above mentioned boundary conditions on the results. A no-slip boundary condition for the

velocity field was imposed at the cylinder's and at the particles' surface. For the scalar field we imposed a zero gradient boundary condition at the cylinder wall, and a fixed value boundary conditions at the particles' surface.

The final mesh consisted of approximately 7 million cells and featured a local grid refinement at the particles surface. The simulation was run until a steady-state solution was obtained.

2.8.3. Results and CPPPO post-processing

The case was run and post-processed using the Vienna Scientific Cluster (VSC-3). The computational domain was decomposed using 128 cores, and the execution of CPPPO's routines required approximately 1.5% of the total calculation time (i.e., 7.5 minutes out of 8.55 hours). CPPPO applied four box filters of different lengths d_f to all the cells and the particles in the domain. In addition, the CPPPO *general sampling* utility was used to evaluate the probability distribution function of the filtered velocity field in the region between $x = 4.5d_p$ and $x = 12.5d_p$.

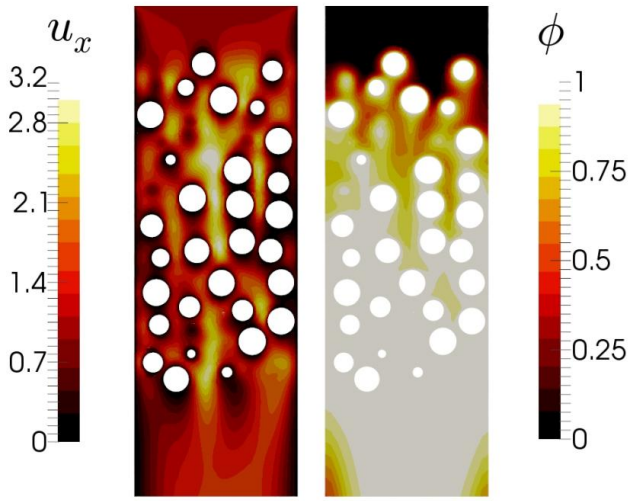


Figure 2.11.: Flow through a particle bed in a cylindrical channel: (unfiltered) velocity field in the axial direction (left panel), as well as scalar field (right panel; $\phi_p = 0.20$).

The resulting unfiltered fields are shown in figure 2.11. For every particle, a filter-size dependent bulk scalar field was defined:

$$\phi_b(d_f) = \frac{\int_{V_f} u_x \phi dV}{\int_{V_f} u_x dV} \tag{2.25}$$

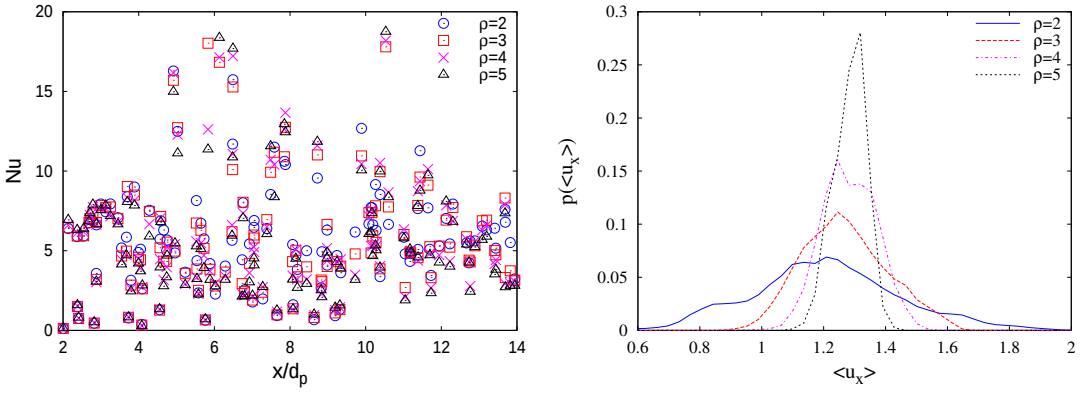
Where the filter volume V_f spans the region between the particle's surface and the filter radius. These quantities are calculated by CPPPO, and subsequently used to calculate a particle-based Sherwood number:

$$Sh_p = Pe \frac{q_p^{s,f}}{1 - \phi_b} \tag{2.26}$$

Here $q_p^{s,f}$ is the dimensionless solid-fluid scalar flux for particle p, which is defined as:

$$q_p^{s,f} = \frac{1}{S_p \, Pe} \int_{S_p} \frac{\partial \phi}{\partial n} dS \tag{2.27}$$

Values of Sh_p as a function of the axial position are shown in Figure 2.12a. The figure only displays values of $Sh_p < 20$, since some particles experience extremely small differences of the scalar quantity, i.e., $1 - \phi_b$, and hence would result in unrealistically large Sh_p values. However, Table 2.3 shows that most of the particles have a particle-based Sherwood number that is smaller than 20. Note, that the use of data from multiple realizations could reduce the standard deviation and result in a constant Sh_p number along the axis of the cylinder as has been already shown in literature [89, 93]. In addition, the data shown in Table 2.3 reveals that increasing the filter size leads to a reduction in the data deviation.



(a) (b)

Figure 2.12.: Particle Sherwood number experienced by a dense particle ensemble in a cylindrical channel as a function of the filter size and the axial position (left panel). Probability distribution function of the filtered axial velocity experienced by the particles (right panel).

$\rho = d_f/d_p$	average Sh_p	standard deviation Sh_p	$\phi_b < 0.99$	$Sh_p < 20$
2	14.380	41.017	98.5%	87.3%
3	13.287	35.223	100%	86.0%
4	12.352	27.937	100%	86.0%
5	11.771	25.194	100%	86.0%

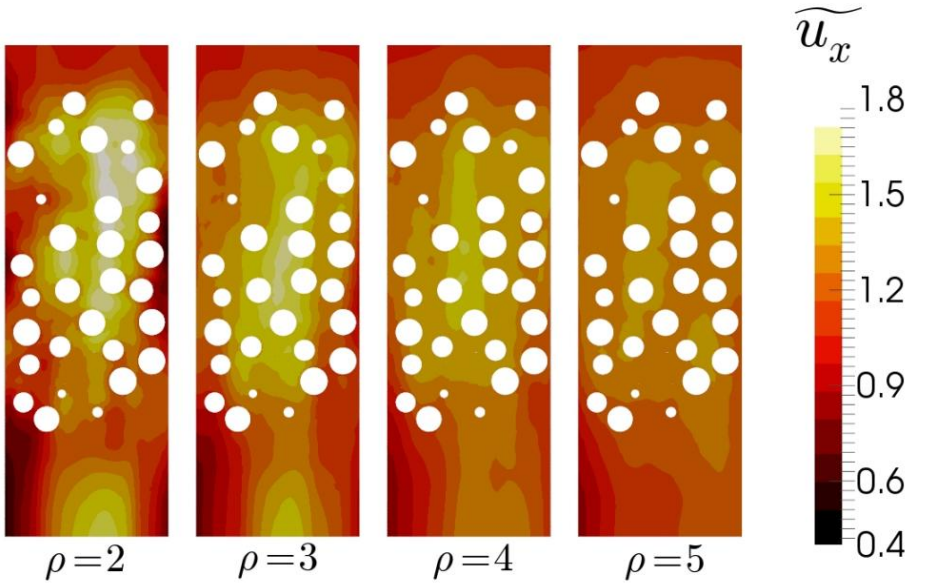
Table 2.3.: Sh_p statistics for different filter parameters in the region $4.5 < x/d_p < 12.5$ (86 particles) and $\phi_b < 0.99$.

Results obtained via CPPPO's *general sampling* module show that the larger the filter is, the more uniform the filtered velocity will be (see Figure 2.12b).

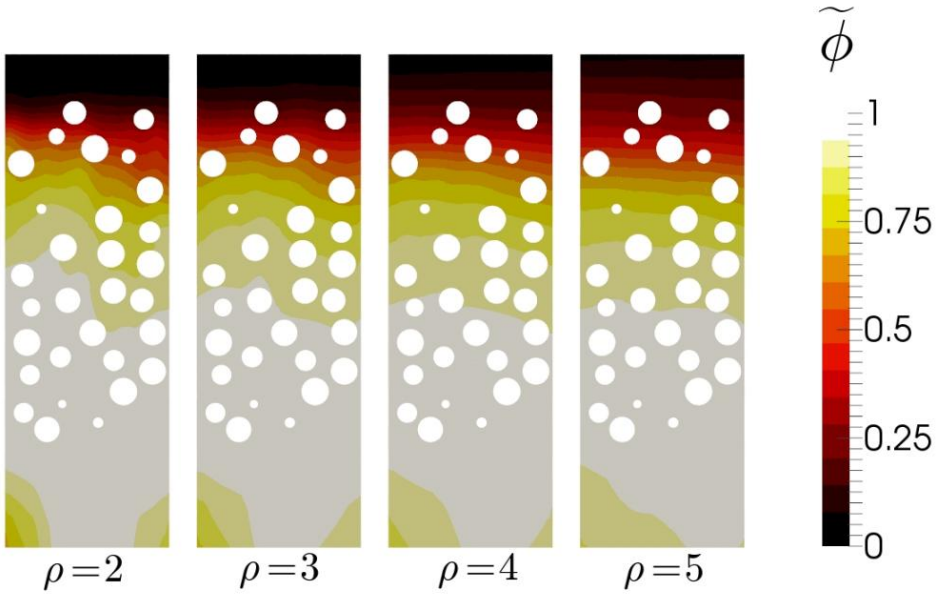
Finally, we show the complete filtered velocity and scalar field in figure 2.13. These fields have been written by the CPPPO-OpenFOAM® interface and are automatically generated. Interestingly, while the filtered velocity field tends to become more uniform by increasing the filter size, the filtered scalar field maintains its dependence on the axial coordinate even for large filters. This is due to the fact that the scalar field is not statistically homogeneous in the axial direction.

2.9. Summary and conclusions

The aim of the CPPPO library is to provide a set of routines for efficient parallel data filtering and processing. These operations are meant to be performed "on the fly" during expensive numerical simulations running on large distributed memory clusters. In order to perform data filtering from parallel simulations on clusters, a novel approach to filtering named "*divergent*" was adopted. The divergent approach showed a linear increase of parallel efficiency with the number of cores, and a major reduction of computational time with respect to the standard convergent approach was demonstrated. Overall, the parallel scalability analysis of CPPPO showed promising results, demonstrating the computational efficiency of our library. Furthermore, the CPU time required by CPPPO was shown to be a small fraction (i.e., less than 2%) of the time required by a typical simulation in the field of dense fluid-particle systems. As recently shown in literature [54], more insight into the governing flow physics of dense particle beds can be gained from the analysis of individual-particle DNS data. We have demonstrated that the filter size should be considered when evaluating such individual-particle data, e.g., (average) fluid quantities experienced by the particles. In addition, the ability to perform variance calculations in CPPPO allows one to extract additional markers that can be helpful to correlate DNS data, and hence establish new closure models. What remains to be done is to develop relevant transport equations for predicting these markers in coarse-grained simulations. Then, we expect that a new gen-



(a)



(b)

Figure 2.13.: Filtered velocity field in the axial direction and filtered scalar field.

eration of closure models, established with the help of tools like CPPPO, will help to refine our predictions for relevant fluid-particle systems in engineering simulations.

CPPPO allows a high flexibility in the filtering operations due to the easy customization of the filtering kernel. This can be achieved either by (i) including an arbitrary number of weights (which can be defined at runtime), or (ii) by implementing the desired kernel function (which requires some coding in C++, and recompilation of CPPPO).

CPPPO comes with instructions for compilation, as well as documentation covering input and usage of every module and sub-module. CPPPO also comes with examples on how to be coupled to OpenFOAM® or CFDEM® applications. A freely available version of the code can be downloaded from the CPC program library. To download the up-to-date version of CPPPO and get additional documentation, the interested reader is referred to `http://www.tugraz.at/en/institute/ippt/downloads-software/`.

3. Consistent closures for Euler-Lagrange models of bi-disperse gas-particle suspensions derived from particle-resolved direct numerical simulations

Particle-Resolved Direct Numerical Simulation (PR-DNS) is employed to simulate momentum and energy transport in bi-disperse gas-solid suspensions by means of a novel hybrid immersed-boundary/fictitious domain (HFD-IB) method. First, we demonstrate the accuracy of the new HFD-IB method against several verification tests. Subsequently, we simulate momentum and energy transfer in bi-disperse suspensions in the limit of high Stokes number, and the predicted flow and temperature fields are used, in conjunction with the open-source parallel data processing library CPPPO [68], to assess the validity of existing closures for momentum and heat transfer in the frame of Particle-Unresolved Euler-Lagrange (PU-EL) models. We propose a correction to the drag force model proposed by Beetstra et al. [11] which consistently takes into account the pressure contribution to the total fluid-particle interaction force in PU-EL models. Also, we propose a stochastic closure model for the per-particle drag coefficient based on a modified log-normal distribution. Finally, we assess the existence of an analogy between the particle-based drag coefficient and the conditionally-averaged Nusselt number. Indeed, our PR-DNS data indicates that a stochastic closure similar to that for the drag can be used to close the particle-based Nusselt number in dense bidisperse suspensions.

3.1. Introduction

Numerical simulations of large scale particle flows, which are widely encountered in industrial applications, are normally performed using averaged equations of motion. In these descriptions the solid and fluid phases are modelled as interpenetrating continua [5]. These models are normally based on the kinetic theory of granular flows [52] and contain unclosed terms that have to be modeled somehow. In a multi-scale

approach [45, 96], these models can be derived from more detailed simulations where particles are described as a discrete phase. In particular, in case the trajectory of each particle is tracked and collisions are resolved, one obtains the so-called Computational Fluid Dynamics-Discrete Elements Method (CFD-DEM) that, in case fluid cells are larger than particle diameters, can also be referred to as the Particle-Unresolved Euler-Lagrange approach (PU-EL) [15–17, 75, 81, 109]. However, even PU-EL equations have several unclosed terms, like the interphase transport coefficients, that account for, for example, fluid-particle heat and momentum transfer. Following the multiscale paradigm in our present contribution, we seek to obtain certain closures from fully resolved simulations, i.e., where the detailed flow and temperature (or concentration) fields are resolved on a sub-particle level. This latter approach can be denoted as Particle-Resolved Euler-Lagrange (PR-EL), or Particle-Resolved Direct Numerical Simulation (PR-DNS) if turbulence models are used or not, respectively. PR-DNS has already been extensively used to derive closures for the drag coefficient in mono- and bi-disperse suspensions [11, 46, 54], or for the Nusselt/Sherwood number in monodisperse suspensions [24, 26, 87, 88, 93]. However, almost the totality of this previous work focused on closures for Euler-Euler-based simulations in a coarse scale. Naturally, the question arises of the same closures can be used for PU-EL simulations, and we will demonstrate that indeed this is not the case.

3.1.1. Upscaling and closure development strategies

A major difference between closures for Euler-Euler and PU-EL models is that the latter require a particle-based description of the interphase transfer processes, while the former (i.e., EE models) require average exchange coefficients. Thus, particle-based models are affected by per-particle fluctuations that arise simply due to the random arrangement of individual particles, and occur even in low-Reynolds number flows. When developing closures for continuum formulations, these quantities, are obtained from averages within each realization, so that the fluctuations in the particle population are lost and the final standard deviation is calculated based on the ensemble of realizations. On the contrary, closures for PU-EL models are based on the whole studied population and thus, they may require stochastic models to take into account the single particle variability [6, 54].

The process of upscaling of fluid quantities that we adopted is known as (spatial) filtering or coarse-graining [99]. The local domain where this operation is performed (in PR-DNS) can be identified with a fluid cell used in PU-EL. In general, closure models derived using this approach have a functional dependence on the filter size, i.e., the support of the filtering kernel or, in other words, the size of the coarse-grained cell [75]. Moreover, while the velocity field is statistically homogeneous in homogeneous particle configurations, the temperature field is generally inhomogeneous [87, 93]. This is in contrast with the assumption of separation of scales required for the development

of continuum formulations, e.g., Euler-Euler models, and poses a challenge also in the development of particle-based models such as PU-EL-based models. In the present work we refer to this issue as *saturation*, since this term does reflect the physical process that is behind.

3.1.2. Immersed-boundary formulations

Studies based on PR-DNS are often performed using immersed-boundary [73] or fictitious domain [86] methods to account for the presence of solid particles, which are often modeled as spheres. A third approach is the "Physalis" method suggested by [84]. This method consists in making use of Lamb's analytical solution for Stokes flow (i.e., using a zero Reynolds number approximation) around a sphere to obtain a high order description of the flow field in the region close to the particle surface. The Physalis method has been successfully applied to moderate Reynolds number suspension flows, and has the key advantages that (i) fluid-particle interaction forces are easily calculated, and that (ii) the numerical error associated with the boundary treatment decreases exponentially. Physalis has been so far applied to flows involvig spherical particles only, and it is questionable if the employed zero Reynolds number approximation has an advantage over other boundary approximations for high Reynolds number flows. Generally, in immersed-boundary methods the particle surface is discretized using a set of nodes, spread uniformly on the surface of each immersed boundary, where appropriate forcing terms are computed to impose a Dirichlet (or Neumann) boundary condition. Subsequently, these forcing terms are extrapolated to the surrounding fluid grid, typically by means of a regularized delta function. While these methods are, in general, rather accurate, they may pose a problem of consistency with the underlining transport equations and introduce an additional discretization (i.e., the surface discretization) in the model. On the contrary, in fictitious domain methods, a rigidity constraint is imposed in the fluid region corresponding to the immersed solid body. This constraint is represented as an additional term in the governing equations, so that there are no issues with consistency or convergence. The mathematical formulation normally follows a Lagrangian multiplier based approach [8, 103]. Generally, all these kind of direct-forcing methods are focused on imposing the boundary conditions at the immersed surface and/or imposing a rigidity constraint in the fluid region occupied by the immersed body. In the present work, we merge a fictitious domain and an immersed-boundary method to obtain an hybrid method that is consistent, convergent and accurate even on relatively coarse grids. The new method can be used to solve both momentum and scalar (thus energy or mass) transport equations on unstructured grids.

3.1.3. Effect of size-polydispersity

Polydispersity (i.e., differences in particle size and/or density) is omnipresent in industrial applications and can have various effects. For example, it was shown that the presence of a small amount of fine particles can greatly enhance fluidization [37]. The study of polydisperse suspensions is complicated further by the different inertia of different particle species, which leads to different average velocities between species, and may ultimately lead to segregation. Thus, fixed bed models can be insufficient to correctly picture the momentum exchange processes in fluidized particle beds [48]. In addition, an extensive study of heat transfer in poly-disperse suspensions requires a large computational effort due to the large set of involved parameters (particle diameters and volume fractions of the single species), and the large number of possible configurations that need to be studied to probe enough statistics.

In the past years, polydispersity has been studied in the frame of fluid-particle momentum transfer [11, 46]. However, the problem of estimating heat/mass transfer coefficients in polydisperse suspensions has been approached only very recently in the scientific community. Specifically, the rather limited study of Tavassoli et al. [90] (considering Reynolds numbers between 30 and 100) is the only attempt towards numerical investigation of heat/mass transfer in a random array of size-disperse spheres we are aware of. Even this recent work failed to establish a conclusive model for the distribution of the per-particle Nusselt, simply because not enough statistics could be collected.

3.1.4. Considerations on numerical simulation of gas-particle suspensions

Gas-particle suspensions are generally characterized by high values of the Stokes number due to the large solid to gas density ratio. This results in the characteristic time of the flow field being much smaller than the characteristic time for the evolution of the particle configuration. In other words, the fluid phase is evolving much faster (eventually reaching a steady state) than the solid phase, allowing the time dependent gas-particle system to be represented as a collection of particle configurations together with steady (or fully developed, if the flow field does not reach a steady state) flow fields. Similar conclusions can be also drawn for the energy transport by noticing that the Prandtl number is close to unity for gases and thus, the temperature field evolves on length and time scales similar to those of the velocity field. In our work we enforce time scale separation to decouple the evolution of the fluid and the particle phase, and therefore we keep the particle cloud fixed while evolving the fluid phase. We do not simulate moving particles. This approach has already been adopted (for the flow field) in the work of Holloway et al. [48]. Furthermore, in fluidized suspensions the effect of particle velocity fluctuations on the drag force is considerably smaller than the effect of the mean fluid-particle slip velocity [108] in the case of low Reynolds number flows.

In our work, we speculate that this holds also for moderate (up to 400) Reynolds numbers. With this in mind, we consider static particle distributions (i.e., particles are not allowed to change their position in time) and neglect the effects of relative velocities between different particles.

3.1.5. Goals and outline

In the present work, we first aim on developing a novel method to account for the presence of immersed bodies in dense particle beds. Second, we apply this method to study transfer of a scalar quantity (i.e., heat and mass) from a random bi-disperse bed of particles. Third, we make use of the open-source library CPPPO [68] to investigate the relationship between (i) closures (for drag coefficient and Nusselt number) developed for continuum models and (ii) particle-based closures that are obtained from coarse-grained fields and individual particle quantities (like interphase heat and drag force). Finally, we aim to explore the effect a bi-disperse particle population has on the Nusselt number to eventually establish an analogy between the coefficients that describe momentum and heat transfer.

Our paper is organized as follows: we give a brief background on the mathematical description and introduce the relevant dimensionless group in Section 3.2. In Section 3.3, we describe the phenomenon of saturation to motivate which range of Reynolds and particle concentration should be probed to derive closure laws. This section also contains an analytical model to predict the length and time scales at which saturation occurs. In Section 3.4 we describe the new hybrid immersed boundary-fictitious domain method, together with some numerical details. We present a set of verification cases for both momentum and heat transfer in 3.5. In Section 3.6, we present the results from PR-DNS. In Section 3.8 we bring our findings into perspective with previous findings in order to highlight the significance of our results.

3.2. Mathematical formulation

In this work, we formulate the framework that governs flow through the intersticises of a bi-disperse particle cloud composed of spherical rigid stationary particles suspended in an incompressible Newtonian fluid. We assume that the fluid density does not depend on the temperature field (i.e., we consider the limit of low temperature differences). Effects of buoyancy and radiation are disregarded as well as viscous heating, so that the momentum equations are decoupled from the energy equation and thus, they can be solved separately. We consider a computational domain Ω subdivided in a purely fluid domain Ω_f, and a fictitious particle domain $\Omega_p = \sum_i \Omega_p^i$, where the summation is carried out over all the particles. In our approach, particles are considered as

regions where the governing equations are equipped with additional forcing terms to impose a Dirichlet boundary condition at the immersed surfaces.

3.2.1. Governing equations

In the present work, we use the library CFDEMCoupling® [53] to solve the following dimensionless transport equations:

$$\boldsymbol{\nabla} \cdot \boldsymbol{u} = 0 \tag{3.1a}$$

$$\frac{\partial \boldsymbol{u}}{\partial t} + \boldsymbol{\nabla} \cdot (\boldsymbol{u}\boldsymbol{u}) = -\boldsymbol{\nabla} p + Re^{-1} \nabla^2 \boldsymbol{u} + \boldsymbol{f}^{\circledcirc} \tag{3.1b}$$

$$\frac{\partial \theta}{\partial t} + \boldsymbol{\nabla} \cdot (\boldsymbol{u}\theta) = Pe^{-1} \nabla^2 \theta + Q^{\circledcirc} \tag{3.1c}$$

In the system of equations 3.1, $\boldsymbol{u}$ is the dimensionless velocity field, p is the dimensionless pressure field, and θ is the dimensionless temperature field. We also introduced the forcing term $\boldsymbol{f}^{\circledcirc}$ used to bend the streamlines to make the flow field consistent with the presence of immersed bodies. The source term $Q^{\circledcirc}$ represents the contribution from immersed particles in the energy equation. These two terms are detailed in section 3.4.

In addition, we introduced the dimensionless group composed by Re, the global Reynolds number, and Pe, the global Peclet number. They are defined as:

$$Re = \frac{U_s d_m}{\nu}, \quad Pe = \frac{\rho_f c_p U_s d_m}{\lambda_f} \tag{3.2}$$

Where U_s is the superficial velocity $U(1-\phi_p)$, and U is the fluid average velocity, as well as ϕ_p is the total particle volume fraction in the domain. ν is the fluid kinematic viscosity, ρ_f is the fluid density, c_p is the fluid thermal capacity, λ_f is the fluid heat conductivity, and d_m is the diameter of the smallest particle. The choice of using d_m as reference is purely arbitrary, and was found useful to interpret the results of our study as discussed in the next paragraphs. One could also introduce the mean Sauter diameter defined as:

$$d_{32} = \frac{\sum_{k=1}^{n_s} N_k d_k^3}{\sum_{k=1}^{n_s} N_k d_k^2} = \left[\sum_{k=1}^{n_s} \frac{\phi_{p,k}}{\phi_p d_k} \right]^{-1} \tag{3.3}$$

Where n_s is the number of species present in the system, N_k is the number of particles of species k and d_k is the diameter of particles of species k, and $\phi_{p,k}$ is the volume fraction of species k.

We assume that the quantities in definition 3.2 do not depend on the temperature or pressure fields, so that the values of Re and Pe can be considered to be constant in space and time. Furthermore, Pe is related to Re by the Prandtl number:

$$Pr = \frac{\rho_f c_p \nu}{\lambda_f} \rightarrow Pe = RePr \tag{3.4}$$

We will limit our present study to the case of $Pr = 1$ so that we can write $Re = Pe$. In this way, the fluid properties (relevant for both momentum and heat exchange) can be fully expressed using just one dimensionless number (Re).

The choice of d_{32} as reference length is consistent with the recent findings in poly-disperse systems [46] and it is a common practice in the field of sprays, in order to quantify an effective drop diameter. However, we made the Navier-Stokes equation dimensionless in such a way that the dimensionless group does not depend on d_{32}. In this way, the values of Re and Pe are not changing with the small to large volume fraction (ϕ_m/ϕ_M) or diameter (d_m/d_M) ratio. In addition, since the particle diameter is made dimensionless with d_m, i.e., $\eta_i = d_i/d_m$, the dimensionless group based on d_m can be straightforwardly converted into dimensionless parameters based on d_{32} if desired.

3.2.2. Filtering operators for coarse graining

The evaluation of particle based transfer coefficients can be performed in different ways for momentum transfer [11, 44, 47] and heat transfer [24, 26, 87, 88]. While previous works aimed to develop such closures for Euler-Euler models, in the present work, we aim to establish correlations for PU-EL models. The main challenge is to provide a description of the velocity and temperature of the fluid phase in regions surrounding each individual particle. This is typically done by averaging the fluid variables around each particle using some kind of Kernel function to apply a statistical weight [5]. We will formally indicate this volume filtering operation as:

$$\overline{(*)} = \iiint_{\Omega} K(\boldsymbol{x} - \boldsymbol{x'})\,(*)\,d^3\boldsymbol{x'} \tag{3.5}$$

Where we denoted the spatial filter Kernel $K(\boldsymbol{x} - \boldsymbol{x'})$ as a function in the space coordinate $\boldsymbol{x}$. Filtered fluid variables are, then, obtained by applying the Favre averaging

operator:

$$\widetilde{(*)} = \frac{\overline{\phi_f (*)}}{\overline{\phi_f}} \tag{3.6}$$

The choice of a suitable functional form of K plays a major role, and should be mainly driven by the nature of the closure model. In PU-EL simulations, the particle diameter is generally much smaller than the size of the fluid cell and thus, only cell-based values (i.e., averaged over the cell volume) are available to be used as variables in a closure model. This requires the filtering Kernel to have a finite support (e.g., the dimensions of the PU-EL fluid cell), and being able to represent the coarse-graining occurring due to the finite volume formulation of the Eulerian phase in PU-EL simulations. These properties are satisfied by the Top-Hat Kernel:

$$K\left(\boldsymbol{x} - \boldsymbol{x'}\right) = \prod_j \frac{\mathcal{H}\left(\frac{\varrho}{2} - |x_j - x'_j|\right)}{\varrho} \tag{3.7}$$

Where $\mathcal{H}$ is the Heaviside step function and ϱ is the dimensionless filter size, defined as ϱ^*/d_p, referring to an equilateral box-shaped filter. In the following, we will refer to the filtered subdomains as coarse grained cells and we will use the notation $\widetilde{(*)}_i$ or $\overline{(*)}_i$ to indicate that filtered quantities are evaluated at $\boldsymbol{x}_i$, i.e., at the center of particle i. Using the functional form defined in 3.7 is equivalent to limit the application of the derived model to (i) structured Cartesian Eulerian grids, and (ii) particles positioned at the exact center of the fluid cell. While the first condition is generally met in well-resolved EL simulations, the second (ii) is generally not, unless particle based values are calculated via linear interpolation from values located in the surrounding cells. Such an interpolation operation is generally cheap (especially for structured grids) when compared to the solution of large linear systems or the integration of Newton equations for a large number of particles [99]. Hence, linear interpolation is typically considered computationally affordable, and indeed used in almost the totality of all recent works that rely on PU-EL simulations [75].

Next, we introduce the so-called bulk or flux-averaged temperature defined as:

$$\widehat{\theta} = \frac{\iint_A \theta \boldsymbol{u} \cdot \mathrm{d}\boldsymbol{A}}{\iint_A \boldsymbol{u} \cdot \mathrm{d}\boldsymbol{A}} \tag{3.8}$$

Where $\boldsymbol{A}$ is a normal vector characterizing a cross sectional area element (i.e., and area element normal to the average flow field). Notice that, unlike other filtering operators, the flux averaging operator removes the dependency with respect to two spatial

components (i.e., that in the plane of the element $\boldsymbol{A}$). It is clear that closure models that are based on flux-averaged quantities are rather useless, since flux-averaged quantities are generally not available in PU-EL or Two Fluids Models. Despite the fact that this problems has been already addressed in literature [87], all available closure models (except that from [87]) are based on flux averaged quantities, and hence not applicable for PU-EL model.

3.2.3. Extraction of conditional averages

In the present study particle based quantities are evaluated using the filtering library CPPPO [68]. A conditional average is then performed for each studied parameter removing those sampled values that deviated more than $\pm 2\sigma$ (σ being the standard deviation) from the average. In addition, only averages containing more than 100 individual samples were considered in the study. Following this strategy, particle based quantities (i.e., the samples) are grouped into bins [68]. Within each bin we also constructed the distribution of the sampled values. Thus, we obtain a set of conditionally averaged values and, for each of them, the corresponding distribution of sampled values. This allows us to represent the particle population using a closure for (i) the conditionally-averaged quantities, and (ii) the distribution of per-particle quantities.

Thus, for a generic sampled quantity Φ_s and a marker field Φ_m we define a closed set U_{Φ_m} of equally spaced values of Φ_m such that:

$$\mathrm{U}_{\Phi_m} = \left\{ \Phi_{m,j} \in \Phi_m, \quad j \in \mathbb{N}^+ \quad | \quad \Phi_{m,j+1} - \Phi_{m,j} = 2\Delta\Phi_m \right\} \tag{3.9}$$

here $\Delta\Phi_m$ is the spacing interval. For each particle i, we define the ensemble conditional average kernel function as:

$$\mathrm{C}^i_{\Phi_{m,j}} = \mathcal{H}(\Phi^i_m - \Phi_{m,j} + \Delta\Phi_m)\mathcal{H}(\Phi^i_m - \Phi_{m,j} - \Delta\Phi_m) \tag{3.10}$$

Which is 1 if the particle marker field value Φ^i_m falls in the discretization interval of $\Phi_{m,j}$ and 0 otherwise. Then, we define the ensemble average of Φ_s conditioned on $\Phi_{m,j}$ as:

$$\langle \Phi_s \rangle_{\Phi_{m,j}} = \frac{\sum_{i=0}^{N_p} \Phi^i_s \mathrm{C}^i_{\Phi_{m,j}}}{\sum_{i=0}^{N_p} \mathrm{C}^i_{\Phi_{m,j}}} \tag{3.11}$$

Furthermore, we define the conditionally-scaled standard deviation $\sigma^{\Phi_s}_{\Phi_{m,j}}$ as:

$$\sigma^{\Phi_s}_{\Phi_{m,j}} = \frac{1}{\langle \Phi_s \rangle_{\Phi_{m,j}}} \sqrt{\sum_{i=0}^{N_p} (\Phi_s^i - \langle \Phi_s \rangle_{\Phi_{m,j}})^2 \frac{\mathrm{C}^i_{\Phi_{m,j}}}{N_p}} \tag{3.12}$$

Which is the standard deviation relative to the mean of the values in bin j characterized by marker m scaled with the respective ensemble average value. In the following, we will omit the index j in the notation and we will simply use the notation $\langle * \rangle_{\Phi_m}$ to indicate conditionally averaged values.

3.3. Saturation phenomena in dense particle beds

In the present work we consider a particle population that is statistically homogeneously distributed in space. Consequently, we expect the velocity field to be statistically homogeneous so that the use of periodic boundary conditions is justified. In contrast, the temperature field will have a statistical biasing due to the advection term $\boldsymbol{\nabla} \cdot (\boldsymbol{u}\theta)$ in the thermal energy transport equation that makes the resulting field statistically inhomogeneous. This inhomogeneity may lead to the conclusion that periodic boundary conditions do not correctly represent the physics of this process. Nevertheless, periodic boundaries have been previously adopted to study heat and mass transfer in monodisperse liquid-particle suspensions [26] exploiting the fact that, before the fluid approaches the particle temperature everywhere in the domain, there exists a temporal window of constant average Nusselt number (Nu). Unfortunately, the approach of Derksen [26] cannot be applied in the present study since this temporal window is too short to probe statistically significant results. This is simply because of the fact that we consider gas-particle systems for which $Pr \approx 1$. Clearly, the fluid will quickly reach a condition that we refer as *saturation*, and which is closely examined in the rest of this section.

3.3.1. Definition of the saturation variable

In our present contribution we define the fluid phase to be *saturated* with respect to particle i, or in a global sense, respectively, if:

$$\Theta^i = \frac{\theta_s^i - \widetilde{\theta}^i}{\theta_s^i} \leq \Theta_{sat}; \quad \Theta = \frac{\theta_s - \widehat{\theta}}{\theta_s} \leq \Theta_{sat} \tag{3.13}$$

Where θ_s^i is the surface temperature of particle i and $\widetilde{\theta}^i$ is the Favre averaged temperature around particle i. We indicate with Θ the saturation margin, and with Θ_{sat} the

threshold after which we consider the fluid to be saturated, i.e., close to the thermal equilibrium with the population.

There exist two main reasons why conditions 3.13 should not occur in the simulated domain.

i Deviations of local values of θ from $\widetilde{\theta}^i$ may induce large fluctuations in the resulting Nusselt number due to the small value of $(\theta_s^i - \widetilde{\theta}^i)$. This causes noise in the resulting statistics for the Nusselt number.

ii If saturation occurs after a short distance (e.g., after $2d_p$), there may be no need for a closure model for the Nusselt number. In fact, it would just be sufficient to set the fluid temperature (assuming that all the particles have the same surface temperature θ_s) to θ_s. Furthermore, the use of a closure in saturated conditions, may lead to numerical instabilities (due to the strong coupling) and thus, to strong oscillations of the temperature field.

In this work, we set the saturation margin Θ_{sat} to 0.05 since we found that local values of θ could experience significant deviations from $\widetilde{\theta}^i$.

It should be noted that, unlike Θ^i, Θ is a function of just one spatial coordinate since it depends only on the flux averaged temperature and the (constant) particle surface temperature. However, in case the particle distribution is homogeneous, the particle surface temperature is constant and identical for every particle, and the filter size ϱ is sufficiently large, then also Θ^i is a function of the stream-wise coordinate only. In what follows we take the z-coordinate as the stream-wise coordinate, i.e., the mean flow direction.

Furthermore, we consider an infinite collection of fluid-particle systems, each of them being a realization of the same macroscopic state (i.e., characterized by identical values for ϕ_p, Pe, θ_s and the particle diameter distribution). We assign to each value of the stream-wise coordinate z all the systems with a particle located at $z^i = z$. This is equivalent to say that, for each value of the coordinate z, it is possible to find a particle configuration such that $\widetilde{\theta}(t, z) = \widetilde{\theta}(t, z^i)$ with $z^i = z$ (since the temperature field is homogeneous in x and y, we dropped the dependence on these coordinates). Thus, under these considerations, Θ^i can be expressed as a continuous smooth function of the stream-wise coordinate.

In the following, we will make no distinction between Θ^i and Θ since the model we propose can be applied to both the definitions of the saturation margin. Thus, we will refer to the filtered temperature as the coarse-grained temperature, where the coarse graining operator is not explicitly specified, but it is assumed to have commutation properties similar to the Favre averaging operator. Also, we introduce the most relevant Nusselt number closures from literature to subsequently build a simple model that allows us to investigate saturation along the flow direction.

3.3.2. Available closures for the Nusselt number

For heat transfer in monodisperse particle beds several closures for Nu exist. The most widely used closure is the one proposed by Gunn in 1978 [43]:

$$Nu_{Gunn} = \left(7 - 10\phi_f + 5\phi_f^2\right)\left(1 + 0.7Re^{0.2}Pr^{1/3}\right) + \left(1.33 - 2.4\phi_f + 1.2\phi_f^2\right)Re^{0.7}Pr^{1/3} \tag{3.14}$$

An alternative closure was suggested by Deen et al. [24], which used DNS data to refit the closure of Gunn:

$$Nu_{Deen} = \left(7 - 10\phi_f + 5\phi_f^2\right)\left(1 + 0.17Re^{0.2}Pr^{1/3}\right) + \left(1.33 - 2.31\phi_f + 1.16\phi_f^2\right)Re^{0.7}Pr^{1/3} \tag{3.15}$$

Another closure obtained from DNS data is the one proposed by Sun et al. [87] :

$$Nu_{Sun} = \left(-0.46 + 1.77\phi_f + 0.69\phi_f^2\right)/\phi_f^3 + \left(1.37 - 2.4\phi_f + 1.2\phi_f^2\right)Re^{0.7}Pr^{1/3} \tag{3.16}$$

Notice that we used ϕ_f to indicate $(1-\phi_p)$ in the above expressions. The last closure is a correction to equation 3.15 for low ϕ_f and will not be used in this work. All these closures where obtained based on a flux averaged temperature and thus, in order to be consistent with coarse-grained models, they must be corrected. Therefore, Sun et al. [87] proposed a correction function that is used to rescale the Nusselt number to obtain $Nu^{(cons)}$ which is consistent with the PU-EL appraoch:

$$Nu^{(cons)} = Nu_{flux}[1 - 1.6\phi_p(1-\phi_p) - 3\phi_p(1-\phi_p)^4\exp(-Re^{0.4}\phi_p)]^{-1} \tag{3.17}$$

Here we used Nu_{flux} to indicate the Nusselt number obtained using any of the proposed closures.

3.3.3. Saturation equation in the case of pure advection

It is possible to derive an analytic expression for the saturation length z_{sat} representing the minimum distance at which condition 3.13 is satisfied. Let us consider a one dimensional time independent model where the local interface heat transfer rate Q balances the convective transport of thermal transport in the bed:

$$Q = U_s\rho_f c_p\frac{\mathrm{d}T(z^*)}{\mathrm{d}z^*} = hA_p(T_s - T(z^*)) \tag{3.18}$$

$$h = \frac{Nu\, k_f}{d_p} \tag{3.19}$$

Here $T(z^*)$ is a coarse-grained fluid temperature, T_s is the particle surface temperature assumed to be constant, and $A_p = 6\phi_p/d_p$ is the specific exchange surface. Then, equation 3.18 describes the steady advection of the fluid temperature field in the particle bed. We can substitute equation 3.19 into equation 3.18 and integrate to obtain:

$$\ln\left(\frac{T_s - T_{sat}}{T_s - T_0}\right) = -\frac{6Nu\phi_p}{Pe} z_{sat} \tag{3.20}$$

Noticing that the argument of the logarithm is precisely the saturation margin Θ with $\theta = (T - T_0)/(T_s - T_0)$, the dimensionless saturation length z_{sat} (made dimensionless using the particle diameter, i.e.$z_{sat} d_p = z^*_{sat}$) can be expressed as:

$$z_{sat} = -\frac{\ln(\Theta_{sat})}{\Lambda}; \quad \Lambda = \frac{6\phi_p Nu}{Pe} \tag{3.21}$$

The above expression gives a measure of the distance (in terms of particle diameters) after which the fluid is nearly in thermal equilibrium with the particle phase. We note in passing that equation 3.18 has been used in other studies to evaluate the mean Nusselt number in mono-disperse particle beds [22, 24, 88]. Specifically, equation 3.21 can be rearranged such that the value of the Nusselt number is computed from a certain saturation level that occurs at a distance z_{sat}:

$$Nu_{sat} = -\frac{\ln(\Theta_{sat}) Pe}{6\phi_p z_{sat}} \tag{3.22}$$

Equation 3.21 predicts $\lim_{Re\to 0} z_{sat} = 0$ which would mean that, for very low Reynolds numbers, no closure for the Nusselt number is needed at any scale since the temperature field will quickly saturate. However, this is purely a consequence from the assumption of a steady state: an analysis considering the transients for the fluid temperature yields the following result (respecting the boundary and initial condition $\Theta(0, z) = 1$ and $\Theta(t, 0) = 1$):

$$\Theta(t, z) = e^{-\Lambda z}\mathcal{H}(t - z) + e^{-\Lambda t}(1 - \mathcal{H}(t - z)) \tag{3.23}$$

Where $\mathcal{H}$ is the Heaviside step function. Comparison of equation 3.20 and 3.23 reveals that:

$$t_{sat} = z_{sat} = -\frac{\ln(\Theta_{sat})}{\Lambda} \tag{3.24}$$

Thus, Λ is both the dimensionless time and space relaxation factor, which is homogeneous in the domain. The latter is a fact of our assumption that the particle distribution is homogeneous in the domain. Notice that t_{sat} is a dimensionless time calculated as $t_{sat} = t^*_{sat} U_s / d_p$, being t^*_{sat} a dimensional time.

3.3.4. Saturation equation with longitudinal dispersion

A more sophisticated model, including an axial dispersion term, can be obtained from direct averaging of the transport equations:

$$\frac{\partial \Theta}{\partial t} + \frac{\partial \Theta}{\partial z} - \frac{1}{Pe_L}\frac{\partial^2 \Theta}{\partial z^2} = -\Lambda \Theta \tag{3.25}$$

Where Pe_L indicates the longitudinal Peclet number defined by the ratio of longitudinal convective and dispersive transport rates. Hence, Pe_L is fundamentally different from Pe, and typically $Pe_L < 2$ and a constant for large Reynolds numbers. When a fluid is flowing through a bed of inert particles, the measured dispersion is due to the combined effects of molecular diffusion and random fluid motion in the intersticises of the bed. The latter phenomenon, also known as pseudo-turbulence, is due to the complex flow pattern induced by the topology of the fluid domain. This results in an anisotropic diffusion tensor since the effects of dispersion are stronger in the direction parallel to the flow field (i.e., in the longitudinal direction) compared to the direction normal to it (i.e., the transverse direction). Several expressions have been suggested for Pe_L, but in this work we follow Delgado [25] and express Pe_L using:

$$\frac{1}{Pe_L} = \frac{1}{\kappa Pe} + \frac{Pe}{5}(1-\mathrm{p})^2 + \frac{Pe^2}{25}\mathrm{p}(1-\mathrm{p})^3\left(e^{-\frac{5}{\mathrm{p}(1-\mathrm{p})Pe}} - 1\right) \tag{3.26}$$

$$\mathrm{p} = \frac{0.48}{Pr^{0.15}} + \left(\frac{1}{2} - \frac{0.48}{Pr^{0.15}}\right) e^{-\frac{0.75Pr}{Pe}} \tag{3.27}$$

Where κ is the tortuosity (which equals $\sqrt{2}$ for spheres). One key feature of Pe_L is that for $Pe > 1$ the relation $Pe_L << Pe$ always holds.

Equation 3.25 can be solved using the same boundary and initial conditions used for 3.23, plus an additional one due to the second order term. Specifically, we choose $\Theta(t, \infty) = 0$, which means that the two phases are in thermal equilibrium far downstream of the inlet. This set of boundary conditions ultimately leads to:

$$\Theta(t,z) = e^{-\Lambda t} + z e^{z\frac{Pe_L}{2}} \int_0^t \frac{1 - e^{-\Lambda(t-\tau)}}{2\sqrt{\frac{\pi\tau^3}{Pe_L}}} e^{\left[\frac{z^2 Pe_L}{4\tau} - \left(\frac{Pe_L}{4} + \Lambda\right)\tau\right]} \mathrm{d}\tau \tag{3.28}$$

Unfortunately, the analysis of the transient response is complicated by the convolution term. However, a useful solution can be obtained for the steady-state situation which is much easier to manipulate:

$$\Theta(z) = e^{-z/\lambda_o} \tag{3.29}$$

$$\lambda_o = \frac{1}{\sqrt{\frac{Pe_L^2}{4} + Pe_L\Lambda} - \frac{Pe_L}{2}} \tag{3.30}$$

Where we introduced the characteristic fluid-particle system length λ_o which represents the distance at which a perturbation in the Θ field is reduced by a factor e in a fluid-particle system with a homogeneous distribution of isothermal particles.

Following equation 3.29, the saturation length becomes:

$$z_{sat} = -\ln\left(\Theta_{sat}\right)\lambda_o \tag{3.31}$$

It is easy to show that equation 3.31 returns equation 3.21 in the limit of infinite Pe_L (i.e., in the absence of longitudinal dispersion) for any finite value of Pe. Also, we note that λ_o is always larger than the corresponding length scale $1/\Lambda$ in a system without dispersion. Thus, longitudinal dispersion alleviates the saturation problem to some degree. Next, it can be shown that in the limit of $Pe \to 0$ equation 3.31 gives:

$$z_{sat} = -\frac{\ln\left(\Theta_{sat}\right)}{\sqrt{6\kappa\phi_p Nu|_{Pe=0}}} \tag{3.32}$$

A major difference between equations 3.28 and 3.23, is that when dispersion is considered, the steady state solution is only reached asymptotically. In fact, no Heaviside function appears in equation 3.28 and thus, the time t_{sat} can only be evaluated for $z > z_{sat}$ instead of $z = z_{sat}$. However, it is still true that, after a time t_{sat} is elapsed, we obtain $\Theta(t, z) > \Theta_{sat}(t_{sat}, z_{sat})$ in $z > z_{sat}$. Thus, also in this case after a dimesionless time t_{sat} has passed, the state of pseudo equilibrium is reached everywhere. In addition, numerical evaluation of equation 3.28 shows that:

$$t_{sat} < z_{sat} \tag{3.33}$$

Which means that for any position z downstream from z_{sat} the state of pseudo-equilibrium is reached (slightly) faster than in the dispersion-free case. We also observed that t_{sat} increases almost linearly with z_{sat}. It is then appropriate to state that

t_{sat} has approximately the same value as z_{sat} and thus the (dimensionless) heat and mass transfer space and time scales are of the same order for this class of fluid-particle systems.

Figure 3.1 summarizes typical values for z_{sat}, and illustrates that the saturation length can be rather small, even at high values of the Reynolds number and in dense beds. Thus, we observe for $\phi_p = 0.5$ that z_{sat} is always smaller than 20 (the saturation length was made dimensionless with the particle diameter) over the full range of Reynolds numbers, i.e., in such situations the fluid domain will saturate very quickly.

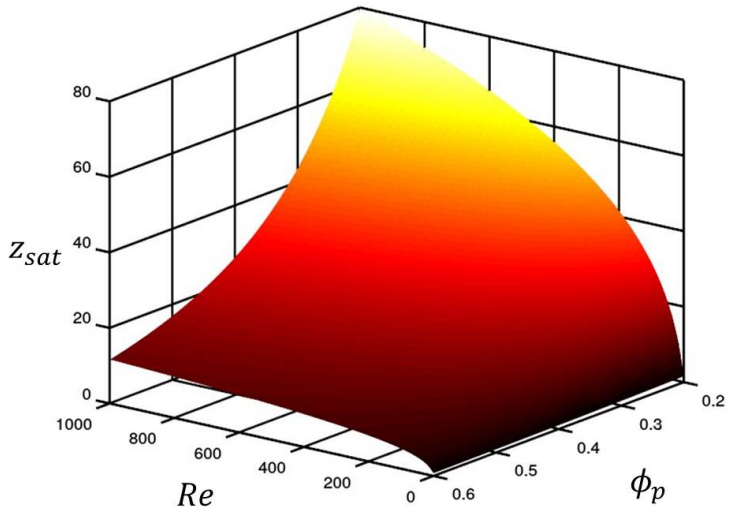


Figure 3.1.: Map showing the value of z_{sat} as a function of Re and ϕ_p. We used equation 3.15 to evaluate the Nusselt number, consider longitudinal dispersion, as well as assume $Pr = 1$.

As shown in figure 3.2, the above model with dispersion is able to reasonably predict the value of Θ and thus, the coarse-grained temperature in the particle bed. The overall agreement is satisfactory, especially considering that the comparison of equation 3.25 is carried out against calculations from ensemble averaged PR-DNS data. Also, Deen et al.'s correlation approximates our PR-DNS reasonably well (see Section 3.6.4).

3.4. Numerical formulation

The software employed in the present work, i.e., OpenFOAM®, uses co-located variables (i.e., all variables are defined at cell centers), and relies on a finite volume method for arbitrarily-shaped cells. The system of equations is solved in a segregated manner,

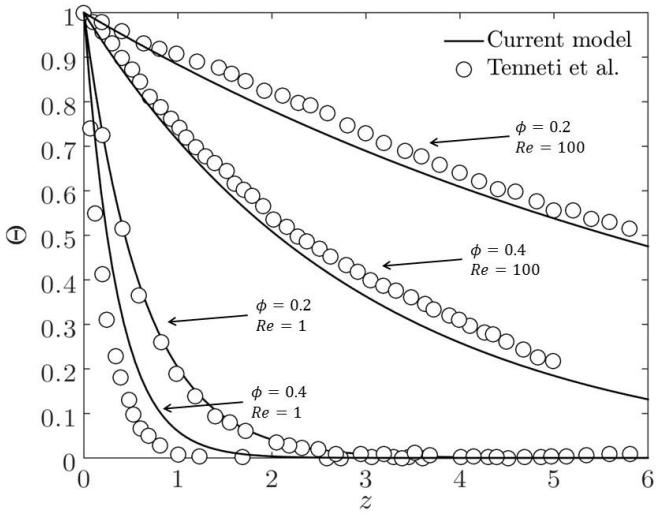


Figure 3.2.: Θ versus dimensionless streamwise coordinate z to compare DNS data from Tenneti et al. [93] with predictions from the one-dimensional convection-dispersion equation, i.e., Eqn. 3.31 using the closure for the Nusselt number from Deen et al. [24].

i..e, it is solved within an iterative sequence. In the present work we use a structured Cartesian mesh where spherical particles are represented by forcing terms in the matrix equations. This greatly reduces the efforts for generating the numerical grid (which is required for more standard body-fitted mesh approaches). Also, the presented approach ensures that no errors arise due to the grid cell topology, making the overall accuracy estimation easier. Local mesh refinement at particle surfaces is performed dividing each hexahedral cell into eight smaller hexahedral cells, and imposing a determinant equal to one on each new cell.

3.4.1. Hybrid fictitious-domain/immersed-boundary method

In a discrete computational domain using non-conformal mesh elements the topology of the fluid-solid interface can not be accurately represented. This lack of topological accuracy is affecting the particle fluid exchange processes and requires an analytic representation of the immersed boundary alongside with the appropriate forcing inside the immersed body. The HFD-IB (Hybrid Fictitious Domain Immersed Boundary) solver allows to represent the immersed surface, and to account for the presence of the immersed body. The idea is to combine the convergence properties of a Lagrangian multiplier fictitious domain method [7, 8, 42, 103, 113] with an immersed boundary method

[73, 98] that acts like a convergence accelerator.

The fictitious-domain/immersed-boundary method combines some elements from both formulations in order to:

- Provide an accurate representation of the particle topology in the discretized domain and, thus applying the corresponding boundary condition at the particle surface (which in general does not coincide with any cell node or cell center).
- Being convergent (i.e., recover the original equations when the grid size tends to zero).
- Ensure consistency of the imposed forcing/source terms with the interphase exchange process.

The latter assertion means that integrating the forcing/source term over the particle volume should give the interphase exchange rate. To be specific, the dimensionless force acting on a particle due to the fluid phase and the total dimensionless heat transferred from a particle i to the surrounding fluid should be respectively expressed by:

$$\boldsymbol{f}_i\left(t\right) = -\int_{\Omega_s^i} \boldsymbol{f}^{\circledcirc}\left(\boldsymbol{x},t\right) d^3\boldsymbol{x} \tag{3.34}$$

$$Q_i\left(t\right) = \int_{\Omega_s^i} Q^{\circledcirc}\left(\boldsymbol{x},t\right) d^3\boldsymbol{x} \tag{3.35}$$

Equations 3.34 and 3.35 represent the mapping of the interface coupling terms from the Eulerian fluid domain to the Lagrangian particle cloud. In order for equation 3.35 to hold, it has to be ensured that no heat transfer occurs inside the immersed body, i.e., the immersed body should be represented by a region of constant fluid temperature θ_i while an appropriate surface temperature θ_s should be imposed in order to satisfy the Dirichlet boundary condition. Equation 3.34 would require a similar condition (i.e., imposed fluid velocity inside the rigid body). However, the required forcing would trigger a flow circulation at the particle surface due to continuity (i.e., the fluid would flow outside the particle). This would result in the hot fluid inside the particle to be artificially transported to the particle surface creating an additional inter-phase heat flux not accounted in equation 3.35. If the immersed body rigidity is not imposed, the forcing term (which would be non-zero only at the particle boundary) would trigger a flow internal to the particle which would dissipate energy by friction, thus overestimating the inter-phase momentum transfer. In fact, the fluid inside the immersed body should be frictionless [78]. In order to avoid this problem, the HFD-IB forces the particle surface

only, but keeps track of the additional forcing that would be needed in order to stop the flow inside the immersed body. Equation 3.34 is then substituted by:

$$\boldsymbol{f}_i\left(t\right) = -\int_{\partial\Omega_s^i} \boldsymbol{f}^{\circledcirc}\left(\boldsymbol{x},t\right) d^3\boldsymbol{x} + \int_{\Omega_s^i - \partial\Omega_s^i} \Delta\boldsymbol{f}^{\circledcirc}\left(\boldsymbol{x},t\right) d^3\boldsymbol{x} \tag{3.36}$$

Where $\Delta\boldsymbol{f}^{\circledcirc}\left(\boldsymbol{x},t\right) = \boldsymbol{f}^{\circledcirc}\left(\boldsymbol{x},t\right) - \boldsymbol{f}^{\circledcirc}\left(\boldsymbol{x},t-\Delta t\right)$ is the change in the forcing term required to impose a fixed value of the velocity field inside the rigid body. Notice that $\delta\Omega_s^i$ is the discretized particle surface (i.e., cells at the particle surface) and thus the second term on the RHS of equation 3.36 represents the forcing only in the interior of particle i. This ansatz allows us to effectively remove the transfer of momentum to the inside of the particles, and Eqn. 3.36 now represents only the fluid-particle interaction force between the particles and the main flow outside of the particle domain. We note that $\Delta\boldsymbol{f}^{\circledcirc}$ does only contribute marginally to the total forcing (it is on the order of 0.01%), especially in case the flow and temperature fields are stationary. Also, we note that in other methods [34] the rigidity condition is still imposed inside the solid domain, possibly resulting in unwanted fluxes through the boundary.

3.4.1.1. Boundary layer reconstruction approach

An accurate imposition of the Dirichlet boundary condition at the particle surface is performed through a second order boundary layer reconstruction procedure. The present approach can be regarded as a generalized second order version of other approaches found in literature [41, 61, 62], since the method we propose is independent of the interpolation scheme used. In fact, our proposed method works with all interpolation schemes available in our software toolkit (i.e., OpenFOAM®). In our boundary layer reconstruction approach, field values in cells that are cut by the particle surface (denoted as "surface cells" below) are forced to assume certain values (i.e., $\boldsymbol{u}_I$ and θ_I). These values are generally different from the corresponding values located at the particle surface ($\boldsymbol{u}_s$ and θ_s). We now consider an interpolation stencil round each surface cell, and use a second order polynomial to reconstruct the fluid phase properties along the radial direction (see figure 3.3). The interpolation points are spaced by the surface distance s defined as:

$$s = \sqrt[3]{3V_c} \tag{3.37}$$

Where V_c is the volume of the surface cells. It is clear that s represents the diagonal of the cubic surface cell, so that the interpolation points will always be located in different cells when the numerical grid is sufficiently regular close to the particle surface, i.e. if the cell shape is approximately cubic and if there is no significant shape or size

difference between two neighboring cells. The surface cell value to be imposed is then evaluated from (for simplicity, we illustrate the interpolation procedure only for θ, similarly this idea can be applied to the velocity field):

$$\theta_{c,I} = \frac{\theta_{P2} - 2\theta_{P1} + \theta_s}{2s^2} d_{cs}^2 + \frac{4\theta_{P1} - \theta_{P2} - 3\theta_s}{2s} d_{cs} + \theta_s \tag{3.38}$$

Where θ_s is the particle surface value, θ_{P1} is the interpolated value of θ at point $P1$, θ_{P2} is the interpolated value of θ at point $P2$, and d_{cs} is the distance between the surface cell center and the particle surface.

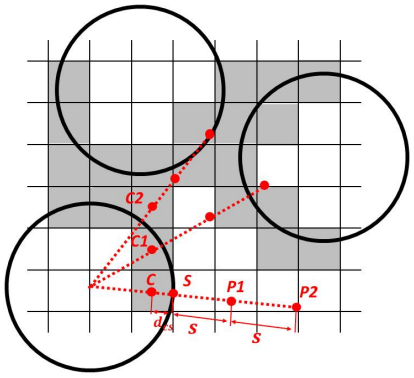


Figure 3.3.: Representation of immersed surfaces on a Cartesian grid (here shown in two dimensions for the sake of simplicity). Circles represent the immersed particle surfaces, while surface cells are colored in grey. Cell value at C is evaluated based on S, $P1$, and $P2$ using a second order polynomial. Conversely, $C1$ and $C2$ are evaluated using a first and zero order approximation, respectively, due to insufficient reconstruction points in the interpolation stencil.

It is clear that, in order for the interpolated field values to have a physical meaning, there should be no other particle within the interpolation stencil. While this condition is almost true for very dilute suspensions and for very fine (compared to the particle diameter) grids, it can not be guaranteed for a general situation. In order to overcome this problem, our proposed method adjust its order (and thus, its interpolation stencil) based on the value of ϕ in the cells close to the interpolation points. In fact, those cells whose center is inside a particle (i.e., $\phi > 0.5$) should not contribute to the boundary layer reconstruction since their value is merely used to satisfy the Dirichlet boundary condition for a certain particle surface cell. Since the interpolation itself may require a (possibly different) computational stencil, we discard an interpolation point if one

of its three closest cells has a value of ϕ higher than 0.5. An example is given in figure 3.3. Whenever an interpolation point is discarded, the order of the boundary layer reconstruction is automaticly descreased. That is, if only the furthest interpolation point is discarded the interpolation is carried out using:

$$\theta_{c,I} = \frac{\theta_{P1} - \theta_s}{s} d_{cs} + \theta_s \tag{3.39}$$

However, if the closest interpolation point is discarded, a zero order approximation is used:

$$\theta_{c,I} = \phi\theta_s + (1 - \phi)\theta_c \tag{3.40}$$

Notice that in equation 3.40 the new cell value is calculated using the field ϕ as a penalty factor like in the standard fictitious domain methods. Thus, a pure fictitious domain approach can be seen as a zero order HFD-IB. In addition, it is clear that in this method the immersed surface is discretized together with the numerical grid (i.e. no separate Lagrangian representation of the particle surface) and in the limit of zero grid spacing Δx we obtatin:

$$\lim_{\Delta x \to 0} s = 0 \quad \rightarrow \quad \theta_{c,I} = \theta_s \tag{3.41}$$

Thus, we impose the exact Dirichlet boundary condition and, in the limit, the method is formally equivalent to a Lagrangian multiplier fictitious domain.

3.4.1.2. Point interpolation methods

Since the HFD-IB is implemented in the OpenFOAM® framework, there are several interpolation schemes that can be used for the reconstruction. We tested all of them, and concluded that the best accuracy is achieved using the *cellPointFace* scheme which calculates the interpolated value as an average of closest vertex, face, and cell center values weighted with the inverse distance.

3.4.1.3. Fluid forcing and source term formulation

The forcing term for the momentum equation is formulated following the work of Blais et al. [13]. This approach is also similar to the one described in [103], and uses a parameter $\omega_f \in [0, 1]$ to adjust the implicitness of the force correction.

$$\boldsymbol{f}^{\circledcirc,n} = I_s^n \left(\boldsymbol{f}^{\circledcirc,n-1} + \omega_f \frac{\boldsymbol{u}_I^n - \boldsymbol{u}^n}{\Delta t} \right) \tag{3.42}$$

Where the superscripts n and $n-1$ refer to the current and previous step (referring to the steppig in the momentum corrector loop) respectively. Δt is the time step and I_s is a function that is 1 in particle surface cells and zero elsewhere. Our algorithm uses a PISO-IB loop, same as in [13], to correct the forcing and pressure field allowing to fulfill the continuity equation, and to enforce now flow through the immersed boundaries.

The source term for the scalar transport is formulated in a slightly different way since no correction loop is required. This is because the scalar and momentum equations are decoupled and the scalar transport equation is solved after the momentum corrector (i.e., using the solenoidal velocity field). Thus, we next consider the linear system arising from the discretization of the scalar transport equation. Each linear equation can be represented by:

$$\mathrm{M}_{ll}\theta_l = Q_l^{\circledcirc} - \sum_m \mathrm{M}_{ml}\theta_m \tag{3.43}$$

Where M is the coefficient matrix of the discretized system of equations and the two subscripts refer to the row and column index. Terms in M are evaluated from the OpenFOAM® matrix assembler following the discretization schemes defined in section 3.4.2. The local interphase heat exchange rate at the step $n+1$ is calculated from:

$$Q_l^{\circledcirc,n+1} = I_p \left(\mathrm{M}_{ll}\theta_{l,I}^n + \sum_m \mathrm{M}_{ml}\theta_m^n \right) \tag{3.44}$$

Being I_p a function that is 1 if the cell belongs to the particle (surface or interior cells) and zero elsewhere. When equation 3.44 is substituted into equation 3.43, wherever $I_p = 1$, we obtain:

$$\theta_l^{n+1} = \theta_{l,I}^n \tag{3.45}$$

Thus, the exact imposition of the interpolated value. Notice that, however, update of the imposed scalar field $\theta_{i,I}$ is performed explicitly.

3.4.2. Numerical solution

The momentum and energy equations were discretized in space using central difference schemes, and a backward scheme was adopted for discretization in time. The pressure coupling was carried out by means of a PISO algorithm that also corrects

the immersed boundary forcing [13]. All the discretization methods we adopted where second order accurate. The Poisson equation for the pressure field was solved using a diagonal incomplete-Cholesky preconditioned conjugate gradient method. The momentum and scalar transport equations were solved using a diagonal incomplete lower-upper preconditioned bi-conjugate gradient. In order to preserve the natural evolution of the simulated fields, no relaxation factors were used for equations or fields, i.e., the simulations were performed fully transient. The transient nature of our simulations is fundamentally due to the formulation of the forcing term in the Navier-Stokes equations (equation 3.42), but we interrupted our simulations when both the velocity and temperature fields reach a steady state conditions, i.e. when the integral carried over all the computational domain of $\boldsymbol{f}^{\circledcirc}$ and $Q^{\circledcirc}$ is not changing significantly with time. Specifically, a simulation ends if, for the last 25% of the simulated time, (i) the average absolute value of the fluctuations was less than 0.5% and (ii) the sum of the fluctuations was zero, i.e. if the integrals are weakly fluctuating but not evolving. It should be noticed that condition (ii) is always satisfied in our computational domains after a sufficiently long time due to the heat sink that balances the interphase heat exchange and the imposed pressure gradient that balances the interphase momentum exchange. The algorithm was tested and verified under several operating conditions against data found in literature. Results from some of these tests are presented in the Chapter 3.5.

3.4.3. Computational setup

We make use of a fully periodic rectangular cuboid domain filled with a bi-dispersed particle population (Figure 3.4). The particle configuration is generated using LIGGGHTS® [53] to generate a homogeneous distribution before the fluid simulation is started. We studied several cases for different values of the overall particle volume fraction ϕ and particle diameters d_1 and d_2.

The fluid flow is driven by imposing a pressure difference between two opposite faces of the domain in the x_1 direction (which we will refer as the streamwise direction). This pressure drop is adjusted such that the fluid superficial velocity $U(1-\phi)$ remains constant.

We included a sink term in equation 3.1c in order to enforce a predefined mean fluid temperature θ_{bulk} at the inlet (see figure 3.4). This is done in order to prevent saturation of the fluid phase with the transferred scalar field. Notice that our approach is similar to the one of Tenneti et al. [93] with the sole difference that we do not modify the periodic boundary, but we rescale the scalar in the first two cell layers downstream the inlet. This approach allows to study flow and scalar fields in fully developed conditions, i.e. without the influence of inlet or outlet boundary conditions. In this sense, our approach is significantly different from that of Tavassoli et al. [90] who used Dirichlet and Neumann boundary conditions for the inlet and outlet respectively.

It is clear that filtering the temperature field across streamwise periodic boundar-

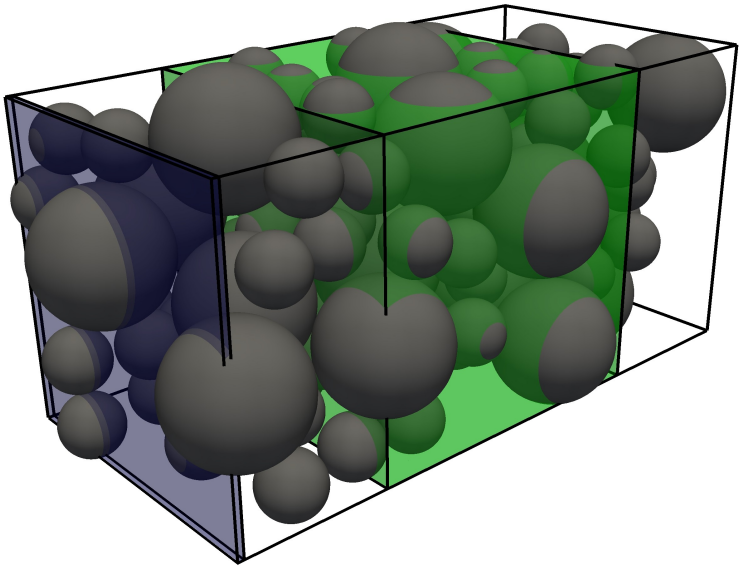


Figure 3.4.: Computational domain for the dense bi-disperse bed ($\phi = 0.5$,$d_2/d_1 = 2$). A mean flow field is imposed, which is aligned with the longest edge and so, the temperature inhomogeneity develops in that direction. The region where particle-based quantities can be studied for values of ρ up to 5 (thus at $2.5d_m$ from the streamwise boundaries) is colored in green. A heat sink is positioned at the domain entrance (colored in blue) to prevent saturation of the fluid with the transferred scalar.

ies produces unphysical results due to the jump in the temperature field. This means that particle-based quantities can only be studied at a certain distance away from the streamwise periodic boundaries (see figure 3.4). Thus, filtering requires relatively long computational domains, which increases the risk of fluid becoming saturated in the studied region. Hence, in order to prevent saturation phenomena, the Reynolds number must be sufficiently high to ensure a significant mean temperature gradient between fluid and particles over the full bed length.

3.5. Verification of the HFD-IB method

In this Chapter we demonstrate the correct implementation and the accuracy of the HFD-IB method by comparison with analytic solutions and existing closure in the frame of momentum, heat and mass transfer past one or more spheres.

3.5.1. Cooling of a sphere immersed in a stationary fluid

We consider the problem of a sphere of diameter d_p immersed in a stationary fluid (i.e., $\boldsymbol{u} = \mathbf{0}$). We compare the radial temperature profile $T(r,t)$ with the analytical solution for $r > d_p/2$. Due to the symmetry of the problem, the diffusion equation in spherical coordinates can be written as:

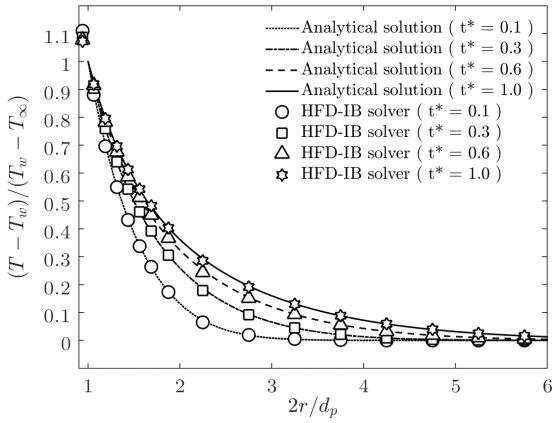


Figure 3.5.: Radial temperature field computed using the HFD-IB solver compared with the analytical solution. The maximum error of 0.85% occurs at the surface of the sphere at $t^* = 0.1$. The error quickly decays below 0.4%. Notice that the HFD-IB is imposing a temperature at the boundary cell that is higher than the wall temperature T_w.

$$\frac{\partial T(r,t)}{\partial t} = \frac{\alpha_f}{r^2}\frac{\partial}{\partial r}\left(r^2\frac{\partial T(r,t)}{\partial r}\right) \tag{3.46}$$

$$T(r,0) = 0, \quad T(d_p/2,t) = T_w, \quad T(\infty,t) = T_\infty \tag{3.47}$$

Where α_f is the heat diffusivity. The well known solution to this problem is:

$$T(r,t) = \frac{1}{r}\left[1 - erf\left(\frac{r - d_p/2}{2\sqrt{\alpha_f t}}\right)\right] \tag{3.48}$$

We considered a cubic box of $8d_p \times 8d_p \times 8d_p$ with the sphere placed in the center. We used a mesh resolution $h = \Delta x/d_p = 16$ and a dimensionless time step $\Delta t^* = \alpha_f \Delta t/d_p^2 = 10^{-3}$. The solver was found be in excellent agreement with the analytical solution with an average deviation of 0.2%.

3.5.2. Forced convection around a sphere

In this test case, transport equations 3.1a, 4.1b, and 3.1c are solved in a cuboid domain containing a single immersed sphere. We used a domain size of $8d_p \times 8d_p \times 16d_p$ and a dimensionless time step $\Delta t = 10^{-4}$. We focused on the local Nusselt number:

$$Nu = \frac{Q^* RePr}{\pi \left(\theta_b - \theta_s\right)} \tag{3.49}$$

Where Q^* is the total interface heat exchange rate calculated using equation 3.35 and θ_b is the dimensionless inlet temperature. We used fixed temperature and velocity boundary condition for the inlet and zero gradient boundary conditions for the remaining boundaries.

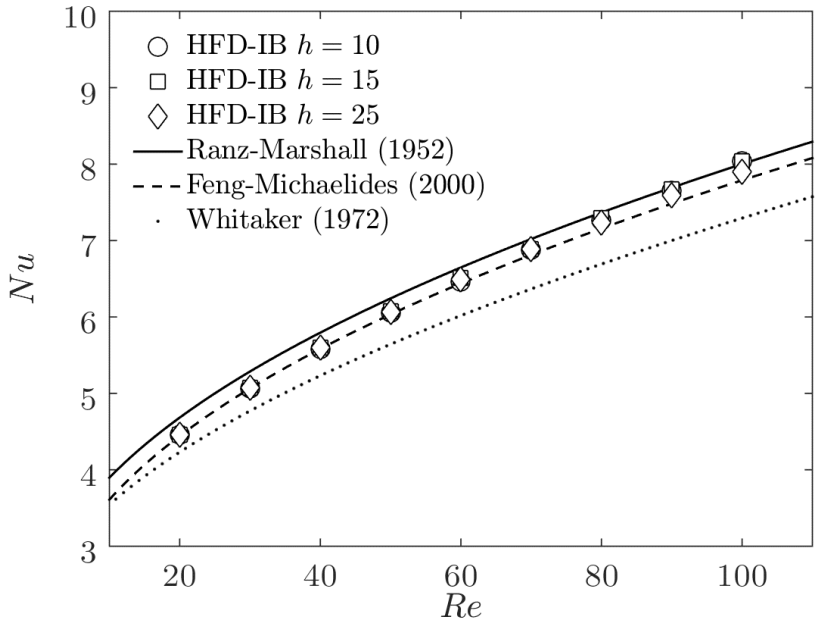


Figure 3.6.: Nusselt number calculated using the HFD-IB solver against results found in literature for different Reynolds numbers (Ranz and Marshall [77], Feng and Michaelides [33] and Whitaker [106]). $h = \Delta x/d_p$ denotes different grid resolutions.

Results showed good agreement with closures found in literature, especially with the numerical work from Feng and Michaelides [33]. In addition, figure 3.6 shows that the Nusselt number is correctly computed (i.e., it is consistent with the existing

closure) even for a relatively coarse mesh. The relative error with respect to Feng and Michaelides was found to be less than 1% for $Re < 70$ and increased up to 1.8% at $Re = 100$. We also examined the drag coefficient defined as:

$$C_d = \frac{8 f_x^{IB}}{\pi \rho_p d_p^2 U_\infty^2} \tag{3.50}$$

Where ρ_p is the particle density, U_∞ is the inlet velocity and f_x^{IB} is the stream-wise component of the dimensionless force acting on the particle calculated using equation 3.36. In figure 3.7 we plotted the values of C_d obtained for different Reynolds numbers against existing closure. The HFD-IB solver shows good agreement with literature for C_d also with relatively coarse grids. The error was found to be on the order of 4% for $Re = 100$ when using the finest grid and around 5% when the coarsest grid was used.

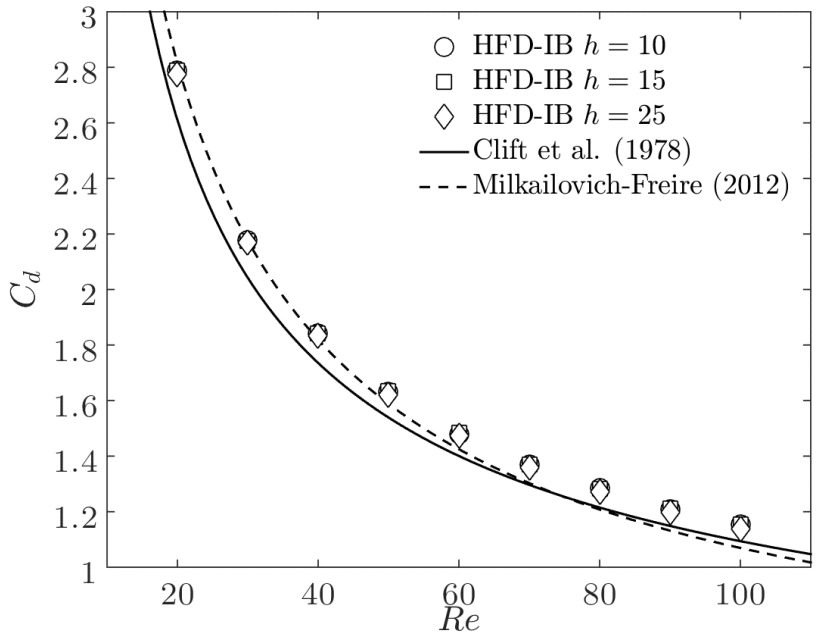


Figure 3.7.: Drag coefficient calculated using the HFD-IB solver against results found in literature [18, 65] for different Reynolds numbers. $h = \Delta x / d_p$ denotes different grid resolutions.

3.5.3. Creeping flow past a periodic static array of spheres

To simulate flow past a periodic array of spheres, it is sufficient to simulate the flow past a single sphere in a fully periodic box. This test case allows to evaluate the accuracy of the method when two or more particles are close or in contact and thus, to assess the effect of the adaptive order of accuracy for the boundary layer reconstruction. For this case, we solved equations 3.1a and 4.1b using a body force to drive a flow field with $Re = 2 \cdot 10^{-5}$ in order to ensure a Stokes flow regime. We then compared the dimensionless force defined as:

$$f^* = \frac{f_x^{IB}}{3\pi\mu_f U_0 d_p} \tag{3.51}$$

Where μ_f is the dimensionless dynamic viscosity of the fluid (set to $0.2\mathrm{Kgm}^2/\mathrm{s}$) and U_0 is the imposed dimensionless fluid velocity (set to 10^{-5}). We varied the dimensionless particle diameter d_p to obtain the desired value of ϕ in the periodic box.

Table 3.1.: Deviation from analytical results for the dimensionless drag force. $\Delta f^*/f^*$ represents the deviation from Zick and Homsy [115]. The settings we used are the same as that employed by Deen et al. [22].

ϕ	$d_p/\Delta x$	f^* Zick and Homsy (1982)	$\Delta f^*/f^*$ Deen et al. (2012)	$\Delta f^*/f^*$ this work
0.5236	32	42.14	1.3 %	1.1 %
0.450	30.4	28.1	-2.8 %	1.4 %
0.343	27.8	15.4	-1.2 %	2.7 %
0.216	23.8	7.442	-0.1 %	-0.8 %
0.125	19.9	4.292	-0.7 %	0.3 %
0.064	15.9	2.81	-1.3 %	-1.6 %
0.027	11.9	2.008	-1.9 %	-2.8 %

The computed values for the dimensionless force (see table 3.1) agree well with the analytical work of Zick and Homsy [115] with a deviation in the range with that obtained by Deen et al. [22]. This may be due to the fact that the forcing term in the immersed boundary they used was also based on a second order interpolation of the fluid variables as was done in the present contribution.

3.5.4. Forced convection past a chain of three spheres

Table 3.2.: Drag coefficient experienced by a chain of three spheres. We used $none$ to indicate that the deviation is below the precision reported in the work of Maheshwari et al. (two significant digits).

s	Re	C_d deviation from Maheshwari et al.		
		1st	2nd	3rd
2	1	0.8 %	0.1 %	0.2 %
2	10	1.0 %	0.4 %	$none$
2	50	2.0 %	$none$	-1.3 %
2	100	3.9 %	$none$	$none$
4	1	0.6 %	-0.8 %	-0.6 %
4	10	0.9 %	0.3 %	$none$
4	50	1.9 %	1.0 %	$none$
4	100	3.7 %	1.6 %	3.5 %

Table 3.3.: Nusselt number in the case of the chain of three spheres.

s	Re	Nu this work			Nu Tavassoli et al.			Nu Ramachandran et al.			Nu Maheshwari et al.		
		1st	2nd	3rd	1st	2nd	3rd	1st	2nd	3rd	1st	2nd	3rd
2	1	2.04	1.54	1.54	2.09	1.58	1.62	2.12	1.81	1.63	2.09	1.83	1.63
2	10	3.31	2.32	2.13	3.45	2.40	2.21	3.37	2.32	2.03	3.32	2.34	2.05
2	50	5.47	3.42	3.10	5.72	3.55	3.19	5.50	3.39	2.98	5.42	3.34	3.08
2	100	7.33	4.24	3.88	-	-	-	7.05	4.18	3.74	6.98	4.23	3.77
4	1	2.23	1.87	1.69	2.31	1.96	1.82	2.17	2.03	1.63	2.20	1.94	1.64
4	10	3.37	2.72	2.53	3.51	2.83	2.62	3.28	2.79	2.49	3.33	2.72	2.53
4	50	5.52	4.05	3.81	5.80	4.21	3.81	5.40	4.18	3.60	5.40	4.11	3.52
4	100	7.13	5.05	4.87	-	-	-	6.96	5.16	4.42	6.91	5.09	4.39

Finally, we consider the case of forced convection past a chain of three spheres. We compared our results for the particle Nusselt number defined in equation 3.49 and for the drag coefficient defined in equation 3.50. Same as Ramachandran et al. [76] and Masheshwari et al. [58], we used two different values for the spacing between the three particles, namely $s = 2d_p$ and $4d_p$. Furthermore, contrarily to the case with a single sphere, we used a cylindrical domain of length $16d_p$ or $20d_p$ (depending on the value of s) and radius 5_dp. The first sphere was centered at a distance $4d_p$ from the inlet. In order to compare with results from Tavassoli et al. [88], we limited the study to a grid resolution of $h = \Delta x/d_p = 15$ (the same grid size he used). We compared the

calculated drag coefficient with the one obtained in the work of Maheshwari et al. [58] (table 3.2). It can be seen that the HFD-IB agrees very well, often showing deviations of less than 1% from Mashewari's work. Benchmark for the Nusselt number (table 3.3), shows that the HFD-IB generally agrees with previous works and predicts the Nusselt number experienced by the first sphere with higher accuracy than the method of Tavassoli et al. [88]. This can be understood by noticing that, for a given grid size, the accuracy of the HFD-IB has a weaker dependency on the Reynolds number (see figure 3.6 in this work and figure 6 in the work of Tavassoli et al. [88]).

3.6. Simulation of bi-dispersed suspensions

3.6.1. Numerical settings

We restricted our study to the parameters shown in table 3.4 and to a domain size of $5d_m \times 5d_m \times 10d_m$ according to the available computational resources. The investigated values of the Reynolds number are relatively high when compared to previous studies [24, 87, 88] so that the heat exchange does not lead to unwanted excessive fluid phase saturation that would deteriorate the statistics of the Nusselt number. Unfortunately, high Reynolds number flow simulations are demanding in terms of mesh resolution, and hence require a fine computational mesh. Therefore, we performed a grid sensitivity analysis on the most critical cases to assess a suitable grid resolution and the associated error. We concluded that this grid resolution is $h = \Delta x/d_m = 50$ at the particle surface, while a larger gid spacing of $h = 25$ between the particles (i.e., in regions in which velocity gradients are less steep) is adequate. Specifically, in the present study we adopted a base mesh with $h = 25$, and then used mesh refinement at the particle surface. The refinement consisted of splitting a coarse cubic cell of the base mesh into eight equally-sized cubic subcells. The error due to the use of mesh refinement (compared to using a uniformly fine grid with $h = 50$) was found to be less than 1%. Overall, we estimated the maximum error connected to the evaluation of the Nusselt number (with respect to a grid independent solution) to be approximately 20%, similar to the error estimated in previous works [87].

3.6.2. Drag coefficient in bidisperse particle beds

In the present work, we imposed a constant average superficial fluid velocity in the simulated domain by means of a global pressure gradient. As a consequence, the total force acting on a particle $\boldsymbol{f}_i$ is next defined as the sum of the fluid drag force $\boldsymbol{f}_i^d$ and a contribution from a mean pressure gradient $\boldsymbol{f}_i^{\nabla p}$ that builds up in the bed of particles.

Table 3.4.: Numerical parameters used in the current study. η_M is the dimensionless particle diameter ratio, ϕ_m/ϕ_M is the ratio of the small particle volume fraction over that of the large particles, N_{pM} is the number of large particles, and N_{pm} is the number of small particles.

ϕ_p	η_M	ϕ_m/ϕ_M	Pe	N_{pM}	N_{pm}
0.20	1.2	0.5	100,250,400	37	33
0.35	1.2	0.5	100,250,400	64	55
0.50	1.2	0.5	100,250,400	92	79
0.20	1.2	1.0	100,250,400	28	48
0.35	1.2	1.0	100,250,400	48	83
0.50	1.2	1.0	100,250,400	69	119
0.20	1.2	2.0	100,250,400	18	62
0.35	1.2	2.0	100,250,400	32	111
0.50	1.2	2.0	100,250,400	46	159
0.20	1.6	0.5	100,250,400	16	33
0.35	1.6	0.5	100,250,400	27	57
0.50	1.6	0.5	100,250,400	39	82
0.20	1.6	1.0	100,250,400	12	49
0.35	1.6	1.0	100,250,400	20	82
0.50	1.6	1.0	100,250,400	29	119
0.20	1.6	2.0	100,250,400	8	66
0.35	1.6	2.0	100,250,400	14	115
0.50	1.6	2.0	100,250,400	19	156
0.20	2.0	0.5	100,250,400	8	32
0.35	2.0	0.5	100,250,400	14	56
0.50	2.0	0.5	100,250,400	20	80
0.20	2.0	1.0	100,250,400	6	48
0.35	2.0	1.0	100,250,400	10	80
0.50	2.0	1.0	100,250,400	15	120
0.20	2.0	2.0	100,250,400	4	64
0.35	2.0	2.0	100,250,400	7	112
0.50	2.0	2.0	100,250,400	10	160

Thus, we write:

$$\boldsymbol{f}_i = \boldsymbol{f}_i^d + \boldsymbol{f}_i^{\nabla p} \tag{3.52}$$

In simulations using a PU-EL model, the contribution from the mean pressure gradient can be explicitely computed based on a known (resolved, but spatially average) pressure field $\widetilde{p}$. We are therefore interested in the remaining force contribution, i.e., the drag contribution in Eqn. 3.52, for which a closure must be profided in PU-EL models. Thus, in what follows we filter the pressure gradient field $\boldsymbol{\nabla} p$ based on our PR-DNS data, and define the per-particle drag force as follows:

$$\boldsymbol{f}_i^d = \boldsymbol{f}_i - \frac{\pi}{6} d_i^3 \widetilde{\boldsymbol{\nabla} p_i} \tag{3.53}$$

We stress that this definition is the most natural choice, since $\widetilde{\boldsymbol{\nabla} p_i}$ is the pressure gradient that would be available in a PU-EL simulation where the CFD cells are of the same size as the filter size. We note that this definition of the drag force is different from that typically used in Euler-Euler simulations, since in the latter the global mean pressure gradient (experienced by all particles) is used to define the drag [46].

The drag force parallel to the main flow direction $f_i^{d\parallel}$ is then made dimensionless (or scaled) with the corresponding Stokes drag force to obtain (dimensional and dimensionless form):

$$F_i = \frac{f_i^{*,d\parallel}}{3\pi d_i \rho_f \nu (1-\overline{\phi}_i)\widetilde{u}_i^{*,\parallel}} = \frac{f_i^{d\parallel} Re}{3\pi \eta_i (1-\overline{\phi}_i)\widetilde{u}_i^{\parallel}} \tag{3.54}$$

Notice that, since we used filtered quantities (i.e., $\widetilde{u}_i^{\parallel}$ and $\overline{\phi}_i$), the dimensionless drag force calculated using Eqn. 3.54 does not respect the usual limit$\lim_{\phi,Re\to 0} F_i = 1$ for a finite value of ϕ. This happens because F_i is particle-based, and not an ensemble average. Thus, it is possible that the $f_i^{d\parallel} = 0$ while $\widetilde{u}_i^{\parallel} \neq 0$ (e.g., in case a particle does not experience a net drag force due to blockage from other particles, but the filtered velocity is non-zero). Our simulation results suggest that this results in F_i to be lower-bounded to 0 instead of 1.

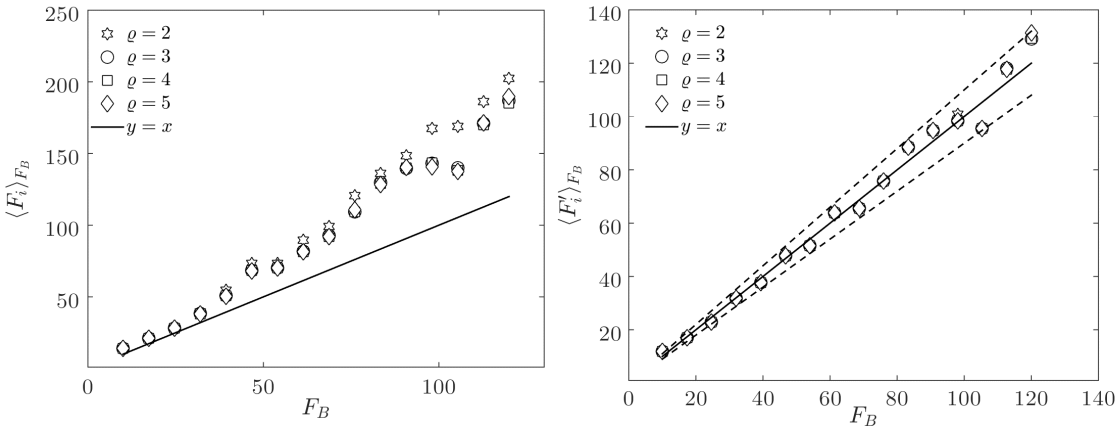
A comparison of the ensemble-averaged drag force evaluated from our PR-DNS data by using equations 3.52 and 3.53 with the closure provided by Beetstra et al. [11] is presented in figure 3.8. Beetstra's closure for drag force in bi-dispersed suspensions can be written as a correction to the mono-dispersed drag force $F_{i,B}^m(Re_i, \overline{\phi}_i)$, i.e.,

$$F_{i,B} = \left[\left(1-\overline{\phi}_i\right) y_i + \overline{\phi}_i y_i^2 + 0.064(1-\overline{\phi}_i)^3\right] F_{i,B}^m(Re_i, \overline{\phi}_i) \tag{3.55}$$

$$F_{i,B}^{m}(Re_i,\overline{\phi}_i) = \frac{10\overline{\phi}_i}{\left(1-\overline{\phi}_i\right)^2} + \left(1-\overline{\phi}_i\right)^2\left(1+1.5\overline{\phi}_i^{\frac{1}{2}}\right) + \frac{0.413Re_i}{24\left(1-\overline{\phi}_i\right)^2}\left[\frac{\left(1-\overline{\phi}_i\right)^{-1}+3\overline{\phi}_i\left(1-\overline{\phi}_i\right)+8.4Re_i^{-0.343}}{1+10^{3\overline{\phi}_i}Re_i^{-(1+4\overline{\phi}_i)/2}}\right] \tag{3.56}$$

$$Re_i = Re(1-\overline{\phi}_i)|\widetilde{u}_i|\eta_i; \quad y_i = \frac{d_i}{d_{32}} \tag{3.57}$$

Notice that, in order to calculate $F_{i,B}$, we only used values obtained by filtering within a box with dimensionless size ϱ. Furthermore, we note that Re_i does not differ significantly if defined using the parallel (to the main flow) filtered velocity $\widetilde{u}_i^{\parallel}$, or the modulus $|\widetilde{u}_i|$



(a) (b)

Figure 3.8.: Parity plot comparing the normalized drag force from the current simulations with values predicted by Beetstra et al. Dashed lines represent 10% deviation from the closure of Beetstra. In panel 3.8a, $\boldsymbol{f}_i^d$ was calculated using equations 3.52 and 3.53. In panel 3.8b the drag force $\langle F_i' \rangle_{F_B}$ accounts for the pressure gradient contribution using equation 3.58 same as in the work of Beetstra et al.

Panel a of figure 3.8 reveals, on average, a significant deviation from the closure proposed in the work of Beetstra et al. Such a deviation can be explained examining the approach they used to calculate the pressure contribution to the drag force. Specifically, two facts need to be considered:

i Since the particle distribution is homogeneous, it is possible to define a global pressure gradient which corresponds to the pressure gradient required to drive the fluid flow through the interstices between particles in an infinitely large bed (or, in other words, through the equivalent porous medium). Therefore, following our method the local filtered pressure gradient should approach the global one in the limit $\varrho \to \infty$. However, as visible from panel a of figure 3.8, our results are insensitive of ϱ already for $\varrho > 3$. This was also confirmed by checking the mean difference between the global pressure gradient (calculated as the pressure gradient required to drive the flow field through the whole computational domain) and that experienced by individual particles (data not shown; more details are, however, provided in 3.7). Thus, the use of the local pressure gradient in place of the global one, contributed significantly only for $\varrho < 3$. A tentative physical explanation is that, for the moderate Reynolds numbers studied here, the fluctuations of the pressure due to acceleration and deceleration of the fluid become already significant, and are noticeable only for small filter sizes. The approach of Beetstra et al. does not account for filter sizes, and only the global pressure gradient was considered (however, as discussed before, it was not subtracted explicitly). This contributes (in part) to the discrepancy observed in Figure 3.8a.

ii Deviations from the correlation of Beetstra et al. for values of $\varrho > 3$ can be explained by considering that the approach used in their work to remove the pressure contribution from the total drag force can not be applied to polydisperse suspensions or particle-based forces. In fact, we show in 3.7 that implicit accounting of this pressure gradient term can only be performed on globally averaged quantities (i.e., ensemble averages over all particles and species). In 3.7 we also show that our data is in good agreement with the correct theoretical predictions for the mean pressure gradient contribution. On the contrary, this term is overestimated in the work of Beetstra et al. and thus, they obtain a smaller drag force.

In 3.7 we show that, under particular assumptions, the ensemble average drag force exerted on all particles can be expressed as the ensemble average total fluid-particle force multiplied by $(1 - \phi_p)$. In the work of Beetstra et al., the pressure contribution to the bi-disperse drag force was (incorrectly) accounted for by exploiting this relation, i.e., they used:

$$F_k(\phi, Re) = -(1 - \phi_p)\frac{\langle F_{g\to s}\rangle}{3\pi\mu d_k v} \tag{3.58}$$

Being $\langle F_{g\to s}\rangle$ the ensemble averaged fluid-particle force (acting on particles of specie k) and v the particle velocity. We find that our results are in good agreement (within 10%) if equation 3.58 was used to account for the pressure gradient force $\boldsymbol{f}_i^{\nabla p}$ and

when using $\overline{\phi}_i$ instead of ϕ_p (see panel b of figure 3.8). In other words, our data suggests that the average pressure gradient seen by the particles is different (i.e., smaller, since we compute higher drag coefficients) from the average pressure gradient in the whole particle bed. A tentative physical explanation is that for the relatively high Reynolds numbers studied here, the fluctuations of the pressure due to acceleration and deceleration of the fluid become already significant. We speculate that this leads to the observed differences visible in panels a and b of figure 3.8). Clearly, the correct way to account for the pressure contribution in PU-EL models is provided by equation 3.53, since each particle is supposed to experience a different local pressure gradient. In addition, the value of $\widetilde{\boldsymbol{\nabla} p_i}$ is directly available in PU-EL simulations and thus, it can be directly used to compute the per-particle pressure gradient force. Hence, we need to refit the expression suggested by Beetstra et al. to provide a drag closure for PU-EL models. Panel b in Figure 3.8 suggest that a simple correction can be applied to $F_{i,B}$ to obtain such a PU-EL-consistent drag force closure. Specifically, we propose:

$$F_{i,corr} = -0.122 + 1.18F_{i,B} + 0.00352F_{i,B}^2 \tag{3.59}$$

Equation 3.59 was obtained by fitting data from $\varrho = 3$ in figure 3.8a and ensuring $F_{i,corr} = 1$ when $F_{i,B} = 1$.

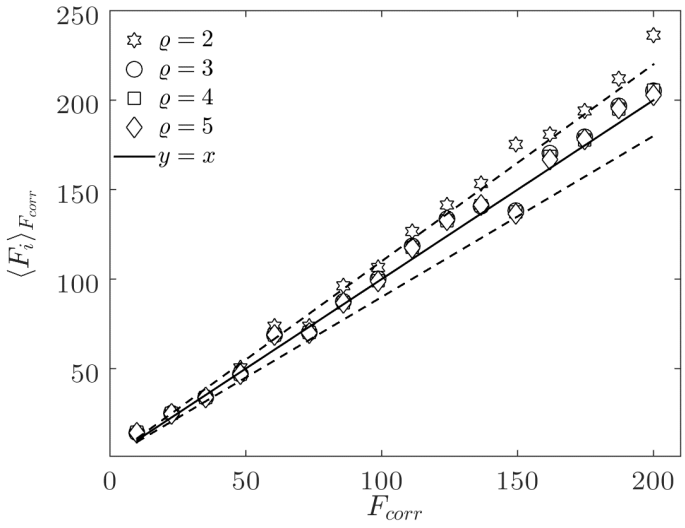


Figure 3.9.: Parity plot comparing the drag force from the current simulations calculated using equation 3.54 with values predicted by equation. Dashed lines represent 10% deviation from the closure.

The resulting agreement has an error of 4.7% on average, as well as a maximum deviation of 19%. We also binned the deviation from equation 3.59 with Re, ϕ and d_i/d_{32}, and found excellent agreement (i.e., absolute deviations between 5% and 7%). It can be noticed that the drag force in case $\varrho = 2$ is significantly deviating from the cases with larger filter sizes. We found that this difference can be entirely attributed to the pressure contribution in equation 3.53, as it is also visible from the different qualitative behavior of the trends as a function of ϱ shown in figure 3.8a and 3.8b. This means that a coarse grained cell with $\varrho = 2$ is still affected by local fluctuations of the pressure field. However, in the case of a non-homogeneous system, the resulting drag force $\boldsymbol{f}_i^d$ calculated according to equation 3.53 will depend on ϱ and the relative position of neighboring particles, so that equation 3.59 will have to take into account sub-grid contributions. This is clearly posing a limit to the grid size in PU-EL since, in case of a non-homogeneous particle distribution, fluid quantities like the velocity field and the pressure gradient may be very different from the homogeneous values used to derive these closures. Thus, specific models for non-homogeneous configurations should be used instead, which are beyond the scope of our present contribution.

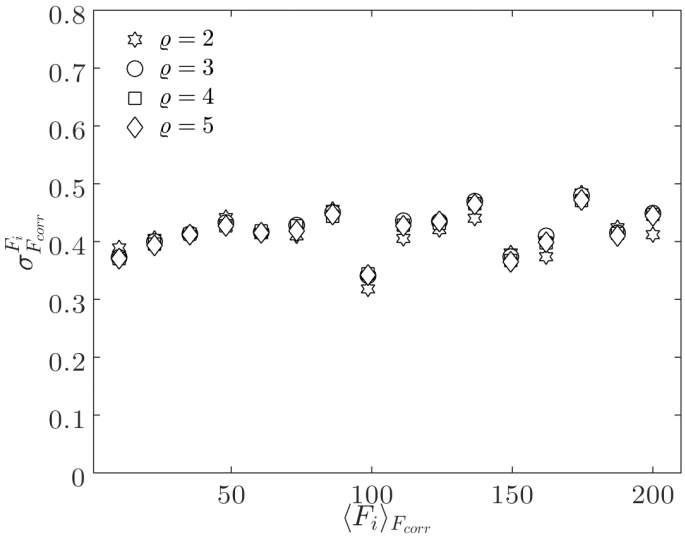


Figure 3.10.: Scaled standard deviation $\sigma^{F_i}_{F_{corr}}$ over $\langle F_i \rangle_{F_{corr}}$.

We now examine the distribution of the dimensionless per-particle drag force, and consider its scaled standard deviation $\sigma^{F_i}_{F_{corr}}$.

As shown in Figure 3.10, we did not observe any significant dependence of $\sigma^{F_i}_{F_{corr}}$ from the mean drag coefficient F_{corr}. This suggests that the relative deviation of F_i from $\langle F_i \rangle_{F_{corr}}$ can be assumed to be approximately constant for the flow regime studied in the present work. Specifically, values of $\sigma^{F_i}_{F_{corr}}$ ranged from 32% to 52% which

is generally attributed to the presence of preferred pathways of the flow through the particle array [24]. We note in passing that this is in line with the previous results of Kriebitzsch et al. [54] once their data is rescaled (see the conclusions sections of the present contribution for an in-depth discussion).

The lack of a functional dependence on the mean drag coefficient results in a similar distribution for the relative deviations at different average drag coefficients $\langle F_i \rangle_{F_{corr}}$ (see figure 3.11). In the present work, our data suggests to represent the stochastic fluctuation of the drag force using a modified log normal distribution, i.e. transposed in such a way that the expectation value is zero. This is in contrast to previous work that considered a Gaussian distribution [54], and motivated by the following facts:

i A log-normal distribution is defined in the range $[0, \infty[$ and, thus, naturally predicts a minimum allowed value for drag coefficient, while the Gauss distribution is defined in $]-\infty, \infty[$ allowing the drag coefficient to assume negative values.

ii As can be seen from figure 3.11, the distribution of relative deviations is skewed. This is also a consequence of the drag coefficient being lower-bounded.

iii Our log-normal distribution can be defined using only one parameter (i.e., the standard deviation), which we have demonstrated being a constant within the range of our study. In fact, physically, the minimum allowable value for the relative deviation of -1 (i.e., zero drag force) and the expectation value is, by definition, equal to zero. Thus, this results in a standard log normal distribution with mean value equal to 1.

Clearly, the support of the log normal distribution has to match with that of the drag force relative deviation. This is performed by translating the distribution. Thus, defining the relative deviation $F_{i,\sigma} = (F_i - \langle F_i \rangle_{F_{corr}})/\langle F_i \rangle_{F_{corr}}$ and the lower bound as $\beta = -1$, the modified log-Normal distribution can be written as:

$$\mathscr{P}(F_{i,\sigma}) = \frac{1}{(F_{i,\sigma} - \beta)\alpha_F \sqrt{2\pi}} \exp\left\{ - \left[\frac{\ln(F_{i,\sigma} - \beta)}{\sqrt{2}\alpha_F} \right]^2 \right\} \tag{3.60}$$

As mentioned in item (iii) above, the shape of $\mathscr{P}(F_{i,\sigma})$ is determined by only one parameter (α_F) which turns out to be a constant. By averaging the values of $\sigma^{F_i}_{F_{corr}}$ as they are shown in Figure 3.10, we conclude that to a first approximation:

$$\alpha_F \approx 0.416 \tag{3.61}$$

The choice of a log-normal distribution also allows the stochastic drag coefficient model to be efficiently implemented in PU-EL codes: if ϵ_i is a randomly generated

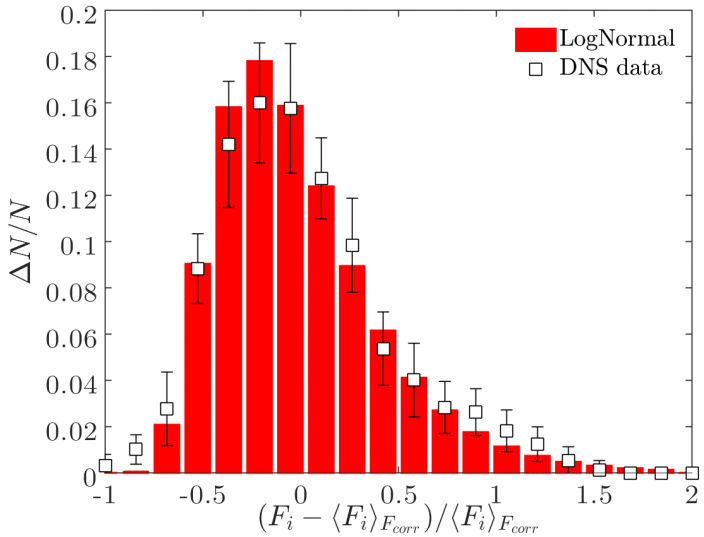


Figure 3.11.: Number distribution of the relative deviation of the conditionally averaged drag force for different values of F_{corr}. The error bar length corresponds to two standard deviation (i.e., the 68.2% error intervall) of the samples in each bin.

number between 0 and 1, the value of $F_{i,\sigma}$ can be calculated by inverting:

$$\epsilon_i = \int_0^{F_{i,\sigma}^p - \beta_F} \mathscr{P}(x)dx \tag{3.62}$$

Where the integral on the right end side is given for a log-normal distribution by:

$$\int_0^{F_{i,\sigma}-\beta} \mathscr{P}(x)dx = \frac{1}{2} + \frac{1}{2}erf\left[\frac{\ln(F_{i,\sigma}-\beta)}{\sqrt{2}\alpha_F}\right] \tag{3.63}$$

Which leads to:

$$F_{i,\sigma} = \beta + \exp\left[erf^{-1}(2\epsilon_i - 1)\sqrt{2}\alpha_F\right] \tag{3.64}$$

The inverse error function can be obtained using pre-computed lookup tables. In case the coefficients are assumed to be constant (as in our study) the whole expression can be tabulated before starting the computation.

While equation 3.62 can be used to compute the per-particle relative deviations from $\langle F_i \rangle_{F_{corr}}$, the important questions on how this random deviation should be evolved in

time remains open. For the time being we note that our model would allow to assign a specific value for $F_{i,\sigma}$ for each particle, and keep this value throughout a PU-EL simulation.

After inverting equation 3.62, the dimensionless drag force acting on a particle i can be computed from:

$$F_i = F_{i,corr}(1 + F_{i,\sigma}) \tag{3.65}$$

However, it should be noted that in the limit $\phi, Re \to 0$, the model for the drag force fluctuation is still predicting a drag force between -1 and 2. Even though a physical interpretation of the above limiting behavior is not as straightforward as one might think (and lies beyond the scope of our present contribution), it is generally accepted that, in this limit, the resulting drag should return Stokes drag law (i.e., $F_i = 1$).

The proposed model for the drag force fluctuation is able to consistently take this limit into account provided that a suitable expression for the standard deviation α_F is provided. In fact, if all the particles tend to experience the same drag force, it follows that the limit $\phi, Re \to 0$ implies $\alpha_F \to 0$, thus making the modified log normal distribution tend to a Dirac delta function. Thus, the future exploration of such a range of parameters (i.e., $\phi, Re \to 0$) is required in order to obtain a suitable expression for α_F.

Another approach that is less consistent, but perhaps more practical and applicable, consists in calculating the particle-based drag force using the following expression:

$$F_i = 1 + \gamma_F F_{i,corr}\left(1 - \frac{1}{F_{i,corr}} + F_{i,\sigma}\right) \tag{3.66}$$

Where a simple functional form for the unknown function γ_F would be the relation:

$$\gamma_F = \frac{F_{i,corr} - 1}{F_{i,corr}} \tag{3.67}$$

It is clear that using definition 3.67 for γ_F in equation 3.66 enforces the Stokes limit (i.e., $F_{i,corr} \to 1$), while it approaches the original closure defined in equation 3.65 for large values of $F_{i,corr}$.

As figure 3.11 demonstrates, our proposed model for the drag force fluctuation is able to capture the general trend of the relative drag force deviation reasonably well. However, since we have probed relatively large Reynolds numbers only in the present study, we lack of data that would justify the above presented drag force model for drag coefficients close to unity.

3.6.3. Nusselt numbers in bidisperse particle beds

The particle-based Nusselt number is defined based on dimensional and dimensionless quantities, respectively, as:

$$Nu_i = \frac{Q_i^*}{\pi d_i \lambda_f (\theta_s^* - \widetilde{\theta^*}_i)} = \frac{Q_i Pe}{\pi \eta_i (\theta_s - \widetilde{\theta}_i)} \tag{3.68}$$

In this work, we do not aim on establishing a closure for the bidisperse Nusselt number as a function of the monodisperse one, but we seek for an analogy between heat/mass and momentum transfer. Thus, we will express the bidisperse Nusselt number as a function of the bidisperse drag force, i.e. F_{corr}. There are some advantages in using this approach to formulate a closure model:

i Existing closures for monodisperse Nusselt numbers are often obtained by flux averaging (in contrast to volume averaging) the temperature field and thus, they have to be corrected using equation 3.17 in order to be applied in coarse-grained models. This operation requires the use of an additional closure and thus, reduces the overall predictive power and range of applicability of the resulting closure model. In addition, we found that scaling the Nusselt number using such closures increases data scattering.

ii Since closures often require evaluation of long expressions, a simple relation between drag and Nusselt number would be more efficient when implemented in PU-EL or Euler-Euler codes. In most applications (e.g., simulation of heat transfer in fluidized beds) the drag coefficient must be evaluated anyhow, such hat a simple relation between drag coefficient an Nusselt number could save computation time.

iii The range of validity of existing closure models for the Nusselt number (generally $Re \leq 100$ and valid only for monodisperse systems) is outside the range of parameters considered in our work. Hence, it is not useful to refit parameters in existing correlations, since the functional form for a Nusselt number closure might be fundamentally different.

Since we limited our study to $Pr = 1$, we must assume that the dependence on the Prandtl number is in the form of $Pr^{1/3}$. Thus, the resulting functional form we adapt is:

$$Nu_{i,bi}(Pr, Re, \phi_p, ...) = Pr^{1/3} G(F_{i,corr}) \tag{3.69}$$

Thus, any information regarding flow variables or particle population is concealed in F_{corr} and correlated with Nu_{bi} by means of the function G. Indeed, and as shown

in Figure 3.12, we find a linear relationship of our PR-DNS data for the mean drag coefficient and the Nusselt number. Specifically, we find that

$$Nu_{i,bi} = Pr^{1/3}(12.2 + 0.312F_{i,corr}) \tag{3.70}$$

The average deviation of equation 3.70 with respect to our data is 4.6%, with a maximum error of 9.5%.

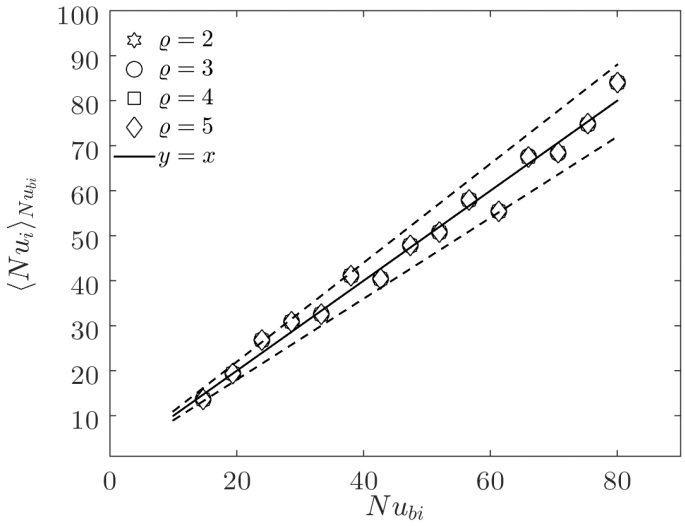


Figure 3.12.: Parity plot comparing the calculated Nusselt number from the PR-DNS evaluated using equation 3.68 with values predicted using equation 3.70. Dashed lines represent a 10% error corridor.

Examing the distribution of the per-per-particle Nusselt number reveals some interesting finding: again we observed (see Figure 3.13) that the scaled standard deviation is approximately constant for the Nusselt number. The standard deviation ranges from approximately 30% to 45%. Following the same approach as for F_{corr} we propose a model for the per-particle Nusselt number based on a log-normally distributed variable as shown in Figure 3.14.

Compared to the drag coefficient the log-normal distribution for the Nusselt number is more peaked (i.e., the standard deviation is smaller), and therefore necessitates the use of a α_{Nu} parameter in place of α_F. In order to represent our PR-DNS data resonably well, the following value for α_{Nu} is recommended:

$$\alpha_{Nu} \approx 0.336 \tag{3.71}$$

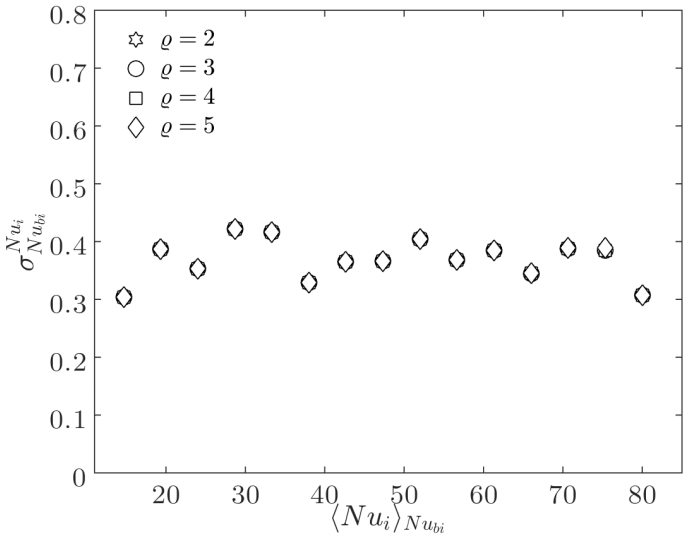


Figure 3.13.: Scaled standard deviation $\sigma^{Nu_i}_{Nu_b i}$ over $\langle Nu_i \rangle_{Nu_{bi}}$. Same as for the drag force F the standard deviation remains approximately constant over the range of mean Nusselt numbers studied.

3.6.4. Global mean Nusselt numbers

We now evaluate the global mean (mixture) Nusselt number which is defined as the Nusselt number that satisfies (with reference to equation 3.21):

$$Nu_{mix} = \frac{\Lambda Pe}{6\phi_p}\frac{d_{32}}{d_m} \tag{3.72}$$

Thus, the above Nusselt number describes the fluid-particle mixture as an homogeneous medium, and is evaluated from a known value for Λ. Notice that we rescaled our Peclet number, defined using d_m, with the mean Sauter diameter d_{32}. Following the discussion in section 3.3, we may express Λ at a certain position $\eta = x/d_m$ as:

$$\Lambda(\eta) = \frac{1}{Pe_L}\left(\frac{Pe_L}{2} - \frac{\ln\Theta(\eta)}{\eta}\right)^2 - \frac{Pe_L}{4} \tag{3.73}$$

Notice that η should be defined in such a way that $\Theta(\eta) = 1$ at $\eta = 0$ to be consistent with the boundary conditions used in equation 3.25.

In order to evaluate the global mixture Nusselt number, we sampled Θ along the

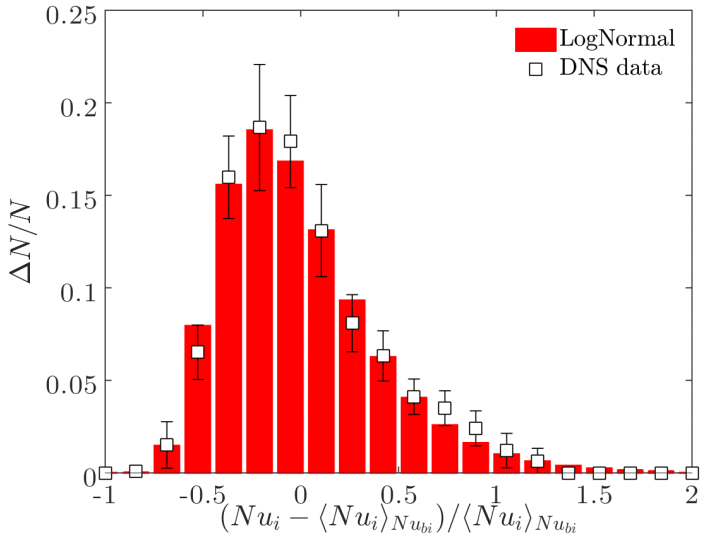


Figure 3.14.: Number-based distribution of the relative deviation of the conditionally averaged Nusselt number for different values of Nu_{bi}. The error bar length corresponds to two standard deviation (i.e., the 68.2% error intervall) of the samples in each bin.

streamwise direction and performed an average over the domain length L:

$$Nu_{bi}^{G} = \frac{1}{L}\int_{0}^{L} Nu_{mix}(\eta)\mathrm{d}\eta \approx \frac{1}{N_s}\sum_{i=1}^{N_s} Nu_{mix}(\eta_i) \tag{3.74}$$

Where N_s is the number of samples taken at different equally spaced locations η_i.

Results from this analysis for the case with and without dispersion (i.e., $Pe_L \to \infty$) are compared with the correlation developed by Deen et al. shown in Figure 3.15. Our data suggest that the Nusselt number calculated using this approach (which for the case without dispersion coincides with the approach used by Tavassoli et al. [90]) approaches the value obtained using the closure provided by Deen et al. [24]. The latter closure has been also confirmed by the recent work of Tavassoli et al. [90], highlighting that our results for the mean heat transfer coefficient are in reasonable agreement with results from literature.

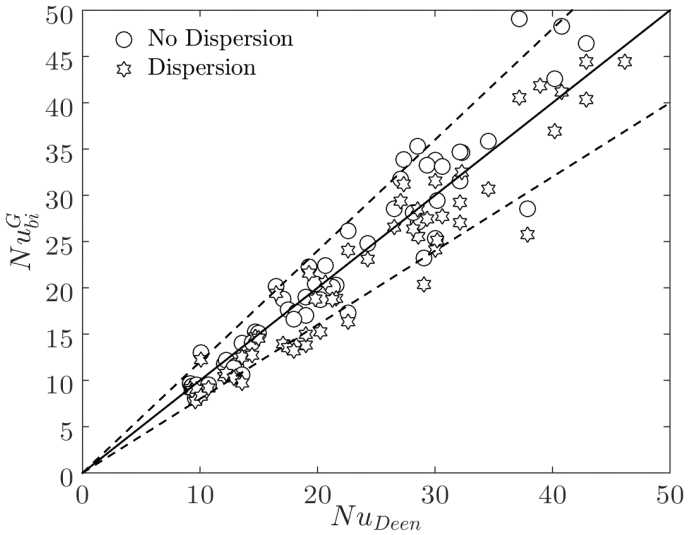


Figure 3.15.: Parity plot comparing the simulated global Nusselt number Nu_{sim} with the corresponding values from Deen et al. [24]. Dotted lines indicate an error corridor of $\pm 20\%$.

3.7. Particle-based and ensemble averaged drag force

The main focus of this work is to establish correlations for particle-based drag force and Nusselt number, and hence the question arises how to account for the pressure contribution when computing drag forces. In this Chapter we show that the standard approach adopted by Beetstra et al. [11] cannot be used to account for the pressure contribution $\boldsymbol{f}_i^{\nabla p}$.

3.7.1. Notation and basic definitions

In this Chapter we will refer to i as the particle index and to j as the particle class index (referred to particles belonging to class number j, e.g., particles having diameter d_j) . Thus, in what follows each particle is characterized by two indexes i and j. Furthermore, we will denote with $\langle(*)\rangle_j$ the ensemble average over particles within their particle class, and with $\{(*)\}$ the ensemble average over all the particle classes. To be more precise, referring to the total force $\boldsymbol{f}_{j,i}$ we define:

$$\langle \boldsymbol{f} \rangle_j = \frac{1}{N_j} \sum_{i=1}^{N_j} \boldsymbol{f}_{i,j} \tag{3.75}$$

$$\{\langle \boldsymbol{f} \rangle\} = \frac{1}{N_{classes}} \sum_{j=1}^{N_{classes}} \langle \boldsymbol{f} \rangle_j \tag{3.76}$$

These averages have to be intended as performed within a configuration, thus an homogeneous distribution of particles experiencing the same global pressure gradient $\boldsymbol{\nabla} p$ and characterized by a global particle volume fraction $\phi = \sum_j \phi_j$, a global Reynolds number Re, and number of particles $N_{tot} = \sum_j N_j$.

We express the total force acting on a single particle using:

$$\boldsymbol{f}_{j,i} = \boldsymbol{f}_{j,i}^{d} + \boldsymbol{f}_{j,i}^{\nabla p} \tag{3.77}$$

Which is the same as equation 3.52 but now generalized to differentiate between particles in multiple classes. Notice that the index i is not the same as equation 3.52 since in Eqn. 3.77 it is conditioned to class j. Furthermore, we account for the pressure term $\boldsymbol{f}_i^{\nabla p}$ by using:

$$\boldsymbol{f}_{j,i}^{\nabla p} = \boldsymbol{f}_{j}^{\nabla p} = -\frac{\pi}{6} d_j^3 \boldsymbol{\nabla} p = -V_j \boldsymbol{\nabla} p \tag{3.78}$$

Which is the same for particles belonging to the same particle class. Notice that we indicated with V_j the volume of a single particle in class j.

Clearly, the following force balance must hold:

$$N_{tot}\{\langle \boldsymbol{f} \rangle\} = -V \boldsymbol{\nabla} p \tag{3.79}$$

Being V the total volume of the domain (i.e., that of the fluid and all particles).

3.7.2. Case with one particle class - monodisperse suspensions

Considering the case of just one particle class, the operator $\{(*)\}$ has no effect, and we can drop the index j. Then, substituting equation 3.78 into equation 3.77 and substituting the resulting pressure gradient into equation 3.79 we obtain:

$$\boldsymbol{f}_i^{d} = (1-\phi)\boldsymbol{f}_i - \phi(\langle \boldsymbol{f} \rangle - \boldsymbol{f}_i) \tag{3.80}$$

Which, after application of operator $\langle (*) \rangle_j$, returns the equation used in Beetstra et al. [11] to account for the pressure contribution in the ensemble averaged drag force.

However, the additional term $\phi(\langle \boldsymbol{f} \rangle - \boldsymbol{f}_i)$ shows that, when extracting statistics for particle-based quantities, the approach of Beetstra et al. [11] cannot be straightforwardly applied, simply because $\boldsymbol{f}_i^d \neq (1-\phi)\boldsymbol{f}_i$. In other words, the pressure term can be regarded as a collective term since it can be indirectly accounted for (i.e., without directly using $\boldsymbol{\nabla} p$) only by considering the average total interphase force in a monodisperse fluid-particle system.

3.7.3. Case with two particle classes - polydisperse suspensions

If we consider an arbitrary number of particle classes, we find that we cannot account for the individual pressure contribution without considering the globally-averaged drag force $\{\langle \boldsymbol{f} \rangle\}$. Thus, only a relation between globally-averaged quantities can be found:

$$\{\langle \boldsymbol{f}_d \rangle\} = (1-\phi)\{\langle \boldsymbol{f} \rangle\} \tag{3.81}$$

Furthermore, an equation equivalent to 3.80, but this time based on class-averaged particle forces rather than individual particles, can be obtained:

$$\langle \boldsymbol{f}_d \rangle_j = \left(1 - \phi_j \frac{N_{tot}}{N_j}\right) \langle \boldsymbol{f} \rangle_j - \phi_j \frac{N_{tot}}{N_j} \left(\{\langle \boldsymbol{f}_d \rangle\} - \langle \boldsymbol{f} \rangle_j\right) \tag{3.82}$$

This equation returns equation 3.81 when averaged over all the particle classes. Notice that these equations imply $\langle \boldsymbol{f}_d \rangle_j \neq (1-\phi)\langle \boldsymbol{f} \rangle_j$, which is the approach used by Beetstra et al. [11]. Also other expressions like $\langle \boldsymbol{f}_d \rangle_j = (1-\phi_j)\langle \boldsymbol{f} \rangle_j$ are clearly erroneous. The reason can be found by examining equation 3.79. The global balance implies that the pressure gradient is connected with the sum of all interphase forces and thus, cannot be expressed as a function of just one subset (i.e., one particle or one particle class) of the fluid-particle system.

3.7.4. Application to the current case

The approach used in the current study (i.e., that presented in section 3.6) is consistent with the above considerations. Specifically, we show that using the Favre averaged pressure gradient is in agreement with the following equation:

$$\boldsymbol{f}_{j,i}^d = \boldsymbol{f}_{j,i} - \phi_j \frac{N_{tot}}{N_j} \{\langle \boldsymbol{f} \rangle\} \tag{3.83}$$

Which can be obtained straightforwardly by substituting equation 3.78 into 3.77 and finally into 3.79. If equation 3.83 is satisfied in our simulations, then our approach is consistent. The term N_{tot}/N_j is, for a bidiperse system, given by from ϕ_m/ϕ_M:

$$N_{tot}/N_j = \left(1 + \frac{d_j^2}{d_k^2}\frac{\phi_k}{\phi_j}\right) \quad k, j = 1, 2 \quad k \neq j \tag{3.84}$$

The term $\{\langle \boldsymbol{f} \rangle\}$ is typically not reported in literature (also not by Beetstra et al.). However, one may use the drag from Beetstra et al. [11] (i.e., as defined in equation 3.55) and consider (the incorrect) equation 3.58 to backup the total iteraction force implied by Beetstra et al.:

$$\{\langle \boldsymbol{f} \rangle\} = \frac{N_1 d_1 F_{B,1} + N_2 d_2 F_{B,2}}{N_{tot}(1-\phi)} \tag{3.85}$$

Where $1, 2$ refer to the two particle classes considered here.

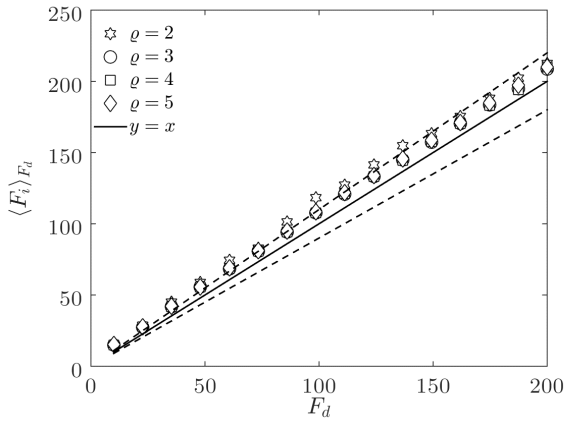


Figure 3.16.: Parity plot comparing the dimensionless drag force from the current simulations (i.e., F_i) calculated using equation 3.54, with predictions based on equation 3.83 (i.e., F_d) that rely on the closure of Beetstra et al. The dashed lines represent a 10% error corridor.

Figure 3.16 shows that equation 3.83 slightly underpredicts the drag force from our simulation but still agrees within 10% for filter sizes $\rho > 2$. This deviation can be interpreted as the true difference between our simulation results and the one of Beetstra et al. after applying the same correction for the pressure gradient.

3.8. Conclusions

In the present contribution we studied momentum and heat transport in gas-solid bi-dispersed suspensions. We made use of PR-DNS to derive closures for particle-based quantities and explored the range of usability of models used to describe heat transfer in dense particle beds. Using a simple one-dimensional model for the gas-phase mixture obtained by analytically averaging the thermal transport equations, we derived an expression for a characteristic length scale that characterizes heat (or mass) transfer process in dense homogeneous fluid-particle systems:

$$\lambda_o = \frac{1}{\sqrt{\frac{Pe_L^2}{4} + Pe_L\Lambda} - \frac{Pe_L}{2}}$$

Furthermore, we applied the idea of a fluid phase saturation length, and showed its relevance for a broad range of Reynolds numbers and particle volume concentrations. We showed that saturation is connected to the λ_o parameter, and occurs within a few particle diameters in certain situations. The effect of axial dispersion, widely neglected in the literature, was demonstrated to be of major importance.

The problem of saturation is already discussed in literature, however, in a more qualitative way as shown in the work of Tenneti et al. [93], as well as Sun et al. [87]. In our work we have shown that the value of the saturation time and length scale, i.e., (t_{sat} and z_{sat}), may be on the order of a few particle diameters. In such a situation the temperature gradient in the fluid will be extremely high, and the fluid will quickly approach a state of thermal equilibrium with the surrounding particles. It is therefore clear that for small values of z_{sat} the exact value for the Nusselt number is of secondary importance for computing the amount of exchanged heat in a particle bed with length $z < z_{sat}$. Also, in a situation in which z_{sat} is small, the time step required to explicitly evaluate the transferred amount of heat in a PU-EL model will be extremely small. Hence, one would adopt an implicit coupling strategy to ensure stability of the numerical algorithm, at the cost of precision with respect to energy conservation (since implicit coupling cannot ensure identical heat fluxes for both phases). In conclusion we recommend to focus on sufficiently high Peclet number for future studies to quantify the Nusselt number, since these situations are of higher practical interest.

In the present work, we also proposed a modified hybrid fictitious domain-immersed boundary (HFD-IB) method to account for the presence of immersed bodies in non-isothermal flows. This method combines the advantageous convergence properties of a Lagrangian multiplier fictitious domain method with a second order accurate immersed boundary method. The HFD-IB was shown to be accurate, even on relatively coarse grids, in predicting momentum and heat transfer gas in gas-particle system.

The final part of our study applied the HFD-IB to investigate flow through bidisperse beds of spherical particles. We identified that the closure provided by Beetstra et al.

[11] for bi-disperse suspensions (which was also shown to be suitable also for tri-disperse suspensions, see [48]) is inaccurate when used in PU-EL-based models. This is because this previous closure does not account for the local mean pressure gradient each particle experiences, but simply accounted for the global pressure gradient in the domain. Since the local mean pressure gradient is known in PU-EL simulations, we proposed a correction to calculate the drag contribution to the total interphase force:

$$F_{i,corr} = -0.1216 + 1.181F_{i,B} + 0.00352F_{i,B}^2$$

The conditionally averaged drag force was shown to agree with this new closure within an average deviation of 4.7%.

We also analyzed the variability of the per-particle drag coefficient, concluding that a log-normal distribution describes this variability reasonably well. This detail is significant, since the recent work of Buist et al. [14] clearly demonstrated that per-particle and mean Nusselt numbers differ (we note in passing that Buist et al. considered the additional complication of a non-uniform particle temperature distribution). Future work might probe whether our proposed model for the per-particle Nusselt number is indeed able to support these findings of Buist et al. [14]. Most important, we showed that the standard deviation relative to the mean drag force is approximately constant over the range of systems studied. This might appear in contrast to previous work [54] on the first view. However, it is important to note that the previous study of Kriebitzsch et al. [54] considered the fluctuation of the total fluid-particle interaction force relative to its mean. This previous study concluded that the fluctuation of the total force decreases with increasing particle concentration. It is now important to consider that (i) the fluctuation of the mean pressure gradient force is weak, as well as (ii) that the pressure gradient increases with increasing particle concentration. Thus, the relative force fluctuations will naturally become smaller with increasing particle concentration, simply because of the larger pressure gradient contribution to the total force. Re-scaling the data in Figure 13 of Kriebitzsch et al. [54] with $1/(1-\phi_p)$, i.e., considering the mean drag force as the reference quantity, suggested that fluctuations relative to the drag force are (to a first approximation, and within the scatter of data presented) insensitive to the mean drag force. This is in line with the present contribution, which clearly demonstrated a constant relative drag force variability over a wide range of drag coefficients.

Finally we assessed the existence of a relationship between the average drag force and Nusselt number. Therefore, we first define the Nusselt number (and in particular the coarse-grained temperature) in a way that is consistent with the corresponding coarse-grained model (i.e., a PU-EL model). Hence we do not require additional closures that relate the coarse-grained temperature with a cup-mixing temperature as proposed by Sun et al. [87]. Based on our PR-DNS data we propose the following closure

to evaluate the particle-based Nusselt number that relies on the following expression:

$$Nu_{i,bi} = Pr^{1/3}(12.2 + 0.312F_{i,corr}) \tag{3.86}$$

This closure fits our data with an average deviation of 4.6%, and we show that our results for a global bed-average Nusselt number is in agreement with the recent study of Tavassoli et al. [90]. Again considering the per-particle variability, however this time for the Nusselt number, we again find that a modified log-normal distribution describes this variability reasonably well.

On a final note, we suggest that the expression we provided for $Nu_{i,bi}$ can be modified to return the value corresponding to an isolated particle immersed in a stationary fluid, i.e., $Nu_{i,bi} = 2$ in the limit $\phi, Re \to 0$. Thus, the particle based Nusselt number can be rewritten (following the same approach used to obtain equation 3.66) as:

$$Nu_{i,bi} = 2 + Pr^{1/3}\frac{F_{i,corr} - 1}{F_{i,corr}}\left(12.2 - \frac{2}{Pr^{1/3}} + 0.312F_{i,corr}\right) \tag{3.87}$$

4. Momentum, heat and mass transfer simulations of bounded dense mono-dispersed gas-particle systems

Particle Resolved Direct Numerical Simulation (PR-DNS) is employed to study momentum, heat and mass transfer in confined gas-particle suspensions. In this work, we show that the presence of wall boundaries induces an inhomogeneous particle distribution, and as a consequence continuous phase fields exhibit peculiar profiles in the wall-normal direction. Therefore, we first propose a correlation for the particle volume fraction as a function of the distance from the wall and the bulk particle concentration. Secondly, we quantify wall effects on flow field and interphase transfer coefficients (i.e., the flow field, a scalar field, as well as the Nusselt number and drag coefficient). We show that these effects do not depend significantly on the Reynolds number in case an appropriate scaling is applied. Finally, we propose correlations to reconstruct the continuous phase fields in the proximity of adiabatic walls. Also, we provide interpolation tables for the correction to the drag force and the Nusselt number that are helpful in unresolved Euler-Lagrange simulations.

4.1. Introduction

Confined suspensions are a topic of active research since they are of use in a wide range of industrial processes like energy storage, heterogeneous catalytic reactors, pulp fibers, separation in micro-channels, or the petroleum industry. Other applications include blood flow in the human body, sediment transport in river beds and pyroclastic flows from vulcanos. Current developments indicate that the confinement effect in suspension flows becomes even more important: for example, 3D printing technology aiming on producing materials capable to be used at high temperatures (e.g., metals or ceramics) is already reality. This enables the use of complex geometries with characteristic dimensions closer to that of the suspended particles. In such systems wall effects will play a central role. In addition, the accurate modeling of momentum, heat and mass transport in dense gas-particle systems is of pivotal importance for designing

chemical reactors [29, 30], and many other systems, e.g., future solar-thermal systems [63, 111]. Again, the effect of confinement plays a central role in most of these applications, and is potentially becoming more important.

Thanks to the continuous increase in the availability of computational resources, hybrid Computational Fluid Dynamics-Discrete Element Method (CFD-DEM) simulations have become a tool for studying such dispersed multiphase systems [15]. A specific example is the so-called Particle-Unresolved Euler-Lagrange (PU-EL) formulation in which each particle trajectory is followed and particle-particle interactions are resolved. The governing equations for descriping continuous phase flow are formulated at a length scale larger than the particle characteristic length. Therefore, one has to solve coarse-grained equations for the continuous phase. Unlike the Euler-Euler (EE) formulation where the dispersed and continuous phases are described as interpenetrating continua, PU-EL formulations allow to directly study intra-particle transport phenomena. This is since the PU-EL formulation still retains the definition of single particles as separate discrete entities. Therefore, PU-EL simulations are best suited for studying complex systems of chemically reacting particles for which it is difficult (or even impossible) to formulate a continuous dispersed phase model with the desired accuracy. Similarly, modeling systems comprised of non-spherical particles is most natural, and pheraphs successful, when using a EL-based model.

However, coarse grained equations in PU-EL formulations have several unclosed terms (e.g., the drag coefficient, the pseudo-turbulent stress, or the interphase heat and mass transfer coefficients) for which one has to provide suitable expressions. In our previous work [69] we showed how such models can be constructed from Particle-Resolved Euler-Lagrange (PR-EL) simulations by means of volume averaging in a way that is consistent with the PU-EL formulation. It was shown that when the filter size is small (i.e., in the order of two times the particle diameter) significant differences arise with respect to EE closures due to local inhomogeneous structures. In other words, EE-based closures cannot be simply used in PU-EL-based simulation models. Furthermore, PU-EL models perform often poorer compared to EE-based models: inaccuracies caused by the interpolation and mapping scheme used to calculate the local voidage may deteriorate the fidelity of PU-EL models [80]. Particularly interesting aspects surface in case walls are present in the region to be modeled:

- the presence of walls induces an inhomogeneous distribution of particles which affects the flow field and the interphase transfer coefficients. This effect is not acounted for in the totality of closures currently used in EE and PU-EL models.

- since the details of the flow field near the wall are not known, typically the slip condition for the fluid is employed in EE and PU-EL models. This leads to significant uncertainties when interpolating the fluid velocity at the particle position near walls. This issue is especially relevant for size-polydisperse suspensions.

- for PU-EL models, the issue of insufficient mesh resolution in case heterogeneous particle structures exist has been systematically explored only in unbounded domains [75]. One would expect that similar issues arise in case the suspension is confined by walls. Conceptually, one could envision treating such wall effects similar to what is done in wall-bounded turbulent flows (e.g., one could employ wall functions). Unfortunately, such concepts are currently not available for dense fluid-particle flows.

A first step to systematically investigate the above aspects would be to quantify wall effects in an isolated fashion, i.e., separate them from the curvature effect that is typically included in the analysis (see Theuerkauf et al. [94], or van Antwerpen et al. [100]). Also, little is known for more dilute and intermediately dense suspensions, since most previous work explored packed beds only. Considering a wider parameter space is, however, essential when building a robust, generally-applicable simulation model. In our present contribution we indeed show that the particle concentration has a pronounced effect on both the velocity and temperature (as a proxy for any scalar) field. This is even the case for the simplest situation of adiabatic walls. We will start our analysis by considering the origin of these effects, namely the particle distribution near the wall.

4.1.1. Particle distribution in wall bounded domains

Extensive studies have been dedicated to the prediction of particle volume concentration of packed beds in the near wall region. In packed beds, the first layer of spherical particles in contact with the walls is characterized by having a well ordered distribution. Most of these wall-near particles are indeed in contact with the wall. Such ordering is progressively lost in the subsequent layers until the particle distribution becomes statistically homogeneous, i.e., the average volume concentration does not vary from one layer to the other anymore.

In earliest works, analytical expressions for the particle volume fraction profile was obtained by volume integration over concentrical annuli in cylindrical packings [64], or over wall-normal layers [55]. More recent studies proposed correlations for these profiles [20, 67], and a comprehensive review of existing correlations is provided by [100]. As discussed above, these studies emphasize on packed beds with cylindrical walls and therefore, they do not distinguish between the effect of the wall curvature and that of the wall alone. Furthermore, most available correlations are based on experimental data. They cover a rather small spectrum of particle volume fractions (typically between 0.35 and 0.65). Surprisingly, there is, to the best of our knowledge, currently no expression for the particle volume fraction as a function of the distance normal to a flat wall for a given particle concentration in the bulk of the particle bed.

4.1.2. Momentum, heat and mass transfer in bounded suspensions

It is not surprising that the peculiar particle ordering near walls leads to a substantial changes in the flow structures: Studies considering cylindrical packed beds revealed that anomalous transport phenomena occur in the fluid that flows in the proximity of a wall boundary [9, 29, 30, 85, 100]. Specifically, it was shown that the inhomogeneous particle distribution leads to characteristic profiles for the flow variables in the direction normal to the wall: For example, the velocity field experiences a parabolic profile (with a characteristic lateral extension of less than one particle diameter) in the vicinity of a wall [29, 40]. Therefore, considering that the particle volume concentration experiences a local minimum for particles in contact with walls, one may expect that the flow rate in the proximity of walls to be significantly larger than that in the bulk of the bed. This would be especially true in geometries which are characterized by a small particle diameter-to-wall distance ratio (i.e., narrow beds or small cylinders) because the the particle volume concentration in the bulk (i.e., center of cylinder or symmetry plane between two walls) will be significantly larger than the average value. Therefore, detailed modeling of near wall perturbations would increase the predictive power of both PU-EL and EE models, since current models cannot account for these effects.

4.1.3. Goals and outline

In the present work, we first quantify the effect of wall boundaries on the particle volume distribution for flat walls. Unlike previous works [94, 100] we do not limit our study to packed beds, but we also cover more dilute suspensions for which comparably little is known. The goal is to obtain an accurate correlation to describe wall induced disturbances for a wide range of average particle volume concentrations. Also, we consider flat walls to eliminate curvature effects. This is motivated by (i) the fact that in a typical PU-EL model only the wall-normal distance is known, but not the curvature, and that (ii) the correlation should be as generally-applicable as possible, and hence wall and curvature effect must be separated.

Secondly, we quantify the effect of such disturbances on the velocity field and a scalar non-reactive field. We put emphasis on quantitative analysis and we provide correlations for all relevant quantities. The ultimate goal of this effort is the replace the primitive slip boundary condition frequently used by a more physical boundary treatment.

Finally, we investigate the effect of walls on the interphase transfer coefficients (drag coefficent and Nusselt number). This is motivated by the current inability of closures to account for the presence of walls in a generic fashion.

This work is structured as follows: background on the mathematical description, together with the key quantities is summarized in Section 4.2. In Section 4.3 we present

details on the numerical solution of the equations presented in Section 4.2. Results from our simulations are presented in Section 4.4. In Section 4.5 we summarize our results and bring them into perspective.

Supplementary material that details the master curves needed to compute the drag and Nusselt numbers as a function of the wall normal distance is available in 4.6.

4.2. Mathematical description

4.2.1. Transport equations

In the present work, we solve the incompressible Navier-Stokes equations together with an advection-diffusion equation for a dimensionless inert scalar $\theta(t, \boldsymbol{x})$ to model momentum, heat and mass transport in dense gas-particle systems. The governing equations are formulated in their dimensionless form, and are defined within the fluid domain Ω_f:

$$\boldsymbol{\nabla} \cdot \boldsymbol{u}\,(t, \boldsymbol{x}) = 0 \tag{4.1a}$$

$$\left[\partial_t + \boldsymbol{u}\,(t, \boldsymbol{x}) \cdot \boldsymbol{\nabla} - Re^{-1}\nabla^2\right] \boldsymbol{u}\,(t, \boldsymbol{x}) = -\boldsymbol{\nabla} p\,(t, \boldsymbol{x}) \tag{4.1b}$$

$$\left[\partial_t + \boldsymbol{u}\,(t, \boldsymbol{x}) \cdot \boldsymbol{\nabla} - Pe^{-1}\nabla^2\right] \theta\,(t, \boldsymbol{x}) = 0 \tag{4.2}$$

Where $\boldsymbol{u}$ is the dimensionless velocity field, p is the dimensionless (dynamic) pressure, Re is the Reynolds number, θ is the dimensionless inert scalar field, and Pe is the Peclet number defined as $Pe = RePr$ being Pr the Prandtl number. In this work, we will always set $Pr = 1$ so that $Pe = Re$ and therefore, the differential operators on the left hand side of 4.1b and 4.2 are the same. Equations 4.1 and 4.2 are valid for $\boldsymbol{x} \in \Omega_f$

4.2.1.1. Dimensional flow variables

In order to avoid confusion in the scaling process of equations 4.1 and 4.2 we briefly introduce the procedure we used for obtaining dimensionless quantities:

i All length scales are made dimensionless with the particle diameter d_p.

ii The dimensionless velocity field $\boldsymbol{u}$ is obtained by scaling the dimensional velocity field with a reference velocity magnitude U_{ref}. The latter is determined from the Reynolds number using $U_{ref} = (Re\mu) / (\rho d_p)$ being μ the dynamic fluid viscosity and ρ the fluid density.

iii Time scales were made dimensionless employing (i) and (ii). Therefore, the dimensional time scales with d_p/U_{ref}.

iv The dimensional (dynamic) pressure scales with ρU_{ref}^2. Similarly, a component of the dimensionless stress tensor $\tau_{xz} = Re^{-1}\partial_z u_x$ (with $\boldsymbol{u} = (u_x, u_y, u_z)$) is related to its dimensional value by a multiplicative factor of ρU_{ref}^2.

v Since we consider a constant surface scalar value c_s, which is the same for each particle, we defined the dimensional scalar as $c = \theta\,(c_s - c_i) + c_i$ where c_i is a reference value of the scalar whose meaning is give from Eqn. 4.6 shown below. We note in passing that such a scaling is possible due to the linearity of equation 4.2.

Following the above procedure, and considering that $Pe = (\rho C_p U_{ref} d_p) / \lambda_f$ (where λ_f is the fluid heat conductivity and C_p is the fluid's specific heat capacity), we can conclude that the interphase heat source for particle i scales as $d_p^{-1}\rho C_p U_{ref}(c_s - c_i)Q_i$. Here Q_i is the dimensionless interphase scalar transfer rate of particle i. Otherwise, if one applies equation 4.2 to the transport of some inert substance in the fluid, then our Peclet number Pe can be expressed as $Pe = (U_{ref} d_p) / \mathscr{D}$, where $\mathscr{D}$ is the diffusion coefficient. The interphase source term is then defined via $d_p^{-1} U_{ref}(c_s - c_i)Q_i$.

Similarly, the interphase force f_i was made dimensionless using a scaling factor of $\rho U_s^2 / d_p$.

4.2.2. Boundary conditions

We define a global domain $\Omega = \Omega_f \cup \Omega_i = [0, L_x] \times [0, L_y] \times [0, L_z]$, where Ω_i is the region occupied by particle i and where L_x,L_y and L_z represent the extension of the global domain in the x, y and z directions, respectively. We can define two kind of boundaries: fluid-particle boundaries Γ_i and a global domain boundary Γ_g. Standard Neumann boundary conditions are employed for the pressure field at walls. At particle surfaces we apply the following boundary conditions:

$$\boldsymbol{u}(t, \boldsymbol{x}) = \boldsymbol{0} \quad \text{and} \quad \theta(t, \boldsymbol{x}) = \theta_s, \quad \forall \boldsymbol{x} \in \Gamma_i, \quad i = 1, 2, \ldots, \mathrm{N}_p \tag{4.3}$$

Where N_p is the number of particles in Ω and θ_s is the particle surface scalar value. In the present study, Γ_g is a semi-periodic boundary with adiabatic walls where the

following boundary conditions apply:

$$\boldsymbol{u}(t,0,y,z) = \boldsymbol{u}(t,L_x,y,z), \quad \theta(t,0,y,z) = \theta(t,L_x,y,z) \tag{4.4a}$$

$$\boldsymbol{u}(t,x,0,z) = \boldsymbol{u}(t,x,L_y,z), \quad \theta(t,x,0,z) = \theta(t,x,L_y,z) \tag{4.4b}$$

$$\boldsymbol{u}(t,x,y,0) = \boldsymbol{u}(t,x,y,L_z) = \boldsymbol{0}, \quad \partial_z\theta|_{z=0} = \partial_z\theta|_{z=L_z} = 0 \tag{4.4c}$$

Naturally, such problem would approach the trivial solution $\boldsymbol{u} = \boldsymbol{0}$ and $\theta = 1$ everywhere in Ω_f within a time that depends on the boundary and initial conditions. In order to obtain a meaningful solution, one has to (i) impose a mean flow, i.e. a pressure gradient and (ii) prevent saturation phenomena in the fluid phase [69, 93]. We therefore impose the following normalization condition for the dimensionless velocity field in the direction $\boldsymbol{n}_x = (1,0,0)$:

$$⨍_{\Omega_f} \boldsymbol{u}\cdot\boldsymbol{n}_x \mathrm{d}x\mathrm{d}y\mathrm{d}z = ⨍_{\Omega_f} U_s \mathrm{d}x\mathrm{d}y\mathrm{d}z = 1, \quad \forall t \in \mathbb{R}^+ \tag{4.5}$$

Where the operator $⨍$ is a shorthand notation for the integral mean in the integration domain, i.e., the volume integral of a quantity normalized with the volume. Constraint 4.5 can be considered as a *mean flow condition* and the scalar $U_s = \boldsymbol{u}\cdot\boldsymbol{n}_x$ indicates the flow in the *streamwise* direction.

In order to prevent saturation phenomena, we adopt the same strategy we used in our previous work [69]: a heat sink is positioned at $0 < x < \epsilon$ where $\epsilon = d_p/10$ and d_p is the particle diameter. This is equivalent to imposing the following condition on $\Omega_{sink} = \Omega_f \cap [0,\epsilon] \times [0,L_y] \times [0,L_z]$:

$$⨍_{\Omega_{sink}} \theta \mathrm{d}x\mathrm{d}y\mathrm{d}z = 0, \quad \forall t \in \mathbb{R}^+ \tag{4.6}$$

4.2.3. Volume averaging operator

In the field of particulate flows and porous media, *volume averaging* is often employed to derive coarse grained equations that describe the system in terms of integral mean values of the original fields [107]. This approach is also known as the *representative volume method* [49] whenever the governing equations are filtered within a sufficiently large volume and homogenized diffusion coefficients are then obtained.

In PU-EL formulations, the volume of averaging corresponds to the Eulerian grid cell which is generally a hexahedron. Therefore, we define the averaging volume as $\Omega_{\boldsymbol{\varrho}}(\boldsymbol{x}) = \{\boldsymbol{x}' \in \Omega \quad | \quad (\boldsymbol{x} - \boldsymbol{\varrho}/2) \leq \boldsymbol{x}' \leq (\boldsymbol{x} + \boldsymbol{\varrho}/2)\}$, where the inequality is valid for each vector component and where $\boldsymbol{\varrho}$ is the dimensionless filter size. In the following, we will consider cubic averaging volumes and therefore, we will use ϱ to indicate a generic component of $\boldsymbol{\varrho}$. Let's take the scalar field $\theta(t, \boldsymbol{x})$ (the extension to the other fluid fields is trivial). We define the respective continuous phase filtered (i.e., volume averaged) field $\widetilde{\theta}(t, \boldsymbol{x})$ as:

$$\widetilde{\theta}(t, \boldsymbol{x}) = \fint\limits_{\Omega_f \cap \Omega_{\boldsymbol{\varrho}}(\boldsymbol{x})} \theta(t, \boldsymbol{x}')\mathrm{d}x'\mathrm{d}y'\mathrm{d}z' \tag{4.7}$$

Definition 4.7 is however not of practical use since (i) Ω_f is generally a function of time, and since (ii) the information regarding the discrete phase (e.g., the particle configuration) remains implicit inside the integral. Therefore, it is general practice to extend the continuous phase fields to the whole Ω and to define the following *indicator function*:

$$\phi(t, \boldsymbol{x}) = \begin{cases} 0 & \boldsymbol{x} \in \Omega_f \\ 1 & \text{otherwise} \end{cases} \tag{4.8}$$

One advantage of PU-EL formulations with respect to EE formulations, is that the indicator function is known from the solution of the Newton's equation of translational motion for the particle cloud. Therefore, we define a *mean particle volume fraction*:

$$\overline{\phi}(t, \boldsymbol{x}) = \fint_{\Omega_{\boldsymbol{\varrho}}(\boldsymbol{x})} \phi(t, \boldsymbol{x}')\mathrm{d}x'\mathrm{d}y'\mathrm{d}z' \tag{4.9}$$

which can be easily calculated from PU-EL results since particle positions and diameters (which are required to define ϕ) are known.

Finally, after defining the fluid volume fraction $\phi_f = 1 - \phi$ we obtain:

$$\fint\limits_{\Omega_f \cap \Omega_{\boldsymbol{\varrho}}} \theta(t, \boldsymbol{x}')\mathrm{d}x'\mathrm{d}y'\mathrm{d}z' = \overline{\phi_f}(t, \boldsymbol{x})^{-1} \fint_{\Omega_{\boldsymbol{\varrho}}(\boldsymbol{x})} \phi_f(t, \boldsymbol{x}')\theta(t, \boldsymbol{x}')\mathrm{d}x'\mathrm{d}y'\mathrm{d}z' \tag{4.10}$$

Which leads to:

$$\widetilde{\theta}(t, \boldsymbol{x}) = \frac{\overline{\phi_f \theta}}{\overline{\phi_f}} \tag{4.11}$$

This operation is equivalent to the Favre averaging operator often used in compressible turbulence. However, it should be noticed that equation 4.11 has nothing to do with compressibility but instead, arises due to the presence of a dispersed phase.

Furthermore, in our notation we deliberately omitted the dependence on ϱ of the volume averaged fields and in the following we will refer to volume averaged fields for different values of the dimensionless filter size.

Another kind of averaging we employ is the *wall normal layer averaging*. This is defined, for the particle volume concentration ϕ, as:

$$< \phi >_{xy} (z) = \fint_{\Omega_{xy}} \phi (x, y, z) \, \mathrm{d}x\mathrm{d}y \tag{4.12}$$

While, for the other flow variables (e.g.,θ:

$$< \theta >_{xy} (z) = \frac{1}{< \phi_f >_{xy} (z)} \fint_{\Omega_{xy}} \phi_f (x, y, z) \, \theta (t, x, y, z) \, \mathrm{d}x\mathrm{d}y \tag{4.13}$$

In the following, we will also define *bulk* quantities (indicated by subscript b) which satisfy (e.g., in the case of θ):

$$\theta_b = \fint_{z_\ell}^{L_z/2} < \theta >_{xy} (z)\mathrm{d}z \quad \text{where} \quad z_\ell \; : \theta_b \gg \; [< \theta >_{xy} (z) - \theta_b], \quad \forall z \in [z_\ell, L_z/2] \tag{4.14}$$

Therefore, bulk quantities are defined in a region located at a larger distance than z_ℓ from the wall. In this region the maximum deviation of the wall normal layer averaged field from a global mean value is negligible. Such a global mean value is what we call the bulk value. In other words, wall effects in the bulk region can only be of zero order as we describe in Section 4.4.1.

Finally, we define the overall mean particle volume concentration ϕ_b as:

$$\phi_p = \fint_{\Omega} \phi(x', y', z')\mathrm{d}x'\mathrm{d}y'\mathrm{d}z' \tag{4.15}$$

4.2.4. Particle-based interphase transfer coefficients

In PU-EL simulations, closures for interphase transfer coefficients must be provide at a particle level, i.e., each particle exchanges a different quantity of momentum (drag

force) and scalar (scalar transfer). Averaged fields experienced by each particle i are computed by interpolating neighbouring cell values at the particle location $\boldsymbol{x}_i$. To account for such interpolation process (typically, a linear interpolation is employed), we will always consider averaging volumes centred at the particle location, i.e., $\Omega_\varrho(\boldsymbol{x}_i)$.

Generally, the (dimensionless) drag coefficient F_i characterizing fluid-particle momentum exchange associated with particle i is obtained by scaling the drag force with a Stokes-like reference drag force. In case a dimensionless drag force f_i is already available (see Section 4.2.1.1), F_i can be computed from a dimensionless Stokes-like reference drag force via:

$$F_i = \frac{\boldsymbol{f}_i \cdot \boldsymbol{n}_x Re}{3\pi\eta_i[1-\overline{\phi}(t,\boldsymbol{x}_i)]\widetilde{\boldsymbol{u}}(t,\boldsymbol{x}_i)\cdot\boldsymbol{n}_x} \tag{4.16}$$

Where η_i is the dimensionless particle diameter, which is unity for all particles in case of monodisperse systems as considered in our present study. We note in passing that 'drag' refers in our study only to force components in the main flow direction $\boldsymbol{n}_x$. Certainly, flow induced fluid-particle interaction forces arise also in other direction. However, our preliminary work showed that these force components are of subordinate importance on average, and hence are not discussed in greater detail in what follows.

The interphase scalar source term Q_i is generally accounted for via the particle-based Nusselt (or Sherwood) number defined as:

$$Nu_i = \frac{Q_i Pe}{\pi\eta_i[\theta_s - \widetilde{\theta}(t,\boldsymbol{x}_i)]} \tag{4.17}$$

4.3. Numerical formulation

In the present work, we used the finite volume library OpenFOAM®[ope] together with Euler-Lagrange library CFDEMCoupling®[cfd] to solve the governing equations. We ran simulations for several values of ϕ_p and Re as shown in Table 4.1. The number of realizations was such that we considered always more than 1,000 individual particles for each combination of ϕ_p and Re. This is necessary to collect meaningful statistics as a function of the wall-normal distance.

4.3.1. Bed generation

Each particle configuration was generated running a granular-only DEM-based simulation using LIGGGHTS ®[53]. Specifically, the desired global-mean particle volume fraction was realized by shrinking a box containing initially randomly distributed particles. Particles were let free to collide (with other particles, as well as the the wall boundaries

Table 4.1.: Cases analysed in the current study.

ϕ_p	Re	Number of realizations
0.1	100,200,300,400	10
0.2	100,200,300,400	5
0.3	100,200,300,400	4
0.4	100,200,300,400	3

in the z-direction) until the final particle configuration was reached. The final size of the shrunken bounding box was set to $8d_p \times 8d_p \times 8d_p$. The so obtained configurations were subsequently used as the input for the CFD simulation, in which particles were arrested with zero speed.

4.3.2. Mesh generation and CFD solution

We used the OpenFOAM®library together with CFDEMCoupling®to assemble a segregated algorithm that solves the governing equations as described in our previous work [69] where we proposed the Hybrid Fictitious Domain-Immersed Boundary method (HFD-IB). The use of an immersed boundary approach allows to solve the governing equations in a global domain Ω with simple shape (which can be efficiently meshed using a simple Cartesian grid) rather than the highly complex fluid domain Ω_f. Therefore, the HFD-IB approach ensures that no errors arise due to highly skewed cells, as well as reduces the effort to build the mesh. Furthermore, such approach follows the decomposition of volume average integrals presented in Section 4.2.3. We found that a grid of size $h = d_p/20$ did not produce significantly different results from a grid of size $h = d_p/50$ as we used in our previous work [69]. This can be attributed to the scaling technique we employed in our statistical analysis, which removes the dependency on the Reynolds number, and therefore alleviates the necessity of an extremely fine grid. To support this with data, we performed a verification study (see Section 4.4.3). Specifically, we show that the bulk value of the particle based Nusselt number agrees favourable with correlations available in literature.

We adopted a second order accurate discretization in space for all terms, and a backward time integration scheme. A diagonal incomplete-Cholesky preconditioned conjugate gradient method was employed to solve the Poisson equation for the pressure field. The momentum and scalar transport equations were solved using a diagonal incomplete lower-upper preconditioned bi-conjugate gradient method. Conjugate gradient methods were preferred to multigrid methods due to the presence of singular forcing terms in our discretised equations.

Convergence and algorithmic details are identical to our previous work [69].

4.3.3. Post processing

We employed the open source library CPPPO [68] to perform all post processing steps. Specifically, this consisted of (i) volume averaging over wall normal layers, or (ii) over $\Omega_\varrho(\boldsymbol{x}_i)$ for each particle. As described in our previous work on bi-dispersed suspensions [69], averaging the scalar field over periodic boundaries poses a problem due to the field being statistically inhomogeneous in the streamwise direction. Therefore, only particles whose related volume averaging domains $\Omega_\varrho(\boldsymbol{x}_i)$ do not cross the periodic boundaries normal to the streamwise direction are considered when probing the quantities related to scalar transport.

4.4. Results

After applying the numerical schemes summarized in Section 4.3, as well as the volume averaging operations from Section 4.2.3, a large set of raw data was collected. This data included per-particle, as well as layer-averaged flow quantities for multiple realizations of each flow situation. With flow situation we refer to a specific arrangement of particles, as well as the steady-state flow profile that is characterized by the Reynolds number and the domain-average particle volume fraction. The following results have been obtained after ensemble averaging over the respective subset of the raw data.

4.4.1. Wall-normal particle distribution

Figure 4.1 shows the wall normal profile of the particle volume fraction for different values of ϕ_b. In current literature, these profiles are often called *wall radial distribution function* due to the cylindrical geometries employed in their evaluation. In what follows it is useful to consider these profiles as perturbations with respect to the homogeneous (bulk) region. We can clearly see that both (i) the absolute amplitude, and (ii) the wave number of the perturbation are larger for larger values of ϕ_b. This can be related to the higher degree of order that the particles experience near the wall at higher packing fractions.

This wall induced perturbation of the particle volume fraction has been studied in literature for the case of freely sedimenting spheres in cylindrical containers [20, 55, 64, 67, 114]. Several correlations were proposed which are all limited to cylindrical coordinates and high packing fractions. Motivated by the previous considerations regarding the behavior of $< \phi >_{xz}$ (that will be referred to, for the sake of simplicity, simply as $\phi(z)$ in what follows) and the work of Mueller [67], we propose the following functional form:

$$\phi(z) = \phi^{(0)}(z) + \phi^{(1)}(z) + \phi^{(2)}(z) + \phi^{(3)}(z) + \dots \tag{4.18a}$$

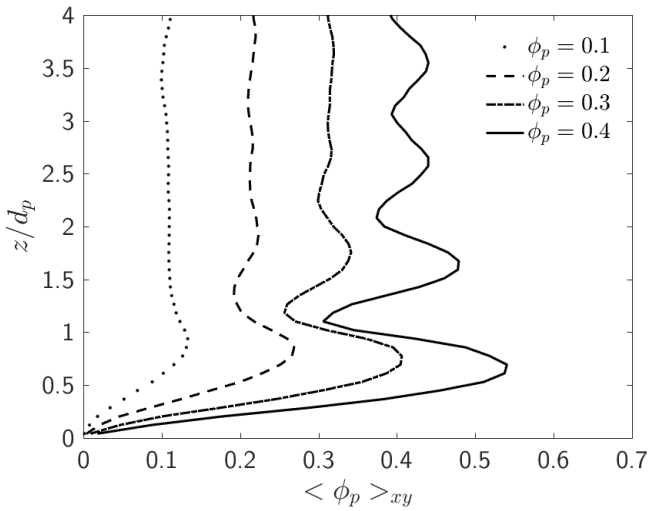


Figure 4.1.: Wall normal profiles of the layer-average ϕ

$$\phi^{(0)}(z) = \phi_b \tag{4.18b}$$

$$\phi^{(n)}(z) = C_n J_{n-1}(\kappa_n z)\exp(\lambda_n z), \quad \forall n = 1, 2, \ldots \tag{4.18c}$$

Where J_n is the n-th Bessel function of the first kind and the constants C_n, κ_n and λ_n are free parameters.

In some sense, equation 4.18a is a perturbation expansion of $\phi(z)$ around the homogeneous value ϕ_b that takes into account the presence of wall induced inhomogeneities. In addition, one shall notice that only J_0 is not-vanishing at $z = 0$. Therefore, $C_1 = -\phi_b$.

Before attempting to evaluate the other free parameters, we have to draw some considerations (and assumptions) about their functional dependence. It is expected, by observing Figure 4.1, that all the parameters are somehow depending on ϕ_b. Specifically:

i C_n represents the amplitude, which scales with ϕ_b same as C_1.

ii κ_n is a wave number, which increases with increasing ϕ_b, and must saturate when approaching the close packing limit. In fact, from $\phi_b = 0.1$ to $\phi = 0.4$ the oscillatory nature of $\phi(z)$ becomes more evident.

iii λ_n is a decay constant of the perturbation and it is therefore expected to be a decreasing function of ϕ_b. This is since the spheres are closer in situations characterized by a higher packing fractions, thus transmitting the perturbation to their neighbours more effectively.

We also assume that the free parameters are independent of z or, in other words, that the arguments of the J_n and $\exp$ functions are linear in z. Finally, we aim to establish if, like for the correlation of Mueller et al., the free parameters are depending upon the vertical domain length L_z.

In order to evaluate the free parameters, we performed a different set of purely DEM-based simulations using LIGGGHTS®. These simulations used a similar configuration as our main study, but with a dimensionless distance $L_z = 12$ between the two walls (therefore, 50% larger than our main study). We probed values of ϕ_b between $\phi_b = 0.05$ and $\phi_b = 0.65$ with increments of $\Delta\phi_b = 0.05$. We first fitted the resulting $\phi(z)$ profiles with equation 4.18a, and subsequently we fitted the obtained parameters with properly chosen functions of ϕ_b. This resulted in the following correlation for the first order perturbed solid volume fraction $\phi^{(1)}(z)$:

$$\phi^{(1)}(z) = -\phi_b J_0(\kappa_1 z)\exp(-\lambda_1 z) \tag{4.19a}$$

$$\kappa_1 = 0.875 + 8.550\sqrt{\phi_b} \tag{4.19b}$$

$$\lambda_1 = 0.461 + 0.409\sin(7.421\phi_b + 0.327) \tag{4.19c}$$

Our formulation based on the perturbation expansion 4.18a allows a to easily generalize our correlation to the case of finite domains where walls are separated by a distance L_z. The average particle concentration on a volume V can be defined as:

$$\phi_p = \frac{1}{V}\sum_{p=1}^{N} m_p = \frac{1}{L_z}\int_0^{L_z} \phi(z)\mathrm{d}z \tag{4.20}$$

Clearly, $\lim_{L_z\to\infty}\phi_b = \phi_p$ must hold. In other words, since ϕ_b can be also defined asymptotically from $\lim_{L_z\to\infty}\phi(z) = \phi_b$, the particle concentration measured in the bulk differs from the (global) average particle concentration.

Substituting the first order perturbation leads to:

$$\phi_p = \frac{1}{L_z}\int_0^{L_z} \left[\phi_b - \phi_b J_0(\kappa_0 z)\exp(\lambda_0 z)\right]\mathrm{d}z \tag{4.21}$$

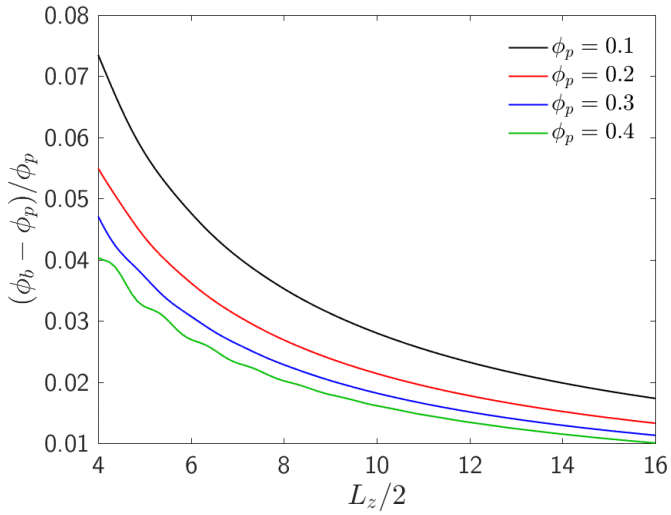


Figure 4.2.: Relative deviation of the bulk particle concentration ϕ_b with respect to the average particle concentration ϕ_p for different values of L_z.

Which leads to the following integral equation for the bulk concentration as a function of L_z:

$$\phi_b(L_z) = \phi_p \left[1 - \frac{1}{L_z} \int_0^{L_z} \frac{1}{L_z} J_0(\kappa_0(\phi_b)z) \exp(\lambda_0(\phi_b)z) \mathrm{d}z \right]^{-1} \tag{4.22}$$

Equation 4.22 is solved iteratively for different values of the domain size ranging from $L_z = 8$ (which is the value used in our main study) to $L_z = 32$. As shown in Figure 4.2, the effect of a finite domain size is relatively small (i.e., a few percent at most) for the values of L_z we considered. Also, the domain size effect decreases to a first approximation exponentially with increasing L_z. Therefore, we do not provide any expression for the bulk particle concentration since in most cases of practical interest the cell size of an EL or EE simulation is sufficiently large (compared to the particle diameter) to accept the approximation $\phi_b \approx \phi_p$.

However, Figure 4.2 provides a measure of the error committed in using finite domains to study wall effects in dense suspensions or, more generally speaking, in granular systems consisting of monodispersed spheres.

Finally, we compared $\phi(z) = \phi_b + \phi^{(1)}(z)$ with results from our main study (see Figure 4.3). We stress that data presented in Figure 4.3 differ from with the data used to determine the free parameters. The two set of simulations are completely separated and performed using a different domain size. Nevertheless, we found excellent agreement between the correlation and the data used for the main study, showing that the domain

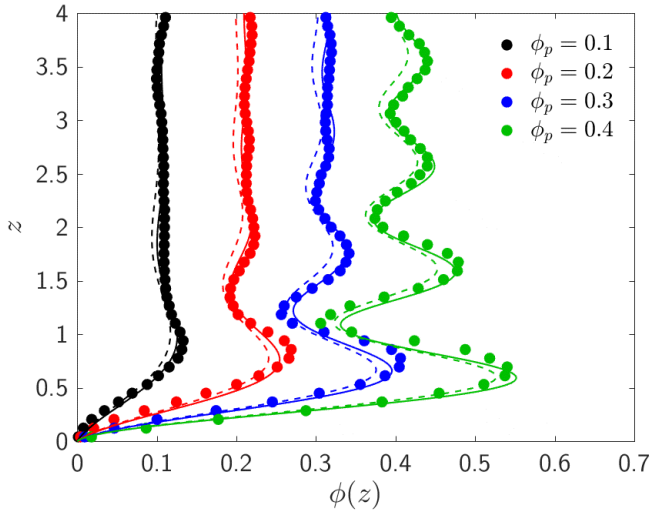


Figure 4.3.: Plot of $\phi(z)$ for four different values of the average particle concentrations. Dots are data from simulations and lines are the first order correlation described by equation 4.19a. Continuous lines are obtained calculating ϕ_b from equation 4.22 while for dashed lines we took $\phi_b = \phi_p$. The average deviation of the correlation from simulations is: $10.5\%(\phi_b = 0.1)$, $9.6\%(\phi_b = 0.2)$, $5.9\%(\phi_b = 0.3)$ and $5.6\%(\phi_b = 0.4)$ when $\phi_b = \phi_p$ (dashed lines). The error decreases when ϕ_p is computed correctly (continuous lines): $7.1\%(\phi_p = 0.1)$, $6.2\%(\phi_p = 0.2)$, $3.9\%(\phi_p = 0.3)$ and $4.2\%(\phi_p = 0.4)$.

size is not a critical parameter for the coefficients (at least, in the relatively large domains we investigated) but it is implicit in the definition of ϕ_b when using equation 4.22. Furthermore, we notice that the error is decreasing with increasing packing fraction. This is in agreement with the previous considerations and in particular with the interpretation of Eqn. 4.18a as a perturbation around the homogeneous value. In fact, the leading term $\phi^{(1)}(z)$ represents the main contribution to the perturbation since it is the one which carries the wall main disturbance (i.e., the one that sets $\phi(z)$ to zero). In other words, additional terms in the expansion can be viewed as contribution due to the number of ways in which particles can be distributed. The higher the order, the more randomness is allowed.

Even if more accurate approximations could be obtained by including higher order contributions, the fitting procedure for the resulting free parameters poses several challenges. Specifically, we found that it becomes increasingly difficult to identify a recognizable functional dependence on ϕ_b. We have therefore suppressed the discussion of these dependencies, and remain with the first to contributions shown on the right hand side of Eqn. 4.18a.

4.4.2. Wall normal velocity and temperature profiles

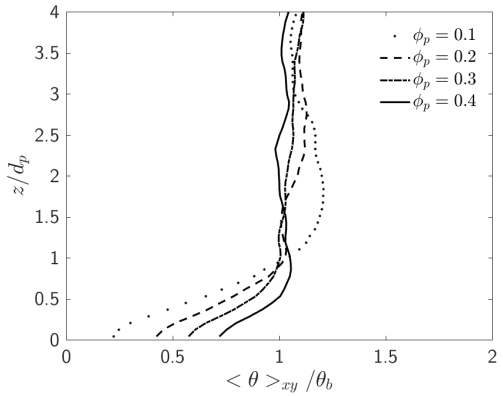
The perturbation to the particle volume fraction shown in Figure 4.1 and described by means of equation 4.18a, gives rise to statistically inhomogeneous velocity and temperature fields as a function of the wall normal distance. Furthermore, considering the profiles shown in Figures 4.4 and 4.5, we can conclude that most of the boundary layer effects are concentrated in the region $z = [0, 1]$, and they are therefore limited to the first particle layer.

This is especially true for the layer averaged temperature field θ shown in Figure 4.4. In addition, we conclude that the effect of the Reynolds number on the velocity profiles is - to a first approximation - negligible with respect to the effect of the bulk particle volume fraction ϕ_b (Figure 4.5). Increasing the fluid speed is therefore equivalent to uniformly scale the velocity field.

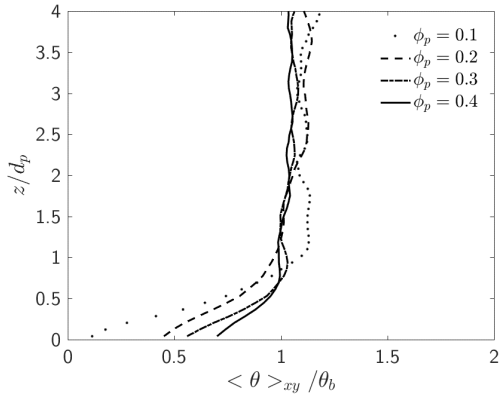
Unlike the velocity field, the temperature field seems to be (weakly) affected by the Reynolds number. However, this can be interpreted in terms of the Peclet number as for low Pe the diffusive nature of heat transport tends to produce more uniform profiles (remember that we are using adiabatic walls). In contrast, for high Pe, the heat is quickly removed by advection in the near wall region, due to the higher fluid speed in this region.

Therefore, it is natural to next attempt defining wall normal temperature and velocity fields $\theta'(z) = \langle\langle\theta\rangle_{xz}\rangle_{Re}$ and $u'(z) = \langle\langle U_s\rangle_{xz}\rangle_{Re}$. This is equivalent to neglecting the effect of the Reynolds number on the wall normal profiles.

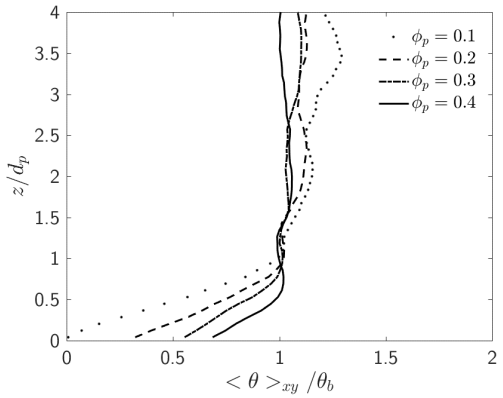
As shown in Figures 4.4 and 4.5, the presence of the discrete phase gives rise to a *particle induced thermal boundary field* $z < z_{\ell,\theta}$ and a *particle induced dynamic*



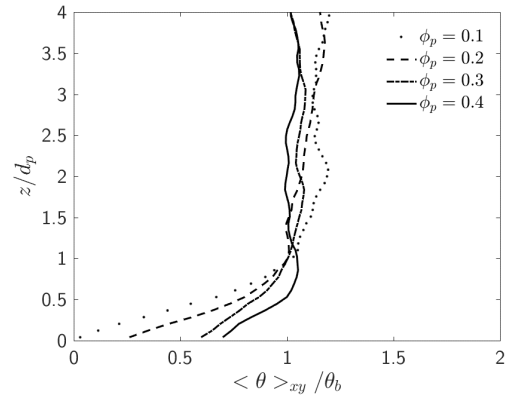
(a) $Re = 100$



(b) $Re = 200$

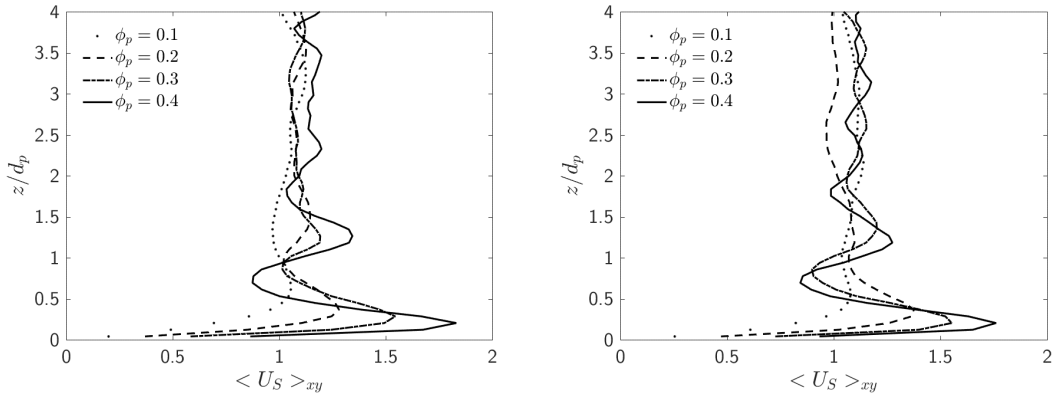


(c) $Re = 300$

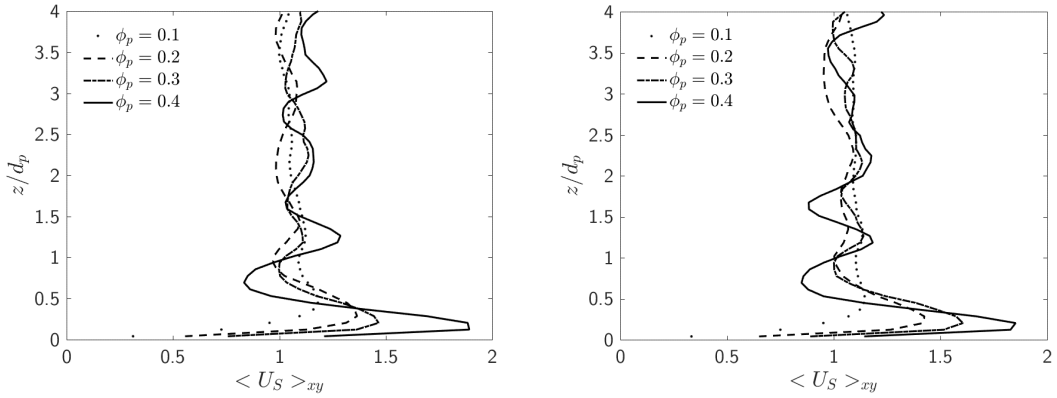


(d) $Re = 400$

Figure 4.4.: Wall normal profiles of θ. $Re = 100$ (4.4a), $Re = 200$ (4.4b), $Re = 300$ (4.4c), $Re = 400$ (4.4d).



(a) $Re = 100$ (b) $Re = 200$



(c) $Re = 300$ (d) $Re = 400$

Figure 4.5.: Wall normal profiles of U_s (streamwise direction). $Re = 100$ (4.5a), $Re = 200$ (4.5b), $Re = 300$ (4.5c), $Re = 400$ (4.5d).

boundary field $z < z_{\ell,u}$ such that to a first approximation one can accept:

$$\theta'(z) = \begin{cases} \theta'_\ell(z) & \text{if } z \leq z_{\ell,\theta} \\ 1 & \text{if } z > z_{\ell,theta} \end{cases} \tag{4.23}$$

$$u'(z) = \begin{cases} u'_\ell(z) & \text{if } z \leq z_{\ell,u} \\ 1 & \text{if } z > z_{\ell,u} \end{cases} \tag{4.24}$$

Where $\theta'_\ell(z)$ and $u'_\ell(z)$ represent the perturbation to the homogeneous bulk field induced by the wall boundaries.

Instead of trying to accurately reconstruct these functions, we next isolate the most physically interesting quantities and use them to build an approximate profile.

4.4.2.1. Wall temperature and wall-normal temperature gradient

It is quite remarkable that despite the fact that we used Neumann boundary conditions at the walls, Figure 4.4 clearly shows the existence of a temperature gradient in the boundary layer. Consequently, particles located within this region experience a mean fluid temperature gradient that superposes the mean fluid-particle temperature difference. The mean temperature gradient is caused by a heat flux pointing from the bulk towards the walls, which is compensated by convective heat transport in the wall-parallel direction.

The gradient in the wall normal direction can be quantified by computing an average gradient in the wall near region:

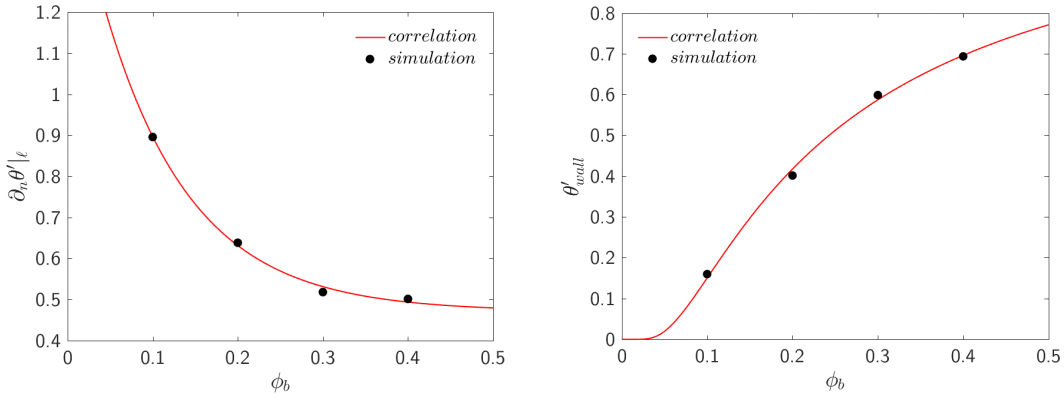
$$\partial_z \theta'|_\ell = z_{\ell,\theta}^{-1} \int_0^{z_{\ell,\theta}} \partial_z \theta' \mathrm{d}z = z_{\ell,\theta}^{-1} \left(\theta'(z_{\ell,\theta}) - \theta'(0)\right) = z_{\ell,\theta}^{-1} \left(1 - \theta'_{wall}\right) \tag{4.25}$$

We propose the following expression for the the wall normal temperature gradient $\partial_z \theta'|_\ell$:

$$\partial_z \theta'|_\ell(\phi_b) = C_1 + C_2 \exp\left(C_3 \phi_b\right) \tag{4.26a}$$

$$C_1 = 0.471, \qquad C_2 = 1.131, \qquad C_3 = -9.763. \tag{4.26b}$$

Figure 4.6a shows that expression 4.26a yields indeed a good approximation to our data.



(a) (b)

Figure 4.6.: Comparison of equations 4.26a (a) and 4.27a (b) with simulation data.

Another quantity of practical interest is the temperature perturbation at the wall $\theta'_{wall} \leq 1$ which is induced by the dispersed phase. Again, we use our data to derive a correlation:

$$\theta'_{wall}(\phi_b) = C_1 \exp\left(\frac{C_2}{\phi_b}\right) \tag{4.27a}$$

$$C_1 = 1.160, \qquad C_2 = -0.204 \tag{4.27b}$$

As shown in Figure 4.6b, the function we used is respecting physically-meaningful limiting conditions (i.e., $\lim_{\phi_b \to 0} \theta'_{wall} = 0$, and $\max(\theta'_{wall}) = 1$, as well as approximates our data convincingly well.

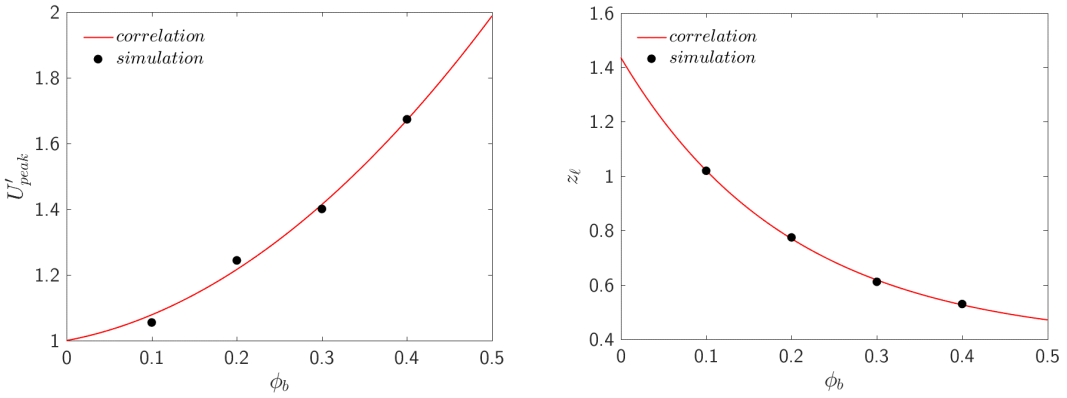
4.4.2.2. Wall shear stress

Regarding the velocity field, one would generally be interested in quantifying the magnitude and height of the peak value u'_{peak} as well as the normalized wall normal shear stress $\tau_{xz,w}$. The latter is proportional to $\partial_z u'|_w$, i.e., the velocity gradient at the wall.

As for the wall temperature and the temperature gradient, we express such quantities as a function of the bulk particle volume fraction ϕ_b:

$$u'_{peak} = 1 + C_1 \phi_b + C_2 \phi_b^2 \tag{4.28a}$$

$$C_1 = 0.486, \qquad C_2 = 2.983 \tag{4.28b}$$



(a) (b)

Figure 4.7.: Comparison of equations 4.28a (a) and 4.29a (b) with simulation data

$$z_{peak} = C_1 + C_2 \exp\left(C_3^{peak} \phi_b\right) \tag{4.29a}$$

$$C_1 = 0.153, \qquad C_2 = 1.151, \qquad C_3 = -11.749 \tag{4.29b}$$

We note in passing that equation 4.28a satisfies the limit $\lim_{\phi_b \to 0} u'_{peak} = 1$, which reflects the obvious fact that no particle induced velocity boundary layer exists in the absence of particles. In addition, we recall that the dimensional wall distance is given by $z^\star = z d_p$. Therefore, no peak will be observed for infinitesimally small particles.

Equations 4.28a and 4.29a can be used to evaluate $\tau_{xz}|_w$. This can be done by representing u' in Taylor series at $z = 0$:

$$u'(z) = \sum_{k=0}^{\infty} \frac{1}{k!} \left.\frac{\partial^k u'}{\partial z^k}\right|_{z=0} z^k \tag{4.30}$$

Our simulations show that the function $u'(z)$ exhibit an almost parabolic behaviour in the interval $z \in [0, z_{peak}]$. Therefore, we seek for a closed expression for $\tau_{xz}|_w$ by retaining only terms $\mathcal{O}(z^2)$ in Eqn. 4.30. Considering the no-slip condition at the wall, we arrive at

$$u'(z) = u'(0) + \partial_z u'|_0 z + \frac{1}{2} \left.\frac{\partial^2 u'}{\partial z^2}\right|_{z=0} z^2 + \mathcal{O}(z^2) = Re^{-1} \tau_{xz}|_w z + \frac{1}{2} \left.\frac{\partial^2 u'}{\partial z^2}\right|_{z=0} z^2 + \mathcal{O}(z^2)$$

(4.31)

This has to be provided with the condition that z_{peak} is a critical point, and should satisfy a concavity condition:

$$\left.\frac{\partial u'}{\partial z}\right|_{z=z_{peak}} = 0, \qquad \partial_{zz}u' < 0 \quad \forall z \in [0, z_{peak}] \tag{4.32}$$

Finally, we obtain the following equation for $u'(z)$:

$$u'(z') = u'_{peak} z' \left(2 - z'\right), \qquad z' = z/z_{peak} \tag{4.33}$$

This ultimately leads to:

$$\tau_{xz}|_w = \frac{2}{Re}\frac{u'_{peak}}{z_{peak}} = \frac{2}{Re}\frac{u'_{peak}}{z_{peak}} = \frac{2}{Re}\frac{1 + 0.486\phi_b + 2.983\phi_b^2}{0.153 + 1.151\exp\left(-11.749\phi_b\right)} \tag{4.34}$$

Equation 4.34 allows to evaluate the (normalized) wall shear stress induced by the fluid-particle system as a function of the bulk particle concentration and the Reynolds number only.

4.4.3. Benchmarks for the Nusselt number prediction

Since we established that wall induced perturbations are mostly relevant within $1d_p$ from the walls, it makes sense to compare the Nusselt number experienced by bulk particles (i.e., particle located beyond $1d_p$ from a wall) with existing correlations. In particular, we compare our result with the correlation from Deen et al. [24]:

$$\begin{aligned} Nu_{Deen,i} = &\left(7 - 10\phi_f + 5\phi_f^2\right)\left(1 + 0.17Re_i^{0.2}Pr^{1/3}\right) + \\ &\left(1.33 - 2.31\phi_f + 1.16\phi_f^2\right)Re_i^{0.7}Pr^{1/3} \end{aligned} \tag{4.35}$$

Notice that we used $\phi_f = 1 - \overline{\phi}(\boldsymbol{x}_i)$ and $Re_i = Re\overline{\phi}(\boldsymbol{x}_i)\widetilde{U_s}(\boldsymbol{x}_i)$ in the above expression. As already pointed out in our previous work [69], expression 4.35 is not consistent with PU-EL since the cup-mixing temperature was used. This temperature is different from the temperature available in a PU-EL simulation. A correction was proposed by Sun et al. [87] that modifies the above correlation to be consistent with PU-EL:

$$\begin{aligned} Nu_{DS,i} = Nu_{Deen,i}\,[1 - &1.6\overline{\phi}(\boldsymbol{x}_i)(1 - \overline{\phi}(\boldsymbol{x}_i)) - \\ &3\overline{\phi}(\boldsymbol{x}_i)(1 - \overline{\phi}(\boldsymbol{x}_i))^4\exp(-Re_i^{0.4}\overline{\phi}(\boldsymbol{x}_i))\big]^{-1} \end{aligned} \tag{4.36}$$

In Figure 4.8, we plot the calculated particle-based Nusselt number over the prediction from equation 4.36. The comparison shows that our methodology is fully capable of reproducing the results of Deen et al. [24] and Sun et al. [87], which are given in terms of an average, and not particle-based, Nusselt number. Discrepancies are observed for (i) high values of Nu (which are, however, outside the range of validity of correlation denoted as Nu_{DS}), and (ii) for $\rho = 2$ (see Figure 4.8a). The latter disagreement can be explained by noticing that, as addressed in our previous work [69], for such filter size local inhomogeneities in the particle bed become relevant. Overall, the agreement can be considered satisfactory and details about average deviation are summarized in Table 4.2.

Table 4.2.: Average relative error and standard deviation of the particle based Nusselt numbers from values predicted by equation 4.36. $Nu_{\mathrm{DS},i}$ is used as a short hand for $Nu_{\mathrm{DS}}(\overline{\phi}(\boldsymbol{x})_i, Re_i)$.

ϱ	$\left\langle \frac{Nu_i}{Nu_{\mathrm{DS},i}} \right\rangle$	$\sigma\left(\frac{Nu_i}{Nu_{\mathrm{DS},i}}\right)$
2	1.37	0.87
3	0.99	0.34
4	0.98	0.36
5	0.95	0.36

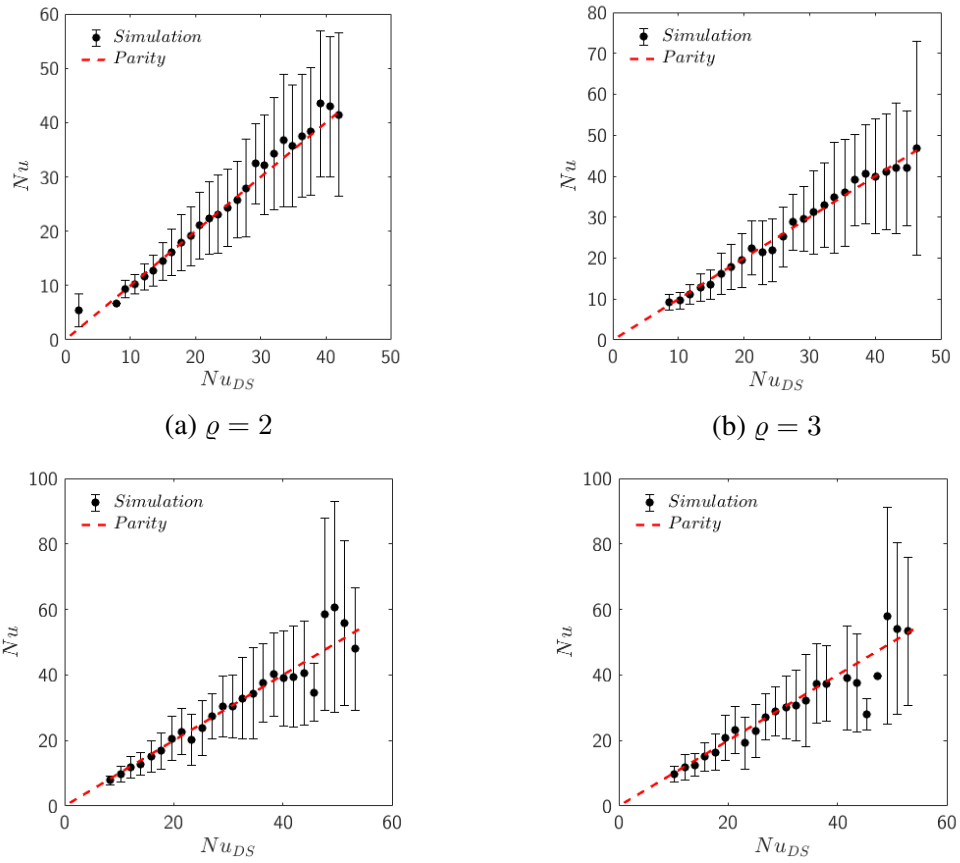
Unlike the approach used in the derivation of Nu_{DS}, our approach allows to evaluate the dispersion of the particle based quantities around the average value (note, only the latter is provided by the correlation). As shown in Table 4.2, the deviation is very high for small filter sizes but seems to approach a constant value of $\sigma\left(Nu_i/Nu_{\mathrm{DS},i}\right) \approx 0.36$ as the filter size is increased. Thus, fluctuations around the mean are substantial, and should be potentailly considered in closures for PU-EL models.

4.4.4. Wall corrections to the drag coefficient and Nusselt number

In order to quantify the effect of wall boundaries on the drag coefficient and Nusselt number, we collected the statistics using the binning approach that is described in one of our previous studies [69].

As a result, we observed that (as for the wall normal velocity and temperature profiles) the dependence on the Reynolds number can be easily dropped using the following scaling:

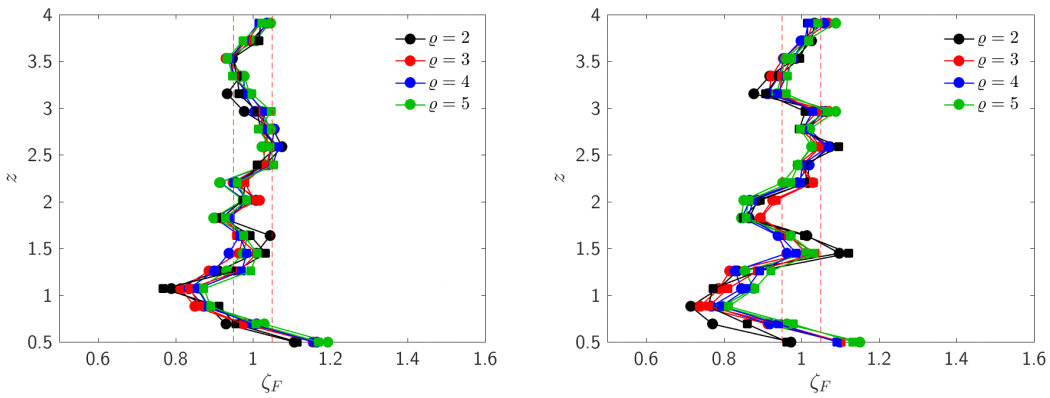
$$F(z, \phi_b, Re) = \zeta_F(z, \phi_b) F_b(z, \phi_b, Re) \tag{4.37}$$



(a) $\varrho = 2$ (b) $\varrho = 3$

(c) $\varrho = 4$ (d) $\varrho = 5$

Figure 4.8.: Parity plot showing the binned particle based Nusselt number compared with the predicion of Deen et al. and Sun et al. Error bars extend to two standard deviations. Different panels refer to different values of the dimensionless filter size ϱ.



(a) $\phi_b = 0.3$ (b) $\phi_b = 0.4$

Figure 4.9.: Comparison between results for $\zeta_F(z, \phi_b)$ from the this work (disks) and our previous work with $Re = 100$ (squares). Red dashed lines represent deviations of $\pm 5\%$ from unity (i.e., the bulk value). The same comparison carried out for $\zeta_{Nu}(z, \phi_b)$ and for different values of ϕ_b leads to a similar agreement between the two studies and is included in 3.5.

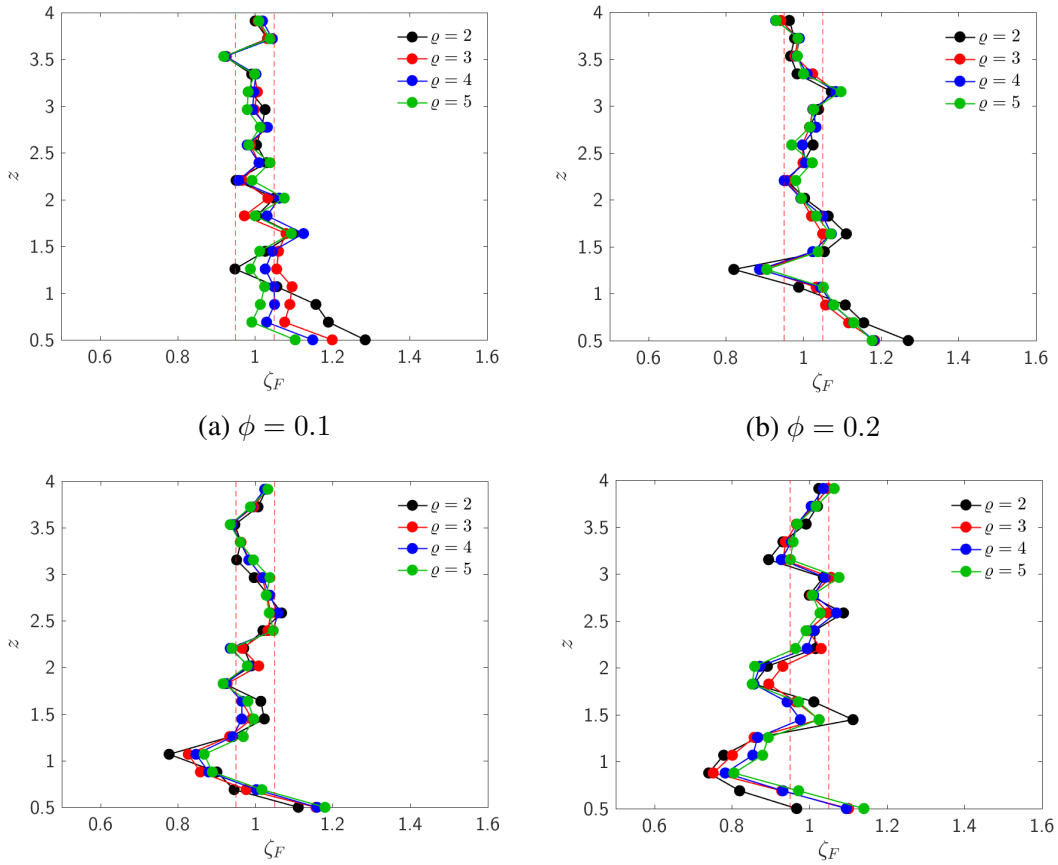
Where $F_b(z, \phi_b, Re)$ is the bulk drag force which is calculated using the approach shown in Section 4.2.3. Therefore, such bulk value represents the homogenous drag calculated using ϕ_b rather than ϕ_p.

The fact that $\zeta_F(z, \phi_b)$ is not a function of the Reynolds number comes from our data analysis and is in agreement with the previous study of the near wall behaviour of the velocity profiles. Therefore, in order to evaluate $\zeta_F(z, \phi_b)$, we consider particles from simulations with different Reynolds numbers in the bins having the same wall normal distance. This strategy allows us to collect more robust statistics on mean exchange coefficients.

In order to further prove that the decomposition expressed in equation 4.37 is valid, we compared results obtained from the current set of simulations with results from one of our previous studies [70]. This latter work featured different particle configurations, but a fixed Reynolds number of 100. The total number of particle is almost the same for both studies. Figure 4.9 shows that decomposition 4.37 leads to an excellent agreement between the two studies, therefore supporting our argument that the Reynolds number does not play an essential role when determining wall profiles.

The data we present in this section are therefore a combination of data from both studies, so that for each value of ϕ_b we use data from simulations involving approximately 4,000 particles.

Figure 4.10 shows results for $\zeta_F(z, \phi_b)$. All wall normal profiles have a maximum corresponding to the first particle layer ($z = 0.5$) and a minimum develops with in-



(a) $\phi = 0.1$ (b) $\phi = 0.2$

(c) $\phi = 0.3$ (d) $\phi = 0.4$

Figure 4.10.: $\zeta_F(z, \phi_b)$ as a function of the filter size ϱ and the particle concentration ϕ_b. Red dashed lines represent deviations of $\pm 5\%$ from unity (corresponding to the bulk value).

creasing ϕ_b. It can also be seen that the intensity of the perturbations becomes larger with larger ϕ_b, which is probably related to the stronger perturbation in the particle concentration. The effect of the filter size ρ is not existing, hinting to the fact that filtered fluid velocities are almost independent of ρ.

Values of $\zeta_F(z, \phi_b)$ are tabulated in tables 4.3 and 4.4. The latter are meant to support the implementation of wall corrections to be used in PU-EL-based simulation models.

We then extend the same approach to the study of Nusselt number profiles by defining:

$$Nu(z, \phi_b, Re) = \zeta_{Nu}(z, \phi_b) Nu_b(z, \phi_b, Re) \tag{4.38}$$

Results for $\zeta_{Nu}(z, \phi_b)$ are plotted in Figure 4.11 and tabulated in Tables 4.5 and 4.6

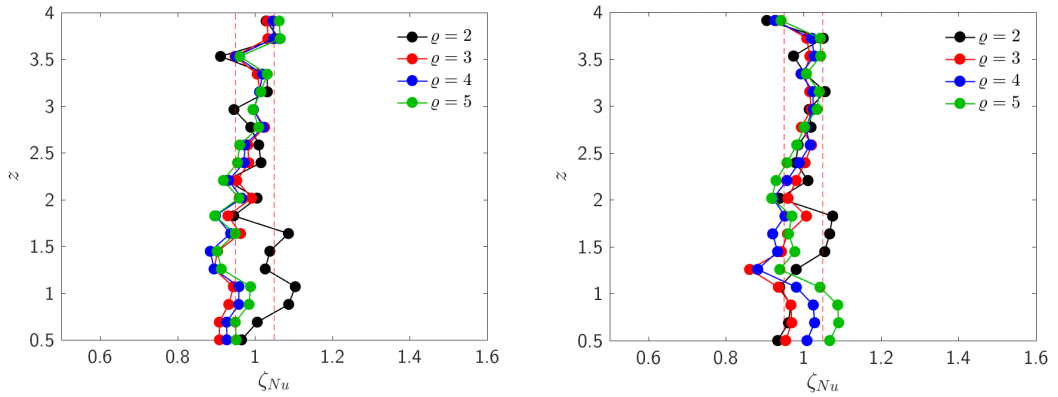
in the Appendix. The results indicate that corrections to the Nusselt number are in general strong as compared to that for the drag. This can be tentatively explained by the combined effect of flow (i.e., the velocity gradient) and the abnormal temperature distribution experienced by wall-near particles.

Futhermore, the filter size ρ has a significant effect on the Nusselt number correction. This effect is extreme for the wall-closest particles, i.e., that located at $z = 0.5$. The enormous sensitivity to ρ observed for the corrections to the Nusselt number is in contrast to the correction for the drag coefficient. The latter are almost insensitive to the filter size. We speculate that the origin of this difference is the abnormal temperature distribution close to the wall: Depending on the filter size, a different *filtered* temperature is experienced by individual particles. Specifically, smaller filter sizes ρ will lead to lower values for $< \theta >_{xy}$ as seen in Figure 4.4. This will lead to a larger difference $\theta_p - < \theta >_{xy}$. In simple words, a smaller filter size (or a smaller grid size in PU-EL models) results in wall-near particles experiencing a different (in the context of our study a cooler) fluid temperature. Thus, for a given fluid-particle heat flux, the Nusselt number must decrease for a decrease in the filter size in case particles are located in the 'temperature boundary layer' observed in Figure 4.4. In Figure 4.11 we observe exactly this trend for the correction of the Nusselt number and for $\phi > 0.1$. In case particles are outside of the 'temperature boundary layer', the above argument on the filter size effect is expected to break down. Also this fact is observable in Figure 4.11 when considered the data for $z > 1.2$. Finally, one should mention that for dilute systems (exemplify by $\phi > 0.1$ in our present study) the 'temperature boundary layer' is only weakly pronounced. This is not directly observable from Figure 4.4, however, a fact since in Figure 4.4 the temperature profile is scaled with the bulk value. The latter is strongly decreasing for decreasing particle concentrations. Hence, we do not observe noticeable corrections to the Nusselt number for $\phi = 0.1$ in Figure 4.11.

4.5. Summary

We studied momentum, heat and mass transfer in dense gas-particle suspensions by means of Particle-Resolved Direct Numerical Simulation (PR-DNS) in wall bounded domains.

We found that the presence of walls induces a particle volume concentration field that is inhomogeneous in the wall-normal direction. We expressed this rather well-known obervation as a series of perturbations with respect to an homogeneous bulk value. We employed particle simulations to obtain a correlation for the first terms of our perturbation expansion. This now allows to accurately predict wall normal profiles of the particle volume concentration without performing expensive DEM-based simulations. Our correlation is valid in the range $0.05 < \overline{\phi} < 0.65$, and provides new insight on wall effects in dilute suspensions.



(a) $\phi = 0.1$ (b) $\phi = 0.2$

(c) $\phi = 0.3$ (d) $\phi = 0.4$

Figure 4.11.: $\zeta_{Nu}(z, \phi_b)$ as a function of the filter size ϱ and the particle concentration ϕ_b. Red dashed lines represent deviations of $\pm 5\%$ from unity (corresponding to the bulk value).

Next, we demonstrated taht perturbations of the particle concentration fields have a significant effect on momentum, heat and mass transfer in the vicinity of walls. This is best illustrated by the peculiar wall normal profiles for the flow variables (i.e., velocity and temperature) documented in our present study. Most remarkably, we found that such profiles do not depend significantly on the Reynolds number when scaled with respect to their bulk value, i.e. the Reynolds number does not affect the shape of the profiles.

Such profiles allow to define a particle induced 'boundary layer' in which the fields differ from the bulk (homogeneous) value. In the present work, we proposed correlations to capture the most relevant physical phenomena happening in this particle induced boundary layer. Specifically, we obtained expressions for the scalar value at the wall, as well as the scalar gradient in the layer. Surprisingly, the latter is non zero despite we used adiabatic walls. Furthermore we provide expressions for (i) the maximum value of the velocity field in the layer, and (ii) the wall normal coordinate z_{max} at which this maximum occurs. These two quantities together with the observation that the velocity field is almost parabolic in $[0, z_{max}]$ allows us to predict the wall shear stress exerted by the gas-particle system.

Finally we studied the interphase transfer coefficients for momentum (drag coefficient) and heat or mass transfer (i.e., the Nusselt or Sherwood number). Also in this case, we were able to obtain scaled wall-normal profiles that are independent on the Reynolds number. Unfortunately, we were not able to obtain a simple correlation for the wall correction functions of drag and heat or mass transfer. However, we tabulated values instead, which provide the basis for the correction of the Nusselt (or Sherwood) number, as well as the drag coefficient of particles situated at $0.5 < z < 2.5$ from the bounding wall. Most remarkably, for the Nusselt number an additional complexity arises, since the corrections are sensitive to the filter size ρ. This is due to the strong temperature gradient near the wall, which - in contrast to the strongly fluctuating mean velocity - causes a dependency of the filtered fluid temperature on the filter size.

4.6. Additional data

4.6.1. Tabulated values for Drag and Nusselt wall corrections

Table 4.3.: Values of $\zeta_F(z, \phi_b)$ shown in figure 4.10a and figure 4.10b.

z	$\phi_b = 0.1$				$\phi_b = 0.2$			
	$\varrho = 2$	$\varrho = 3$	$\varrho = 4$	$\varrho = 5$	$\varrho = 2$	$\varrho = 3$	$\varrho = 4$	$\varrho = 5$
0.500	1.270	1.184	1.181	1.177	1.285	1.200	1.150	1.104
0.689	1.155	1.117	1.129	1.130	1.190	1.077	1.030	0.992
0.879	1.108	1.057	1.076	1.078	1.158	1.090	1.051	1.014
1.068	0.988	1.033	1.044	1.051	1.057	1.096	1.050	1.025
1.258	0.820	0.891	0.887	0.905	0.949	1.056	1.027	0.989
1.447	1.054	1.038	1.024	1.038	1.027	1.061	1.046	1.012
1.637	1.111	1.050	1.072	1.070	1.098	1.080	1.126	1.094
1.826	1.064	1.021	1.050	1.033	1.004	0.973	1.032	1.000
2.016	1.003	0.993	0.993	0.995	1.051	1.033	1.063	1.076
2.205	0.958	0.956	0.950	0.980	0.952	0.967	0.958	0.993
2.395	1.000	0.999	1.005	1.023	1.030	1.011	1.011	1.039
2.584	1.025	0.997	0.998	0.970	1.004	0.988	0.980	0.984
2.774	1.016	1.016	1.032	1.016	1.018	1.030	1.032	1.014
2.963	1.039	1.024	1.025	1.026	1.027	0.999	0.994	0.980
3.153	1.072	1.084	1.083	1.097	0.989	1.007	0.997	0.983
3.342	0.983	1.024	1.009	1.000	0.992	0.999	1.003	0.999
3.532	0.967	0.978	0.982	0.984	0.922	0.927	0.924	0.920
3.721	0.977	0.986	0.989	0.986	1.033	1.034	1.045	1.039
3.911	0.963	0.942	0.928	0.930	1.001	1.008	1.020	1.011

Table 4.4.: Values of $\zeta_F(z, \phi_b)$ shown in figure 4.10c and figure 4.10d.

z	$\phi_b = 0.3$				$\phi_b = 0.4$			
	$\varrho = 2$	$\varrho = 3$	$\varrho = 4$	$\varrho = 5$	$\varrho = 2$	$\varrho = 3$	$\varrho = 4$	$\varrho = 5$
0.500	1.111	1.157	1.159	1.180	0.967	1.101	1.095	1.140
0.689	0.945	0.976	1.003	1.017	0.820	0.928	0.931	0.972
0.879	0.901	0.858	0.880	0.889	0.740	0.752	0.783	0.805
1.068	0.777	0.827	0.847	0.868	0.779	0.801	0.854	0.879
1.258	0.937	0.934	0.942	0.969	0.867	0.857	0.866	0.894
1.447	1.023	0.991	0.965	0.996	1.112	1.025	0.977	1.025
1.637	1.014	0.965	0.966	0.981	1.011	0.966	0.942	0.972
1.826	0.920	0.927	0.924	0.917	0.857	0.895	0.856	0.852
2.016	0.990	1.009	0.983	0.979	0.891	0.931	0.872	0.859
2.205	0.970	0.965	0.935	0.940	1.016	1.030	0.994	0.964
2.395	1.020	1.033	1.046	1.046	1.012	0.996	1.010	0.991
2.584	1.068	1.040	1.059	1.036	1.088	1.048	1.069	1.027
2.774	1.034	1.031	1.038	1.028	1.000	1.005	1.011	1.008
2.963	0.997	1.017	1.020	1.038	1.036	1.055	1.040	1.076
3.153	0.952	0.985	0.983	0.995	0.895	0.933	0.927	0.951
3.342	0.963	0.963	0.962	0.961	0.931	0.938	0.953	0.958
3.532	0.947	0.937	0.940	0.935	0.991	0.966	0.968	0.968
3.721	1.007	0.995	0.989	0.987	1.021	1.011	1.004	1.017
3.911	1.029	1.029	1.024	1.032	1.024	1.047	1.036	1.064

Table 4.5.: Values of $\zeta_{Nu}(z, \phi_b)$ shown in figure 4.11a and figure 4.11b.

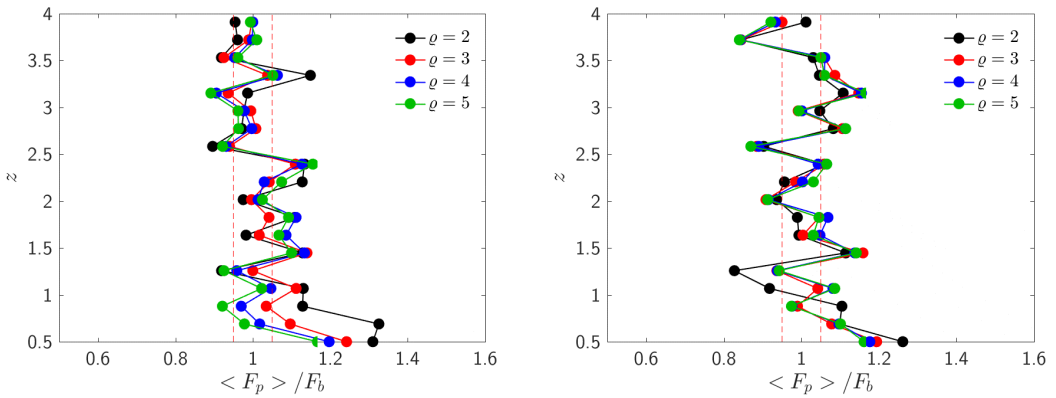
z	$\phi_b = 0.1$				$\phi_b = 0.2$			
	$\varrho = 2$	$\varrho = 3$	$\varrho = 4$	$\varrho = 5$	$\varrho = 2$	$\varrho = 3$	$\varrho = 4$	$\varrho = 5$
0.500	0.966	0.908	0.928	0.952	0.934	0.954	1.009	1.068
0.689	1.006	0.908	0.927	0.950	0.961	0.970	1.029	1.091
0.879	1.087	0.933	0.958	0.985	0.968	0.967	1.025	1.088
1.068	1.104	0.944	0.959	0.989	0.939	0.935	0.982	1.043
1.258	1.027	0.894	0.895	0.913	0.981	0.861	0.882	0.939
1.447	1.039	0.904	0.885	0.903	1.055	0.943	0.933	0.978
1.637	1.087	0.963	0.937	0.950	1.068	0.959	0.921	0.961
1.826	0.945	0.930	0.898	0.896	1.075	1.008	0.952	0.970
2.016	1.006	0.992	0.966	0.959	0.938	0.960	0.922	0.918
2.205	0.927	0.953	0.932	0.919	1.012	0.981	0.957	0.929
2.395	1.016	0.985	0.972	0.955	0.980	1.004	0.989	0.957
2.584	1.010	0.982	0.974	0.961	0.986	1.021	1.017	0.983
2.774	0.989	1.025	1.022	1.010	1.019	0.995	1.006	1.002
2.963	0.946	0.996	0.996	0.995	1.015	1.020	1.025	1.035
3.153	1.032	1.014	1.012	1.015	1.056	1.016	1.025	1.041
3.342	1.032	1.006	1.020	1.032	1.004	1.002	0.994	1.008
3.532	0.911	0.949	0.951	0.961	0.974	1.017	1.028	1.045
3.721	1.063	1.033	1.048	1.066	1.051	1.008	1.021	1.044
3.911	1.029	1.032	1.047	1.063	0.905	0.930	0.926	0.942

Table 4.6.: Values of $\zeta_{Nu}(z, \phi_b)$ shown in figure 4.11c and figure 4.11d.

z	$\phi_b = 0.3$				$\phi_b = 0.4$			
	$\varrho = 2$	$\varrho = 3$	$\varrho = 4$	$\varrho = 5$	$\varrho = 2$	$\varrho = 3$	$\varrho = 4$	$\varrho = 5$
0.500	0.820	0.974	1.046	1.127	0.738	1.035	1.169	1.350
0.689	0.794	0.965	1.050	1.141	0.706	0.944	1.074	1.232
0.879	0.744	0.847	0.917	0.993	0.589	0.668	0.752	0.869
1.068	0.794	0.858	0.911	0.995	0.651	0.617	0.692	0.799
1.258	0.894	0.893	0.929	1.007	0.715	0.656	0.701	0.797
1.447	0.849	0.846	0.828	0.896	0.745	0.745	0.719	0.820
1.637	0.910	0.930	0.868	0.920	0.863	0.869	0.744	0.831
1.826	1.016	0.962	0.869	0.894	0.935	0.943	0.759	0.821
2.016	0.938	0.937	0.865	0.854	0.943	0.950	0.790	0.782
2.205	0.990	1.003	0.954	0.900	0.984	0.992	0.933	0.822
2.395	1.012	1.037	1.034	0.972	1.066	1.040	1.032	0.937
2.584	1.060	1.018	1.025	1.001	1.026	0.975	0.978	0.912
2.774	0.997	0.972	0.978	0.978	1.082	1.068	1.075	1.078
2.963	0.998	1.006	1.017	1.040	0.980	1.087	1.096	1.157
3.153	0.990	1.013	1.030	1.053	0.938	0.946	0.945	0.982
3.342	1.004	0.994	1.001	1.021	0.992	0.959	0.975	1.019
3.532	0.971	0.939	0.938	0.966	0.932	0.937	0.945	0.978
3.721	0.937	0.959	0.958	0.976	0.986	1.000	1.001	1.027
3.911	1.030	1.054	1.060	1.088	1.028	1.037	1.052	1.119

4.6.2. Drag and Nusselt profiles for different Reynolds numbers

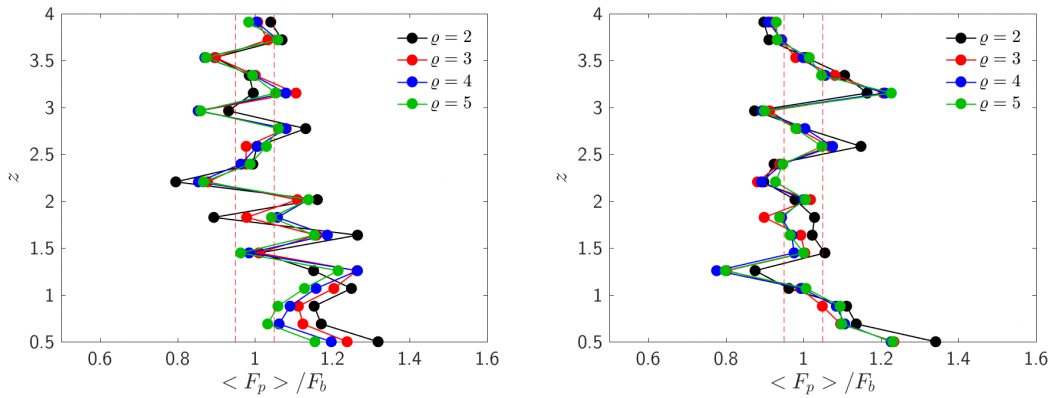
In this section we show the scaled profiles for drag and Nusselt number at different Reynolds numbers. The peculiar characteristics are visible for different values of ϕ_p rather than for different Reynolds numbers.



(a) $\phi = 0.1$ (b) $\phi = 0.2$

(c) $\phi = 0.3$ (d) $\phi = 0.4$

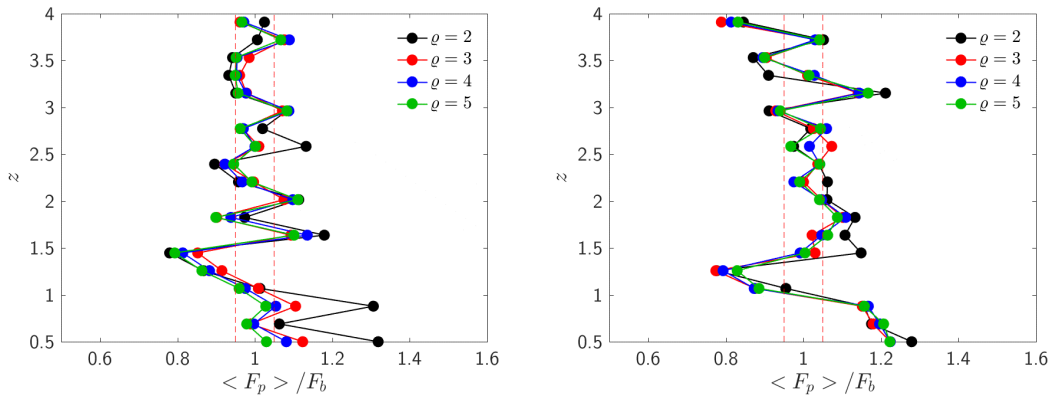
Figure 4.12.: Scaled drag for $Re = 100$.



(a) $\phi = 0.1$ (b) $\phi = 0.2$

(c) $\phi = 0.3$ (d) $\phi = 0.4$

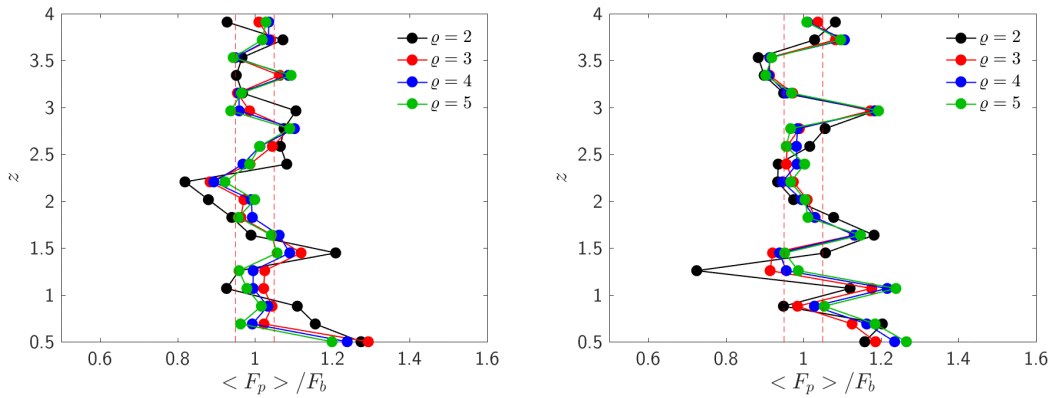
Figure 4.13.: Scaled drag for $Re = 200$.



(a) $\phi = 0.1$ (b) $\phi = 0.2$

(c) $\phi = 0.3$ (d) $\phi = 0.4$

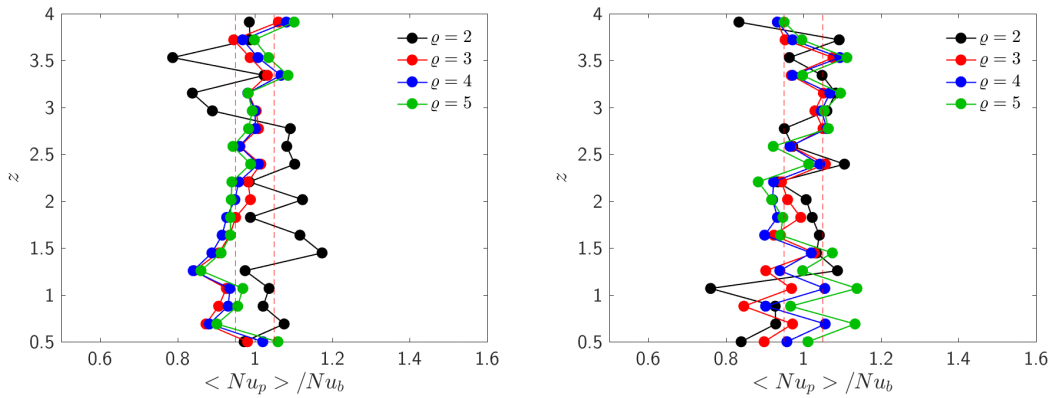
Figure 4.14.: Scaled drag for $Re = 300$.



(a) $\phi = 0.1$ (b) $\phi = 0.2$

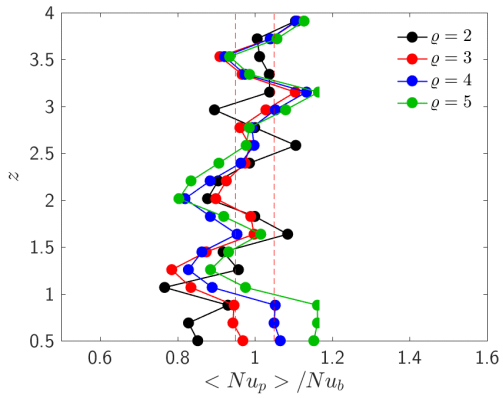
(c) $\phi = 0.3$ (d) $\phi = 0.4$

Figure 4.15.: Scaled drag for $Re = 400$.

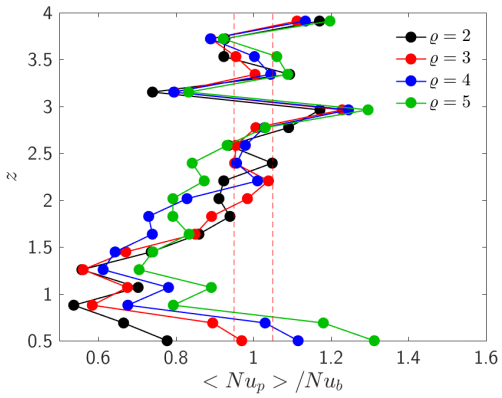


(a) $\phi = 0.1$

(b) $\phi = 0.2$

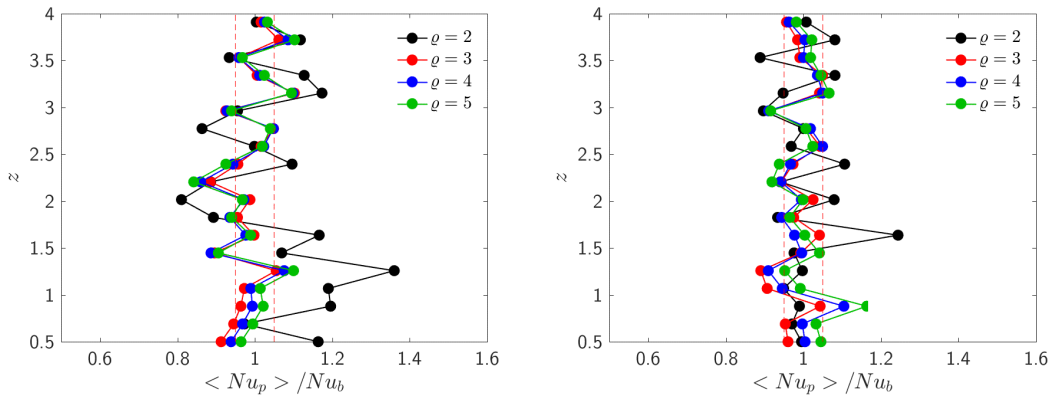


(c) $\phi = 0.3$



(d) $\phi = 0.4$

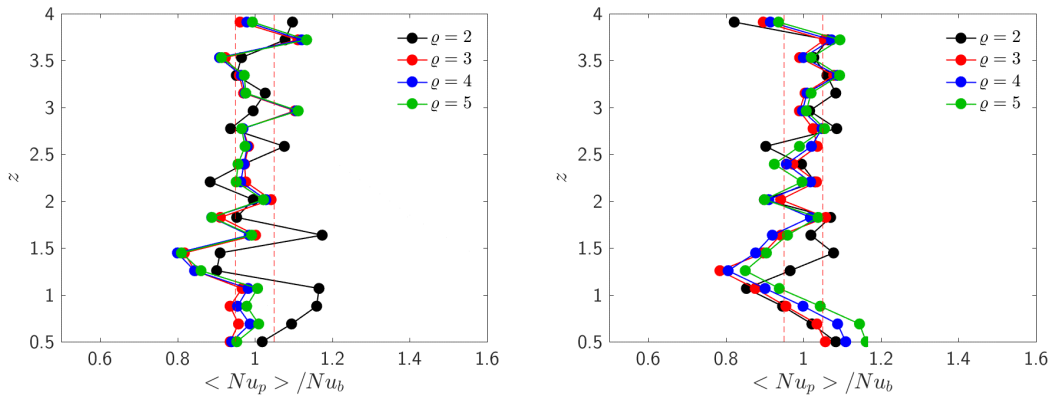
Figure 4.16.: Scaled Nusslet for $Re = 100$.



(a) $\phi = 0.1$ (b) $\phi = 0.2$

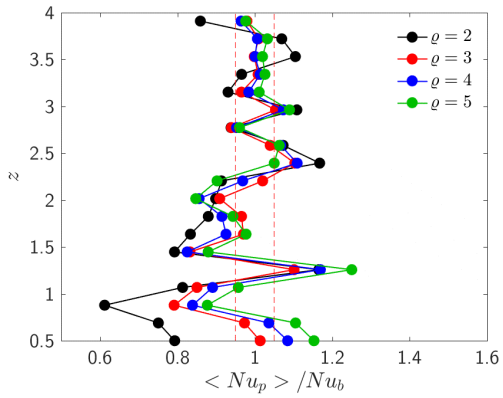
(c) $\phi = 0.3$ (d) $\phi = 0.4$

Figure 4.17.: Scaled Nusslet for $Re = 200$.

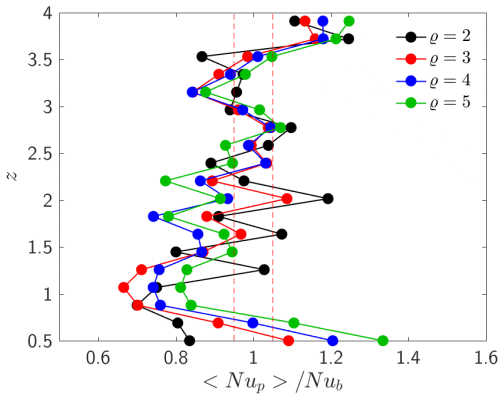


(a) $\phi = 0.1$

(b) $\phi = 0.2$

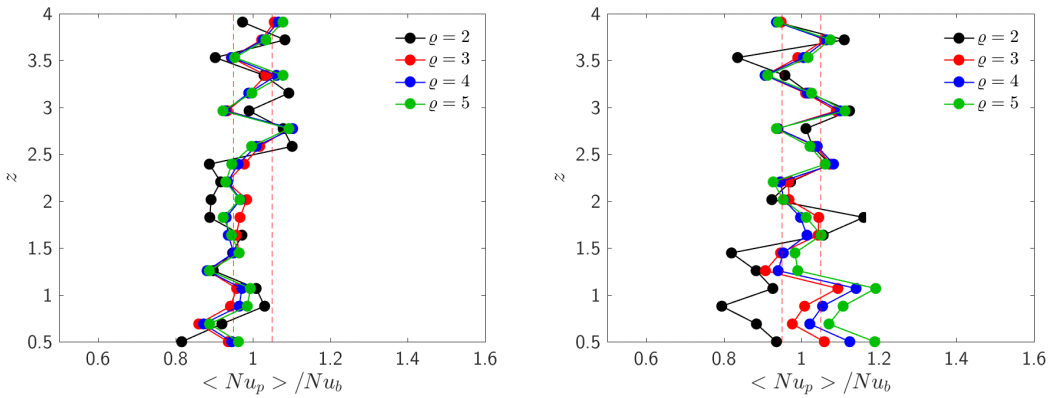


(c) $\phi = 0.3$



(d) $\phi = 0.4$

Figure 4.18.: Scaled Nusslet for $Re = 300$.



(a) $\phi = 0.1$ (b) $\phi = 0.2$

(c) $\phi = 0.3$ (d) $\phi = 0.4$

Figure 4.19.: Scaled Nusslet for $Re = 400$.

5. Conclusion and outlook

5.1. Conclusion

The individual studies which compose the present thesis contain several major findings and contributions to the actual state of the art in the field of gas-particle suspension modelling.

In Chapter 2 we developed a novel library for parallel data filtering and processing which can be link to a variety of simulators. Such library has been proven to be of major importance for the subsequent studies conducted in the frame of this thesis and features some novel algorithms and a unique code structure that allows high parallel efficiency and low RAM consumption.

A second essential tool for Particle Resolved Direct Numerical Simulation (PR-DNS) was developed in Chapter 3. The HFD-IB method combines the Immersed Boundary approach [73] which aims on imposing the boundary conditions at the immersed surfaces, with the Fictitious Domain approach [86], where a rigidity condition is imposed within the region occupied by the immersed bodies. The algorithm was shown to be capable of accurately imposing Dirichlet boundary conditions and compute interphase transfer coefficients on relatively coarse grids compared to existing algorithms. An extension to general boundary conditions is provided in Appendix A.

In Chapter 3 we also introduced the concept of saturation as a consequence of the fast interphase heat and mass transfer which results in strong variation of a non-reactive scalar field defined in the fluid phase. In PU-EL simulations, saturation would give rise to large interphase transfer coefficients and the steep scalar gradient in the microscopic field would not be captured in the PU-EL simulation. Therefore, we provided an accurate analytical model based on the pseudo-turbulent dispersion models proposed by Delgado [25] and the Nusselt number correlation from Deen et al. [23] that can predict the occurrence of a saturated regime. Wherever such situation occurs, models based on the heat transfer coefficient are not viable. To a first approximation, one could just enforce the (fluid's) cell scalar value to equal the particle surface value.

Again, in Chapter 3 we perform a large array of PR-DNS of bi-disperse gas-solid suspensions at moderate Reynolds number. We demonstrated - using results from our simulations together with theoretical analysis - that the widely used correlation of Beetstra et al. [11] fails in separating a meaningful drag force from the total fluid-particle interaction force. Therefore, this correlation leads to significant underestimation of the

interphase momentum exchange between phases in bi- and poly-disperse suspensions in certain situations. We proposed a simple modifications to the correlation of Beetstra et al. that matches our results. Additionally, we proposed a correlation to compute the Nusselt number as a function of the drag force. We also studied the deviation of individual particle-based interphase transfer coefficients from the mean value and we found that they behave like a lognormal distribution with constant scaled standard deviation.

In Chapter 4 we focussed on the effect of adiabatic walls on mono-dispersed suspensions. We first proposed a novel expression for the wall normal profile of the particle volume fraction which, unlike the models proposed in previous works [100], is applicable to flat walls over a broad range of bulk particle volume fractions. We also provided a formal description of the effect of the domain extension on the bulk particle volume concentration.

Furthermore, we observed that a perturbation of the particle volume fraction induces peculiar wall normal profiles for the flow and scalar fields in the region close to the wall. We named this region the "particle-induced boundary layer". Specifically, in this region the scalar field shows a wall-normal gradient despite the walls are adiabatic. We speculate that such gradient is due to the high velocity in the region very close to the wall. The flow in this region transports the scalar field further downstream at an higher rate with respect to the homogeneous region far from the walls. In fact, the velocity field shows a parabolic wall normal profile in the particle-induced boundary layer with a maximum that is a function of the particle volume fraction. We proposed correlations for the scalar gradient, as well as the wall scalar value together with the location and magnitude of the velocity field maximum. Such quantities allow to approximate the wall normal flow and scalar profiles in the particle-induced boundary layer with sufficiently accuracy. Finally, we investigated the effect of wall disturbances on the interphase transfer coefficients. This investigation highlights significant and systematic wall effects for dense systems.

5.2. Outlook

In the present thesis we focussed on the study and modelling of particle-based interphase transfer coefficients (ITC) for gas-particle suspension. As described in Chapter 3, we followed a standard approach defining an average ITC for a specific class of particles, i.e, particles experiencing the same value of voidage, Reynolds number and referring to the same filter size. Furthermore, we quantified the observed deviation by means of a distribution function, therefore closing the second order statistics for the particle-based ITC. However, such approach is only valid on a statistical basis, i.e., the closure would yield the correct mean and standard deviation of transfer coefficients when applied to a large particle ensemble. However, our results do *not* provide a deterministic description of per-particle transfer coefficients. In fact, deviations from the average ITC are not originating from a pure random process in PU-EL. This is because the relative position of particles is known, same as the average values of the continuous phase fields. This is in contrast with EE models, where the description of the dispersed phase is purely statistical. Specifically, one can speculate that ITCs deviations are fundamentally related to: (i) the relative position of particles and (ii) transport phenomena in the continuous phase happening on the sub-grid scale.

The first cause of deviation (i) has been recently studied by Akiki et al. [4] who proposed the "Pairwise Interaction Point-Particle" model (PIEP). This model makes use of the Faxe theorem to evaluate the drag force perturbation due to neighbouring particles. Even though the model predicts more accurate drag forces, it is still not adequate to account for high order particle interactions in complex systems due to its pairwise interaction approximation. Furthermore, it is not clear if such model can be somehow derived (or inferred) by using some approximated form of the governing equations for the continuous phase. This poses some serious questions on the generality of the PIEP model. Currently, the PIEP model is still limited to particle-based drag force, and no extension to general ITCs exists. Therefore, the quest for a general approach to deterministically predict per-particle ITCs as functions of the neighbouring particles is still ongoing. It appears of pivotal importance for the development of more accurate and reliable PU-EL simulators to shed more light on this issue in the near future.

The effect of sub-grid scales on ITC deviations (i.e., item ii) is by nature impossible to describe using a deterministic approach. This is because the coarse graining process conceals the information of the flow details in sub-grid stresses and fluxes. However, one can speculate that such deviation will be more prominent as the filter size (i.e., the PU-EL cell) becomes larger with respect to the particle diameter. This is expected because of the high non equilibrium of the sub-grid flow and the lack of scale separation. Furthermore, the particle tendency to form clusters creates strongly inhomogeneous sub-grid flows within a region of the coarse grained cell. In the extreme case, the force acting on a particle can have opposite sign with respect to the average fluid-particle relative speed. Such inhomogeneities are fundamentally different from pseudo-turbulent

phenomena [25] which have a diffusive nature, while sub-grid inhomogeneities are advective. Perhaps, a more advanced mathematical description of the coarse graining process is necessary to fully understand the role and the modelling of such sub-grid contributions.

In Chapter 4, we proposed an approach to model sub-grid inhomogeneities arising from adiabatic walls. Such work can be extended to develop proper boundary conditions for both PU-EL models and EE models, with the latter requiring our description of the wall-induced perturbations on the particle volume fraction in addition to the flow and scalar correlations. Therefore, wall functions - analogous to the ones found in turbulence modelling - could be implemented by exploiting the research work conducted within this thesis. Also, extension to situations involving fixed-temperature boundary conditions at the wall, as well as zero flux boundary conditions at the particle surfaces, would be helpful. This would help to quantify fluid-wall transfer coefficients in dense suspensions. Last, extending the current study to flowing suspensions could shed light into the effect of the particles' granular temperature on ICTs.

Finally, with the current increase in supercomputing capabilities which allow several billion particles to be tracked in PU-EL simulations, the role of on-the-fly data processing tools like CPPPO will get more and more prominent. Such filtering tools can be used to develop closure models as we did in the present thesis. Filtering is becoming increasingly interesting in the community, as shown by the recent attempts of Cloete et al. [19], or Tryggvason [95]. Another usage of such tools, which is directly connected with on-the-fly data processing, is the dynamic adjustment of coefficients in closure models. This strategy has been widely exploited in Large Eddy Simulations following the seminal work of Germano et al. [38]. It is currently applied in filtered two-fluid models (FTFM) for disperse multiphase flows [82]. However, such models generally rely on a similarity between the fine and coarse filtered scales. This assumption may be of questionable validity in the case of highly non-equilibrium flows. For example, the dynamic model from Germano et al. requires that the sub-grid stresses are described using the same model e.g., the Smagorinsky model [79], which is valid in the universal inertial range. Therefore, further studies on the accuracy and formulation of such filtered corrections are needed. Tools to perform such investigations are precisely those like CPPPO.

6. Scientific output

6.1. Peer-reviewed journals

4. F. Municchi and S. Radl, “Momentum, heat and mass transfer simulations of bounded dense mono-dispersed gas-particle systems", *International Journal of Heat and Mass Transfer* **Under Review**, (2017).

3. F. Municchi and S. Radl, “Consistent closures for Euler-Lagrange models of bi-disperse gas-particle suspensions derived from particle-resolved direct numerical simulations", *International journal of heat and mass transfer* **94**, p. 171-190 (2017).

2. J.H. Cloete, S. Cloete, F. Municchi, S. Radl and S. Amini, “The sensitivity of filtered Two Fluid Model to the underlying resolved simulation setup", *Powder Technology* **316**, p. 265-277 (2017).

1. F. Municchi, C. Goniva and S. Radl, “Highly efficient spatial data filtering in parallel using the opensource library CPPPO", *Computer Physics Communications* **207**, p. 400-414 (2016).

6.2. Talks and conferences

10. F. Municchi and S. Radl , “Closures for discrete suspension flow models: new insight from particle-resolved direct numerical simulations ad data filtering", *International Conference on Numerical Methods in Multiphase Flows, ICNMMF-III, University of Tokyo, Japan*, (2017)

9. F. Municchi, S. Radl and C. Goniva, “Near-wall effects for momentum, heat and mass transport in gas-particle suspensions at moderate Reynolds numbers", *Proceedings of the APS DFD 2016, Portland, United States*, (2016)

8. F. Municchi,M. Askarishahi,M. Salehi, C. Goniva, S. Radl, “On the Relevance of Closure Laws for Momentum, Heat and Mass Transfer in Gas-Particle Suspensions", *AIChE Annual Meeting 2016, San Francisco, United States*, (2016)

7. T. Forgber, F. Municchi, T. Puffitsch and S. Radl, “Intra- and Inter-Particle Resolved Simulations and Experiments on Thermal Transport in Confined Particle Beds", *Poster session presented at AIChE Annual Meeting 2016, San Francisco, United States*, (2016)

6. F. Municchi and S. Radl, “Fictitious Domain and Immersed Boundary methods in OpenFOAM: Application to Complex Geometries", *PFAU XIII: OpenFOAM user meeting, Vienna, Austria*, (2016)

5. F. Municchi and S. Radl, “Architecture and application of the data filtering library ”CPPPO” to transport phenomena in dense gas-particle flows", *PDESoft-2016, University of Warwick, UK*, (2016)

4. F. Municchi, S. Radl, C. Goniva, “Development of closure models for unresolved particle simulations of heat and mass transfer in dense particle beds using CPPPO: a library for fast parallel data filtering", *International Conference on Multiphase Flows, ICMF-2016, Florence, Italy*, (2016)

3. F. Municchi and S. Radl, “Highly efficient "on-the-fly" data processing using the open-source library CPPPO", *Workshop HPC enabling of OpenFOAM®, CINECA, Casalecchio di Reno, Italy*, (2016)

2. F. Municchi, S. Radl and C. Goniva, “Fast spatial filtering using CPPPO: a compilation for particle-fluid data post processing", *Austrian HPC meeting, Bad Ausse, Austria*, (2016)

1. F. Municchi, S. Radl and C. Gonival, “A Hybrid Fictitious Domain-Immersed Boundary Method for the Direct Simulation of Heat and Mass Transport in Fluid-Particle Systems", *CFDEMProject user meeting, Linz, Austria*, (2016)

6.3. GitHub repositories

2. **openHFDIB**
   link:GitHub
   Open source implementeation of the Hybrid Fictitious-Domain/Immersed-Boundary Method in OpenFOAM®

1. **C3PO-PUBLIC**
   link:GitHub
   Public repository for the compilation of fluid/particle parallel processing routines.

A. HFD-IB: extension to general boundary conditions

In this appendix we show how the HFD-IB method can be extended to impose general boundary conditions on a general partial differential equation.

A.1. Partial differential equations and boundary conditions

We consider a computational domain C and let $\Omega \subset C$ be the *physical domain*, which is assumed to be smooth and not self-intersecting. Let $B = C \setminus \overline{\Omega} = \cup_i B_i$, where B_i are the *immersed domains* assumed not to be interpenetrating, i.e. $B_i \cap B_j = \partial B_i \cap \partial B_j$ or $\overline{B}_i \cap \overline{B}_j = \emptyset$ for all $i \neq j$. Finally, we define the *immersed surface* $\Gamma = \cup_i \Gamma_i$ where $\Gamma_i = B_i \cap \Omega$ are the immersed surfaces (or interfaces) relative to the immersed body i.

We consider a partial differential equation of k^{th} order (that can be linear or non-linear) for the scalar field $\phi(t, \mathbf{x})$ (extension to vector fields is straightforward) together with a proper set of boundary conditions at the immersed surfaces. For simplicity, we will assume $\partial\Omega \subset \Gamma$ (every boundary in the physical domain is an immersed boundary) but extension to the general case of mixed standard and immersed boundaries is straightforward. Let $\mathcal{P}^k$ be a partial differential operator of order k in space and let $\Phi(t, \mathbf{x})$ be a function of space and time (a source term, in case $\mathcal{P}^k\phi(t, \mathbf{x})$ represents a transport equation). Thus, the differential problem for N_b immersed bodies can be written as:

$$\mathcal{P}^k\phi(t, \mathbf{x}) = \Phi(t, \mathbf{x}) \qquad \forall \mathbf{x} \in \Omega, \tag{A.1a}$$

$$\sum_{n=0}^{k-1} \alpha_{n,i}(t, \mathbf{x}) \left[\mathbf{n}_i(t, \mathbf{x}) \cdot \nabla\right]^n \phi = \gamma_i(t, \mathbf{x}) \qquad \forall \mathbf{x} \in \Gamma_i, \quad \forall i = 0, 1, \ldots, N_b. \tag{A.1b}$$

Where the immersed boundary conditions are expressed in general form as functions of the boundary coefficients $\alpha_{n,i}(t, \mathbf{x})$ and $\gamma_i(t, \mathbf{x})$. We also denoted the vectors normal

to the immersed surface Γ_i as $\mathbf{n}_i(t, \mathbf{x})$. Note that the time dependence of normal vectors and boundary coefficients implies that immersed bodies are allowed to move.

Remark 1 *Notice that equations A.1a and A.1b are formulated in* ***dimensionless form*** *for the sake of generality and simplicity.*

The next step consist in formulating equation A.1a in C. This implies that equation A.1a is solved in the entire computational domain, thus requiring additional explicit boundary conditions $BC(\phi) = 0$ on ∂C. In the following, we will not discuss these boundary conditions, that are specific for each application and pose no relevant issue to the illustrated method. Concerning the immersed boundary conditions, they are imposed by modifying equation A.1a with an additional term that acts like a Lagrangian multiplier.

Let $\chi_i(t, \mathbf{x})$ be an indicator function defined as:

$$\chi_i(t, \mathbf{x}) = \begin{cases} 1 & \text{if } \mathbf{x} \in \Gamma_i \\ 0 & \text{otherwise} \end{cases} \tag{A.2}$$

Then, the immersed boundary problem can be stated as:

$$\mathcal{P}^k \phi(t, \mathbf{x}) - \Phi(t, \mathbf{x}) = \sum_{i=0}^{N_b} \chi_i(t, \mathbf{x}) f_i(t, \mathbf{x}) \qquad \forall \mathbf{x} \in C, \tag{A.3a}$$

$$BC(\phi) = 0 \qquad \forall \mathbf{x} \in \partial C. \tag{A.3b}$$

We express the forcing terms $f_i(t, \mathbf{x})$ as:

$$f_i(t, \mathbf{x}) = \left[\mathcal{P}^k \phi(t, \mathbf{x}) - \Phi(t, \mathbf{x})\right] + \left[\phi(t, \mathbf{x}) - \psi_i(t, \mathbf{x})\right] \tag{A.4}$$

In equation A.4, the first term on the RHS is just used to equate the LHS of equation A.3a, so that when $\chi_i = 1$ the partial differential equation is reduced to $\phi(t, \mathbf{x}) = \psi_i(t, \mathbf{x})$. Thus, $\psi_i(t, \mathbf{x})$ represents the value that field ϕ should have at the boundary to satisfy the immersed boundary condition for the immersed body i. Therefore, in the present method we assume that the problem given by equation A.1a with boundary conditions A.1b has a solution that is unique.

If we define the operator $A_i = \sum_{n=0}^{k-1} \alpha_{n,i}(t, \mathbf{x}) \left[\mathbf{n}_{\Gamma_i}(t, \mathbf{x}) \cdot \nabla\right]^n$, we can express $\psi_i(t, \mathbf{x})$ as follows:

$$\psi_i(t, \mathbf{x}) = A_i^{-1} \gamma(t, \mathbf{x}) \tag{A.5}$$

Thus, substituting in the forcing terms:

$$f_i(t,\mathbf{x}) = \left[\mathcal{P}^k\phi(t,\mathbf{x}) - \Phi(t,\mathbf{x})\right] + \left[\phi(t,\mathbf{x}) - A_i^{-1}\gamma(t,\mathbf{x})\right] \tag{A.6}$$

Thus, the inverse operator A_i^{-1} plays a major role in the present method since allows to reduce any boundary condition to the imposition of a specific value for the field ϕ. This allows to use standard immersed boundary approaches developed for Dirichlet boundary conditions with minimal modifications.

A.2. Discretization of the immersed boundary condition: evaluation of ψ_i

In the following, we assume that the function ϕ is analytic everywhere. Therefore, we can express the field in the proximity of the immersed boundaries in Taylor series:

$$\phi(t,\mathbf{x}) = \sum_{k=0}^{\infty} \frac{1}{k!}\left[(\mathbf{x}-\mathbf{z_i})\cdot\nabla\right]^k \psi|_{\mathbf{z}_i} \quad \forall \mathbf{z_i} \in \Gamma_i \tag{A.7}$$

In particular, we are interested in an expression involving the derivatives normal to the immersed surface i. Therefore, we perform a change of variable and rewrite A.7 as:

$$\phi(t, s\mathbf{n}_i + \mathbf{z}_i) = \sum_{k=0}^{\infty} \frac{s^k}{k!} \left.\frac{\partial^k \psi}{\partial s^k}\right|_{\mathbf{z}_i} \quad \forall \mathbf{z_i} \in \Gamma_i \tag{A.8}$$

Notice that the normal derivatives are now calculated using the parameter s, which represents the distance from the immersed boundary point $\mathbf{z}_i$ in the direction normal to the immersed surface. Clearly, expansion A.8 can be written only if we assume that $\phi = \psi_i$ at the immersed boundary, thus if the field ϕ is constrained by A.6. Therefore, defining the Taylor operator $T_i = \sum_{k=0}^{\infty} \frac{s^k}{k!} \left.\frac{\partial^k}{\partial s^k}\right|_{\mathbf{z}_i}$ we can summarize:

$$\psi_i(t,\mathbf{z}_i) = A_i^{-1}\gamma_i(t,\mathbf{z}_i) = T_i^{-1}\phi(t, s\mathbf{n}_i + \mathbf{z}_i) \quad \forall \mathbf{z_i} \in \Gamma_i \tag{A.9}$$

Which relates ψ_i,γ_i and ϕ by means of the operators T_i and A_i. However, the normal derivatives of ϕ that satisfy $\phi = \psi_i$ are unknown and can be evaluated (for each $\mathbf{z_i} \in \Gamma_i$)

from the following system of equations:

$$\begin{cases} T_i^N \psi_i(t, \mathbf{z}_i) = \phi(t, \Delta s \mathbf{n}_i + \mathbf{z}_i) \\ T_i^N \psi_i(t, \mathbf{z}_i) = \phi(t, 2\Delta s \mathbf{n}_i + \mathbf{z}_i) \\ \quad \vdots \\ T_i^N \psi_i(t, \mathbf{z}_i) = \phi(t, N\Delta s \mathbf{n}_i + \mathbf{z}_i) \\ A_i \psi_i(t, \mathbf{z}_i) = \gamma_i(t, \mathbf{z}_i) \end{cases} \tag{A.10}$$

Where Δs is a finite interval and T_i^N is the Taylor operator truncated at the term N, being N the order of the immersed boundary condition.

Remark 2 *The immersed boundary operator can be of any order depending on the values of the coefficients $\alpha_{n,i}$. In fact, in the limit one could take $\alpha_{0,i} = 1$ and $\alpha_{n,i} = 0, \forall n > 0$ therefore allowing for $N = \infty$, which would require an infinite number of equations. Thus, the boundary condition is not limiting the order of the Taylor series, which may be used to reconstruct the field in the proximity of the immersed boundary with arbitrary precision.*

Substantially, system A.10 is expressed in term of the matrix-vector product:

$$\mathbf{M} = \begin{bmatrix} 1 & \Delta s & \frac{\Delta s^2}{2} & \cdots & \frac{\Delta s^N}{N!} \\ 1 & 2\Delta s & \frac{(2\Delta s)^2}{2} & \cdots & \frac{(2\Delta s)^N}{N!} \\ 1 & 4\Delta s & \frac{(4\Delta s)^2}{2} & \cdots & \frac{(4\Delta s)^N}{N!} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \alpha_0 & \alpha_1 & \alpha_2 & \cdots & \alpha_N \end{bmatrix} \quad \mathbf{d} = \begin{bmatrix} \psi_i(t, \mathbf{z}_i) \\ \frac{\partial \psi}{\partial s}\Big|_{\mathbf{z}_i} \\ \frac{\partial^2 \psi}{\partial s^2}\Big|_{\mathbf{z}_i} \\ \vdots \\ \frac{\partial^N \psi}{\partial s^N}\Big|_{\mathbf{z}_i} \end{bmatrix} \quad \mathbf{t} = \begin{bmatrix} \phi(t, \Delta s) \\ \phi(t, 2\Delta s) \\ \phi(t, 3\Delta s) \\ \vdots \\ \gamma_i(t, \mathbf{z}_i) \end{bmatrix} \tag{A.11}$$

$$\mathbf{M}\mathbf{d} = \mathbf{t} \tag{A.12}$$

Therefore, derivatives of ψ_i at $\mathbf{x} = \mathbf{z}_i$ are obtained performing $\mathbf{d} = \mathbf{M}^{-1}\mathbf{t}$. Clearly, the computational cost increases with increasing order of the truncated Taylor series or the immersed boundary condition.

Remark 3 *Notice that for typical CFD applications most boundary condition have $N < 1$ and differential operators are normally discretized using second order accurate methods (in the case of finite volume simulators). Therefore, matrix $\mathbf{M}$ is generally 3×3 so that $\mathbf{M}^{-1}$ can be pre-computed and the number of floating point operations required to apply this method is negligible when compared to the algorithms required for solving the linear system arising from the discretization of equation A.1a.*

A.3. Discretization of the immersed boundary condition: discretized computational domain

Discretization of the computational domain implies two major issues:

i Location $\Delta s\mathbf{n}_i + \mathbf{z}_i$ may not correspond to any computational node so that a strategy to evaluate $\phi(t, \Delta s\mathbf{n}_i + \mathbf{z}_i)$ is required.

ii Location $\mathbf{z}_i$ may not correspond to any computational node and therefore the boundary value $\psi(t, \mathbf{z}_i)$ can not be straightforwardly applied to the discretized domain.

The first issue (i) can be solved by defining general interpolation operators $\mathcal{I}^*$ that map the field ψ from the computational grid to location $\Delta s\mathbf{n}_i + \mathbf{z}_i$. We will not discuss these operators in detail, we simply notice that they may affect the accuracy of the immersed boundary method since generally $\phi(t, \Delta s\mathbf{n}_i + \mathbf{z}_i) \approx \mathcal{I}^*\phi$.

The second issue (ii) is posing the opposite problem: extrapolation from a specific node to the surrounding grid. In order to preserve the N accuracy of the solution we may take advantage of the just computed derivatives of $\psi(t, \mathbf{z}_i)$ to calculate ϕ at the closest node or cell center. Therefore, each point $\mathbf{z}_i$ must lie on a line that is normal to the immersed surface and passing through the grid point (or cell center) $\mathbf{p}_{\mathbf{z}_i}$. Thus, the computational grid naturally discretizes the immersed surface by defining the boundary nodes $\mathbf{z}_i$ as shown in figure A.1. Specifically, the density of boundary nodes increases with decreasing cell spacing, i.e., the discretized surface is refined together with the mesh.

The discrete forcing term $f_i^*(t, \mathbf{p}_{\mathbf{z}_i})$ is then modified as:

$$f_i^*(t, \mathbf{p}_{\mathbf{z}_i}) = \left[\mathcal{P}^k\phi(t, \mathbf{p}_{\mathbf{z}_i}) - \Phi(t, \mathbf{p}_{\mathbf{z}_i})\right] + \left[\phi(t, \mathbf{p}_{\mathbf{z}_i}) - \psi_i^*(t, \mathbf{p}_{\mathbf{z}_i})\right] \tag{A.13}$$

Where $\psi_i^*(t, \mathbf{p}_{\mathbf{z}_i})$ is evaluated from:

$$\psi_i^*(t, \mathbf{p}_{\mathbf{z}_i}) = \sum_{k=0}^{N} \frac{1}{k!} \left|\mathbf{p}_{\mathbf{z}_i} - \mathbf{z}_i\right| \left.\frac{\partial^k \psi}{\partial s^k}\right|_{\mathbf{z}_i} \tag{A.14}$$

Therefore, for each boundary cell, i.e. a cell that is cut by an immersed surface, the solver has to perform the following steps:

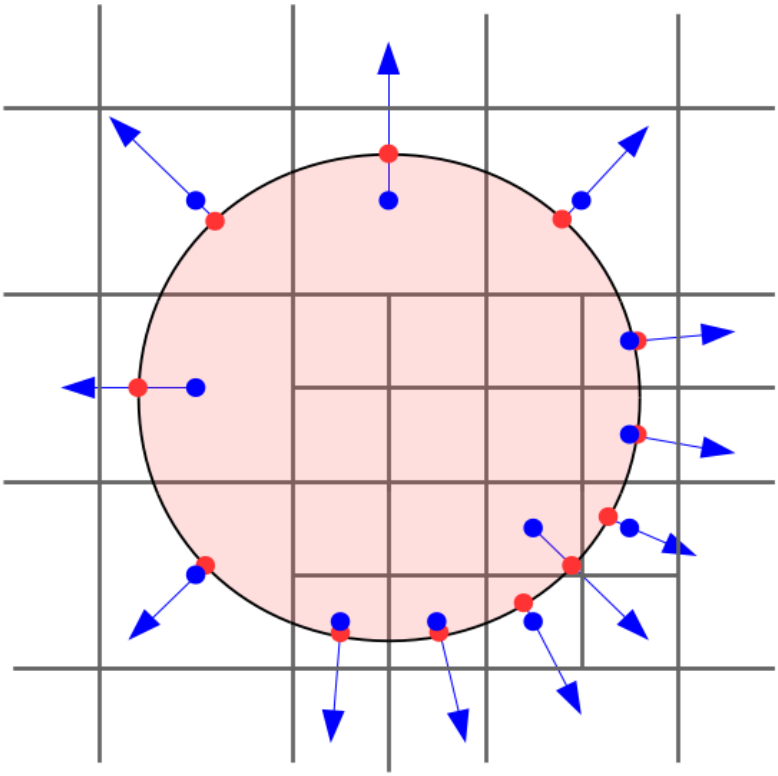


Figure A.1.: Immersed body (shaded red) discretized over non-uniform grid. Red nodes represent the boundary nodes $\mathbf{z}_i$ and blue nodes represents boundary cell centres (where the forcing term is applied). Blue arrows indicate the direction normal to the surface $\mathbf{n}_i$ and illustrate how boundary nodes are computed as projection of boundary cell centres over $\mathbf{n}_i$.

- Assemble the matrix $\mathbf{M}$ if the boundary coefficients are varying with time and evaluate vector $\mathbf{t}^* \approx \mathbf{t}$ using the interpolators:

$$\mathbf{t}^* = \begin{bmatrix} \mathcal{I}^*\phi(t, \Delta s) \\ \mathcal{I}^*\phi(t, 2\Delta s) \\ \mathcal{I}^*\phi(t, 3\Delta s) \\ \vdots \\ \gamma_i(t, \mathbf{z}_i) \end{bmatrix} \tag{A.15}$$

- Solve the system: $\mathbf{d} = \mathbf{M}^{-1}\mathbf{t}^*$.
- Evaluate $\psi_i^*(t, \mathbf{p}_{\mathbf{z}_i})$ from equation A.14.
- Update the forcing term defined in A.13.

The modified partial differential equation A.3a together with the boundary conditions A.3b can then be solved using appropriate numerical algorithms.

Remark 4 *Notice that the present formulation for the forcing term f_i^* as described in equation A.13 is equivalent to the formulation for Q presented in Chapter 3. Such formulation is preferable to the one we employed for the velocity field in Chapter 3 and described by Blais et al. [13] since it can be shown to produce stable results at higher Reynolds numbers.*

A.4. Verification of the mixed boundary condition

Finally, we propose a simple verification case to test the accuracy of the proposed algorithm when dealing with Robin boundary conditions. Specifically, we will solve the 1-dimensional ordinary differential equation:

$$\frac{d^2\phi(x)}{dx^2} = 20\cos(3\pi x) \tag{A.16}$$

With the following boundary conditions:

$$-\phi + 5\frac{d\phi}{dx} = 3, \quad x = 0 \tag{A.17a}$$

$$\phi = 1, \quad x = 1 \tag{A.17b}$$

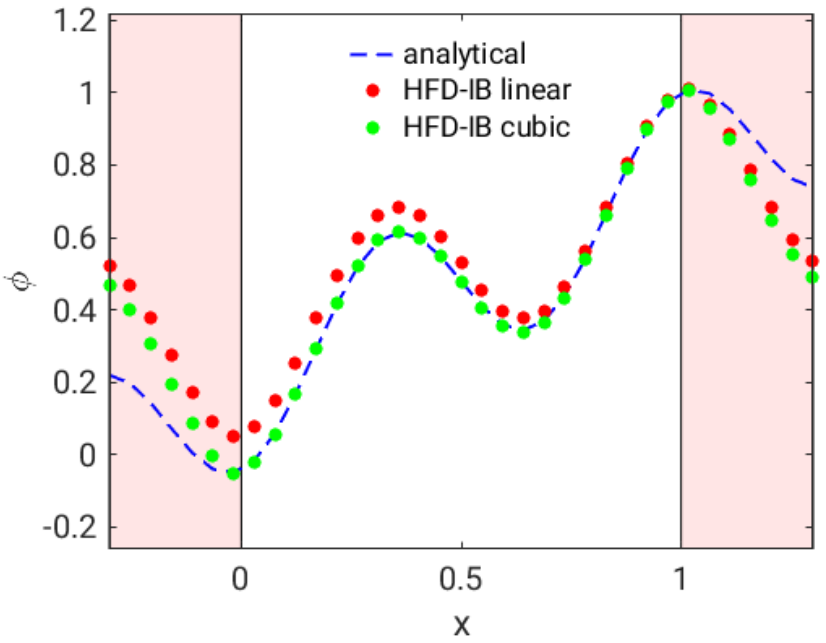


Figure A.2.: Comparison between the exact solution of equation A.16 and the HFD-IB algorithm using linear interpolation (red dots) and cubic interpolation (green dots). Red shaded areas represent the extended domain.

Equation A.16 is discretized using a second order finite difference scheme (central differencing) and solved using a Gauss-Seidel algorithm. The forcing term arising from the HFD-IB method is updated at each iteration. In order to apply the HFD-IB method we solve Equation A.16 in $x \in [-0.3, 1.3]$ and we apply periodic boundary conditions. Therefore, intervals $[0.3, 0]$ and $[1, 1.3]$ represent the extended domain that in the case of external flows would corresponds to the domain occupied by an immersed body.

Two interpolation points are employed in the Taylor expansion of ψ_i, which is the minimum requirement to impose A.17a. Results obtained using linear and cubic interpolators $\mathcal{I}^*$ are presented in Figure A.2.

Cubic interpolators are performing better than linear interpolators when high order (i.e., Robin) boundary conditions are employed while Dirichlet boundary conditions are less susceptible to the accuracy of the interpolation. This is because a Dirichlet boundary condition provides the leading term in the Taylor expansion, while in higher order boundary conditions the leading term is obtained from the solution of $\mathbf{d}^* = \mathbf{M}^{-1}\mathbf{t}^*$ and it is therefore a function of the interpolated values.

Furthermore, increasing the order of the Taylor series (and thus the size of $\mathbf{M}$) does not lead to significant improvements in the solution compared to the use of more accurate interpolation schemes.

Overall, the method is capable of correctly impose general boundary conditions with an accuracy that depends on the order of the interpolation schemes adopted

The MATLAB®code employed to produce this verification case can be found at author's gitHub repository.

Bibliography

[cfd] CFDEMproject. https://www.cfdem.com/.

[ope] OpenFOAM. https://openfoam.org/.

[swa] swak4foam. https://openfoamwiki.net/index.php/Contrib/swak4Foam.

[4] Akiki, G., Jackson, T. L., and Balachandar, S. (2017). Pairwise interaction extended point-particle model for a random array of monodisperse spheres. *Journal of Fluid Mechanics*, 813:882–928.

[5] Anderson, T. B. and Jackson, R. O. Y. (1967). A Fluid Mechanical description of fluidized beds. *Ind. Eng. Chem. Fundamen.*, 6(4):527–539.

[6] Andrews IV, A. T., Loezos, P. N., and Sundaresan, S. (2005). Coarse-grid simulation of gas-particle flows in vertical risers. *Industrial and Engineering Chemistry Research*, 44(16):6022–6037.

[7] Angot, P. (2005). A unified fictitious domain model for general embedded boundary conditions. *Comptes Rendus Mathematique*, 341(11):683–688.

[8] Auricchio, F., Boffi, D., Gastaldi, L., Lefieux, A., and Reali, A. (2015). On a fictitious domain method with distributed Lagrange multiplier for interface problems. *Applied Numerical Mathematics*, 95:36–50.

[9] Bale, S., Sathe, M., Ayeni, O., Berrouk, A. S., Joshi, J., and Nandakumar, K. (2017). Spatially resolved mass transfer coefficient for moderate Reynolds number flows in packed beds: Wall effects. *International Journal of Heat and Mass Transfer*, 110:406–415.

[10] Batchelor, G. K. (2000). *An Introduction to Fluid Dynamics*. Cambridge University Press, Cambridge.

[11] Beetstra, R. (2009). Drag Force of Intermediate Reynolds Number Flow Past Mono- and Bidisperse Arrays of Spheres. *IFAC Proceedings Volumes (IFAC-PapersOnline)*, 7(2):405–410.

[12] Berselli, L. C. (2012). Analysis of a Large Eddy Simulation model based on anisotropic filtering. *Journal of Mathematical Analysis and Applications*, 386(1):149–170.

[13] Blais, B., Lassaigne, M., Goniva, C., Fradette, L., and Bertrand, F. (2016). A semi-implicit immersed boundary method and its application to viscous mixing. *Computers and Chemical Engineering*, 85:136–146.

[14] Buist, K., Backx, B., Deen, N., and Kuipers, J. (2017). A combined experimental and simulation study of fluid-particle heat transfer in dense arrays of stationary particles. *Chemical Engineering Science*, 169:310–320.

[15] Capecelatro, J. and Desjardins, O. (2013). An Euler-Lagrange strategy for simulating particle-laden flows. *Journal of Computational Physics*, 238:1–31.

[16] Capecelatro, J., Desjardins, O., and Fox, R. (2014). Numerical study of collisional particle dynamics in cluster-induced turbulence. *Journal of Fluid Mechanics*, 747:R2.

[17] Capecelatro, J., Desjardins, O., and Fox, R. O. (2015). On fluid–particle dynamics in fully developed cluster-induced turbulence. *Journal of Fluid Mechanics*, 780:578–635.

[18] Clift, R., Grace, J., and Weber, M. (1978). *Bubbles, Drops and Particles*.

[19] Cloete, J. H., Cloete, S., Municchi, F., Radl, S., and Amini, S. (2017). The sensitivity of filtered Two Fluid Model to the underlying resolved simulation setup. *Powder Technology*, 316:265–277.

[20] de Klerk, A. (2003). Voidage variation in packed beds at small column to particle diameter ratio. *AIChE Journal*, 49(8):2022–2029.

[21] De Stefano, G. and Vasilyev, O. V. (2002). Sharp cutoff versus smooth filtering in large eddy simulation. *Physics of Fluids*, 14(1):362–369.

[22] Deen, N. G., Kriebitzsch, S. H. L., van der Hoef, M. a., and Kuipers, J. a. M. (2012). Direct numerical simulation of flow and heat transfer in dense fluid-particle systems. *Chemical Engineering Science*, 81:329–344.

[23] Deen, N. G. and Kuipers, J. a. M. (2014). Direct Numerical Simulation (DNS) of mass, momentum and heat transfer in dense fluid-particle systems. *Current Opinion in Chemical Engineering*, 5:84–89.

[24] Deen, N. G., Peters, E. a. J. F., Padding, J. T., and Kuipers, J. a. M. (2014). Review of direct numerical simulation of fluid-particle mass, momentum and heat transfer in dense gas-solid flows. *Chemical Engineering Science*, 116:710–724.

[25] Delgado, J. M. P. Q. (2006). A critical review of dispersion in packed beds. *Heat and Mass Transfer/Waerme- und Stoffuebertragung*, 42(4):279–310.

[26] Derksen, J. (2014a). Simulations of solid–liquid mass transfer in fixed and fluidized beds. *Chemical Engineering Journal*, 255:233–244.

[27] Derksen, J. J. (2014b). Simulations of solid-liquid scalar transfer for a spherical particle in laminar and turbulent flow. *AIChE Journal*, 60(3):1202–1215.

[28] Derksen, J. J. and Sundaresan, S. (2007). Direct numerical simulations of dense suspensions: wave instabilities in liquid-fluidized beds. *Journal of Fluid Mechanics*, 587:303–336.

[29] Dixon, A. G. (2017). Local transport and reaction rates in a fixed bed reactor tube: Endothermic steam methane reforming. *Chemical Engineering Science*, 168:156–177.

[30] Dong, Y., Sosna, B., Korup, O., Rosowski, F., and Horn, R. (2017). Investigation of radial heat transfer in a fixed-bed reactor: CFD simulations and profile measurements. *Chemical Engineering Journal*, 317:204–214.

[31] Favre, A. (1965). Review on space-time correlations in turbulent fluids. *J. Appl. Mech.*, 32:241–257.

[32] Favre, A. (1992). Formulation of the Statistical Equations of Turbulent Flows with Variable Density. In *Studies in turbulence*, pages pp 324–341.

[33] Feng, Z.-G. and Michaelides, E. E. (2000). A numerical study on the transient heat transfer from a sphere at high Reynolds and Peclet numbers. *International Journal of Heat and Mass Transfer*, 43(2):219–229.

[34] Feng, Z.-G. and Michaelides, E. E. (2009). Heat transfer in particulate flows with Direct Numerical Simulation (DNS). *International Journal of Heat and Mass Transfer*, 52(3-4):777–786.

[35] Feng, Z.-G. and Musong, S. G. (2014). Direct numerical simulation of heat and mass transfer of spheres in a fluidized bed. *Powder Technology*, 262:62–70.

[36] Forgber, T. and Radl, S. (2018). A novel approach to calculate radiative thermal exchange in coupled particle simulations. *Powder Technology*, 323:24–44.

[37] Geldart, D., Baeyens, J., Pope, D., and Van De Wijer, P. (1981). Segregation in beds of large particles at high velocities. *Powder Technology*, 30:195–205.

[38] Germano, M., Piomelli, U., Moin, P., and Cabot, W. H. (1991). A dynamic subgrid-scale eddy viscosity model. *Physics of Fluids A: Fluid Dynamics*, 3(7):1760–1765.

[39] Germano M. (1980). Differential filters for the large eddy numerical simulation of turbulent flow. *Phys. Fluids*, 29(6):1755–1757.

[40] Giese, M., Rottschäfer, K., and Vortmeyer, D. (1998). Measured and modeled superficial flow profiles in packed beds with liquid flow. *AIChE Journal*, 44(2):484–490.

[41] Gilmanov, A. and Sotiropoulos, F. (2005). A hybrid Cartesian/immersed boundary method for simulating flows with 3D, geometrically complex, moving bodies. *Journal of Computational Physics*, 207(2):457–492.

[42] Glowinski, R., Pan, T., Hesla, T., Joseph, D., and Périaux, J. (2001). A Fictitious Domain Approach to the Direct Numerical Simulation of Incompressible Viscous Flow past Moving Rigid Bodies: Application to Particulate Flow. *Journal of Computational Physics*, 169(2):363–426.

[43] Gunn, D. (1978). Transfer of heat or mass to particles in fixed and fluidised beds. *International Journal of Heat and Mass Transfer*, 21(4):467–476.

[44] Hill, R. J., Koch, D. L., and Ladd, A. J. C. (2001). The first effects of fluid inertia on flows in ordered and random arrays of spheres. *Journal of Fluid Mechanics*, 448:243–278.

[45] Hoef, M. a. V. D., Beetstra, R., and Kuipers, J. a. M. (2005). Lattice-Boltzmann simulations of low-Reynolds-number flow past mono- and bidisperse arrays of spheres: results for the permeability and drag force. *Journal of Fluid Mechanics*, 528:233–254.

[46] Holloway, W., Benyahia, S., Hrenya, C. M., and Sundaresan, S. (2011). Mesoscale structures of bidisperse mixtures of particles fluidized by a gas. *Chemical Engineering Science*, 66(19):4403–4420.

[47] Holloway, W. and Sundaresan, S. (2014). Filtered models for bidisperse gas-particle flows. *Chemical Engineering Science*, 108:67–86.

[48] Holloway, W., Yin, X., and Sundaresan, S. (2009). Fluid-particle drag in inertial polydisperse gas-solid suspensions. *AIChE Journal*, 59(2):1995–2004.

[49] Hornung, U. (1997). *Homogenization and Porous Media*, volume 6 of *Interdisciplinary Applied Mathematics*. Springer New York, New York, NY.

[50] Igci, Y., Andrews, A. T., Sundaresan, S., Pannala, S., and O'Brien, T. (2008). Filtered two-fluid models for fluidized gas-particle suspensions. *AIChE Journal*, 54(6):1431–1448.

[51] Jackson, R. (2000). *The Dynamics of Fluidized Particles*. Cambridge Monographs on Mechanics.

[52] Kelbaliyev, G. and Ceylan, K. (2001). A theoretical model for the particle distribution in a polydispersed solid mixture under hydrodynamic and gravitational effects. *Powder Technology*, 115(1):84–89.

[53] Kloss, C., Goniva, C., Hager, A., Amberger, S., and Pirker, S. (2012). Models , algorithms and validation for opensource DEM and CFD-DEM. *Progress in Computational Fluid Dynamics*, 12:140–152.

[54] Kriebitzsch, S. H. L., van der Hoef, M. A., and Kuipers, J. A. M. (2013). Drag force in discrete particle models-Continuum scale or single particle scale? *AIChE Journal*, 59(1):316–324.

[55] Kubie, J. (1988). Influence of containing walls on the distribution of voidage in packed beds of uniform spheres. *Chemical Engineering Science*, 43(6):1403–1405.

[56] Li, J., Ge, W., Wang, W., Yang, N., Liu, X., Wang, L., He, X., Wang, X., Wang, J., and Kwauk, M. (2013). *From Multiscale Modeling to Meso-Science*. Springer Berlin Heidelberg, Berlin, Heidelberg.

[57] Ma, T., Bosilca, G., Bouteiller, A., and Dongarra, J. J. (2013). Kernel-assisted and topology-aware MPI collective communications on multicore/many-core platforms. *Journal of Parallel and Distributed Computing*, 73(7):1000–1010.

[58] Maheshwari, A., Chhabra, R. P., and Biswas, G. (2006). Effect of blockage on drag and heat transfer from a single sphere and an in-line array of three spheres. *Powder Technology*, 168(2):74–83.

[59] Mamidala, A. R., Kumar, R., De, D., and Panda, D. K. (2008). MPI collectives on modern multicore clusters: Performance optimizations and communication characteristics. *Proceedings CCGRID 2008 - 8th IEEE International Symposium on Cluster Computing and the Grid*, pages 130–137.

[60] Marchisio, Daniele L., Fox, R. O. (2007). *Multiphase reacting flows: modelling and simulation*. Springer.

[61] Mark, a., Rundqvist, R., and Edelvik, F. (2011). Comparison Between Different Immersed Boundary Conditions for Simulation of Complex Fluid Flows. *Fluid Dynamics and Materials Processing*, 7(3):241–258.

[62] Mark, A. and van Wachem, B. G. (2008). Derivation and validation of a novel implicit second-order accurate immersed boundary method. *Journal of Computational Physics*, 227(13):6660–6680.

[63] Marti, J., Haselbacher, A., and Steinfeld, A. (2015). A numerical investigation of gas-particle suspensions as heat transfer media for high-temperature concentrated solar power. *International Journal of Heat and Mass Transfer*, 90:1056–1070.

[64] M.H. Govindarao, V. and Froment, G. F. (1986). Voidage profiles in packed beds of spheres. *Chemical Engineering Science*, 41(3):533–539.

[65] Mikhailov, M. and Freire, a. S. (2013). The drag coefficient of a sphere: An approximation using Shanks transform. *Powder Technology*, 237:432–435.

[66] MPI Forum (2012). *MPI: A Message-Passing Interface Standard Version 3.0*.

[67] Mueller, G. E. (1993). Radial void fraction distributions in randomly packed fixed beds of uniformly sized spheres in cylindrical containers. *Powder Technology*, 77(3):313–319.

[68] Municchi, F., Goniva, C., and Radl, S. (2016a). Highly efficient spatial data filtering in parallel using the opensource library CPPPO. *Computer Physics Communications*, 207:400–414.

[69] Municchi, F. and Radl, S. (2017). Consistent closures for Euler-Lagrange models of bi-disperse gas-particle suspensions derived from particle-resolved direct numerical simulations. *International Journal of Heat and Mass Transfer*, 111:171–190.

[70] Municchi, F., Radl, S., and Goniva, C. (2016b). Near-wall effects for momentum, heat and mass transport in gas-particle suspensions at moderate Reynolds numbers. In *APS Division of Fluid Dynamics Meeting*.

[71] Nakariyakul, S. (2013). Fast spatial averaging: An efficient algorithm for 2D mean filtering. *Journal of Supercomputing*, 65:262–273.

[72] Nijemeisland, M., Dixon, A. G., and Stitt, E. H. (2004). Catalyst design by CFD for heat transfer and reaction in steam reforming. *Chemical Engineering Science*, 59(22-23):5185–5191.

[73] Peskin, C. (1972). Flow patterns around heart valves: A numerical method. *Journal of Computational Physics*, 10(2):252–271.

[74] Radl, S., Girardi, M., and Sundaresan, S. (2012). Effective Drag Law for Parcel-Based Approaches - What Can We Learn From Cfd-Dem ? *European Congress on Computational Methods in Applied Sciences and Engineering (ECCOMAS 2012), Vienna, Austria*, (Eccomas):1–15.

[75] Radl, S. and Sundaresan, S. (2014). A drag model for filtered Euler–Lagrange simulations of clustered gas–particle suspensions. *Chemical Engineering Science*, 117:416–425.

[76] Ramachandran, R. S., Kleinstreuer, C., and Wang, T. (1989). Forced convection heat transfer of interacting spheres. *Numerical Heat Transfer, Part A: Applications*, 15(4):471–487.

[77] Ranz, W. and Marshal, W. (1952). Evaporation from drops. *Chem. Eng. Prog.*, (48):141–146.

[78] Sabetghadam, F. (2015). Exact imposition of the regular rigid immersed surfaces on the solution of the incompressible Navier – Stokes equations. *ArXiv*.

[79] Sagaut, P. (2006). *Large Eddy Simulation for Incompressible Flows*. Scientific Computation. Springer-Verlag, Berlin/Heidelberg.

[80] Salehi, M.-S., Askarishahi, M., and Radl, S. (2017). Analytical solution for thermal transport in packed beds with volumetric heat source. *Chemical Engineering Journal*, 316:131–136.

[81] Sankaran Sundaresan, Stefan Radl, Christian C. Milioli, F. E. M. (2013). Coarse-Grained Models for Momentum, Energy and Species Transport in Gas-Particle Flows. *The 14th International Conference on Fluidization – From Fundamentals to Products*.

[82] Schneiderbauer, S. (2017). A spatially-averaged two-fluid model for dense large-scale gas-solid flows. *AIChE Journal*, 63(8):3544–3562.

[83] Schneiderbauer, S. and Pirker, S. (2014). A Coarse-Grained Two-Fluid Model for Gas-Solid Fluidized Beds. *The Journal of Computational Multiphase Flows*, 6(1):29–48.

[84] Sierakowski, A. J. and Prosperetti, A. (2016). Resolved-particle simulation by the Physalis method: Enhancements and new capabilities. *Journal of Computational Physics*, 309:164–184.

[85] Singhal, A., Cloete, S., Radl, S., Quinta-Ferreira, R., and Amini, S. (2017). Heat transfer to a gas from densely packed beds of monodisperse spherical particles. *Chemical Engineering Journal*, 314:27–37.

[86] Smagulov, S. S. (1979). Fictitious domains method for Navier-Stokes equations. *Preprint CC SA USSR*, 68.

[87] Sun, B., Tenneti, S., and Subramaniam, S. (2015). Modeling average gas–solid heat transfer using particle-resolved direct numerical simulation. *International Journal of Heat and Mass Transfer*, 86:898–913.

[88] Tavassoli, H., Kriebitzsch, S., van der Hoef, M., Peters, E., and numerical simulation of particulate flow with heat transferJ.a.M. Kuipers, D. (2013). Direct numerical simulation of particulate flow with heat transfer. *International Journal of Multiphase Flow*, 57:29–37.

[89] Tavassoli, H., Peters, E., and Kuipers, J. (2015). Direct numerical simulation of fluid–particle heat transfer in fixed random arrays of non-spherical particles. *Chemical Engineering Science*, 129:42–48.

[90] Tavassoli, H., Peters, E., and Kuipers, J. (2017). Direct numerical simulation of non-isothermal flow through dense bidisperse random arrays of spheres. *Powder Technology*, 314:291–298.

[91] Tenneti, S. (2013). *Momentum , energy and scalar transport in polydisperse gas-solid flows using particle-resolved direct numerical simulations*. PhD thesis.

[92] Tenneti, S. and Subramaniam, S. (2014). Particle-Resolved Direct Numerical Simulation for Gas-Solid Flow Model Development. *Annual Review of Fluid Mechanics*, 46:199–230.

[93] Tenneti, S., Sun, B., Garg, R., and Subramaniam, S. (2013). Role of fluid heating in dense gas-solid flow as revealed by particle-resolved direct numerical simulation. *International Journal of Heat and Mass Transfer*, 58(1-2):471–479.

[94] Theuerkauf, J., Witt, P., and Schwesig, D. (2006). Analysis of particle porosity distribution in fixed beds using the discrete element method. *Powder Technology*, 165(2):92–99.

[95] Tryggvason, G. (2017). Direct Numerical Simulations of Gas-Liquid Flows. In *International Conference on Numerical Methods for Multiphase Flows*.

[96] Tryggvason, G., Thomas, S., Lu, J., and Aboulhasanzadeh, B. (2010). Multiscale Issues in DNS of Multiphase Flows. *Acta Mathematica Scientia*, (2):551–562.

[97] Tsai, C.-M. and Yeh, Z.-M. (2012). Fast Spatial Averaging Filter. *2012 International Symposium on Computer, Consumer and Control*, (1):153–156.

[98] Uhlmann, M. (2005). An immersed boundary method with direct forcing for the simulation of particulate flows. *Journal of Computational Physics*, 209(2):448–476.

[99] Usabiaga, F. B., Pagonabarraga, I., and Delgado-Buscalioni, R. (2013). Inertial coupling for point particle fluctuating hydrodynamics. *Journal of Computational Physics*, 235:701–722.

[100] van Antwerpen, W., du Toit, C., and Rousseau, P. (2010). A review of correlations to model the packing structure and effective thermal conductivity in packed beds of mono-sized spherical particles. *Nuclear Engineering and Design*, 240(7):1803–1818.

[101] van der Hoef, M., van Sint Annaland, M., and Kuipers, J. (2005). Computational fluid dynamics for dense gas-solid fluidized beds: a multi-scale modeling strategy. *China Particuology*, 3(1-2):69–77.

[102] Van der Hoef, M.A., van Sint Annaland, M., Deen, N.G., Kuipers, J. (2008). Numerical simulation of dense gas–solid fluidized beds: a multiscale modeling strategy. *Annu. Rev. Fluid Mech.*, 40:47–70.

[103] Wachs, A., Hammouti, A., Vinay, G., and Rahmani, M. (2015). Accuracy of Finite Volume/Staggered Grid Distributed Lagrange Multiplier/Fictitious Domain simulations of particulate flows. *Computers & Fluids*, 115:154–172.

[104] Wehinger, G. D., Eppinger, T., and Kraume, M. (2015). Detailed numerical simulations of catalytic fixed-bed reactors: Heterogeneous dry reforming of methane. *Chemical Engineering Science*, 122(27):197–209.

[105] Welford, B. P. (1962). Note on a method for calculating corrected sums of squares and products. *Technometrics*, 4(3):419–420.

[106] Whitaker, S. (1972). Forced Convection Heat Transfer Correlations for Flow In Pipes, Past Flat Plates, Single. *AIChE Journal*, 18(2):361–371.

[107] Whitaker, S. (1999). *The Method of Volume Averaging*, volume 13 of *Theory and Applications of Transport in Porous Media*. Springer Netherlands, Dordrecht.

[108] Wylie, J. J., Koch, D. L., and Ladd, A. J. C. (2003). Rheology of suspensions with high particle inertia and moderate fluid inertia. *Journal of Fluid Mechanics*, 480(2003):95–118.

[109] Ye, M., Van Der Hoef, M. A., and Kuipers, J. A. M. (2004). A numerical study of fluidization behavior of Geldart A particles using a discrete particle model. *Powder Technology*, 139(2):129–139.

[110] Zaidi, A. A., Tsuji, T., and Tanaka, T. (2014). A new relation of drag force for high Stokes number monodisperse spheres by direct numerical simulation. *Advanced Powder Technology*, 25(6):1860–1871.

[111] Zanganeh, G., Pedretti, A., Zavattoni, S., Barbato, M., and Steinfeld, A. (2012). Packed-bed thermal storage for concentrated solar power – Pilot-scale demonstration and industrial-scale design. *Solar Energy*, 86(10):3084–3098.

[112] Zhang, H. W., Zhou, Q., Xing, H. L., and Muhlhaus, H. (2011). A DEM study on the effective thermal conductivity of granular assemblies. *Powder Technology*, 205(1-3):172–183.

[113] Zhang, L., Liu, K., and You, C. (2015). Fictitious domain method for fully resolved reacting gas–solid flow simulation. *Journal of Computational Physics*, 299:215–228.

[114] Zhang, W., Thompson, K. E., Reed, A. H., and Beenken, L. (2006). Relationship between packing structure and porosity in fixed beds of equilateral cylindrical particles. *Chemical Engineering Science*, 61(24):8060–8074.

[115] Zick, A. A. and Homsy, G. M. (1982). Stokes flow through periodic arrays of spheres. *Journal of Fluid Mechanics*, 115:13–26.