Developing New Methodologies to the Simulation of Emulsion Mixtures with Transport of Particles in OpenFOAM

by

Mingze Gao

A thesis submitted in partial fulfillment of the requirements for the degree of

Master of Science

Department of Mechanical Engineering
University of Alberta

Abstract

Sand separators are commonly equipped inline in the oil and gas production to separate produced sand from the extracted fluids, usually a mixture of oil, water and gas. To ensure the equipment can satisfy the work conditions, Computational Fluid Dynamics (CFD) is widely used. However, the presence of water-oil emulsions can significantly affect the flow regime, mainly because of its viscosity. The emulsion viscosity tends to increase dramatically (10 to 100 times) when the water cut is before the inversion point and decreases sharply after the inversion point. This phenomenon cannot be correctly captured with current multiphase mixture models. Also, the CFD simulation results of the sand separation cases can have a large discrepancy in terms of the separation efficiency of up to 100%, compared to experimental results. As a result, the overall objective of this project is to develop a multiphase, incompressible CFD-DEM solver that can accurately simulate the flow regime of water-oil emulsions with the transport of sand. To achieve this goal, two solvers, *denseParticleFoam* and *multiphaseInterFoam*, were selected from OpenFOAM for modification. Also, the entire project was split into three phases. The first phase mainly focused on modifications of the *denseParticleFoam*, which included converting the governing equations to the strong conservation form and improving its efficiency. In the second phase, a new viscosity model was created based on experimental results, and the new viscosity equation was implemented in the Volume of Fluid method solver of OpenFOAM (*multiphaseInterFoam*). In addition, the modified Volume of Fluid method was validated to ensure it is bug-free and able to return the correct viscosity for emulsion mixtures. Lastly, the third phase consisted in developing a new set of governing equations, so

that the modified *denseParticleFoam* from phase one can be coupled with the Volume of Fluid method from *multiphaseInterFoam*. For this thesis, the first, second phase, and the theory part of the third phase have been accomplished.

Preface

This thesis is an original work by Mingze Gao. The new viscosity model was created based on the experimental data obtained in Dr. Sean Sanders' lab. The simulations were numerically compared with the results from Fernandes et al. [1].

"Life is like a box of chocolates, you never know what you're going to get." –Forrest Gump.

To my mother, Guihong Wang, and my father Ganbo Gao, who always give me all their supports.

Acknowledgments

I would like to gratefully thank my supervisor Dr. Carlos Lange, for his excellent guidance, encouragement and valuable recommendations during my Master's degree program.

I would also like to express my gratitude to Mitacs Acceleration and Sand Separation Technologies Inc. for their financial support.

I would like to give special thanks my group member, Stanley John. He helped me a lot in learning and coding in C++ language.

I would like to thank my group members, Seyedehfatemeh Razavi, Jean Luc Olsen. We learned from each other.

I would like to offer my thanks to Mechanical Engineering department staff, particularly Gail Dowler for her high discipline and enormous support during my journey.

Table of Contents

List of Tables

List of Figures

List of Symbols

Latin

$\bar{H_P}$	Average height of all particles
$\bar{r_1}$	Refinement ratio of the coarse mesh and medium mesh
$\bar{r_2}$	Refinement ratio of the fine mesh and medium mesh
$\bar{r}$	Average refinement ratio
Δp_f	Laplace pressure
δ_N	Small value defined in OpenFOAM
$\dot{V}_{flow}$	Total volume flow rate being injected into the domain
d_{32}	Sauter mean diameter
Re	Reynolds number
$\vec{F}^c$	Contact forces that act on a specific particle
$\vec{F}^f$	Fluid forces that act on a specific particle
$\vec{F}^g$	Gravitational force
$\vec{F}^{nc}$	Non-contact forces
$\vec{f}_\sigma$	Surface tension force
$\vec{F}_\tau$	Viscous force
$\vec{F}_d$	Drag force
$\vec{f}_P$	Particle forces that act on the fluid

$\vec{F}_{\nabla p}$ Pressure gradient force

$\vec{F}_{Ml}$ Magnus lift force

$\vec{F}_{vm}$ Virtual mass force

$\vec{g}$ Gravitational acceleration

$\vec{h}$ Distance to the reference level

$\vec{M}$ Momentum

$\vec{Q}_p$ One of the source terms for Volume of Fluid method that is multiplied with α_c

$\vec{Q}_u$ One of the source terms for Volume of Fluid method that is not multiplied with α_c

$\vec{S}_P$ Resultant particle forces

$\vec{U}^s$ Superficial velocity

$\vec{U}_c$ Velocity of the continuous phase

$\vec{U}_P$ Velocity of the particles

^{n} Represents the current time step value

$_{i,j,k}$ Count index

A_P Cross section area of a particle

C_k Richardson coefficient

C_m Coefficient of the transient term used in MULES method

D Depth of the domain

D_P Diameter of the pipe

d_P Diameter of particles

E Elastic Constant (Young's Modulus)

e_n Coefficient of normal restitution for particle-particle interactions

$e_{n,w}$ Coefficient of normal restitution for particle-particle interactions

f_s Distance to the reference level

H Height of the domain

h_i Height of each individual particle

I Momentum of inertia of a specific particle

k Spring stiffness used in the spring dashpot model

k_{tur} Turbulent kinematic energy

m Mass of a particle

N_P Total number of particles

$N_{\mathrm{Re},P}$ Reynolds number that is calculated by using the Sauter mean diameter

N_c Number of nodes of the coarse mesh

N_f Number of nodes of the fine mesh

N_m Number of nodes of the medium mesh

p Static pressure

p_{rgh} Summation of the Static pressure and hydrostatic pressure

R_1, R_2 Radius of curvature in each of the axes that are parallel to the surface

s Ratio of the viscosity of the dispersed phase and continuous phase

T Transpose operator

t Time

t_r Residence time of the flow

V_P Volume of a particle

V_{cell} Volume of a cell in the computational domain

V_{geo} Total volume of the geometry

W Width of the domain

Greek

α_c Volume fraction of the continuous phase in the particle-fluid system

α_d Volume fraction of the dispersed phase in the particle-fluid system

β_{Pf} Inter-phase momentum exchange coefficient

χ Compressibility of a mixture

δ Overlaps when two particles collide

$\dot{\gamma}$ Shear rate

ϵ_o Order of the error

η Packing fraction

γ Damping coefficient used in the spring dashpot model

κ Curvature

μ_c Effective dynamic viscosity of the continuous phase

μ_e Effective dynamic viscosity of the emulsion

μ_o Dynamic viscosity of oil

μ_w Dynamic viscosity of water

ν_c Kinematic viscosity of the continuous phase

ν_e Effective kinematic viscosity of the emulsion

$\nu_{c,i}$ Kinematic viscosity of a specific fluid

ν_{tur} Poisson's ratio

ω_{tur} Specific rate of dissipation

Φ_c The summation of the convective and the compression terms of the VOF equation

ϕ_c Volume fraction of the continuous phase in the fluid-fluid system

ϕ_d Volume fraction of the dispersed phase in the fluid-fluid system

ϕ_w Absolute volume fraction of water

$\phi_{d,\max}$ The maximum volume fraction of the dispersed phase

ϕ_{ne} Volume fraction of all non emulsion fluids

ϕ_{wr} Relative volume fraction of water

ψ Packing factor

ρ_c Effective density of the continuous phase

ρ_o Density of oil

ρ_P Density of particles

ρ_w Density of water

$\rho_{c,i}$ Density of a specific fluid

σ Surface tension

$\sigma_{w,o}$ Surface tension between water and oil

τ Shear stress

τ_{PP} Coefficient of friction between particles and particles

τ_{PW} Coefficient of friction between particles and walls

τ_{st} Shear rate tensor

$\vec{\omega}$ Angular velocity

$\vec{I}$ Kronecker delta

Abbreviations & Acronyms

AH The averaged height at each time ste.

CLSVOF Coupled Level Set and Volume of Fluid.

CSF Continuum surface force.

DoF Degree of Freedom.

EE Eulerian-Eulerian.

EL Eulerian-Lagrangian.

LS Level Set.

MAH The average value of AH.

MPD The average value of MPD.

RHS Right-hand side.

SST Sand Separation Technology Inc..

VOF Volume of Fluid.

VPD The maximum pressure drop at each time step.

Chapter 1

General Introduction

This chapter is an overview of the problems and objectives of this project, and the literature review will be given in chapter 2. In chapter 3, the new methodologies that are dedicated to this project will be explained. Also, the data analysis and results will be shown in chapter 4. Moreover, the conclusion and summary of this project will be given in chapter 5.

1.1 Introduction

Nowadays, simulating fluid-particle interactions has become a more and more interesting topic, which can be applied to various kinds of real problems, such as predicting the sand filtration efficiency of a filter, riverbed erosion, fluidized bed. These applications have significant meaning to the oil industries in Alberta because most crude oil and gas wells are located in sand and shale formation, and the crude oils extracted from the wells are usually heavily loaded with solids, such as sand, clay, and silt. Transporting solids in pipelines can cause not only serious risks to the surface infrastructure but also economic loss to the companies. Collisions between particles and pipes would accelerate the wear of pipes and cause the ground equipment to have a shortened life span. Therefore, it is essential to add inline sand separation equipment to protect the surface infrastructure to the largest extent. From the 2019-2020 Alberta Energy Report [2], 1,755 new crude oil wells were drilled in Alberta. Of

these, it is estimated that approximately 85% of these wells require surface-level sand separation equipment or desanders.

Sand Separation Technology (SST) is a sand separator manufacturing company located in Grande Prairie, Alberta. They use both numerical methods and experiments to ensure their products can properly function under different kinds of operating conditions. In terms of the numerical method, they are using a one-way coupled Computational Fluid Dynamics (CFD) and Discrete Element Method (DEM) solver to evaluate their equipment performance. Also, the simulation results are calibrated with experiments to ensure the results are reliable and have errors in an acceptable range. In the SST indoor lab, two different 3D printed 18" models of their sand separators are connected to a scaled flow loop, which is shown in figure 1.1. During the experiments, the fluid used in the experiments is premixed in the fluid reservoir and then pumped into the separators. After reaching a steady-state, sand particles are introduced to the flow loop via a bypass. It is observed in the lab that most sands will stay in the separator, and only a small amount will follow the flow towards the outlet and get collected in sandbags. In this way, the sand separation efficiency of the separators can be measured and used for simulation validations.

Figure 1.1: Overview of test flow loop at Sand Separation Technologies.

The one-way coupled solver currently used by SST consists of two parts, the CFD

solver, *interFoam* or *multiphaseInterFoam*, which are two multi-fluid solvers in OpenFOAM (Open Field Operation And Manipulation), and the DEM solver is chosen from LIGGGHTS (LAMMPS Improved for General Granular and Granular Heat Transfer Simulations). The simulations are performed under various kinds of conditions and with different fluids in order to replicate the real situations in oil fields as close as possible. Therefore, the fluids used in simulations are usually present as two (water-air) or three (water-air-oil) fluid mixtures. It can be seen from figure 1.2, that the simulation results for the two phases case can reach a good agreement with experiments, although the relative errors are approximately 10-20% for both designs. However, there is a significant difference for the three phases trial, and the relative errors are all above 100%. This shows that, if the accuracy of the CFD-DEM solver can be improved, the design of the separators can be improved before making any actual models, which could lead the design process to be more efficient and economical. Hence, SST is urgently looking for a new numerical solver that can help them to get a more accurate prediction of their product performance.

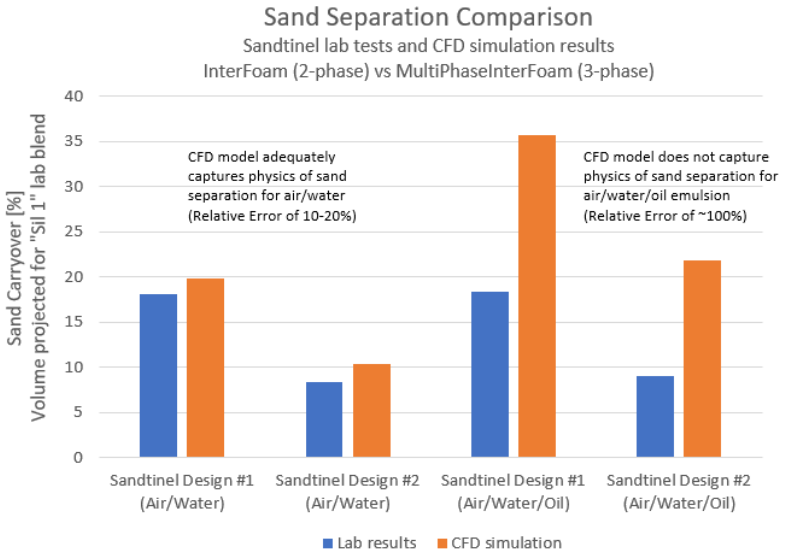


Figure 1.2: Simulations and experiments results for sand transportation. (courtesy of Sand Separation Technologies)

There are two major problems in the current solvers that may result in a large dis-

crepancy in the results compared to the experiments. First, the mixture of water and oil, also known as W/O emulsion, has a significantly different rheological behaviour from either water or oil. Detailed descriptions will be demonstrated in the following paragraphs. Second, the one-way coupled solver might not be suitable for simulating fluid-particle interactions. For such case, the particles are not only affected by the presence of fluids but also influenced by the collisions with neighbouring particles and walls. Also, the fluids are impacted by the particles, especially the interface between two fluids. To predict the particle movements in fluids, a two-way coupled CFD-DEM solver should be a better option, where there are exchanges of information between the CFD and the DEM solvers.

To overcome the first problem, it is essential to have a good understanding of emulsions. An emulsion is when one immiscible fluid is dispersed as droplets that are stably suspended in another immiscible fluid and forms a kinetically stable but thermodynamically unsteady mixture [3]. The viscosity of W/O emulsions can be strongly affected by many factors, such as volume fraction, temperature, shear rate, droplet size, and interfacial tension [4]. Among all these factors, emulsion viscosity is most sensitive to volume fraction as a result of the non-Newtonian behaviour caused by droplet "crowding" or structural viscosity of the emulsions [5].

It has been shown that emulsification can significantly increase the effective viscosity of the mixture (10 to 100 times) [4]. Furthermore, Kokal has noted that the W/O emulsion exhibits Newtonian fluid behaviour at low water cut[1] [5, 6]. In this region, the emulsion viscosity stays as a constant value for all shear rates, and the slope of measured viscosity curves is zero. After the low water cut region, produced oilfield emulsions can be classified as shear thinning or pseudoplastic fluids, which means as the shear rate increases, their viscosity decreases. This is also the region in which the W/O emulsion viscosity tends to increase remarkably as the water cut increases.

[1]Water cut in this thesis refers to the volume fraction of the water, which is equal to the ratio of the volume of water and the total volume.

After a certain water cut, which is the so-called inversion point, the W/O emulsion switches to the O/W emulsion. In this region, water starts dominating the rheological behaviour of emulsions, and the viscosity decreases as the water cut increases [5]. One example of the light crude oil emulsion viscosity measurements with an increase in water cut is shown in figure 1.3 [7]. It is obvious that the emulsion viscosity increases dramatically with an increase in water cut initially and then decreases sharply after the inversion point and finally reaches the water viscosity. Also, the effective viscosity of a W/O emulsion can be several times higher than the viscosity of oil alone, depending on the volume fraction of water in the emulsion.

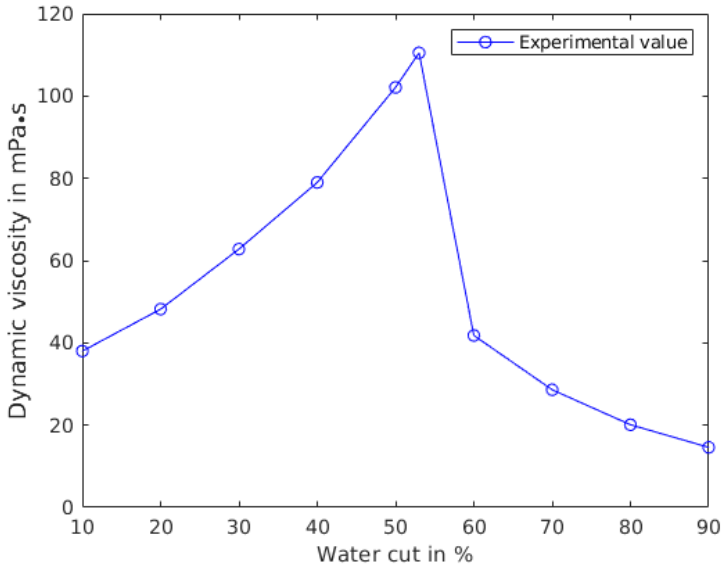


Figure 1.3: Measured viscosity values for the Dagang light crude oil emulsion as the water content increased, based on data from [7].

As a result, current multi-fluid models in most CFD software cannot correctly calculate the effective viscosity of an emulsion. When a control volume contains two or more phase fluids, the resulting effective kinematic viscosity is usually estimated by a weighted average combination of the individual viscosity values based on the volume fraction of each component. To address this problem, implementing a new viscosity function that can represent the entire range of water cut emulsion into a

dedicated solver will be the best option. This would require an open-source CFD software so that the new emulsion viscosity model can be successfully implemented.

In terms of the two-way coupled solvers, few studies have been done previously. For example, Li et. al coupled the *twoPhaseEulerFoam* with *DPMFoam* in OpenFOAM [8], and Lu et. al and Sun et. al coupled a CFD solver, *interFoam*, from OpenFOAM and a DEM solver from LIGGGHTS [9, 10]. In their studies, several three-dimensional cases are presented, such as particle sedimentation, dam break with a moving bed and particle movements in a rotating drum. These cases reached good agreement with either theoretical prediction or experimental results, which demonstrates the capability of their solvers. With those open-source resources, it is possible to create a two-way coupled multi-fluid CFD-DEM solver to satisfy the project needs, and the most important step is to understand and implement the governing equations. In addition, there are two limitations associated with the current existing two-way coupled CFD-DEM solvers in OpenFOAM. First, the maximum number of fluids is limited. Second, LIGGGHTS has no longer been an open-source software for many years. To tackle these problems, two strategies are proposed in the following paragraphs.

In terms of the former problem, the solver that can only run with a maximum of two fluids would not meet the requirements of the current project. Hence, a multi-fluid model that can solve more than two fluid mixtures should be used. In this way, the capability of the modified solver can be extended to solving N number of fluid phases with the transport of particles. To address the latter concern, the DEM solver obtained from LIGGGHTS can be replaced by other open-source software, such as a DEM solver based on OpenFOAM, which allows users to modify and create new solvers based on their own requirements.

To take all concerns into account, the modifications of this project will be performed in OpenFOAM-dev, which was released on 8th May, 2020. One of the CFD-DEM solvers in OpenFOAM is called *denseParticleFoam*, which is a single-phase transient solver for the coupled transport of particle clouds, including the effect of the volume

fraction of particles on the continuous phase. Moreover, Fernandes et al. [1] have validated this solver and stated *denseParticleFoam* is valid to solve particle-particle and particle-fluid interactions. In addition, to implement the new viscosity equation and ensure the solver is able to solve N number phases of fluids, the multi-fluid model implemented in *multiphaseInterFoam* is selected. Detailed implementations and modifications will be discussed in chapter 3.

1.2 Research Objectives

The overall objective of this project is to extend *denseParticleFoam* to a two-way coupled multi-fluid CFD-DEM solver that can accurately predict the rheological behaviour of emulsions with the transport of particles. In order to accomplish this objective, it requires modifications on *denseParticleFoam*, creating a viscosity model that can represent all range of water cut emulsions, and coupling the modified *denseParticleFoam* with a modified multi-fluid model. In addition, to ensure that major modifications made on the software are correct, this project is divided into three phases.

In the first phase, the governing equations of *denseParticleFoam* are reviewed and modified, which is an essential step to couple *denseParticleFoam* with the multi-fluid model. Also, the modified solver will be numerically validated with the experimental data from Fernandes et al. [1]. The second phase is to improve the accuracy of calculating the viscosity of an emulsion. Hence, an advanced viscosity equation that can calculate the emulsion viscosity at different water cuts and shear rates will be created based on the measured data. Also, this new viscosity model will be implemented in the VOF method of *multipahseInterFoam*, and the different rheological behaviour caused by the viscosity will be studied. Moreover, the last objective is to investigate how the multi-fluid model from the Eulerian perspective and the solid model from the Lagrangian perspective can be coupled together. To achieve this goal, the theory of a two-way coupled multi-fluid CFD-DEM solver will be developed, especially its

governing equations.

Note that the governing equations derived from phase three will not be implemented in OpenFOAM due to constrains in time and resources. Therefore, implementing and validating the phase three solver will be left for future works.

1.3 Thesis Layout

The work in this thesis is divided into five chapters. Chapter one (the current chapter) illustrates how this project is proposed by SST and the large discrepancy between the numerical results and experiments that motivates the project. Also, a short description of the rheological behaviour of W/O emulsions is presented. Moreover, the main disadvantages of current existing solvers are demonstrated, which gives a clear view of the objective of this project.

Chapter two is an overview of the numerical models that have been developed to predict particle movements, multiple fluid mixtures and emulsion viscosity.

Chapter 3 focuses on the development of new models to simulate the W/O emulsions with the transport of particles. As mentioned in this chapter, the emulsion viscosity cannot be correctly represented with the existing viscosity models that are implemented in CFD software at the current stage. Therefore, a new viscosity model is developed and implemented in OpenFOAM based on the experimental results.

Chapter 4 shows validations of the implemented new models. The validation mainly focuses on the first two phases, which include the new viscosity model and the modified solver *DPConFoam*. In terms of the third phase, only the theory has been developed for the couple multi-fluid CFD-DEM solver. Also, the simulation and experimental results will be discussed here.

Finally, chapter 5 is a conclusion of the study of this project, including a few points of recommendation that are proposed for future works.

Chapter 2

Review of Methodologies to Couple CFD-DEM Solvers for Non-Newtonian Mixture Flows

2.1 Introduction

This chapter is a comprehensive overview of the existing methods that may be applicable to this project, and it is divided into four sections. First, the mathematical models that can be applied to evaluate the water-oil (W-O) emulsions will be reviewed, which include the current method implemented in OpenFOAM. Second, the governing equations of the Volume of Fluid (VOF) method will be discussed. Finally, the DEM model and CFD models are reviewed, and the approaches to couple CFD and DEM will also be illustrated.

2.2 Review of Existing Emulsion Viscosity Models

In OpenFOAM, the effective density is calculated based on the weighted average volume fraction of each fluid. However, this method cannot reflect the real change of the emulsion viscosity with respect to water cut, as it can be seen in figure 2.1. The emulsion viscosity tends to adopt higher values compared with the simple weighted average between two values, especially in the region near the inversion point. In this specific case, the maximum measured viscosity value is more than six times larger

than the one obtained from the OpenFOAM method. In addition, the functional behaviour of the viscosity change on each side of the inversion point is completely different from a linear function. Therefore, more advanced viscosity models developed by previous researchers will be reviewed in this section[1], and one will be selected for later modification and implementation in chapter 3.

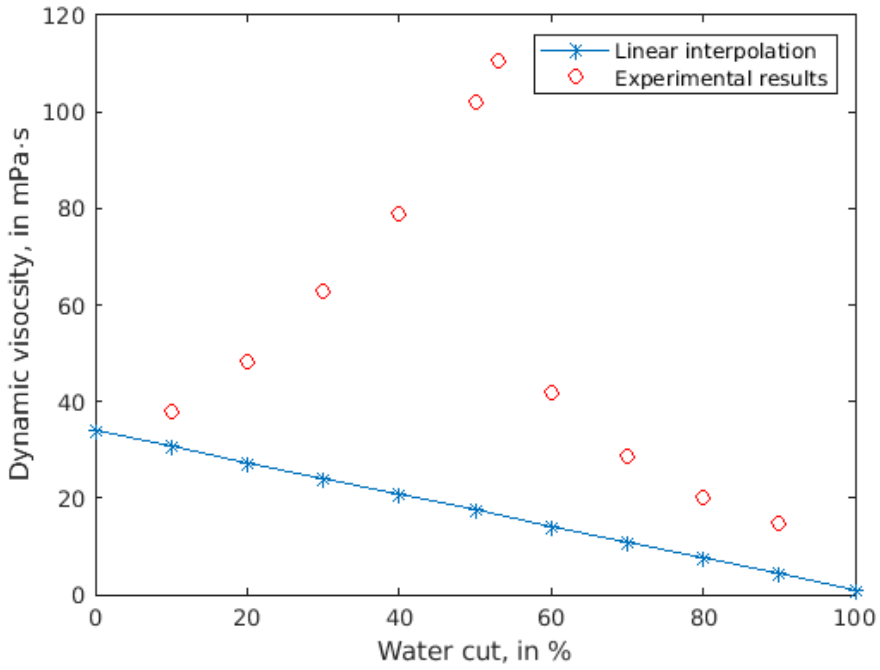


Figure 2.1: A comparison between the viscosity obtained by weighted averaging (linear interpolation) and by experimental values [7] at different water cuts.

Water-oil (W-O) emulsions can be categorized into three groups based on the water cut [5]:

- Water-in-oil (W/O) emulsion: this type of emulsion contains the water droplets as the dispersed phase in a continuous oil phase. This is the most common type of emulsion in industry, and many correlations have been found for this region.

[1]Note that the dispersed phase and continuous phase in this section mainly refer to two fluids. For example, when water droplets are dispersed into oil, water and oil will be in the dispersed phase and the continuous phase, respectively. To distinguish the dispersed phase (solid particles) and the continuous phase (fluid) that will be used in later sections, different sets of symbols are used. The volume fraction will be represented by ϕ in this section and α in other sections.

- Oil-in-water (O/W) emulsion: this type of emulsion consists of oil droplets in a continuous water phase, and it is usually referred to as a "reverse" emulsion.

- Multiple or complex emulsions: this type of emulsion is more complex. For example, a water-in-oil-in-water (W/O/W) emulsion consist of tiny water droplets suspended in a larger oil droplet that is suspended in a continuous water phase.

The viscosity of W-O emulsions tends to reach a higher value than the viscosity of either oil or water. This phenomenon is a result of the non-Newtonian behaviour caused by the droplet "crowding" or structural viscosity of the emulsions [5]. Kokal et al. [5][6] have stated the W/O emulsion exhibits Newtonian fluid behaviour at low water cut. In this region, the emulsion viscosity stays constant for all shear rates, and the slope of measured viscosity curves is zero. After the low water cut region, the produced oilfield emulsions can be classified as shear thinning or pseudoplastic fluids, which means that, as the shear rate increases, their viscosity decreases. This is also the region where the W/O emulsion viscosity tends to increase remarkably as the water cut increases. After a specific water cut, which is the so-called inversion point, the W/O emulsion switches to an O/W emulsion, and its viscosity decreases as the water cut increases [5]. Other factors can influence the emulsion viscosity, such as viscosity and density of the fluids and average droplet size (r) [11]. From the previous research, many W-O emulsion viscosity models have been developed based on the water cut, shear rate, and temperature, which will be discussed in the following sections.

2.2.1 Water Cut Related Models

As mentioned before, the viscosity of the emulsion is highly related with water cuts. At the stage of water-in-oil emulsion, its viscosity will increase dramatically with an increase in water cut and reach the maximum value at the inversion point. After the inversion point, the continuous phase will be switched from oil to water, and a

sudden drop in viscosity may cause discontinuities for the viscosity equation. When the mixture becomes an O-W emulsion, the viscosity value will decrease significantly as the water cut increases. The following correlations have been developed for the W/O emulsions.

In order to simplify the viscosity equation, many researchers use relative viscosity in their models, which is defined as:

$$\mu_r = \frac{\mu}{\mu_c} \tag{2.1}$$

In 1906, Einstein studied the viscosity of the fluid that consists of small spherical particles as a problem in theoretical hydrodynamics [12]. And in 1911, Einstein proposed a positive linear relationship between the relative viscosity of the emulsion and the volume fraction of the dispersed phase (ϕ_d). However, this correlation, as shown in equation 2.2, is only valid for a dilute suspension system [11]:

$$\mu_r = 1 + 2.5\phi_d \tag{2.2}$$

Later, many researchers tried to extend Einstein's work to liquids containing small droplets of another liquid in suspension. However, when liquid droplets replace solid particles, a complete theory can barely be accomplished. This is mainly caused by unknown boundary conditions between the continuous phase and dispersed phase and partly because that various types of forces can lead to droplet deformation, such as surface tension and viscous forces [12]. In 1932, Taylor [12] extended Einstein's equation to include the case of liquid drops by considering the effect of both dispersed phase and continuous phase:

$$\mu_r = 1 + \left[2.5\left(\frac{s+0.4}{s+1}\right)\right]\phi_d \tag{2.3}$$

where s is defined as $\frac{\mu_d}{\mu_c}$, which is the ratio of the viscosity of the dispersed phase and continuous phase. However, this equation is only valid when the droplets of the dispersed phase are very small to remain nearly spherical, which requires the surface

tension between droplets to be sufficiently large. It is essential to notice that when the dispersed phase is solid spherical particles, μ_d tends to be infinitely large so that equation 2.3 will become Einstein's equation, equation 2.2, by taking the L'Hopital's rule.

In 1950, Krieger and Dougherty [13] developed a new empirical equation, equation 2.4, based on Mooney's work, equation 2.5, to solve intense concentrations of dispersed phase systems. This further development makes the function an excellent fit for their viscosity data over the entire range of water cuts. Plus, they introduced the packing fraction 'ψ', which represents the maximum concentration of the dispersed phase. When the system reaches a close-packed system, the viscosity value tends to approach infinite.

$$\mu_r = \left(1 - \frac{\phi_d}{\phi_{d,\max}}\right)^{-[\psi]\phi_{d,\max}} \tag{2.4}$$

where $[\psi]$ is the intrinsic viscosity and $\phi_{d,\max}$ is the maximum volume fraction of the dispersed phase, which is defined by

$$[\psi] = \frac{\mu_d}{\mu_c} - 1 \tag{2.5}$$

However, equation 2.4 can be easily applied because there is no simple way to measure the packing factor for emulsions [14].

In the same year, Richardson [15] has developed another equation for emulsions based on compressibility. He noticed the viscosity curve of the benzene-in-water emulsion increased exponentially with respect to the volume fraction of water. Therefore, he introduced a variable 'χ' to represent the mixture compressibility, which depends on the relative compressibility of the two fluid phases. Finally, the correlation proposed by Richardson can be written as:

$$\mu_r = \exp(\chi\phi_d) \tag{2.6}$$

This equation showed good agreement with the data obtained by those researchers, who measured the apparent viscosity of such emulsions as albumin and blood. Also,

he pointed out that Broughton and Squires obtained a linear relationship based on the $\log(\mu_r)$ against ϕ_w plot [15]. A linear relationship was observed between these two criteria. And the correlation equation can be written as:

$$\log(\mu_r) = C_k(\phi_d + \phi_0) \tag{2.7}$$

Herein, C_k is the Richardson coefficient that depends on the emulsifying agent, and ϕ_0 is the ratio of the y-intercept found in the $\log(\mu_r)$ against ϕ_w plot to C_k.

2.2.2 Shear Rate Related Models

To describe the viscosity change to shear rate, Hasan et al. [16] tested three commonly used viscosity models (Power-law model, Bingham model, and Casson model) for heavy oils, based on experimental values. Among all these models, the Power-law model fits the experimental data best with a regression correlation coefficient, R^2, of 0.994. Meriem-Benziane et al. [17] tested the Ostwald de Waele model and the Herschel-Bulkley model for W-O emulsions at different water cuts (30%, 50%, and 70%).

- Ostwald de Waele model (Power law model):

$$\mu_e = \xi\dot{\gamma}^{z-1} \tag{2.8}$$

- Herschel-Bulkley model:

$$\mu_e = \frac{\tau_0}{\dot{\gamma}} + \xi\dot{\gamma}^{z-1} \tag{2.9}$$

where μ_e is the dynamic viscosity of the emulsion, and ξ and z are the consistency index and flow index, respectively. These parameters need to be obtained from curve fitting. τ_0 and $\dot{\gamma}$ are the yield stress and the shear rate, respectively.

Based on the R^2, Meriem-Benziane et al. concluded that the Power-law model is more suitable for high water cut emulsions, like the 70% case, whereas the Herschel-Bulkley model can be applied to cases with low water cut, 30% and 50% in their

study [17]. This is also validated by Ariffin et al. [18], who curve-fitted the measured emulsion viscosity at low water cut (20%, 30%, and 40%) with the two models above. In their study, the Herschel-Bulkley model showed a better correlation. Meanwhile, the Power-law model shows a slight offset to the measured points, but the correlation coefficients are all above 0.94, which means the model can still be used when the yield stress of the emulsion cannot be determined.

2.2.3 Temperature Related Models

Temperature is another factor that can significantly affect the W-O emulsion viscosity. In 1995, Ronningsen et al. [6] proposed a linear correlation function to link the temperature (T), continuous volume fraction (ϕ_c) and the relative emulsion viscosity (μ_r), which is defined as:

$$\ln(\mu_{\mathrm{r}}) = a_1 + a_2 T + a_3 \phi_w + a_4 T \phi_w \tag{2.10}$$

This work is an extension of the equation 2.6 based on the analysis of experimental results adopted at different temperatures and water cuts. However, since the shear rate is not taken into account in this case, $a_1 \sim a_4$ should have multiple definitions to ensure the correlation coefficients are reasonable for the entire range of shear rate.

Another model recommended by ASTM (2001) is modified from Walther's model, which forms a connection between the kinematic viscosity (ν_e) and temperature [4]. This method is commonly used in the petroleum industry, and the correlations can be found in [4].

In 2015, Li et al. [11] analyzed their experiments based on Richardson's equation (equation 2.6) and the Broughton and Squires' equation (equation 2.7). They stated that, when the water cut is lower than 20%, the relative viscosity is exponentially related to water concentration only. And when the water cut is greater than 20%, the effect of the water cut, temperature, and shear rate need to be taken into account.

Their modified equations are defined as the following:

$$\ln(\mu_r) = \begin{cases} b_1 + b_2\phi_w & (\phi_w < 20\%) \\ b_3 + b_4 T + b_5\phi_w + b_6 T\phi_w + b_7 \ln\dot{\gamma} + b_8 T \ln\dot{\gamma} & \\ \quad + b_9\phi_w \ln\dot{\gamma} + b_{10} T\, \phi_w \ln\dot{\gamma} & (\phi_w > 20\%) \end{cases} \tag{2.11}$$

where $b_1 \sim b_{10}$ are the coefficients gained from curve-fitting emulsion viscosity. Even though this step function fits their experimental results well, it cannot be applied to other W-O emulsions.

There are also other empirical models that have been developed, but they are not straightforward to be implemented in a solver as a result of either the variables being very hard to be defined or the form of the equation being overly complicated, and its solution is not trivial. Therefore, they are not considered candidate models for this project. For anyone who is interested in this topic, those models can be found in [19][20][21][22][14]. In this project, two factors (shear rate and water cut) were considered when performing the measurement of emulsion viscosity. More details of the development of the new viscosity model will be provided in chapter 3, and the analysis of the experimental results and implementation of the new model will be demonstrated in chapter 4. Also, when implementing the new viscosity model, a control of the new viscosity model will be created for users to determine if there is emulsion in the system.

2.3 Multiphase Mixture Models

Another objective of this project is to ensure that the complex rheological behaviour of the W/O emulsion can be correctly predicted. Several multiphase mixture models have been developed in the last few decades, such as the Standard Level Set (SLS) method [23], Volume-of-Fluid (VOF) method [24], the Coupled LS-VOF (CLSVOF) method [25]. For the LS method, the interface is captured implicitly, and it is set to zero level set. The outside and the inside of the interface are set to be positive and negative, respectively. When the interface moves, the zero level set will change

accordingly [26]. This method offers more accurate results in representing the interface with higher-order schemes, hybrid, and adaptive techniques, but the mass conservation is not guaranteed [27][28]. The VOF method does not allow the fluid phases to be interpenetrating and requires an additional transport equation to solve the volume fraction of the continuous phase [29]. With this additional equation, each control volume will obtain an indicator. When the indicator is zero or one, the control volume is filled with one of the continuous phases. Additionally, when a control volume contains a mixture of multiple fluids, the indicator is between zero and one. A comprehensive comparison between different mixture models can be found in [27].

The multiphase mixture can also be solved by utilizing Euler-Euler models. The fluid phases are treated separately, and each of them is governed by a set of governing equations [29]. For this reason, this method allows each fluid to obtain its own velocity field. In addition, the fluid phases are coupled with a shared pressure field and interphase exchange coefficients [29].

Among all different methods, the VOF method is considered to be the most suitable for this project due to the following reasons [30]:

- It ensures mass conservation by its nature.
- It is robust and easy to implement in CFD software.
- It can handle the interface connections at different size scales.
- It updates the phase volume fraction in a computational cell based on neighbouring cells.
- It can solve a mixture with more than two fluids and create multiple free surfaces, e.g. liquid-air, for each fluid-fluid combination.
- It can be extended to solve three-dimensional, unstructured meshes.

- Its governing equations can be solved in the Finite Volume Method (FVM) framework.

Figure 2.2: An example of a not fully separated 80% water cut emulsion (top view). (courtesy of Sand Separation Technologies)

Also, in the case where the water and oil are not fully separated, as shown in figure 2.2, there is a transition region that may present as a W/O emulsion or an O/W emulsion or a W/O/W emulsion. By using the VOF method, the surface tension force at the interface can be taken into account and effective viscosity of the flow can be calculated. These are not able to be obtained from the two-fluid model.

For these reasons, the VOF method is widely used in commercial CFD software to simulate the free surfaces and is selected as the multi-fluid model for this project. In OpenFOAM, a modified VOF method based on Hirt and Nichols' work [24] is implemented. A comprehensive overview of the VOF method will be given in the following section.

2.3.1 Review of the VOF Method

When multiple fluids are present in a system, the location of the free surface of two immiscible fluids needs to be determined. This can be resolved by employing the VOF method proposed by Hirt and Nichols [24], and an additional transport equation is required:

$$\frac{\partial \alpha_i}{\partial t} + \nabla \cdot \left(\alpha_i \vec{U}_c \right) = 0 \tag{2.12}$$

In OpenFOAM, the governing equations for a system that includes N incompressible fluids, fluid i is defined by:

- VOF equation:

$$\frac{\partial \alpha_i}{\partial t} + \nabla \cdot \left(\alpha_i \vec{U}_c \right) + \sum_{j=1}^{n} \nabla \cdot \left(\alpha_i \alpha_j \vec{U}_r \right) = 0 \tag{2.13}$$

- Continuity equation:

$$\nabla \cdot \vec{U}_c = 0 \tag{2.14}$$

- Momentum equation:

$$\frac{\partial \rho_c \vec{U}_c}{\partial t} + \nabla \cdot (\rho_c \vec{U}_c \vec{U}_c) = -\nabla p_{rgh} - \vec{g} \cdot \vec{h} \nabla \rho_c + \nabla \cdot (\rho_c \tau_{st}) + \vec{f}_\sigma \tag{2.15}$$

- τ_{st}:

$$\tau_{st} = \nu_c \left(2\mathrm{D} - \frac{2}{3} (\nabla \cdot U) \vec{I} \right) \tag{2.16}$$

$$\mathrm{D} = \frac{1}{2} \left[\nabla U + (\nabla U)^T \right] \tag{2.17}$$

where, $\vec{U}_r$ is the relative velocity between two fluids, τ_{st} is the shear stress tensor, and D is the strain rate [31]. The discretization of this non-linear term can be found in [30]. Also, the last term in equation 2.13 is used to ensure the sharp region on the free surface can be properly represented [32]. As stated by Berberović, the most critical issue of the VOF method proposed by Hirt and Nichols is that the free surface is unable to have a sharp change while ensuring the volume fraction of the fluid is

bounded and conserved [30]. This implies the resolution of the sharp region on the free surface is extremely dependent on the mesh size. Especially in the cases where the fluid density ratio is high, small errors in the volume fraction can result in large errors not only for the continuous phase properties but also for the final simulation results, as the errors are accumulating at every time step [30].

To accurately simulate the interface between fluids, Berberović also noted that numerical diffusion is the most essential factor that leads the interface to be artificially smeared. Therefore, the convective term needs to be properly discretized. To reduce the effect of the numerical diffusion and remedy the smearing of steep gradients, Weller proposed the additional convective term, which is the third term in equation 2.13.

Note that the summation of all α_i obtained from equation 2.13 should be unity. Also, the pressure term used in equation 2.15 is a modified pressure (p_{rgh}) that is equal to the summation of the static pressure and hydrostatic pressure. The benefit of using this pressure term is that each fluid phase can have a different normal component of the pressure gradient when the phases are separating at a stationary non-vertical solid wall with no-slip boundary condition [30]. This is mainly due to the hydrostatic pressure, which is proportional to the density of the fluid.

In addition, $\vec{f}_\sigma$ is the surface tension force at the free surface, which is caused by the pressure difference on the two sides of the interface. When the free surface is at a stable condition, the pressure difference can be calculated by using the Young-Laplace equation [33]:

$$\Delta p_f = \sigma \left(\frac{1}{R_1} + \frac{1}{R_2} \right) \tag{2.18}$$

where σ is the surface tension between two fluids. For example, the surface tension between water and air is 72 $\mathrm{mN{\cdot}m^{-1}}$. R_1 and R_2 are radius of curvature in each of the axes that are parallel to the surface. In OpenFOAM, the curvature is defined by:

$$\kappa = -\nabla \cdot \left(\frac{\nabla \alpha_i}{|\nabla \alpha_i|} \right) \tag{2.19}$$

Based on the continuum surface force (CSF) model developed by Brackbill et al. [34], the volumetric surface tension force is evaluated by:

$$\vec{f}_{\sigma} = \sigma \kappa \nabla \alpha_i f(x) \tag{2.20}$$

This equation is only valid for the cases with constant surface tension. A varying surface tension, additional shear stress caused by surface tension gradient needs to be taken into account [30]. Also, Brackbill et al. [34] recommended that the indicator function ($f(x)$) in the equation above should be smoothed before determining the normal direction of the free surface. For example, $f(x)$ is defined as $\rho_{f,i}/(\sum_i^N \rho_{f,i})$, which ensures the surface tension force towards regions of higher density fluids without influencing the magnitude of the force. The indicator function can also be replaced by another function to locate the free surface to the center of the cell or towards the largest phase fraction gradient [30]. In this study, this indicator function will be treated as unity, since Lafaurie et al. [35] have proved the indicator functions have only a slight influence on the surface tension force based on their experiments. In this study, the built-in surface tension force model will be directly used because the investigation of the surface tension force is out of the scope of this project.

In OpenFOAM, the equation 2.13 is solved in the MULES (Multidimensional Universal Limiter with Explicit Solution) explicit solver [32], which is an approach to ensure the solutions of the volume fraction equation are bounded by limiting the α_i values to lie in between 0 and 1 [36]. This can prevent the round-off error from accumulating after many iterations. The general form of the function to calculate the fluid volume fraction can be expressed as the following:

$$\alpha_i^n = \frac{\frac{C_m \alpha_i^{n-1}}{\delta t} + Q_u - \Phi_c}{\frac{C_m}{\delta t} - Q_p} \tag{2.21}$$

where Φ_c is the summation of the convective and the compression terms of the VOF equation and C_m is the coefficient of the transient term, which is one by default. In

addition, Q_p and Q_u are two source terms, predefined in OpenFOAM. This allows users to explicitly solve terms if it is needed. The difference between these two terms is that Q_p is multiplied by α_i^n, whereas Q_u is not.

After obtaining the volume fraction of each fluid phase, the effective density and viscosity will be updated by:

$$\rho_c = \sum_{i=1}^{N} \alpha_i \rho_{c,i} \tag{2.22}$$

$$\nu_c = \sum_{i=1}^{N} \alpha_i \nu_{c,i} \tag{2.23}$$

$$\mu_c = \sum_{i=1}^{N} \alpha_i \rho_{c,i} \nu_{c,i} \tag{2.24}$$

where μ_c, ν_c, and ρ_c are the effective dynamic viscosity, kinematic viscosity, and density of the continuous phase; $\nu_{c,i}$ and $\rho_{c,i}$ are the kinematic viscosity and density of a fluid component of the mixture.

When there are particles in the computational domain, equation 2.13 to 2.15 need to be further modified. This will be discussed in Chapter 3.

2.4 Review of the Governing Equations for a Single Fluid CFD-DEM Solvers

To achieve the objective of this project, those models that can solve the particle movements and predict the rheological behaviour of a mixture were reviewed.

The approaches to solving a particulate flow can be classified into two groups, Eulerian-Eulerian (EE) and Eulerian-Lagrangian (EL). In terms of the fluid phases, both solvers treat them as the continuous phase, which implies the fluid phases are solved based on computing the locally averaged Navier-Stokes equation [37]. The substantial difference between the two methods lies on how they solve the particle phase. In terms of the EE method, the particles are treated as a continuum, and they are solved based on the Boltzmann equation derived from the Reynolds Transport Theorem [38].

In contrast, the EL approach, proposed by Cundall et al. [39], calculates the motion of particles from a Lagrangian perspective. In this way, the particles are treated as a discrete particle phase, and they are solved by integrating Newton's second law. Also, the information of particles, such as their forces, velocities, and positions, can be tracked. For example, the Discrete Element Method (DEM) is one of the methods developed from the Lagrangian perspective. It calculates the particle movements based on the forces acting on it. These forces are usually caused by the interactions between particles, particles and fluid, and particles and walls.

In 1993, Tsuji et al. [40] coupled Cundall's DEM model with CFD and performed a 2D fluidized bed simulation. In their simulation, the interaction between particles and fluids was taken into account, and the results prove that particle movements can be accurately predicted. The advantage of performing the DEM simulations is that it can easily obtain the dynamic information of the particles, such as positions and forces that are acting on each particle, which is extremely difficult in the sense of performing experiments [41]. Depending on the size of the particles and the cells in the computational domain, the CFD-DEM method can be distinguished as resolved and unresolved. The resolved CFD-DEM method discretizes the fluid domain to the particle scale and resolves the flow around each particle [42]. For this reason, the particle size can be greater than the grid size. On the contrary, the unresolved method can only be applied to those cases where the particle size is smaller than the grid size. Even though the resolved method can provide more accurate results, it requires intensive computational power with an increased number of particles. This would limit the capability of the resolved method of solving large industry problems. Therefore, the unresolved CFD-DEM method will be applied to this project to solve the flow with the transport of particles.

2.4.1 Coupling Approaches

There are three methods that can be applied to couple CFD and DEM [43], which include:

- Zero-way coupling: The fluids and particles are computed independently and there is no interaction between the continuous phase and the dispersed phase.
- One-way coupling: The particles are affected by the fluid, but the dispersed phase influence acting on the fluids are negligible, so that it is not considered.
- Two-way coupling: The particles and fluids influence each other by exchanging information, such as momentum, velocity and forces. Also, the particle-particle interactions are considered in this case.

The final version of the solver will eventually be used to simulate sand separations, which usually involve large number of particles. This would require it to take both particle-particle interactions and particle-fluid interactions into consideration. Therefore, the two-way coupled method will be used in this project.

2.4.2 Void Fraction Calculation and Grid Size Selection

When simulating particulate flows, the most important parameter that needs to be determined is the volume fraction of the continuous phase. Thus, the particle volume and the interaction forces between particles and fluids must be projected onto the CFD mesh. In addition to the projection, the scheme must ensure mass conservation, and the produced α_i can result in a stable simulation with affordable computational cost [44]. The standard counting method assigns the entire particle to the corresponding CFD volume if its centroid is located in that cell. Marshall and Sala [45] have tested different particle counting models, which include the concentration blob, the momentum preserving approach, the conservation blob, and moment-preserving method. They stated that, if the ratio of the grid size to particle diameter is less than

three ($\frac{\Delta x}{d_P} < 3$), the particle counting method and momentum preserving approach will generate intensive noise in the concentration field. In OpenFOAM, the particle tracking algorithm developed by Macpherson et al. [46] is used. The implemented algorithm is very robust, efficient, and effective in solving particulate flow even with an unstructured, 3D arbitrary polyhedral mesh geometries [46].

2.4.3 Review of the DEM method in OpenFOAM

The movements of a particle can be classified into two groups: translational and rotational. These movements are mainly caused by the momentum exchange resulting from the interactions between a particle and other or walls and surrounding fluids. Therefore, the particle movements are affected by its immediate contact particle or fluid and by the waves far away from it. For this reason, the numerical time step of the DEM method is usually required to be smaller than a critical value [1]. For example, in a single degree of freedom (DoF) system, the critical time for a mass m that is connected to a spring with stiffness of k can be estimated as $2\sqrt{m/k}$ [39].

The DEM allows small overlaps ($< 5\% d_P$) to exist when solving collisions between two particles. When two particles collide with each other, they are contacting on a finite area instead of a single point as a result of the deformation of the particles [41]. The contact forces in this region are decomposed into normal and tangential directions with contact models. The elastic and dissipative components contained in the contact models can account for the inelastic collisions between particles. A schematic diagram that shows the forces acting on a particle can be seen in figure 2.3.

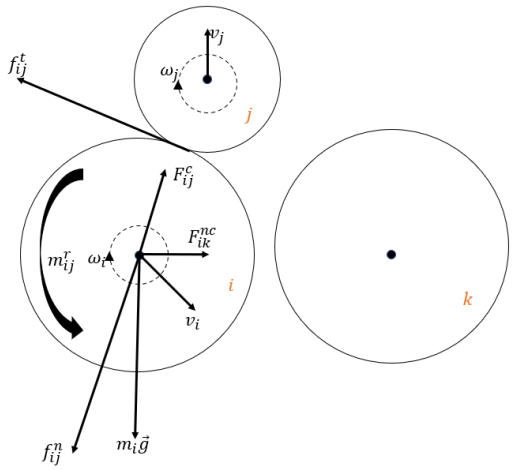


Figure 2.3: A Schematic diagram that demonstrates the forces acting on particle i from contacting particle j and non-contacting particle k.

Based on [1] and [39], the movements of each particle in the computational domain is governed by:

$$m_i \frac{\mathrm{d}\vec{U}_i^p}{\mathrm{d}\,t} = \sum_{j=1}^{n_i^c} \vec{F}_{ij}^c + \sum_{k=1}^{n_i^{nc}} \vec{F}_{ik}^{nc} + \vec{F}_i^f + \vec{F}_i^g \tag{2.25}$$

$$I_i \frac{\mathrm{d}\vec{\omega}_i}{\mathrm{d}\,t} = \sum_{j=1}^{n_i^c} \vec{M}_{ij} \tag{2.26}$$

where $\vec{U}_i^p$ and $\vec{\omega}_i$ are the translational and angular velocities of particle i, respectively. Also, $\vec{F}_{ij}^c$ and $\vec{M}_{ij}^c$ denotes the forces and momentum due to the contact of another particle j or wall to particle i. $\vec{F}_{ij}^{nc}$ represents the non-contact forces caused by particle k to i. $\vec{F}_i^f$ and $\vec{F}_i^g$ are the particle-fluid interaction forces and gravitational forces acting on particle i. Moreover, m_i and I_i are the mass and momentum of inertia of particle i.

In terms of the body force of a spherical particle, it is evaluated by:

$$\vec{F}_i^g = m_i\vec{g} = \frac{1}{6}\pi\rho_{P,i}d_{P,i}\vec{g} \tag{2.27}$$

Other contact and non-contact forces will be explained and listed in section 2.4.4.

2.4.4 Particle-Particle and Particle-Fluid Interaction Forces

As mentioned in the previous section, particle movement is affected by other particles and surrounding fluids. The contact force due to particle collisions is decomposed to normal and tangential components in the particle-particle interactions. Based on the spring-dashpot model proposed by Cundall and Strack [39], the elastic deformation and viscous dissipation can be represented by the spring and dashpot, respectively. A schematic diagram of the spring dash model with small overlaps between particles is shown in figure 2.4. In 1990, Tsuji et al. extended Cundall and Strack's model to take the three-dimensional motion of spheres into account [47]. Therefore, the Hertz Mindlin contact model will be used in this thesis.

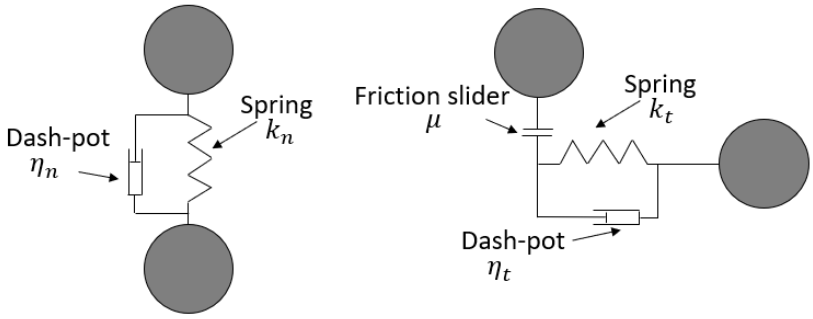


Figure 2.4: Schematic plot of the spring-dashpot model.

.

The normal and tangential contact forces between particle-particle or particle-wall[2] are formed by two parts: elastic and dissipative, which are mathematically defined as [44]:

$$\begin{aligned}\vec{f}_{ij}^{c} = k_{n,ij}||\delta_{n,ij}||^{a}\delta_{n,ij} + \gamma_{n,ij}||\delta_{n,ij}||^{b}\dot{\delta}_{n,ij} \\ + k_{t,ij}||\delta_{n,ij}||^{c}\delta_{t,ij} + \gamma_{t,ij}||\delta_{n,ij}||^{d}\dot{\delta}_{t,ij}\end{aligned} \tag{2.28}$$

where $k_{n,ij}$ and $k_{t,ij}$ are the normal and tangential stiffness coefficients, $\gamma_{n,ij}$ and $\gamma_{t,ij}$ are the normal and tangential damping coefficients, $\delta_{n,ij}$ and $\delta_{t,ij}$ are the normal

[2]When solving particle-wall interactions, the radius and mass of the wall are assumed to be infinity.

and tangential overlaps, and $\dot{\delta}_{n,ij}$ and $\dot{\delta}_{t,ij}$ are the time derivatives of the normal and tangential overlaps. Depends on the contact model, a, b, c and d are determined. For example, the four exponents are zero for the linear contact model. k, γ, δ are material properties, and they are defined based on specific problems. In the equation above, the first two terms represent the normal elastic and dissipative forces, and the last two terms are the tangential elastic, and dissipative forces [44].

Except for the contact forces, the non-contact forces also have an essential influence on the packing and movements of particles when dense particles are involved. These non-contact forces are usually calculated by empirical equations, which include van der Waals force [48], electrostatic force [49], and liquid bridge force [50][51][52]. These forces are summarized and listed in [41].

The particle-fluid interactions also contribute significant influence to particle movements since the particle is always surrounded by fluids. Various kind of force models have been implemented in the DEM module of OpenFOAM, which include drag force ($\vec{F}_d$), pressure gradient force ($\vec{F}_{\nabla p}$), viscous force ($\vec{F}_\tau$), virtual mass force ($\vec{F}_{vm}$), lift force ($\vec{F}_L$), Magnus lift force ($\vec{F}_{Ml}$), and etc. [1] [41] [53]. These are all belong to the particle-fluid interaction force, which can be written as:

$$\vec{F}_i^f = \vec{F}_d + \vec{F}_{\nabla p} + \vec{F}_\tau + \vec{F}_L + \vec{F}_{vm} + \vec{F}_{Ml} + ... \tag{2.29}$$

In this thesis, only the drag force and pressure gradient force are considered as the particulate flow validation case is a simulation of the fluidized bed, which only includes gas and solid. As stated by Zhu et al., these two forces play the most important roles in a gas-solid system [41]. Therefore, the rest of this section will primarily focus on the drag force model and pressure gradient force that is implemented in OpenFOAM, and equation 2.29 is simplified to:

$$\vec{F}_i^f = \vec{F}_d + \vec{F}_{\nabla p} \tag{2.30}$$

First, the drag force depends on the superficial velocity $\vec{U}^s$, which is proportional to the relative velocity between the continuous phase and the dispersed phase [1]. It is defined as:

$$\vec{U}^s = \alpha_d \left(\vec{U}_c - \vec{U}_P \right) \tag{2.31}$$

Herein, $\vec{U}_c$ and $\vec{U}_P$ are the dispersed phase and continuous phase velocity, respectively. α_d is the volume fraction of the dispersed phase in a certain cell.

$$\alpha_d = \frac{1}{V_{cell}} \sum_{\forall_i \in V_{cell}} V_{P,i} \tag{2.32}$$

In this equation, V_P is the volume of a particle that is located in a specific volume (V_{cell}). Besides, the summation of α_d and α_c is unity. Based on Newton's equation, the drag force that is experienced by a spherical particle with a diameter of d_P moving in a gas flow with a density of ρ_c can be calculated by:

$$\vec{F}_d = \frac{1}{2} C_d A_P \rho_c \left| \vec{U}^s \right| \vec{U}^s = C_d \pi d_P^2 \rho_c \left| \vec{U}^s \right| \frac{\vec{U}^s}{8} \tag{2.33}$$

where C_d is the drag coefficient, which can be split into two regions depending on the Reynolds number (0.2 and 0.5 for laminar and turbulent flows, respectively [54]). A_p is the cross-section area of the particle. However, when a particle is surrounded by others, the fluid volume reduces, and equation 2.33 is no longer applicable. This is because the surrounded particle would generate a sharp fluid velocity gradient, which leads to increased shear stress on the particle surface [41].

To determine the drag force experienced by a particle in a particulate system, the model developed by Ergun and Wen and Yu [41] can be used. The empirical correlations contained in this model are based on either bed pressure drop or bed expansion experiment [41]. In addition, the drag equation and the correlations are defined as:

$$\vec{F}_d = V_{P,i} \, \beta_{Pf} \, \frac{\vec{U}^s}{\alpha_d} \tag{2.34}$$

Herein, β_{Pf} is the inter-phase momentum exchange coefficient. Based on the volume fraction of the fluid, this parameter has two correlations:

$$\beta_{Pf} = \begin{cases} 150\frac{(1-\alpha_d)^2}{\alpha_c}\frac{\mu_c}{(d_{P,i})^2} + 1.75\alpha_d\frac{\rho_c}{d_{P,i}}\left|\vec{U}^s\right| & (\alpha_c < 0.8) \\ \frac{3}{4}C_d\frac{\alpha_d\alpha_c}{d_{P,i}}\rho_c\left|\vec{U}^s\right|\alpha_c^{-2.65} & (\alpha_c \geq 0.8) \end{cases} \tag{2.35}$$

where μ_c is the dynamic viscosity of the continuous phase, and C_d in this case also depends on the particle Reynolds number, which is given by:

$$C_d = \begin{cases} \frac{24}{\mathrm{Re}_P}\left(1 + 0.15\mathrm{Re}_P^{0.687}\right) & (\mathrm{Re}_P \leq 1000) \\ 0.44 & (\mathrm{Re}_P > 1000) \end{cases} \tag{2.36}$$

and Re_P is defined as:

$$\mathrm{Re}_P = \frac{\rho_c d_P \alpha_c |\vec{U}^s|}{\mu_c} \tag{2.37}$$

In addition to drag force, the pressure gradient force is defined as:

$$\vec{F}_{\nabla p} = -V_i \nabla p \tag{2.38}$$

This force will also be further investigated in the next section. Equations of other particle-fluid interaction forces can be found in [41][44].

2.4.5 Review of the Continuous Phase Governing Equations in a Single Fluid CFD-DEM solver

For a particulate flow, the presence of particles obstructs the path of the flow, and forces the fluid to detour its path by flowing through the gaps between particles. Concurrently, the fluid generates drag force and other particle-fluid interaction forces, which can lead to movements of particles. Also, the motion of particles will in turn influence the fluid flow again. Compared to the general form of the momentum and continuity equations, the volume fraction of the dispersed phase needs to be taken into consideration. Moreover, the mass and momentum conservation of the continuous phase are governed by the local mean variables technique developed by Anderson and Jackson [55]. Therefore, Gidaspow [56] proposed two models to ensure mass

(continuity equation) and momentum conservation (Navier-Stokes (NS) equations) for an incompressible fluid with particles in the computational domain. These two models are labelled as Model A and Model B, and they are defined as [57]:

- Model A:

$$\frac{\partial \alpha_c \rho_c \vec{U}_c}{\partial t} + \nabla \cdot (\alpha_c \rho_c \vec{U}_c \vec{U}_c) = -\alpha_c \nabla p + \nabla \cdot (\alpha_c \rho_c \tau_{st}) + \alpha_c \rho_c \vec{g} - \vec{S}_P^A \tag{2.39}$$

- Model B:

$$\frac{\partial \alpha_c \rho_c \vec{U}_c}{\partial t} + \nabla \cdot (\alpha_c \rho_c \vec{U}_c \vec{U}_c) = -\nabla p + \nabla \cdot (\alpha_c \rho_c \tau_{st}) + \alpha_c \rho_c \vec{g} - \vec{S}_P^B \tag{2.40}$$

Herein, $\vec{U}_c$, α_c and ρ_c are the continuous phase velocity, volume fraction and density, respectively. p is the static pressure, and $\vec{S}_P$ is the particle-fluid interaction forces. It can be seen that the pressure term is different in the two models, because Model A assumes that both gas and solid phases share the pressure drop, whereas in model B the pressure drop is only applied to the gas phase. Note, the particle-fluid interaction forces in equation 2.39 and 2.40 are defined differently to ensure the momentum equations in both models are equivalent. For example, $\vec{S}_P^A$ and $\vec{S}_P^B$ are connected via $\vec{S}_P^B = \vec{S}_P^A/\alpha_c - \rho_c(1-\alpha_c)\vec{g}$ in the Two Fluid Model [56]. In addition, both models use the same continuity equation:

$$\frac{\partial \alpha_c}{\partial t} + \nabla \cdot (\alpha_c \vec{U}_c) = 0 \tag{2.41}$$

In *denseParticleFoam*, the non-conservative form of the Model B is implemented. Also, it is a transient solver that solves single phase incompressible fluid with particles. This implies the density of the continuity phase will be the same everywhere in the domain over time. For this reason, equation 2.40 can be divided by the density. The transient and the convective terms of the *denseParticleFoam* solver are defined by the following codes:

```
fvm::ddt(alphac, Uc) + fvm::div(alphaPhic, Uc)
- fvm::Sp(fvc::ddt(alphac) + fvc::div(alphaPhic), Uc)
```

and the momentum equation of *denseParticleFoam* can be written as:

$$\begin{aligned}\frac{\partial \alpha_c \vec{U}_c}{\partial t} + \nabla \cdot (\alpha_c \vec{U}_c \vec{U}_c) - \left(\vec{U}_c \frac{\partial \alpha_c}{\partial t} + \vec{U}_c \nabla \cdot (\alpha_c \vec{U}_c) \right) \\ = -\frac{1}{\rho_c} \nabla p + \nabla \cdot (\alpha_c \tau_{st}) + \alpha_c \vec{g} - \frac{1}{\rho_c} \vec{S}_P^B\end{aligned} \tag{2.42}$$

With the additional two terms, $\vec{U}_c \frac{\partial \alpha_c}{\partial t}$ and $\vec{U}_c \nabla \cdot (\alpha_c \vec{U}_c)$, the following derivation can prove that equation 2.44 is in the non-conservative form, since one of the velocity terms is outside of the convective term, which is the essential characteristic of the non-conservative form of the momentum equation. Also, the summation of these two additional terms should be zero theoretically because of the continuity equation, equation 2.41.

$$\begin{aligned}&\frac{\partial \alpha_c \vec{U}_c^n}{\partial t} + \nabla \cdot \left(\alpha_c \vec{U}_c^{n-1} \vec{U}_c^n \right) - \left(\vec{U}_c^n \frac{\partial \alpha_c}{\partial t} + \vec{U}_c^n \nabla \cdot \left(\alpha_c \vec{U}_c^{n-1} \right) \right) \\ &= \vec{U}_c^n \frac{\partial \alpha_c}{\partial t} + \alpha_c \frac{\partial \vec{U}_c^n}{\partial t} + \alpha_c \vec{U}_c^{n-1} \cdot \nabla \vec{U}_c^n + \vec{U}_c^n \nabla \cdot \left(\alpha_c \vec{U}_c^{n-1} \right) \\ &\quad - \left(\vec{U}_c^n \frac{\partial \alpha_c}{\partial t} + \vec{U}_c^n \nabla \cdot \left(\alpha_c \vec{U}_c^{n-1} \right) \right) \\ &= \alpha_c \frac{\partial \vec{U}_c^n}{\partial t} + \alpha_c \vec{U}_c^{n-1} \cdot \nabla \vec{U}_c^n\end{aligned} \tag{2.43}$$

where the superindex n and $n-1$ means the current time step and last time step, respectively. Therefore, equation 2.42 can be simplified to:

$$\alpha_c \frac{\partial \vec{U}_c}{\partial t} + \alpha_c \vec{U}_c \cdot \nabla \vec{U}_c = -\frac{1}{\rho_c} \nabla p + \nabla \cdot (\alpha_c \tau_{st}) + \alpha_c \vec{g} - \frac{1}{\rho_c} \vec{S}_P^B \tag{2.44}$$

Herein, $\vec{S}_P$ is the volumetric momentum exchange term between particles and fluids, it is dominated by the drag force and pressure gradient force in a gas-solid flow as mentioned in the previous section. The formula is defined as:

$$\begin{aligned}\vec{S}_P^B &= \frac{\sum_{i=1}^{N_P} \left(\vec{F}_d + \vec{F}_{\nabla p} \right)}{V_{cell}} \\ &= (1 - \alpha_c) \left(\vec{F}_d + \vec{F}_{\nabla p} \right)\end{aligned} \tag{2.45}$$

In contrast, the interaction force only takes the drag force into account in Model A [58].

$$\vec{S}_P^A = \frac{\sum_{i=1}^{N} \vec{F}_d}{V_{cell}} = (1 - \alpha_c)\vec{F}_d \tag{2.46}$$

Therefore, if equation 2.45 is substituted into equation 2.44, equation 2.44 will be transferred to the equivalent form of Model A.

2.4.6 The Algorithm of two-way coupled CFD-DEM solvers

For a two-way coupled CFD-DEM solver, the order of solving the governing equations of particles and fluids is important. In general, the dispersed phase is solved first. At this stage, the forces that are acting on the particles, positions, locations of particles, and volume fraction of the continuous phase are updated. Next, transferring information to the CFD solver, and the fluid flow will be solved based on the momentum equation and continuity equation. The steps in each time interval are illustrated in the following list, and the algorithm of a two-way coupled solver can be seen in figure 2.5 [1].

- Use the DEM solver to solve the particles' position and velocities.
- Particles are projected onto the CFD mesh based on their new positions.
- The volume fraction of the continuous phase is updated in each cell.
- The particle-particle and particle-fluid interaction forces are calculated.
- Pass the information to the CFD solver.
- Update the velocity and the pressure field for the fluid with taking the interaction forces and volume fraction of the dispersed phase into account.

The algorithm that is used in this project will be discussed in chapter 3.

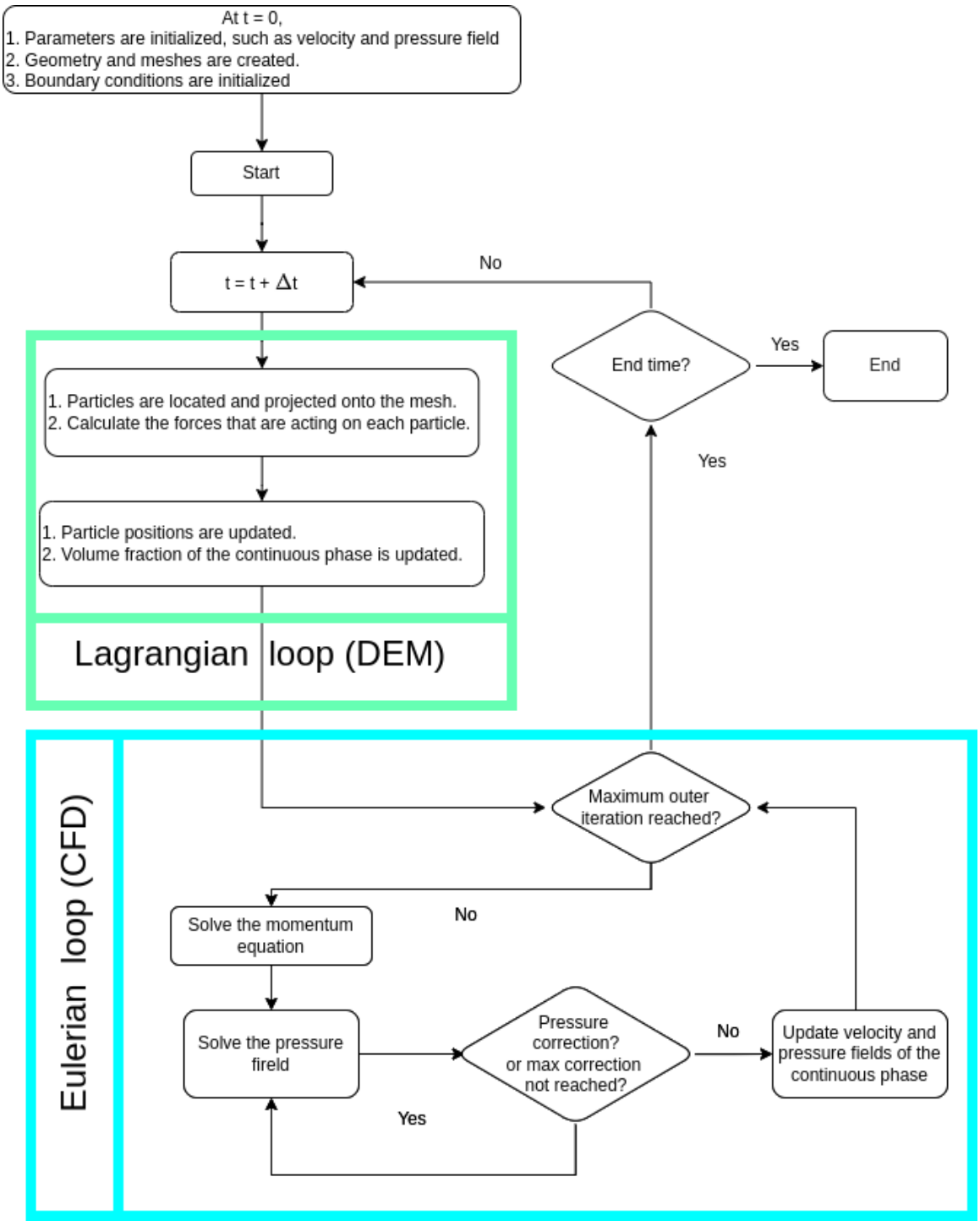


Figure 2.5: Algorithm of a two-way coupled CFD-DEM solver.

2.5 Conclusions

This chapter is a review of the prior research, which mainly focuses on the previous W-O emulsion viscosity models and CFD and DEM models for a single fluid phase particulate flow. Based on these reviews, it can be concluded the following two problems may lead to inaccurate simulation results for a particulate emulsion.

- The viscosity model in the current version of OpenFOAM is a linear function, which can not represent the W-O emulsion viscosity correctly.
- The project sponsor SST is using a one-way coupled method. However, when introducing a large number of particles, the flow will be affected by the particle phase, which requires a two-way coupled method.

To solve the problems above, this project is split into three phases and new methodologies will be illustrated in chapter 3. These include the development of a new W-O emulsion viscosity model, the modification of the *denseParticleFoam*, and the theory of combining the VOF method with *DPConFoam*.

Chapter 3

New Methodology & Implementation of OpenFOAM Codes

3.1 Introduction

Coupling the VOF method with an unresolved, incompressible, single fluid phase CFD-DEM model has been validated in terms of solving fluid mixtures with the presence of dense particles [9][10][59][60]. In these studies, they coupled the VOF method with LIGGGHTS and simulated many representative cases, such as the dam break, the circulating flow in a rotating drum, and the particle sedimentation. All of these cases demonstrated well the particle-fluid interactions, and the simulation results of these cases show satisfactory agreement with the experimental results or the empirical values. However, these solvers can only handle a maximum of two fluid phases and one particle phase, whereas the sand separation simulations usually include at least three fluids, namely water, oil, and air, and one particle phase. In this project, the proposed new methods will extend the number of fluids to any arbitrary value.

The new methods developed in this project are described in this chapter, which includes creating and implementing a new viscosity model for W-O emulsion mixtures, modifications of the *denseParticleFoam*, and deriving a new set of governing equations

for a two-way coupled multi-fluid CFD-DEM solver. All of these are performed in OpenFOAM-dev, which is one of the OpenFOAM version that include the most up-to-date libraries and was released in 2019. The advantages of using OpenFOAM can be concluded from the following perspectives. First, it is open-source software. This allows users to modify solvers without restrictions. Second, the solvers are continuously maintained and updated, which gives users opportunities to submit bugs and recommendations to professionals to enhance the quality of the solvers. Third, the DEM model implemented in OpenFOAM has been validated by Fernandes et al. [1]. This opens up the possibility that the coupling process can be directly performed in OpenFOAM instead of coupling the CFD solver with the DEM software LIGGGHTS, which has been converted from open-source to close-source software for a few years. Therefore, two solvers were selected from OpenFOAM to be the base solvers of this study, namely *denseParticleFoam* (contains the DEM model) and *multiphaseInterFoam* (contains the VOF and multi-fluid models).

In section 3.2, the new W-O emulsion viscosity equation model will be explained. The implementation of the new viscosity model will be performed in chapter 4, and a new solver, *multiphaseEmulsionFoam*, will be obtained. Also, three test cases will be described to test the new viscosity model. The first case will be used to test the functionalities of the new viscosity model. Case two is a W-O mixture simulation in a horizontal pipe, and case three is based on the SST indoor flow loop. Section 3.3 mainly focuses on the modifications of the *denseParticleFoam* solver, which include changing the governing equations to the strong conservation form, restoration of the density to the momentum equation, and replacing the body force term and pressure term. The new, modified solver is named DPConFoam. Also, a fluidized bed case is described to validate the *DPConFoam*. In addition, section 3.4 shows the derivation for the governing equations of the two-way coupled multi-fluid CFD-DEM solver, especially the VOF method. The implementation and validation of this set of governing equations will be performed in future works.

3.2 Modification of the Multiphase Mixture Model

As mentioned in chapter 1, the W-O emulsion viscosity tends to change dramatically with respect to water cut, shear rate, and temperature. In this study, the temperature effect is neglected since the flow loop is placed in an indoor lab, where the temperature is constant. Also, the temperature of the fluid was measured while performing the experiments, and the temperature change was unnoticeable. Therefore, the new W-O emulsion viscosity model only takes the water cut and shear rate into account in this project.

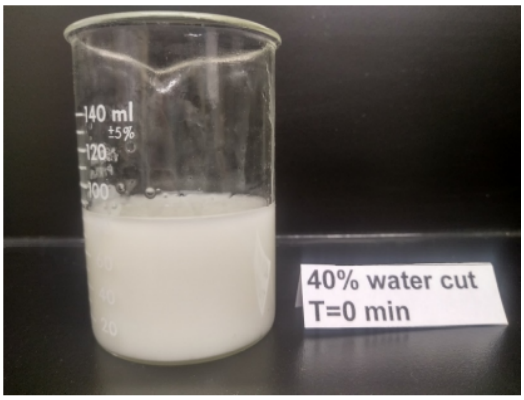
Based on the reviews in section 2.2, most of the existing numerical models are empirical equations obtained by curve fitting the experimental data. Also, the W-O emulsion viscosity cannot be represented by a unique function since the oil properties would change for different oils. The oil used in this project is Shell Tellus S2 VX 22[1], and there is no viscosity data of the W-O emulsions that are made by this type of oil. Therefore, before creating the new viscosity model, the W-O emulsion viscosity needs to be measured.

3.2.1 Viscosity Measurement

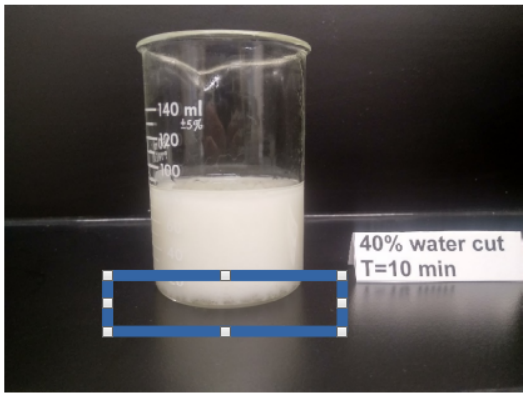
The emulsion used in this study is produced by mixing the Shell Tellus S2 VX 22 with water, and its viscosity was measured at different water cuts and shear rates. Before conducting the measurements, the stability of different water cut W-O emulsions were tested. Different water cut emulsions were well mixed and settled for 30 min. It was observed that the separations gradually occurred over time, and an example of the 40% water cut emulsion can be seen in figure 3.1. Therefore, 0.5% of Span 80 was added to ensure the emulsions were stable during the measurements.

Five different water cut emulsions were prepared by using a VWR® 25D Digital Homogenizer with an operating speed of 10000 rpm for 3 min. Meanwhile, tap water was added drop by drop. Also, mixing the emulsion at the same speed for another 3

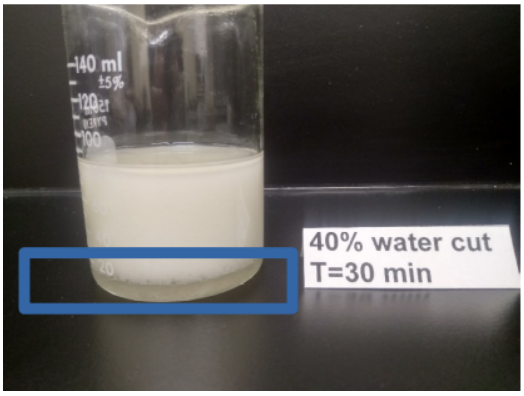
[1]Detailed information of Shell Tellus S2 VX 22 can be found in Appendix A



(a)



(b)



(c)

Figure 3.1: The stability test for 40% water cut emulsion after a) 0, b) 10, and c) 30 minutes. (courtesy of Gustavo Cifuentes Dias and Dr. Sean Sanders)

min just before the measurement. An example of 40% water cut emulsion with 0.5% of Span 80 after 10 min is shown in figure 3.2.

The emulsion viscosity was measured with a TA Instruments AR-G2 rheometer at 20°C. During each measurement, the shear rate was increasing from 10 s^{-1} to 280 s^{-1} with increments of 30 s^{-1} and then decreasing to 10 s^{-1}. By adding 0.5% of the emulsifier, the emulsion was stable with water cut up to 70%. After 70% water cut, the phase inversion happened, and water started to dominate the properties of the emulsion. Therefore, the emulsion viscosity after 70% were not used in this study. The measuring process was repeated three times for each emulsion to ensure the precision and accuracy of the results. In total, 50 averaged viscosity values were

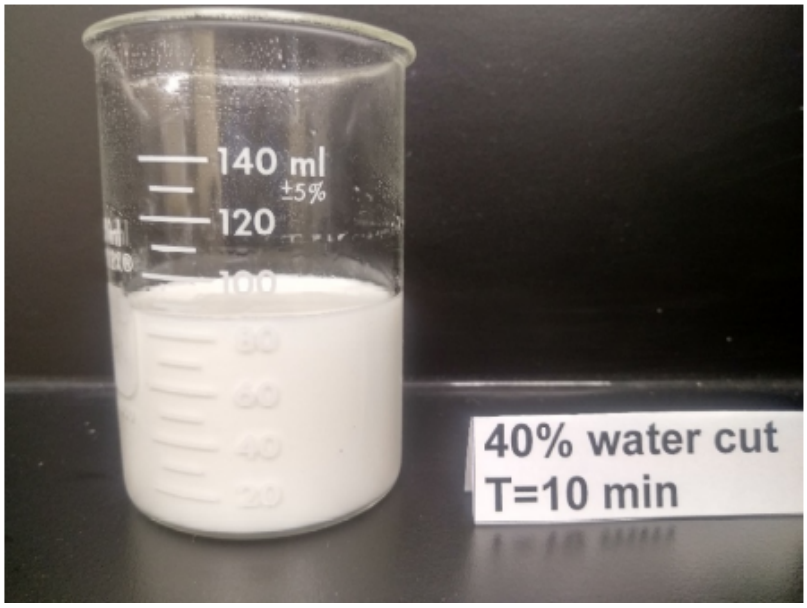


Figure 3.2: 40% water cut emulsion with 0.5% Span 80.
(courtesy of Gustavo Cifuentes Dias and Dr. Sean Sanders)

used to analize the relationship between the viscosity of different water emulsions and shear rate, and the results are listed in section 4.2.1.

3.2.2 Proposed new viscosity models for emulsions

In this project, a new empirical viscosity model that considers both shear rate and water cut was proposed. Equations in this model are obtained by curve-fitting the experimental results. However, it is known that the emulsion viscosity for water cut and shear rate show different tendencies, which means they must be curve-fitted by different base functions. Therefore, the challenge of this new model is to define a function that can include both parameters at the same time. To overcome this problem, either the Bidirectional Interpolation Method (BIM) or the representational functions can be used. In both methods, the Heaviside step function will be applied to deal with the viscosity change due to the phase inversion. In this project, the functions that depends on the shear rate and water cut are called shear rate functions and water cut functions, respectively.

As it can be seen in figure 3.3, the Heaviside step function returns zero and one

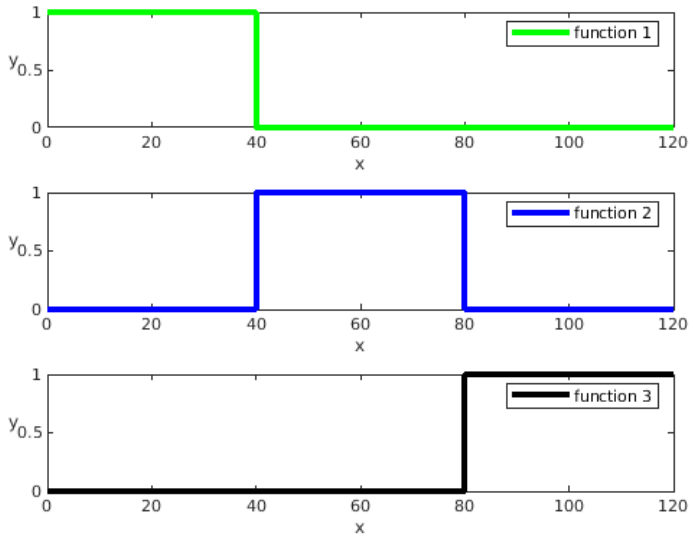


Figure 3.3: An example of the Heavisde step function.

for true and false arguments, respectively. Depending on the value of the variable, different functions will be used. In OpenFOAM, the Heaviside step function can be controlled by two functions, 'pos' and 'neg'. These two functions return one if the inpu statement is true and zero otherwise. For example, 'pos(1)' and 'pos(-1)' will return 1 and 0, respectively. Therefore, the Heaviside step function for emulsion viscosity with respect to water cut can be defined as:

$$\mu_e = \begin{cases} f_1(\phi_w) & \phi_w \leq \text{inversion point} \\ f_2(\phi_w) & \phi_w > \text{inversion point} \end{cases} \tag{3.1}$$

To evaluate the emulsion viscosity at a specific shear rate ($\dot{\gamma}_1$) and water cut (ϕ_{w1}) with the BIM, one possible method is to curve-fit all the shear rate functions first. Next, substitute $\dot{\gamma}_1$ into these functions, and a series of viscosity values will be obtained. Then, these values can be curve-fitted to an intermediate water cut function. Finally, the effective viscosity will be gained by substituting ϕ_{w1} into the intermediate function. The advantage of this method is that it is relatively accurate. However, implementing this method in OpenFOAM would require additional files used to curve-fit functions. Also, every time the viscosity is evaluated in the domain,

an intermediate water cut function is required, and the viscosity needs to be computed cell by cell. For this reason, this method is computationally costly and complicated to be implemented, which will not be used in this project.

For the representational function method, the water cut functions could be split into a few groups based on their coefficients. Then, one function will be used to represent all the functions in this zone. The coefficients of the representational function need to ensure that it is highly correlated to the measured data in its zone. This method will not affect the overall structure of the OpenFOAM code since it only requires replacing the original viscosity function with a Heaviside step function. Even though this method is less accurate than the BIM, its efficiency gains more popularity in solving industry problems.

After the new viscosity is implemented, a quick test will be performed, and its setup and results will be shown in section 4.2.4.

3.2.3 Case Two: Water-Oil Mixtures in Horizontal Pipe

In this case, a W-O mixture was simulated in a horizontal pipe with both *multiphaseEmulsionFoam* (new viscosity model) and *multiphaseInterFoam* (original linear viscosity function), and the simulation results were compared with results from the literature [61][62]. The geometry and the mesh of the horizontal pipe can be seen in figures 3.4 and 3.5, respectively. There are two inlets and one outlet. In this way, water and oil can be injected into the domain separately. Fluids used in this case are water and oil, and an artificial density was used for oil to exaggerate the stratification phenomenon. Also, the entire domain was filled with water initially, and then gradually injected water and oil from two inlets. In addition, the mesh was generated by snappyHexMesh method, which is a mesh generation tool implemented in OpenFOAM. The mesh is dominated by hexahedra type of mesh (96.7%), and a few places are polyhedra and prisms (3.3%). The information of the geometry and the mesh can be found in table 3.1. The length of the pipe is 20 times the diameter

of the pipe, which gives the flow sufficient space to become fully developed at the downstream of the pipe.

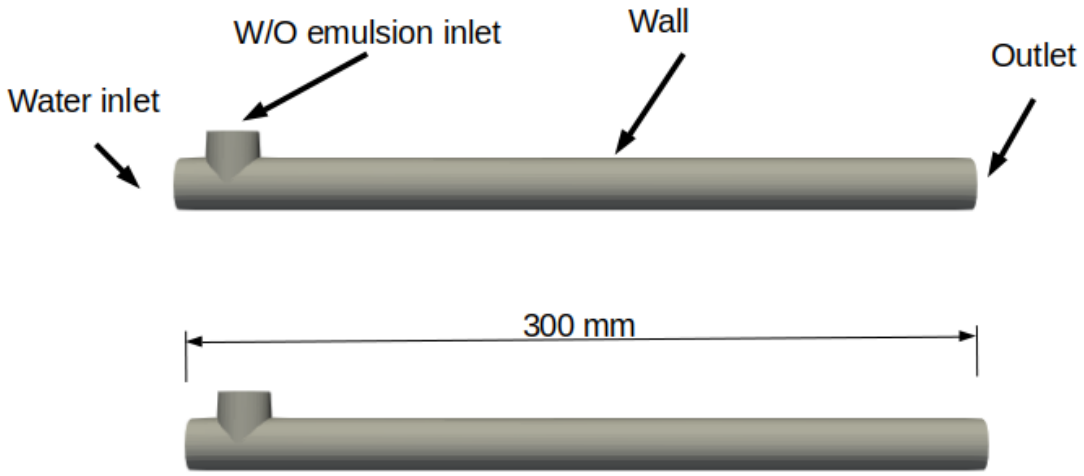


Figure 3.4: The geometry of the horizontal pipe.

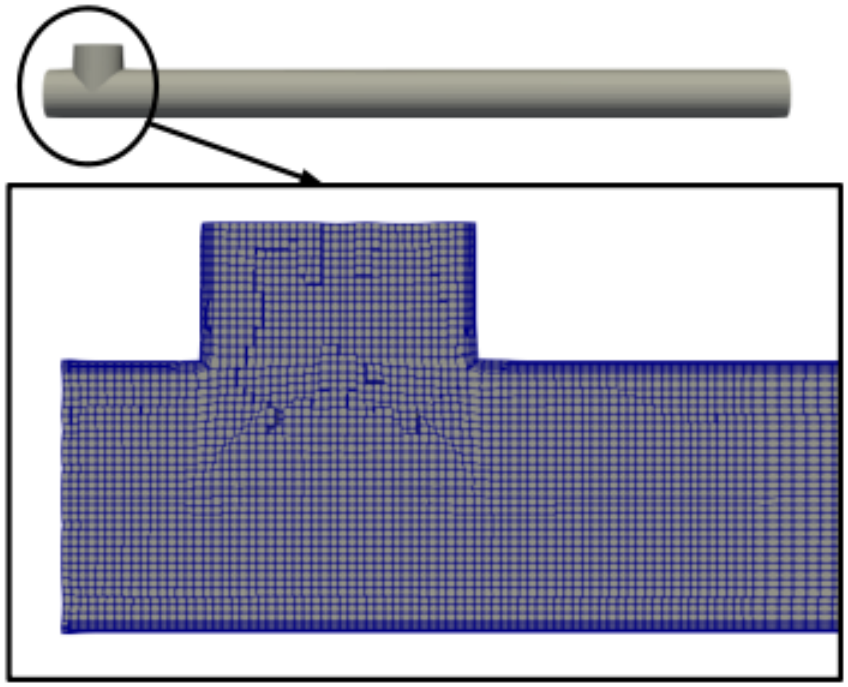


Figure 3.5: The geometry of the horizontal pipe.

Table 3.1: Geometry and mesh information of the horizontal pipe

Variables	Value
Simulation geometry	
Overall W×D×H	300×20×40 mm
Diameter of the pipe	20 mm
Mesh	
Number of cells	330184
First inflation layer thickness	0.000128 m
Expansion ratio	1.15
Number of layers	5

The Reynolds number for a pipe flow can be calculated by:

$$\mathrm{Re} = \frac{U\, D_P}{\nu} \tag{3.2}$$

where D_P is the diameter of the pipe. Based on this equation, the Reynolds numbers obtained for water and W/O emulsion inlets are 2000 and 23, respectively. Since the Reynolds number for water and oil is under 2300, the flow will probably be laminar. Regarding the boundary conditions, the outlet was set to 'inletOutlet' boundary condition [63], which plays two roles. First, it sets an inlet velocity boundary condition for the reverse flow. Second, the outflow boundary condition was set to zero gradient. In this study, the reverse flow velocity was set to zero to avoid any flow flowing back to the domain. Detailed boundary conditions and the parameters used in the setup are listed in table 3.2 and 3.3, respectively.

Table 3.2: Boundary conditions of the horizontal pipe case

Boundaries	**Boundary conditions**
Water inlet	$\vec{U}_x = 0.1$ m·s^{-1}
	p_{rgh}: zero gradient
	ϕ_w: uniform 1
	ϕ_o: uniform 0
W/O emulsion inlet	$\vec{U}_y = -0.1$ m·s^{-1}
	p_{rgh}: zero gradient
	ϕ_w: uniform 0.3
	ϕ_o: uniform 0.7
Outlet	$\vec{U}_c$: inletOutlet
	p_{rgh}: 0 Pa
	ϕ_w and ϕ_o: zero gradient
Walls	no-slip walls
	p_{rgh}: zero gradient
	ϕ_w and ϕ_o: zero gradient

Table 3.3: Parameters used in setup of the horizontal pipe case

Variables	Value
Fluid properties	
ρ_w	1000 kg·m^{-3}
μ_w	10^{-6} kg·m^{-1}·s^{-1}
ρ_o	500 kg·m^{-3}
μ_o	$5.5{\cdot}10^{-5}$ kg·m^{-1}·s^{-1}
$\sigma_{w,o}$	0.07
Turbulent model	k-omega SST
Discretization schemes	
Transient term	Euler
Convection term	Linear linear
Diffusion term	Gauss linear
Convergence criteria	
$\vec{U}_c$	10^{-6}
p_{rgh}	10^{-7}
Total time	16 s for coarse mesh 5 s for medium and fine meshes
Time step	Depends on the maximum Co
Maximum Co	0.5
Internal field	
ϕ_w	1
ϕ_o	0

For this case, the velocity profile at the downstream of the pipe, where the flow is fully developed, and the dynamics of the interface, will be investigated. Results and discussion can be found in section 4.3.

3.2.4 Case Three: Flow Loop Simulation

To investigate the influence caused by the new viscosity model, simulations were performed in a modified SST indoor flow loop, which can be seen in figure 3.6. There

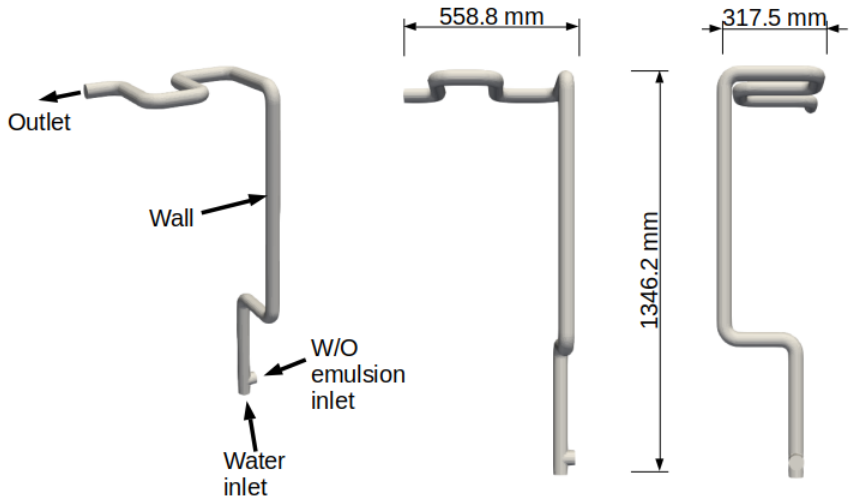


Figure 3.6: The geometry of the modified indoor flow loop.

are two inlets and one outlet. In this way, water and oil can be injected into the domain separately, and the mixing and separation process of the mixture would be clearer. Fluids used in this case are water and oil, and the density of the oil was fabricated so that the gravity effect will be exaggerated. Also, the entire domain was filled with water initially, and then gradually injected water and oil from two inlets. The mesh was generated by snappyHexMesh method. The mesh is dominated by hexahedra type of mesh (88.3%), and a few places are polyhedra and prisms (11.7%). The mesh of a section of the pipe is shown in figure 3.7. Some geometry and mesh information can be found in table 3.4. Simulations were performed by the *multiphaseInterFoam* solver (original linear viscosity function) and the modified *multiphaseInterFoam* solver (new viscosity model), and the results were numerically compared.

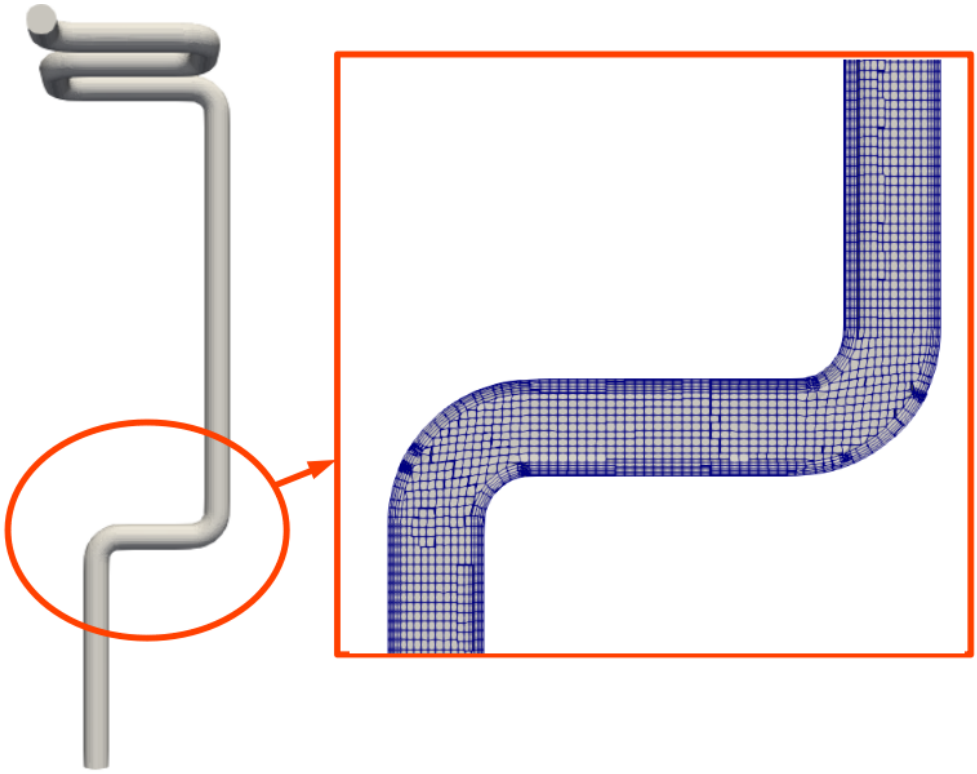
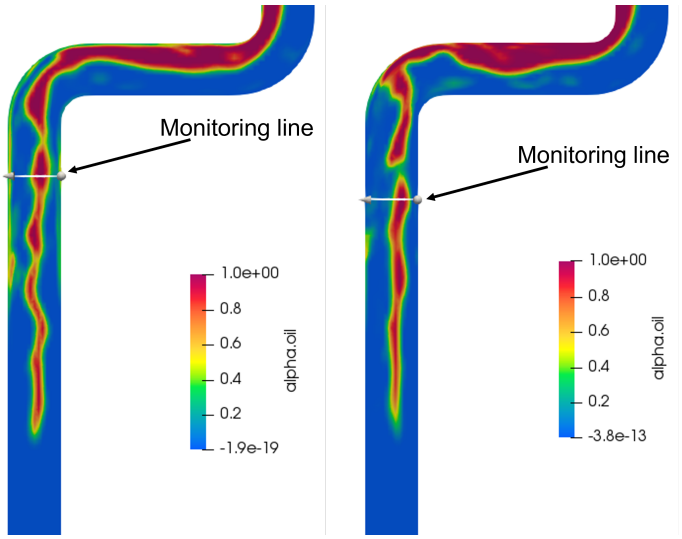


Figure 3.7: The mesh of the modified indoor flow loop.

Table 3.4: Geometry and mesh information of the modified flow loop case

Variables	Value
Simulation geometry	
Overall W×D×H	558.8×317.5×1346.2 mm
Diameter of the pipe	50.8 mm
Mesh	
Number of cells	109288
First inflation layer thickness	0.001 m
Expansion ratio	1.15
Number of layers	5

Based on equation 3.2, the Reynolds numbers obtained for water and W/O emulsion inlets are 5080 and 92.36, respectively. Since the Reynolds number for water is above 4000, the flow will be turbulent. Therefore, the k-omega SST (Shear Stress Transport) model was utilized. Also, the outlet was set to 'inletOutlet' boundary condition to avoid any flow flowing back to the domain. Detailed boundary conditions and the parameters used in the setup are listed in table 3.5 and 3.6, respectively.



(a) (b)

Figure 3.8: Locations of the monitoring lines used in *multiphaseEmulsionFoam* (left) and *multiphaseInterFoam* (right).

Furthermore, two lines across the mixture region were placed in the pipe, as shown in figure 3.8. In this way, the viscosity change along the lines can be compared for two models. Other comparisons, such as flow patterns, separation of the fluids, and the flow development will be discussed in chapter 4.

Table 3.5: Boundary conditions of modified flow loop

Boundaries	**Boundary conditions**
Water inlet	$\vec{U}_z = 0.1$ m·s^{-1}
	p_{rgh}: zero gradient
	k_{tur}, ν_{tur} and ω_{tur} : fixed value
	ϕ_w: uniform 1
	ϕ_o: uniform 0
W/O emulsion inlet	$\vec{U}_x = -0.1$ m·s^{-1}
	p_{rgh}: zero gradient
	k_{tur}, ν_{tur} and ω_{tur} : fixed value
	ϕ_w: uniform 0.3
	ϕ_o: uniform 0.7
Outlet	$\vec{U}_c$: inletOutlet
	p_{rgh}: 0 Pa
	k_{tur} and ω_{tur}: zero gradient
	ν_{tur}: fixed value
	ϕ_w and ϕ_o: zero gradient
Walls	no-slip walls
	p_{rgh}: zero gradient
	k_{tur}: with kqRWallFunction
	ν_{tur}: nutkWallFunction
	ω_{tur}: omegaWallFunction
	ϕ_w and ϕ_o: zero gradient

Table 3.6: Parameters used in setup of the modified flow loop

Variables	Value
Fluid properties	
ρ_w	1000 kg·m^{-3}
μ_w	10^{-6} kg·m^{-1}·s^{-1}
ρ_o	500 kg·m^{-3}
μ_o	$5.5{\cdot}10^{-5}$ kg·m^{-1}·s^{-1}
$\sigma_{w,o}$	0.07
Turbulent model	k-omega SST
Discretization schemes	
Transient term	Euler
Convection term	Linear linear
Diffusion term	Gauss linear
k_{tur} **and** ω_{tur}	Gauss upwind
Convergence criteria	
k_{tur}, $\vec{U}_c$, ω_{tur}	10^{-6}
p_{rgh}	10^{-7}
Total time	25 s
Time step	Depends on the maximum Co
Maximum Co	0.5
Internal field	
ϕ_w	1
ϕ_o	0

3.3 Modifications of the *denseParticleFoam*

The purpose of performing modifications in *denseParticleFoam* is to prepare a new intermediate solver, to be called *DPConFoam*, to couple with the VOF model implemented in *multiphaseInterFoam*. As mentioned in section 2.4.6, the information of the continuous phase is used to evaluate the particle movements for a two-way coupled solver. Therefore, it is essential to use the same set of variables in *DPConFoam* and *multiphaseInterFoam*. Also, the form of the continuity equations in both solvers should be the same. Therefore, a series of modifications will be shown in this section to obtain the intermediate step solver, which was named as *DPConFoam*.

The continuity equation, equation 2.41, is still valid for *DPConFOAM*. No modification will be performed on it.

3.3.1 Modification of the governing equations in *denseParticleFoam* to strong conservation form

As stated in section 2.4.5, the momentum equation used in *denseParticleFoam* is in the non-conservative form as a result of the additional two terms, $\vec{U}_c \frac{\partial \alpha_c}{\partial t}$ and $\vec{U}_c \nabla \cdot (\alpha_c \vec{U}_c)$. To keep the form of the governing equations consistent with *multiphaseInterFoam*, equation 2.44 has to be transformed to the strong conservation form. Thus, these two terms should be removed. In this way, the strong conservation form of the momentum equation can be obtained as:

$$\frac{\partial \alpha_c \vec{U}_c}{\partial t} + \nabla \cdot (\alpha_c \vec{U}_c \vec{U}_c) = -\frac{1}{\rho_c} \nabla p + \nabla \cdot (\alpha_c \tau_{st}) + \alpha_c \vec{g} - \frac{1}{\rho_c} \vec{S}_P^B \tag{3.3}$$

In addition, *denseParticleFoam* is dedicated to solving for a single incompressible fluid with the transport of particles. In this case, the fluid density would not change in both space and time. For this reason, all the terms in equation 2.44 can be divided by the continuous phase density. However, when solving the multi-fluid system, the sharp changes in density need to be taken into account in equation 3.3. Thus, it needs

further modifications, which will be explained in the next section.

3.3.2 Restoration of density to the governing equations

The density of a multi-fluid system may vary between two adjacent cells. Also, the density in a specific cell depends on the volume fraction of each fluid, which means it may change over time. This implies that the density of a multiphase mixture does not only change in space but also in time. Therefore, it is necessary to restore density to its original place when solving multiphase mixture cases, and equation 3.3 is modified to:

$$\frac{\partial \alpha_c \rho_c \vec{U}_c}{\partial t} + \nabla \cdot (\alpha_c \rho_c \vec{U}_c \vec{U}_c) = -\nabla p + \nabla \cdot (\alpha_c \rho_c \tau_{st}) + \alpha_c \rho_c \vec{g} - \vec{S}_P^B \tag{3.4}$$

3.3.3 Replacement of the body force and the pressure term

The pressure term and body force term in equation 2.44 can be replaced by a modified pressure term (p_{rgh}), which is the summation of the kinematic pressure and hydrostatic pressure. With this step, the body force term, $\rho\vec{g}$, does not need to be included in the momentum equation, which will make the solver numerically convenient in solving buoyant and multiphase cases [64]. p_{rgh} is defined as:

$$p_{rgh} = p - \rho_c(\vec{g} \cdot \vec{h}) \tag{3.5}$$

Substituting this pressure into the gradient operator, the pressure gradient term with p_{rgh} can be derived to:

$$\begin{aligned} -\nabla p &= -\nabla p_{rgh} - \nabla(\alpha_c \rho_c (\vec{g} \cdot \vec{h})) \\ &= -\nabla p_{rgh} - \alpha_c \rho_c \vec{g} \cdot \nabla \vec{h} - \vec{h} \cdot \nabla \alpha_c \rho_c \vec{g} \\ &= -\nabla p_{rgh} - \alpha_c \rho_c g \cdot \nabla \vec{h} - \alpha_c \rho_c \vec{h} \cdot \nabla \vec{g} - \vec{g} \cdot \vec{h} \nabla \alpha_c \rho_c \end{aligned} \tag{3.6}$$

$$-\nabla p_{rgh} - \vec{g} \cdot \vec{h} \nabla(\alpha_c \rho_c) = -\nabla p + \alpha_c \rho_c \vec{g} \tag{3.7}$$

This derivation is based on [65], and $\nabla\vec{h}$ is equivalent to the gradient of Kronecker delta ($\nabla\vec{I}$), which is equal to zero so that this term can be cancelled. Also, $\nabla\vec{g}$ can

be cancelled because the gravitational acceleration is constant in the domain. In addition, the static pressure can be obtained by subtracting $\alpha_c \rho_c \vec{g}$ from p_{rgh} after solving the momentum equation and continuity equation.

Taking all these changes into account, equation 3.4 can be written as:

$$\frac{\partial \alpha_c \rho_c \vec{U}_c}{\partial t} + \nabla \cdot (\alpha_c \rho_c \vec{U}_c \vec{U}_c) = -\nabla p_{rgh} + \nabla \cdot (\alpha_c \rho_c \tau_{st}) - \vec{g} \cdot \vec{h} \nabla (\alpha_c \rho_c) - \vec{S}_P^B \quad (3.8)$$

3.3.4 New set of governing equations for *DPConFoam*

In summary, the governing equations of the *DPConFoam* solver were obtained after performing three modifications on *denseParticleFoam*, and the continuity equation and the momentum equation are defined in equations 2.41 and 3.8, respectively.

3.3.5 Case Four: Validation of *DPConFoam*

To evaluate the accuracy of *DPConFoam* when solving a single-phase flow with particles, the simulation results were numerically compared with ones obtained from the unmodified solver *denseParticleFoam*. The selected validation flow problem should have the following characteristics:

- The simulation contains a single phase air, and it is incompressible.
- The process is isothermal.
- Particles have uniform size.
- The gravitational acceleration is constant at 9.81 $\mathrm{m \cdot s^{-1}}$ downward.
- The process is transient.

Both solvers simulated a fluidized bed case with an initial bed height of 0.0465 m and 3982 particles under the same setup. The geometry and mesh for the simulations are employed as in Fernandes [1] and shown in Figure 3.9. Figure 3.9 also shows the initial positions of all particles and mesh of the domain. To decrease the computational time,

some setup parameters were modified, specifically the number of particles reduced from 24750 to 3982. In addition, parameters used in the setup for the current case and boundary conditions are listed in table 1 and 2, respectively. Note that the interstitial inlet velocity boundary condition can keep the overall fluid flow rate constant over time according to the non-constant solid fraction at the boundary faces by adjusting the fluid inlet velocity [1]. Moreover, the k-omega SST model was used for solving turbulence. The simulation time for both cases is 2 s with the same time step of $5{\cdot}10^{-5}$ s. The parameters used in the setup and boundary conditions are listed in table 3.8 and 3.7.

Table 3.7: Boundary conditions of the fluidized bed case.

Boundaries	**Boundary conditions**
bottom (inlet)	interstitial inlet velocity (U:fixed value) p and p_{rgh}: fixed flux pressure k_{tur}, ν_{tur} and ω_{tur} : fixed value
top (outlet)	U: fixed value p and p_{rgh}: 0 Pa k_{tur} and ω_{tur}: zero gradient ν_{tur}: fixed value
left & right	no-slip walls p and p_{rgh}: zero gradient k_{tur} = with kqRWallFunction ν_{tur} = nutkWallFunction ω_{tur} = omegaWallFunction
front & back	symmetry

Table 3.8: Parameters used in setup of the fluidized bed case.

Variables	**Value**
Simulation geometry	
D×W×H	0.015×0.15×0.45 m
Number of cells	5400
Nodes in x, y, z directions	2×30×90
Particle properties [1]	
ρ_P	2526 kg·m^{-3}
d_P	2.5·10^{-3} m
$e_n/e_{n,w}$	0.97/0.97
τ_{pp}/τ_{pw}	0.1/0.09
ν	0.35
E	10^8 Pa
$\vec{U}_P$	1.875 m ·s^{-1}
Fluid properties	
ρ_c	1.2 kg·m^{-3}
μ_c	10^{-5} kg·m^{-1}·s^{-1}
Turbulent model	k-omega SST
Discretization schemes	Ergun Wen-Yu drag model Gravity Pressure gradient
Discretization schemes	
Transient term	Euler
Convection & diffusion terms	Gauss linear
k_{tur} **and** ω_{tur}	Gauss upwind
Convergence criteria	
k_{tur}, U_c **and** ω_{tur}	10^{-5}
p **and** p_{rgh}	10^{-6}

To check the performance of *DPConFoam*, the time-averaged results obtained from

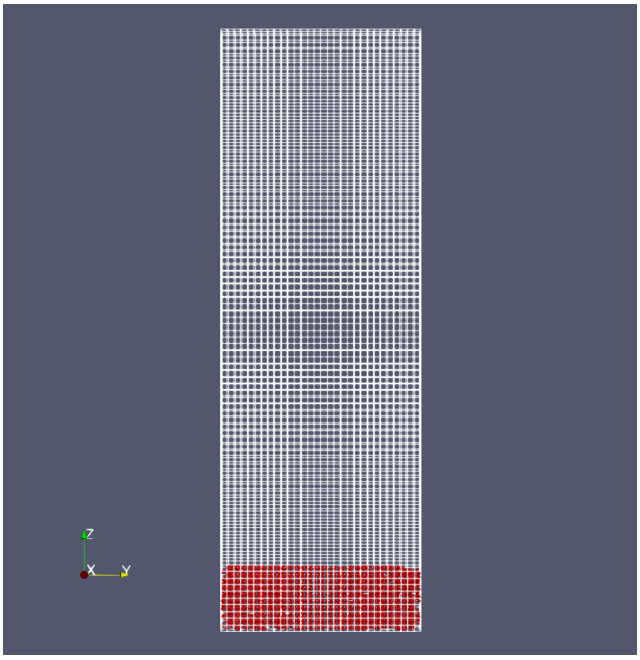


Figure 3.9: Initial positions of particles and mesh of the domain.

two solvers are numerically compared. The pressure drop and averaged height of all particles will be used to validate this solver. The purpose of monitoring the average height of all particles is to evaluate the prediction of the bed expansion dynamics [1]. The results of the comparison will be shown in section 4.5. The equation to calculate the monitored variable at each time step is given by:

$$\bar{H_P}|_t = \frac{\sum_{i=1}^{N_P} h_i}{N_P} \tag{3.9}$$

3.4 A new set of governing equations for a Two-way Coupled Multi-fluid CFD-DEM Solver

The last objective of this project is to develop a theory of coupling the multi-fluid model with *DPConFoam*. In the previous sections, the method of creating a new W-O emulsion viscosity model has been proposed. Also, the new solver *DPConFoam*, which uses the strong conservation form governing equations, has been created. With these modifications, the emulsion viscosity can be accurately calculated, and the continuity and momentum equations are ready for merging. However, there is still

one problem: merging the multi-fluid model (Eulerian model) and the DEM model (Lagrangian model). For a two-way coupled CFD-DEM solver, the information needs to be transferred between fluids and particles, which means the continuity equation, the momentum equation and the multi-fluid models need to be aware of the presence of particles. To overcome this challenge, a modified VOF equation was proposed. Therefore, the modifications of the Volume of Fluid equation (equation 2.13) and the momentum equation of *DPConFoam* (equation 3.8) will be presented.

First, the VOF method was modified by taking the continuous phase volume fraction into account. This is because the actual volume being transported in a cell is proportionally decreased by α_c. Therefore, the governing equation for one specific fluid, i, in an N fluid phases with particles can be written as:

$$\frac{\partial \alpha_c \alpha_i}{\partial t} + \nabla \cdot \left(\alpha_c \alpha_i \vec{U}_c \right) + \sum_{j=1}^{N} \nabla \cdot \left(\alpha_c \alpha_i \alpha_j \vec{U}_r \right) = 0 \tag{3.10}$$

As it is shown in equation 2.21, the default coefficient for the transient term is one, which does not fit the form of equation 3.10 because of the presence of α_c. However, when α_c is inside of the transient term, the MULES method will solve $\alpha_c \alpha_i$ instead of α_i, which will lead to wrong results of the fluid volume fractions. Therefore, one possible solution is to move the α_c outside the transient term and make it the coefficient of α_i, which is the C_m in equation 2.21. In this way, the MULES method can be applied again, and α_c will be evaluated individually. To achieve this goal, the transient term of equation 3.10 can be split into two terms by using the product rule, and equation 3.10 can be rewritten as:

$$\alpha_c \frac{\partial \alpha_i}{\partial t} + \alpha_i \frac{\partial \alpha_c}{\partial t} + \nabla \cdot \left(\alpha_c \alpha_i \vec{U}_c \right) + \sum_{j=1}^{N} \nabla \cdot \left(\alpha_c \alpha_a \alpha_j \vec{U}_r \right) = 0 \tag{3.11}$$

This equation cannot be directly solved as there are two transient terms, but $\alpha_i \frac{\partial \alpha_c}{\partial t}$ can be replaced by $-\alpha_i \nabla \cdot \left(\alpha_c \vec{U}_c \right)$ as a result of the continuity equation 2.41. In this way, the number of transient terms decreased to one, and $-\alpha_i \nabla \alpha_c \vec{U}_c$ can be moved to the right-hand side (RHS) of the VOF equation and solved explicitly. In the MULES

method, this term can be represented by $\alpha_i Q_p$, which implies that $Q_p = \nabla \cdot \left(\alpha_c \vec{U}_c\right)$. Therefore, the final form of the VOF equation for multi-fluid flows with particles can be expressed as:

$$\alpha_c \frac{\partial \alpha_i}{\partial t} + \nabla \cdot \left(\alpha_c \alpha_i \vec{U}_c\right) + \sum_{j=1}^{N} \nabla \cdot \left(\alpha_c \alpha_a \alpha_j \vec{U}_r\right) = \alpha_i \nabla \cdot \left(\alpha_c \vec{U}_c\right) \tag{3.12}$$

and its discretized form of equation 3.12 can be found as:

$$\frac{\alpha_c \alpha_i^n - \alpha_c \alpha_i^{n-1}}{\delta t} + \underbrace{\sum_f \left[\alpha_c \alpha_i U\right]_f \cdot \vec{S}_f + \sum_{j=1}^{n} \sum_f \left[\alpha_c \alpha_i \alpha_j \vec{U}_r\right]_f \cdot \vec{S}_f}_{=\Phi_c} = \alpha_i^n Q_p \tag{3.13}$$

In MULES solver, the volume fraction of each fluid at the current time step is solved by:

$$\alpha_i^n = \frac{\frac{\alpha_c \alpha_i^{n-1}}{\delta t} - \Phi_c}{\frac{\alpha_c}{\delta t} - Q_p} \tag{3.14}$$

In terms of the momentum equation, equation 3.4 needs to take the surface tension force into account, and the final version of the momentum equation can be written as:

$$\frac{\partial \alpha_c \rho_c \vec{U}_c}{\partial t} + \nabla \cdot (\alpha_c \rho_c \vec{U}_c \vec{U}_c) = -\nabla p_{rgh} + \nabla \cdot (\alpha_c \rho_c \tau_{st}) + \alpha_c \vec{f}_s + \vec{g} \cdot \vec{h} \nabla (\alpha_c \rho_c) - \vec{S}_P^B \tag{3.15}$$

where $\vec{f}_s$ is the surface tension force, and it is defined in equation 2.20. Also, when particles are present in a cell that has the interface of two fluids, it is not possible to determine if they are located on the surface or not. However, when particles are located on the free surface, the contact area for two fluids would decrease. Therefore, the surface tension force is assumed to be proportionally decreased by the continuous phase volume fraction.

This new set of governing equations will be implemented in OpenFOAM in future works and be validated by testing the sand separation efficiency of the separator at different flow rates and water cuts. Also, the pressure drop and flow rate along the pipe will be monitored. The sand separation efficiency obtained from the experiments can be found in Appendix D.

3.5 Conclusion

In this chapter, a new viscosity model that is dedicated to evaluating the W-O emulsion viscosity at different water cuts and shear rates was proposed. Also, the *denseParticleFoam* was modified to *DPConFoam*. *DPConFoam* uses the strong conservation form of governing equations, includes the density in the momentum equation, and uses the modified pressure p_{rgh}. Finally, the theory of a two-way coupled multi-fluid CFD-DEM solver was developed.

To validate the new viscosity model and the *DPConFOAM*, four cases were proposed in this chapter. The results and discussion of these cases can be found in chapter 4.

Chapter 4

Results and Discussion

4.1 Introduction

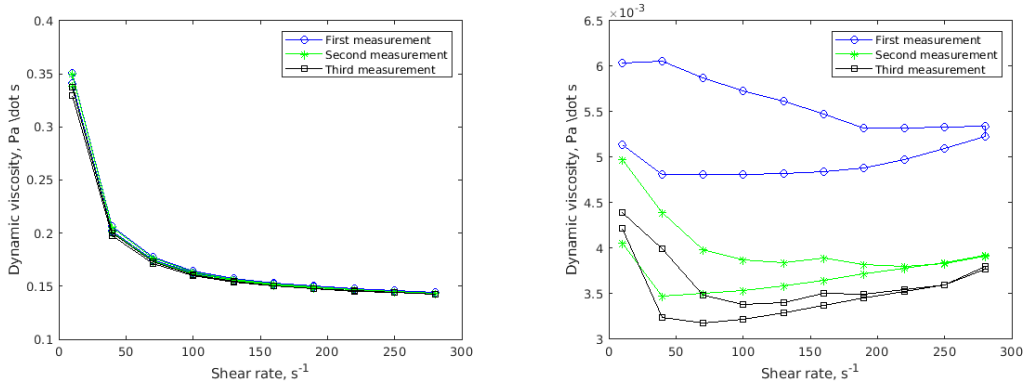
This chapter emphasizes the analysis of the experimental and simulation results, which will be explained in the next four sections. First, the measured viscosity data will be analyzed, and the functions of the new viscosity model will be determined in section 4.2.1. Also, the new viscosity model will be implemented in the OpenFOAM (*multiphaseInterFoam*), and the solver with this model will be named as *multiphaseEmulsionFoam*. In addition, a quick test will be performed to validate the correct implementation and proper working of (Case One) the new viscosity model. In case two, the W-O mixture will be simulated in a horizontal pipe with both *multiphaseInterFoam* and *multiphaseEmulsionFoam*, and their velocity profile and dynamics of the interface will be compared with results from the literature [61][62]. In case three, the simulation results of the modified flow loop case will be numerically compared. As mentioned in the previous chapter, the same simulation case is performed with both *multiphaseInterFoam* and *multiphaseEmulsionFoam*. Therefore, the different in the solution fields caused by the new viscosity model will be discussed in section 4.5. Finally, a fluidized bed case simulated by *DPConFoam* will be numerically compared with the results obtained by Fernandes et al.[1].

4.2 New Viscosity Model Development

The new viscosity model was developed based on the data measured by Gustavo Cifuentes Dias, a master's student of Dr. Sean Sanders in the Department of Chemical and Material engineering, University of Alberta.

4.2.1 Viscosity Data Analysis

The viscosity of six different water cut emulsions (0%, 20%, 40%, 60%, 70%, 80%) was measured at ten different shear rates. The measurements were repeated three times, and the average values were calculated for later analysis. Some emulsions were stable over time, but others were unstable, a comparison between a stable and an unstable emulsion viscosity with respect to the shear rate can be seen in figure 4.1. When the emulsion was stable during measurements, the shear rate viscosity curves are close to each other. Otherwise, a clear separation was observed. This was used to determine where the phase inversion happened. In this case, the emulsion was stable until a water cut of 70%. Therefore, the 80% water cut emulsion viscosity was not used in this study. In total, 50 average data points were obtained and are listed in table 4.1.



(a) (b)

Figure 4.1: The plot of measured viscosity against shear rate for 40% emulsions (a) and 80% emulsions (b).

Table 4.1: Average value of the W/O emulsion viscosity at different water cut and shear rate

Shear rate (s^{-1})	**10**	**40**	**70**	**100**	**130**
Water cut (%)	**Dynamic viscosity (Pa·s)**				
0	0.04431	0.04432	0.04432	0.04433	0.04439
20	0.16102	0.10883	0.09697	0.09086	0.08687
40	0.34099	0.20217	0.17429	0.16217	0.15565
60	0.94197	0.52058	0.44037	0.40329	0.38121
70	1.63336	0.91382	0.76683	0.69417	0.64759
Shear rate (s^{-1})	**160**	**190**	**220**	**250**	**280**
Water cut (%)	**Dynamic viscosity (Pa·s)**				
0	0.04442	0.04447	0.04452	0.04457	0.04463
20	0.08396	0.08174	0.08004	0.07874	0.07774
40	0.15159	0.14872	0.14658	0.14492	0.14356
60	0.36611	0.35493	0.34600	0.33843	0.33190
70	0.61371	0.58722	0.56518	0.54617	0.52937

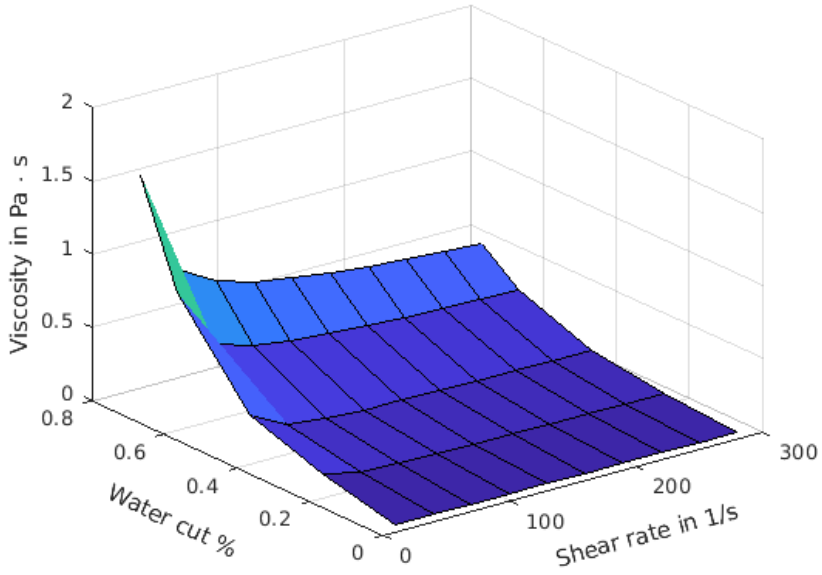


Figure 4.2: 3D plot based on the measured emulsion viscosity.

Based on the measured data, a 3D plot was generated and shown in figure 4.2. It can be seen that the Shell Tellus S2 VX 22 oil shows a Newtonian behaviour, which means its viscosity does not vary much with respect to the shear rate. In addition, the viscosity of emulsion at other water cuts tends to increase when the water cut increases and the shear rate decreases. Furthermore, the maximum viscosity was obtained at 70% water cut for all shear rates, and the shear rate affects the viscosity of emulsion more significantly at lower water cut.

To further analyze the measured data, exponential equations and power-law functions were used to curve fit the measured data from the perspective of water cut and shear rate, respectively. On top of that, the obtained shear rate and water cut functions and their correlation coefficients are listed in table 4.2 and 4.3.

Table 4.2: Shear rate functions obtained from curve fitting the measured viscosity.

Water cut (%)	**Viscosity vs shear rate**	**Correlation coefficient**
0	0.044	0.9902
20	$0.259\dot{\gamma}^{-0.2199}$	0.9902
40	$0.572\dot{\gamma}^{-0.258}$	0.9719
60	$1.783\dot{\gamma}^{-0.310}$	0.983
70	$3.220\dot{\gamma}^{-0.331}$	0.9944

Herein, $\dot{\gamma}$ is the shear rate and ϕ_w is the volume fraction of water.

It can be seen that the correlation coefficients for all conditions are above 0.97, especially the equations that are related to water cut have correlation coefficient above 0.99. This indicates that the exponential function is the proper base function to calculate the viscosity of different water cut emulsions.

By comparing the correlation functions in the tables above, the water cut functions have similar coefficients, whereas the coefficients of the shear rate functions do not have a specific tendency. Therefore, the representational function method can be utilized, which consists of splitting the water cut functions into a few groups based

Table 4.3: Water cut functions obtained from curve fitting the measured viscosity.

Shear rate (s^{-1})	**Viscosity vs water cut**	**Correlation coefficient**
10	$0.049185e^{4.9858\phi_w}$	0.9969
40	$0.043595e^{4.2043\phi_w}$	0.9966
70	$0.042127e^{3.9752\phi_w}$	0.9953
100	$0.041402e^{3.8502\phi_w}$	0.9947
130	$0.040999e^{3.7652\phi_w}$	0.9944
160	$0.040722e^{3.702\phi_w}$	0.9944
190	$0.040545e^{3.6504\phi_w}$	0.9944
220	$0.040443e^{3.6053\phi_w}$	0.9944
250	$0.040406e^{3.5641\phi_w}$	0.9945
280	$0.040421e^{3.5254\phi_w}$	0.9947

on their coefficients. In this study, the water cut functions were divided into three groups, and each region is governed by a representational function, which is listed in table 4.4. Also, the viscosity functions after the inversion point can be represented by exponential functions as stated in [7]. As a result, exponential functions were used to represent the region between the maximum value of each representational function and the water viscosity. Thus, the emulsion viscosity at the entire water cut range can be represented by a Heaviside step function, equation 3.1. The complete form of the new equation will be demonstrated in the next section.

4.2.2 Viscosity Functions of the New Viscosity Model

In table 4.4, the viscosity functions for the W/O cases are listed. To optimize the correlation between the representational functions and measured data and the efficiency of the solver, three functions were obtained for different shear rate ranges.

Note that the coefficients of the viscosity functions after the inversion point are very large and the exponent coefficients are very small. This is mainly due to the lack of measured data after the inversion point, and it is known that the trend of the

Table 4.4: Representational function of each shear rate zone and water cut

Function	**Shear rate (s^{-1})**	ϕ_w **<=0.7**	ϕ_w **>0.7**
1	**<40**	$0.04919e^{4.986\phi_w}$	$4.922{\cdot}10^{7}e^{-24.6195\phi_w}$
2	**>=40 & <100**	$0.04237e^{4.010\phi_w}$	$3.075{\cdot}10^{6}e^{-21.8464\phi_w}$
3	**>=100**	$0.04071e^{3.666\phi_w}$	$1.205{\cdot}10^{6}e^{-20.9096\phi_w}$

viscosity change after the inversion point should also follow the trend of an exponential function. But, if a linear function was used, the viscosity of the emulsion after the inversion point would be significantly overestimated. Therefore, the exponential function was selected instead of a linear function to connect the maximum viscosity and the viscosity of the water.

Table 4.5: Correlation coefficient of the representational functions at each measured shear rate

Function	**Shear rate (s^{-1})**	**Correlation coefficient**
1	**10**	0.999
2	**40**	0.957
	70	0.993
	100	0.988
3	**130**	0.976
	160	0.987
	190	0.993
	220	0.995
	250	0.994
	280	0.991

The correlation coefficients of each representational function to the measured viscosity in their covered region are shown in table 4.5. It is shown that the correlation coefficients for all different shear rates are above 0.95, which indicates these three water cut functions are sufficient to represent the viscosity change in the entire range

of shear rates.

4.2.3 Implementation of the New Viscosity Model

The equations in table 4.4 have been implemented in OpenFOAM, which can be found in appendix B. The emulsion viscosity in the control volume is calculated based on the local water cut and shear rate. The algorithm of the new model is shown in figure 4.3.

To optimize the efficiency of the solver, users need to specify if they want to use the new viscosity model. This is controlled by a variable called 'emulsion' defined in the *fvSolution* dictionary file. If it is set to 'yes', the new viscosity model will be used, and otherwise, the linear function will be utilized. Also, to prevent the order of the continuous phases listed in *transportProperties* dictionary file affecting the viscosity evaluation, three if statements were implemented to distinguish the emulsion components (water and oil). Depending on the conditions, the viscosity of emulsions is calculated by using the new viscosity model and then linearly combined with other non-emulsion components.

In addition, the viscosity equations listed in table 4.4 are dedicated to the cases that only have water and oil. If there is non-emulsion component in the system, such as air, the relative water cut needs to be known for viscosity calculation. The relative water cut is defined as:

$$\phi_{wr} = \frac{\phi_w}{1 - \sum_i^N \phi_{ne}} \tag{4.1}$$

where ϕ_{wr} and ϕ_w are the relative and absolute volume fraction of water, and $\phi_{ne,i}$ is volume fraction of each non emulsion component. Also, a small value close to the numerical precision (10^{-37}) needs to be added in the denominator to avoid numerical failure.

As seen from the following codes (emulsion viscosity functions), small values are used in statements to ensure each representational function covers the correct range of water cut or shear rate. Also, a plot of this new viscosity model is shown in figure

4.4.

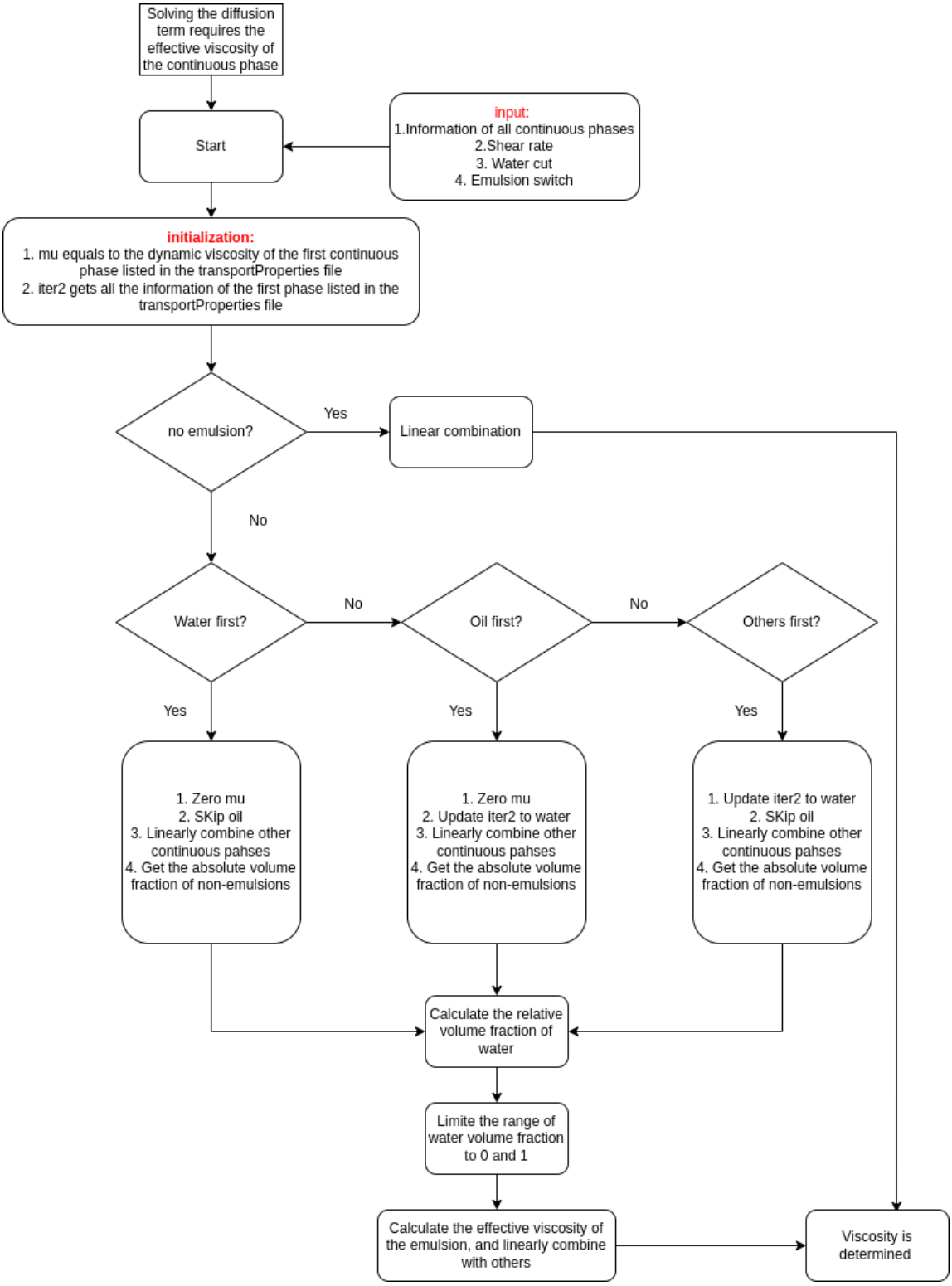


Figure 4.3: Algorithm of the new viscosity model.

```
mu += (1-nonemulsion)*
(
    neg(dimensionedScalar(dimTime, 1.0)*strainRate()-40-VSMALL)*
    (
        neg(alpha_w - 0.7+VSMALL)*0.04919*exp(4.986*alpha_w)
       +pos(alpha_w-0.7+VSMALL)*49216934.2723*exp(-24.6195*alpha_w)
    )
    +pos(dimensionedScalar(dimTime, 1.0)*strainRate()-40-VSMALL)*
    neg(dimensionedScalar(dimTime, 1.0)*strainRate()-100-VSMALL)*
    (
        neg(alpha_w-0.7+VSMALL)*0.04237*exp(4.010*alpha_w)
       +pos(alpha_w-0.7+VSMALL)*3074612.3957*exp(-21.8464*alpha_w)
    )
    +pos(dimensionedScalar(dimTime, 1.0)*strainRate()-100-VSMALL)*
    (
        neg(alpha_w - 0.7+VSMALL)*0.04071*exp(3.666*alpha_w)
       +pos(alpha_w - 0.7+VSMALL)*1204812.7792*exp(-20.9096*alpha_w)
    )
)*dimensionedScalar(dimViscosity,1)*dimensionedScalar(dimDensity,1);
```

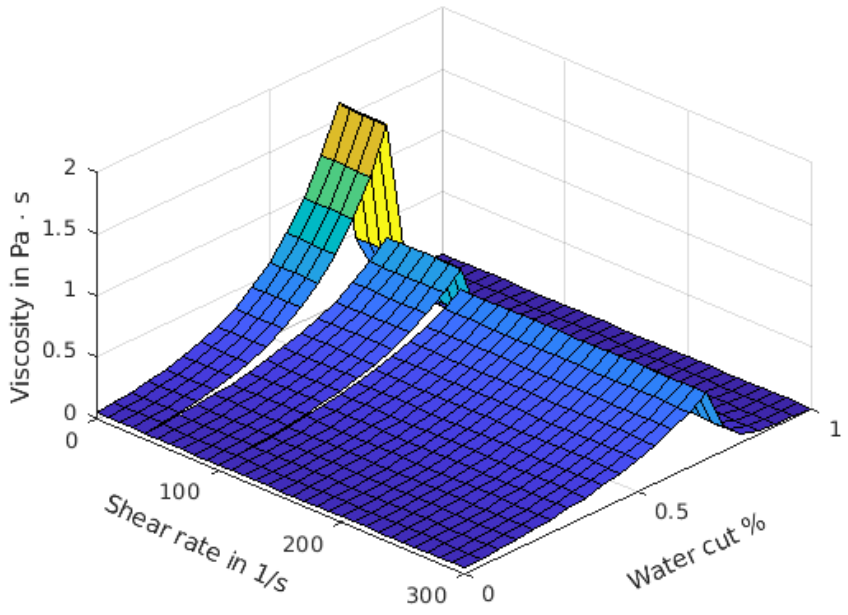


Figure 4.4: The emulsion viscosity plot obtained from the new viscosity model.

4.2.4 Case One: Tests for the New Viscosity Model

This is a quick test for the new viscosity model, which involves six cases and is used to check the following two criteria. First, ensure the order of the phases listed in the

transportProperties dictionary file does not affect the viscosity evaluation. Second, check if the implemented function returns the correct values. The simulation results were obtained after one time step and compared with empirical values obtained from the new viscosity equation. In this way, the algorithm of the implemented new viscosity model can be tested if it is bug-free. Here are the assumptions for these simulations:

- Fluids are incompressible and immiscible.
- The flow is steady state.
- The flow is isothermal.
- The gravitational acceleration is constant at 9.81 $\mathrm{m{\cdot}s^{-1}}$ downward.

The simulations were performed in a 2D domain, as shown in figure 4.5. The mesh, in this case, is randomly generated since the mixture's viscosity should not be affected by the size of the mesh. Thus, no mesh independence check will be performed in this case. In addition, the four sides of the domain were set to no-slip wall boundary conditions.

The fluids in the domain are the same and at a stationary state for each test, and the fluid mixtures are combinations of water(W), oil(O) and air(A). The order of these three phases listed in the *transportProperties* dictionary file and their volume fractions (VF) are shown in table 4.7. In addition, these three phases' properties were fabricated to catch errors in the code easily, and they are listed in table 4.6.

Table 4.6: Properties of fluids that are used in the test cases

Name	**Density ($\mathbf{kg{\cdot}m^{-3}}$)**	**Kinematic viscosity ($\mathbf{m{\cdot}s^{-1}}$)**
Water	1	10^{-6}
Oil	1	$5.5{\cdot}10^{-5}$
Air	1	0.01

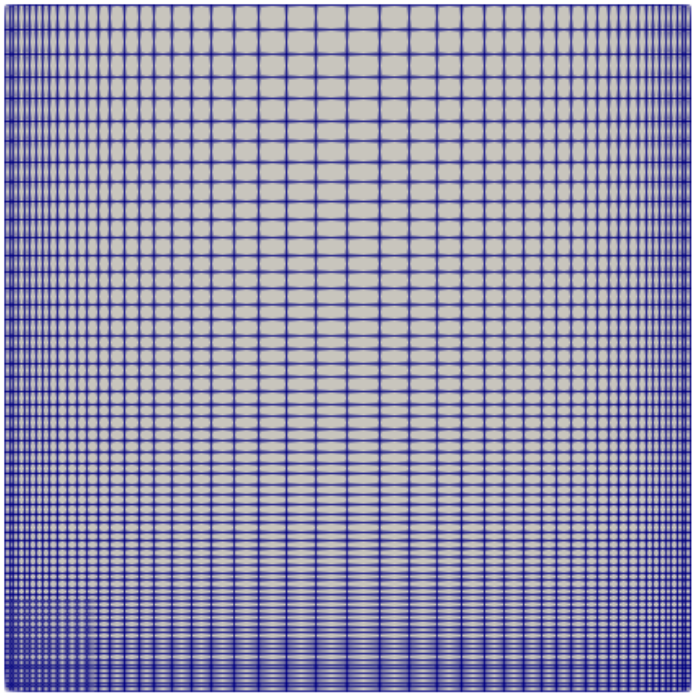


Figure 4.5: The geometry and mesh used in the new viscosity model test case.

Table 4.7: Viscosity equation test cases results [viscosity listed in this table are kinematic viscosity, in $m^2 \cdot s^{-1}$]

Case	Combination	VF (%)	Simulated results	Empirical value
1	**W-A-O**	**50-50-0**	0.0050005	0.0050005
2	**W-O-A**	**40-60-0**	0.36144	0.36144
3	**O-A-W**	**50-50-0**	0.0050275	0.0050275
4	**O-W-A**	**30-20-50**	0.18572	0.18572
5	**A-W-O**	**50-20-30**	0.18572	0.18572
6	**W-O-A**	**20-30-50**	0.18572	0.18572

Based on the obtained results, the implemented functions can return the expected effective viscosity, and no bug is found when running the code. This indicates the viscosity functions are properly and correctly implemented in the OpenFOAM.

4.3 Case Two: W-O mixtures in a Horizontal Pipe

4.3.1 Convergence Test and Mesh Independence Check

The simulations performed on the finest mesh used the coarse mesh results as initial conditions. In this way, the simulations are able to reach steady-state faster. The overall volume fraction of oil is monitored over time, shown in figure 4.6. It can be seen from the figure that the simulation results obtained from different meshes reached a steady-state regime. Even though there are oscillations in results, the magnitude of the change is within 1%.

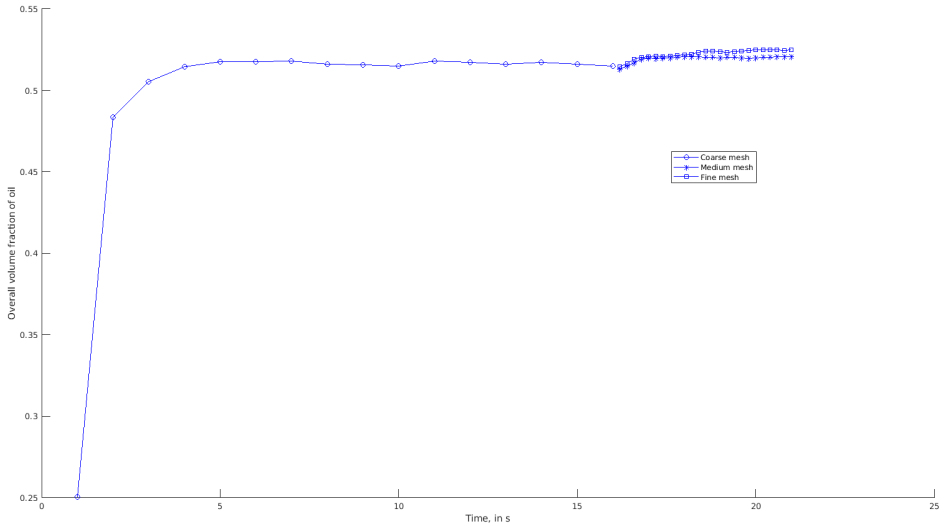


Figure 4.6: Overall volume fraction of oil over time obtained from the *multiphaseEmulsionFoam* case.

In this case, the mesh independence check is based on the overall volume fraction of the oil at three mesh sizes after the flow reaches a steady state. The results are shown in figure 4.7 and table 4.8.

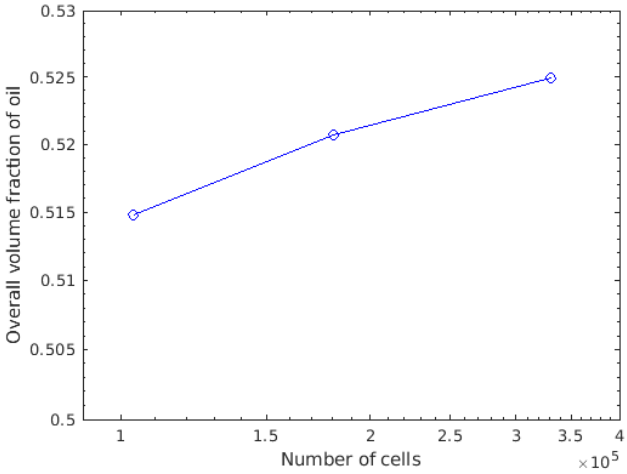


Figure 4.7: The oil volume fraction of the entire domain obtained from the *multiphaseEmulsionFoam* case.

Table 4.8: Mesh refinement test.

Mesh	**Number of cells**	**Oil volume fraction**
Coarse	$N_c = 103602$	0.5130
Medium	$N_m = 180726$	0.5204
Fine	$N_f = 330184$	0.5249

The error order of the simulations is calculated based on the Richardson extrapolation, where in this case can be calculated as:

$$\epsilon_o = \frac{\log\left(\frac{\phi_{o,m}-\phi_{o,c}}{\phi_{o,f}-\phi_{o,m}}\right)}{\log\bar{r}} \tag{4.2}$$

where $\bar{r}$ is the average refinement ratio, and it is calculated by:

$$\bar{r} = \overline{r_1 + r_2} = \overline{\left(\frac{N_m}{N_c}\right)^{1/3} + \left(\frac{N_f}{N_m}\right)^{1/3}} \tag{4.3}$$

In this case, it is equal to 1.21. The order of error obtained for this case is 1.67, which is close to the expected order of 2. Thus, the simulation results show that they have reached mesh independence. Also, the expected true value (ϕ_{exact}) for this case can be calculated by:

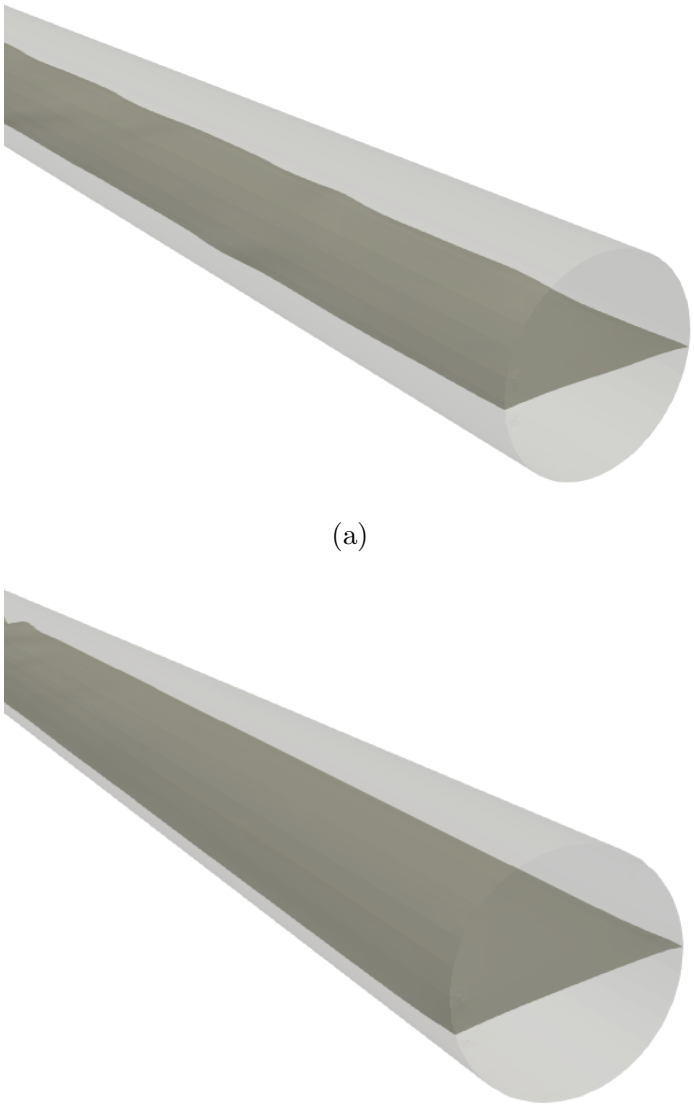
$$\epsilon_{\mathrm{h}}^{\mathrm{d}} = \frac{\phi_{o,f} - \phi_{o,m}}{\bar{r}^{\epsilon_o} - 1} \tag{4.4}$$

$$\phi_{\mathrm{exact}} \approx \phi_{f,o} + \epsilon_{\mathrm{h}}^{\mathrm{d}} \tag{4.5}$$

where $\epsilon_{\mathrm{h}}^{\mathrm{d}}$ is the discretization error, and ϕ_{exact} is calculated as 0.536. The relative difference between the expected true value and the fine mesh result is 2.05%, which is acceptable. The discussion in the following sections will be based on the fine mesh results.

4.3.2 Results and Discussion

As mentioned in section 3.2.3, the velocity profile at the downstream of the pipe and the flow regime will be compared. Edomwonyi et al. [62] performed experiments to observe the flow regimes of W-O mixtures under different conditions. Their experimental results show that there are always waves present at the interface. When the velocity of the fluids is over 0.1 $\mathrm{m{\cdot}s^{-1}}$ ($\mathrm{Re} > 210$), the waves are easily noticed. In contrast, when the velocity of the flow is small, the waves tend to have a small magnitude and long wavelength. Even though the Re of the oil, in this case, is much smaller than the experimental condition, the water phase may dominate the dynamics of the flow since the Re of the water phase is much greater than the oil phase in both experiments and simulations. Thus, small waves are expected to be observed. Section views of the simulation results obtained from the *multiphaseInterFoam* and *multiphaseEmulsionFoam* can be seen in figure 4.8.



(a)

(b)

Figure 4.8: The interface between water and oil obtained from the *multiphaseEmulsionFoam* (a. top) and *multiphaseInterFoam* (b. bottom).

It can be seen from the figures above that the interface obtained from the *multiphaseEmulsionFoam* case shows wave structures at the downstream of the pipe. In contrast, the *multiphaseInterFoam* case shows a flat interface. This indicates that the *multiphaseEmulsionFoam* can properly predict the dynamics of the interface and lead to a better estimation of the flow regime for W-O mixtures.

Also, a monitoring line was placed in the pipe to check the velocity profile. The location of the monitoring line can be seen in figure 4.9.

Figure 4.9: The location of the monitoring line in the horizontal pipe case.

As stated in Huang et al.'s [61] paper, the velocity gradient should change at the interface of two fluids. This is mainly because of the shear stress, which is defined in equation 4.6. The shear stress should be the same on both sides of the interface. Thus, the velocity gradient will change if two fluids have different viscosity. In this case, the water viscosity is lower than the oil viscosity, and the velocity gradient on the water side should be sharper than on the oil side.

$$\tau = \mu_c \frac{\mathrm{d}\vec{U}_c}{\mathrm{d}y} \tag{4.6}$$

The velocity profiles along the monitoring line for two cases can be seen in figure 4.10. It is noticed that the change of the velocity gradient is only observed in the *multiphaseEmulsionFoam* case, whereas the *multiphaseInterFoam* case shows a similar velocity gradient on both sides of the interface. Thus, the *multiphaseEmulsionFoam* shows a better prediction of the dynamics of the flow.

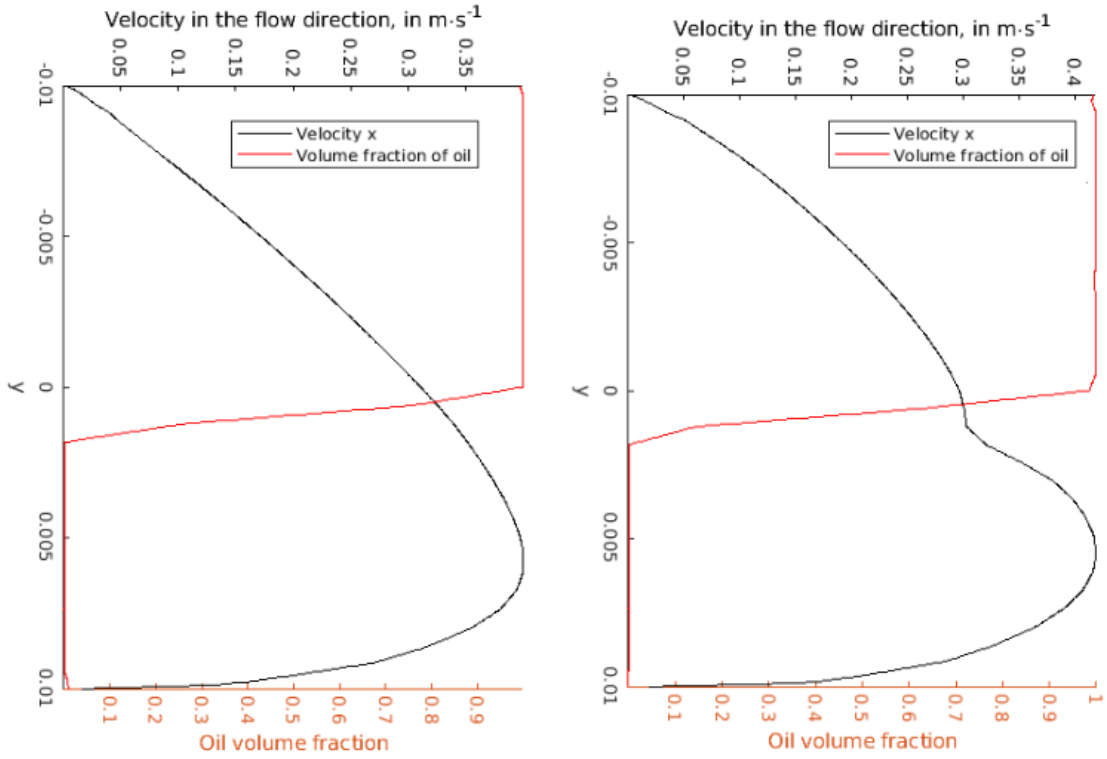


Figure 4.10: The velocity profile along the monitoring line obtained from the *multiphaseInterFoam* case (left) and the *multiphaseEmulsionFoam* case (right).

4.4 Case Three: Flow Loop Simulation

4.4.1 Convergence Test and Mesh Independence Check

The residence time of the flow in the loop is around 14 s, which is calculated based on equation 4.7. However, the horizontal inlet injected 30% water cut emulsions instead of pure water, which means it would take longer time for the oil volume fraction to reach a constant value. Therefore, the simulations were running to 40 s to reach a fully steady state.

$$t_r = \frac{V_{geo}}{\dot{V}_{flow}} \tag{4.7}$$

where t_r is the residence time of the flow, $\dot{V}_{flow}$ is the total inlet volume flow rate, and V_{geo} is the total volume of the geometry.

To check the convergence of the simulation, two criteria were monitored:

- Convergence plot of the velocity field, as shown in figure 4.11.

- Oil volume fraction of the entire domain over time, as shown in figure 4.12.

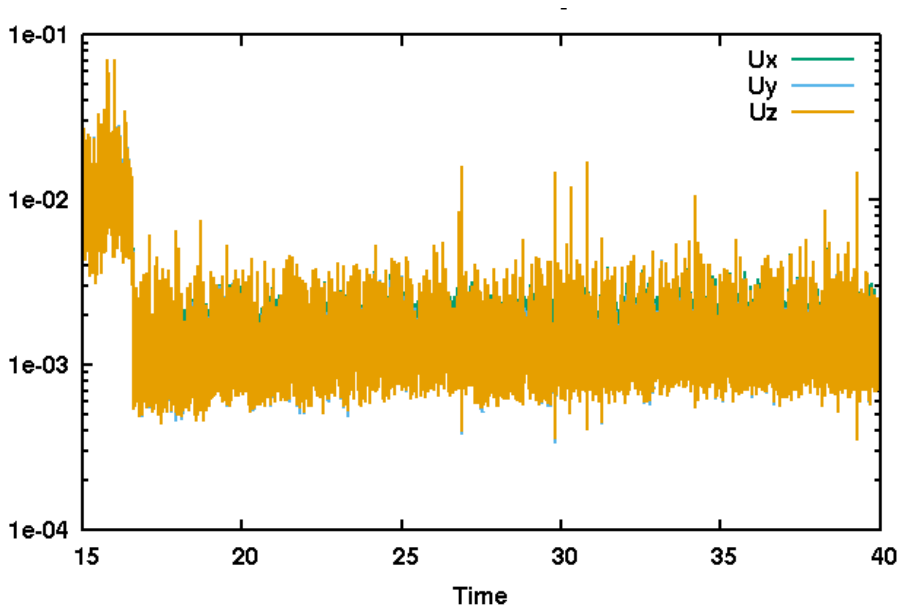


Figure 4.11: The residual plot of the velocity obtained from the *multiphaseEmulsionFoam* case.

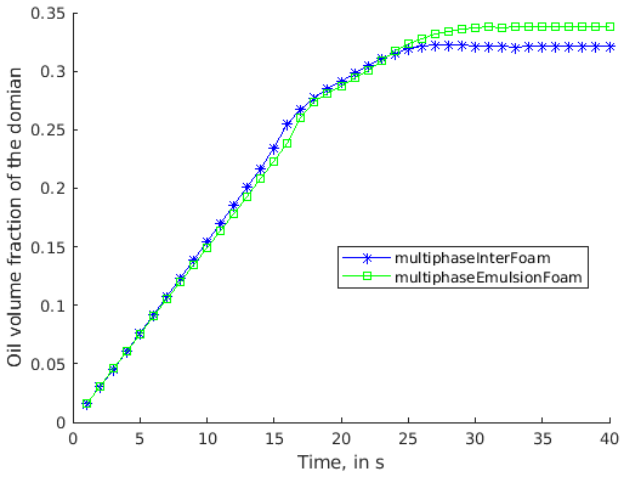


Figure 4.12: The oil volume fraction of the entire domain obtained from the *multiphaseEmulsionFoam* case.

As it can be seen from the figures 4.13, both the convergence plot and the oil volume fraction plot have reached a steady state region. Therefore, it can be concluded that the simulation at 40 s has reached steady state.

The mesh independence check is based on the overall volume fraction of the oil at three mesh sizes after the flow reach steady state. The results are shown in figure 4.13 and table 4.9

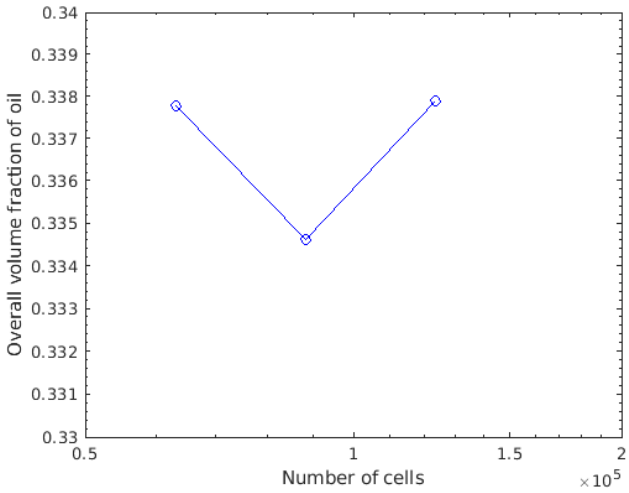


Figure 4.13: The oil volume fraction of the entire domain obtained from the *multiphaseEmulsionFoam* case.

Table 4.9: Mesh refinement test.

Mesh	**Number of cells**	**Oil volume fraction**
Coarse	$N_c = 63187$	0.33779
Medium	$N_m = 88298$	0.33460
Fine	$N_f = 123765$	0.33789

The order of error was not able to be obtained as the results were not in the asymptotic region. This is mainly due to the limited resources, and the current fine mesh has reached the maximum computational power that can be provided. The simulation will be performed on a finer mesh in future works. On top of this, there is only a small difference of 1% between the results obtained from the fine mesh and medium mesh. Therefore, the discussion in the following sections will be based on the fine mesh results.

4.4.2 Results and Discussion

In this section, the simulation results of the modified flow loop obtained from *multiphaseInterFoam* and *multiphaseEmulsionFoam* are compared. To get a clear view of the flow, the pipe was rotated 35° forward. Also, a surface contour plot was created where the volume fraction of water is 50%, representing the free surface between oil and water, which are shown in figure 4.16 and 4.17 for *multiphaseInterFoam* and *multiphaseEmulsionFoam*, respectively.

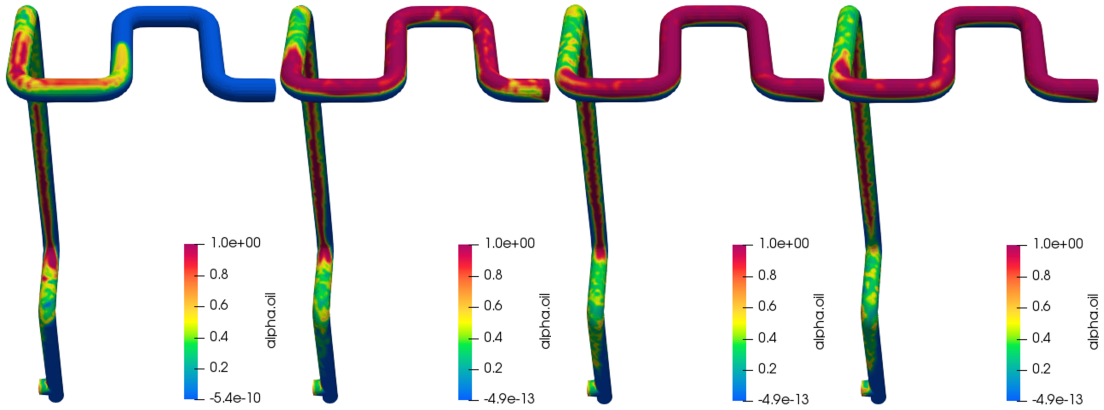


Figure 4.14: The oil volume fraction plot obtained from *multiphaseEmulsionFoam* case at 10, 20 , 30 and 40 s, respectively.

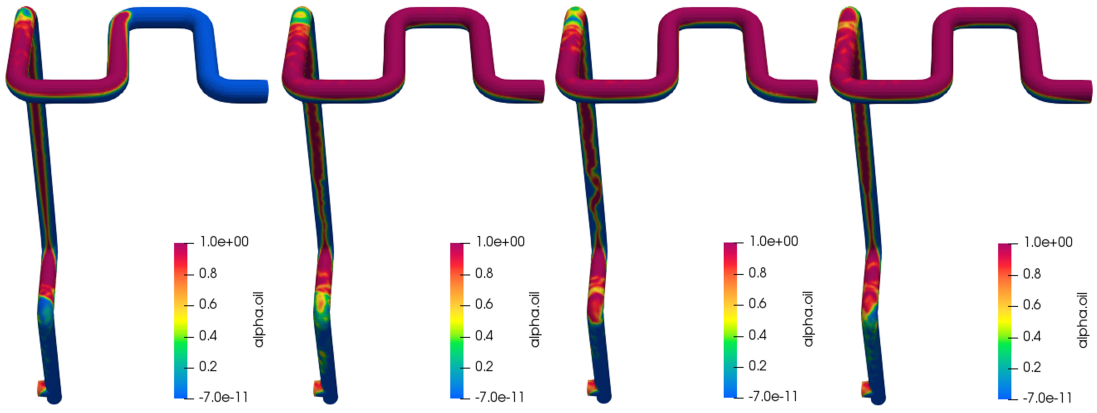


Figure 4.15: The oil volume fraction plot obtained from *multiphaseInterFoam* case at 10, 20 , 30 and 40 s, respectively.

Figure 4.16: The contour plot of the free surface obtained from *multiphaseEmulsionFoam* case at 10, 20, 30 and 40 s, respectively.

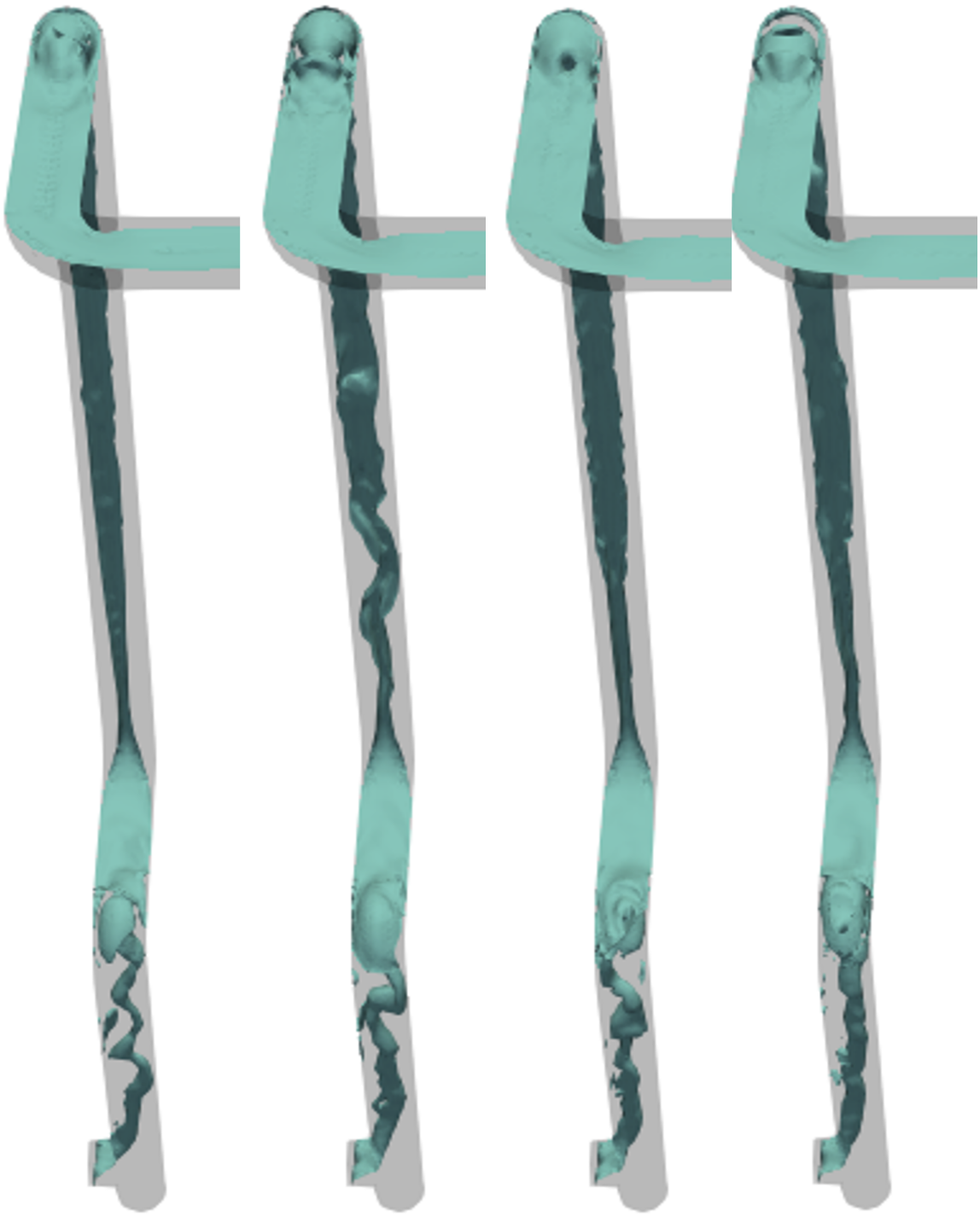
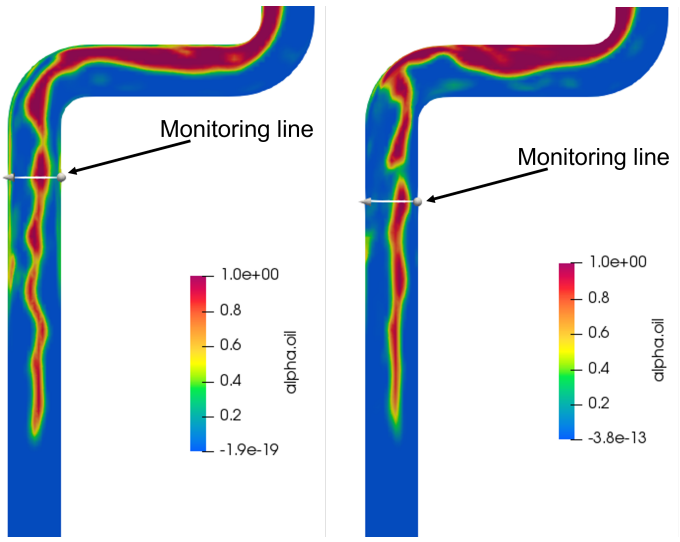


Figure 4.17: The contour plot of the free surface obtained from *multiphaseInterFoam* case at 10, 20, 30 and 40 s, respectively.

It can be seen from the figures above, that the flow developed slower in the *multiphaseEmulsiuonFoam* case than in the *multiphaseInterFoam* case. In figure 4.12, when the slope of the curve starts changing, the oil flow reaches the outlet. This took approximately 17 s and 16 s for the *multiphaseEmulsiuonFoam* case and the *multiphaseInterFoam* case, respectively. This implies that the *multiphaseEmulsionFoam* requires a longer time to develop. After reaching a steady state, oil and water separation is found since the maximum oil volume fraction changed from 0.7 to 1. The separation mainly happened in the horizontal pipe due to the large density ratio, especially after the pipe changes from vertical to horizontal. In the original *multiphaseInterFoam* case, the oil and the water tend to separate immediately after entering the horizontal section. In contrast, the flow in the new *multiphaseEmulsionFoam* case shows a slow separation, and it requires a longer distance for the flow to get fully separated. This could be a result of the high viscosity of the emulsion. In addition, it was observed that the oil mixture tended to move at the center of the pipe in both cases, which is mainly because of the effect similar to drag reduction agents [66]. However, the flow regime near the inlet obtained in the *multiphaseEmulsionFoam* case is more like a churn flow [67].

To investigate the dynamic viscosity change in the domain, monitoring lines were used, which are shown in figure 4.18. In total, the viscosity values were collected from 45 control volumes with a resolution of 1000.

In terms of the *multiphaseEmulsionFoam* case, the dynamic viscosity of each control volume located on the line was compared with the empirical value evaluated from the new viscosity equation. It is found that the values are matched, which means the implemented viscosity model is correctly implemented in OpenFOAM and it is working properly. The viscosity change and the water cut along the monitoring lines for the *multiphaseEmulsionFoam* case and the *multiphaseInterFoam* case can be seen in figures 4.19 and 4.20, respectively.



(a) (b)

Figure 4.18: Locations of the monitoring lines used in *multiphaseEmulsionFoam* (left) and *multiphaseInterFoam* (right).

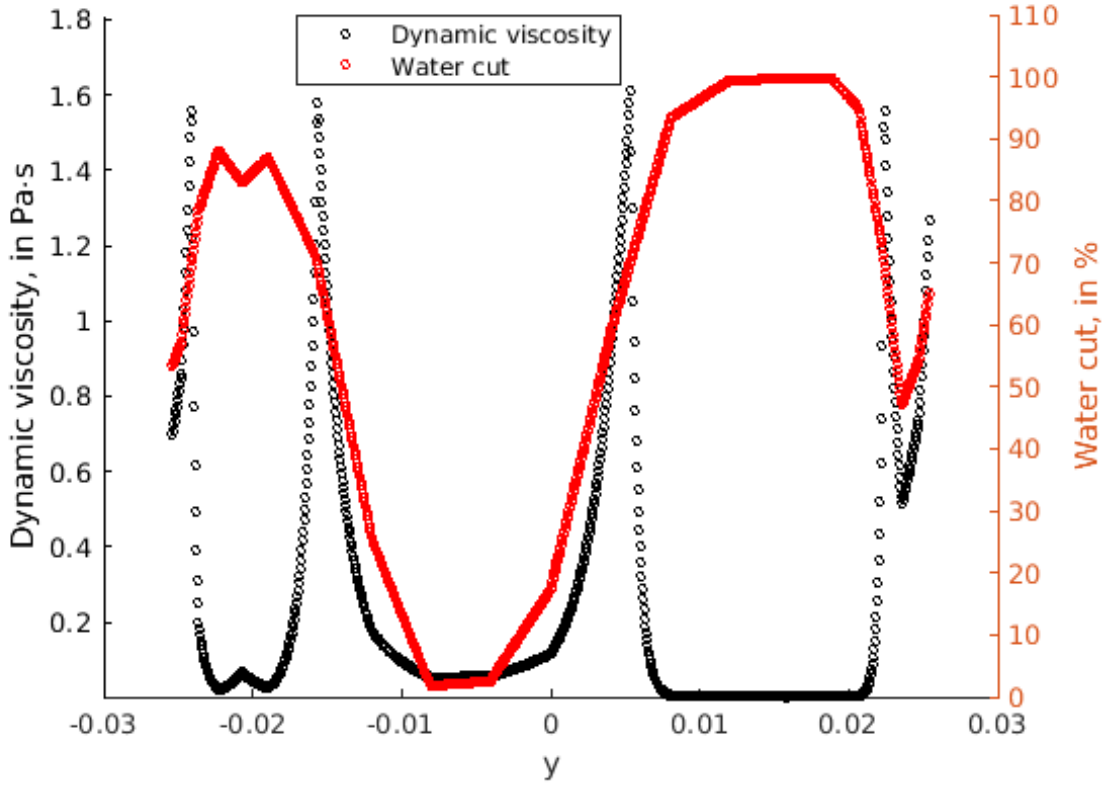


Figure 4.19: Dynamic viscosity along the monitoring lines obtained from the new *multiphaseEmulsionFoam* case.

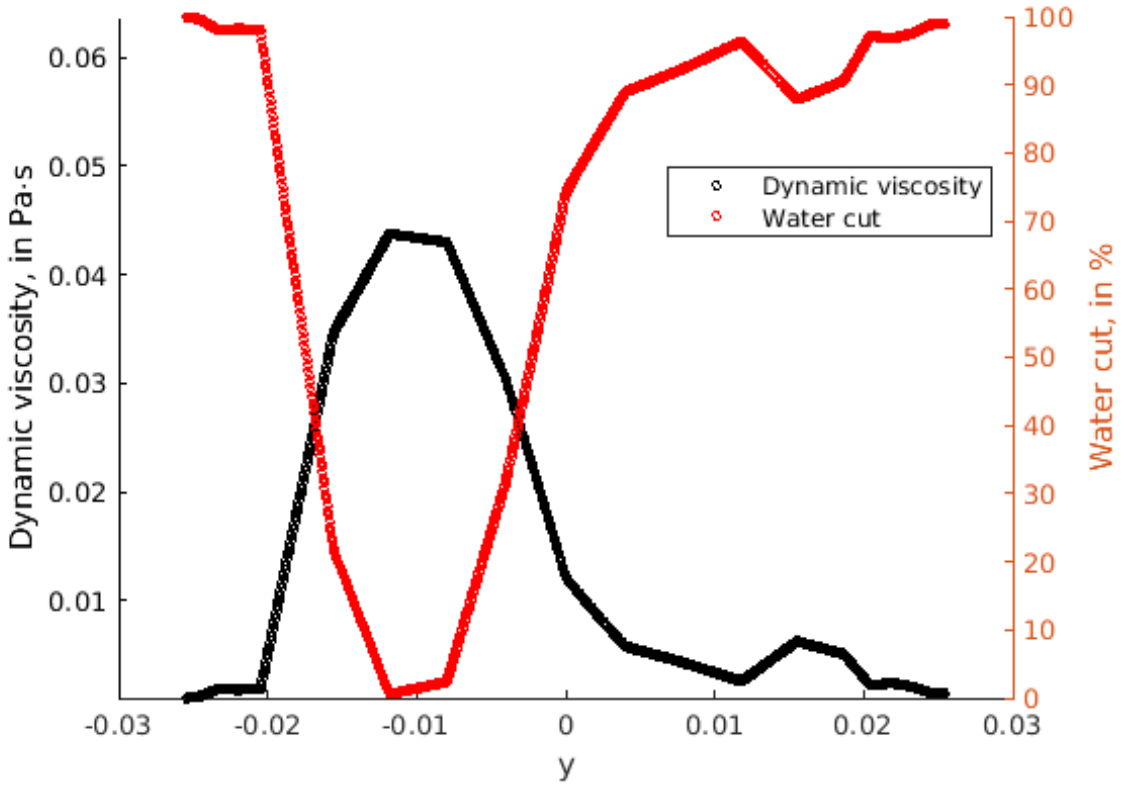


Figure 4.20: Dynamic viscosity along the monitoring lines obtained from the original *multiphaseInterFoam* case.

The figures above show the relationship between the viscosity and water cut in table 4.10. When the water cut is below 70%, the emulsion viscosity increases rapidly when the water cut is increased. When the water cut is around 70%, there would be a large viscosity gradient. Also, when the water cut is above 70%, there would be a sharp decrease in the viscosity. These phenomena correspond to the expected viscosity change. The maximum viscosity in figure 4.19 and 4.20 are 1.61 and 0.044 Pa·s, respectively. This shows the maximum viscosity obtained in the *multiphaseEmulsionFoam* case could be approximately 36.6 times higher than the *multiphaseInterFoam* case. Also, these figures reveal that the maximum viscosity in the *multiphaseEmulsionFoam* case is located near the interface. In contrast, the maximum viscosity in the *multiphaseInterFoam* case is located in the region where there is pure oil.

Table 4.10: Relationship of the viscosity change with respect to water cut for *multiphaseEmulsionFoam* and the *multiphaseInterFoam* case.

Solvers	**Water cut**	**Dynamic viscosity**
multiphaseEmulsionFoam	Increase (<= 70%)	Increase
	Increase (> 70%)	Decrease
multiphaseInterFoam	Increase	Decrease

The pressure drop of the entire domain was calculated, which is listed in table 4.11. When the new viscosity model was used, the pressure drop increased by approximately 13.2%, satisfying the expectations that an emulsion's presence can lead to a significant pressure drop.

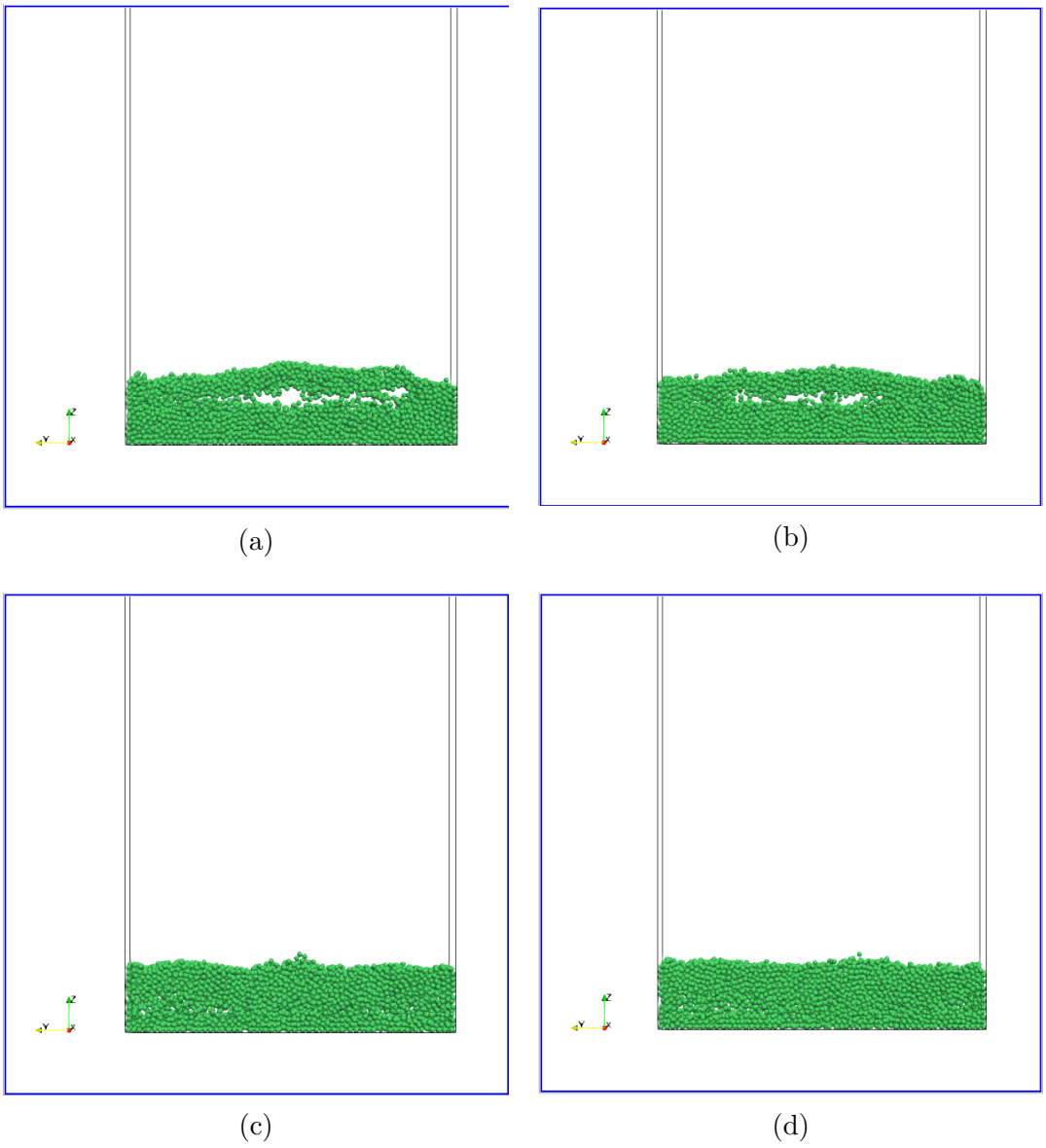
Table 4.11: Maximum and minimum pressure obtained from the *multiphaseEmulsionFoam* and the *multiphaseInterFoam* case.

Solvers	**Pressure (Pa)**	**Pressure drop (Pa)**
multiphaseEmulsionFoam	Max: $6.9 \cdot 10^3$	$1.89 \cdot 10^4$
	Min: $-1.2 \cdot 10^3$	
multiphaseInterFoam	Max: $5.7 \cdot 10^3$	$1.67 \cdot 10^4$
	Min: $-1.1 \cdot 10^4$	

Furthermore, the efficiency of the two solvers was compared on the coarse grids. There were no other influential factors when running the coarse grid cases, and they were running individually. Under the same conditions, the time for *multiphaseEmulsionFoam* and *multiphaseInterFoam* to complete the 40 s simulations was 36,521 s and 30,452 s, respectively. This indicates that the *multiphaseEmulsionFoam* requires more computational power becuase of the implemented new viscosity function.

4.5 Case Four: Fluidized Bed Simulation

The particle movements obtained from *DPConFoam* and *denseParticleFoam* are shown in figure 4.21 and 4.22. It can be observed that both cases obtained similar particle patterns, and air bubbles are found during the simulation.



(a) (b)

(c) (d)

Figure 4.21: Particle movements obtained from *DPConFoam* (a and c) and *denseParticleFoam* (b and d) at 0.5 and 1 s.

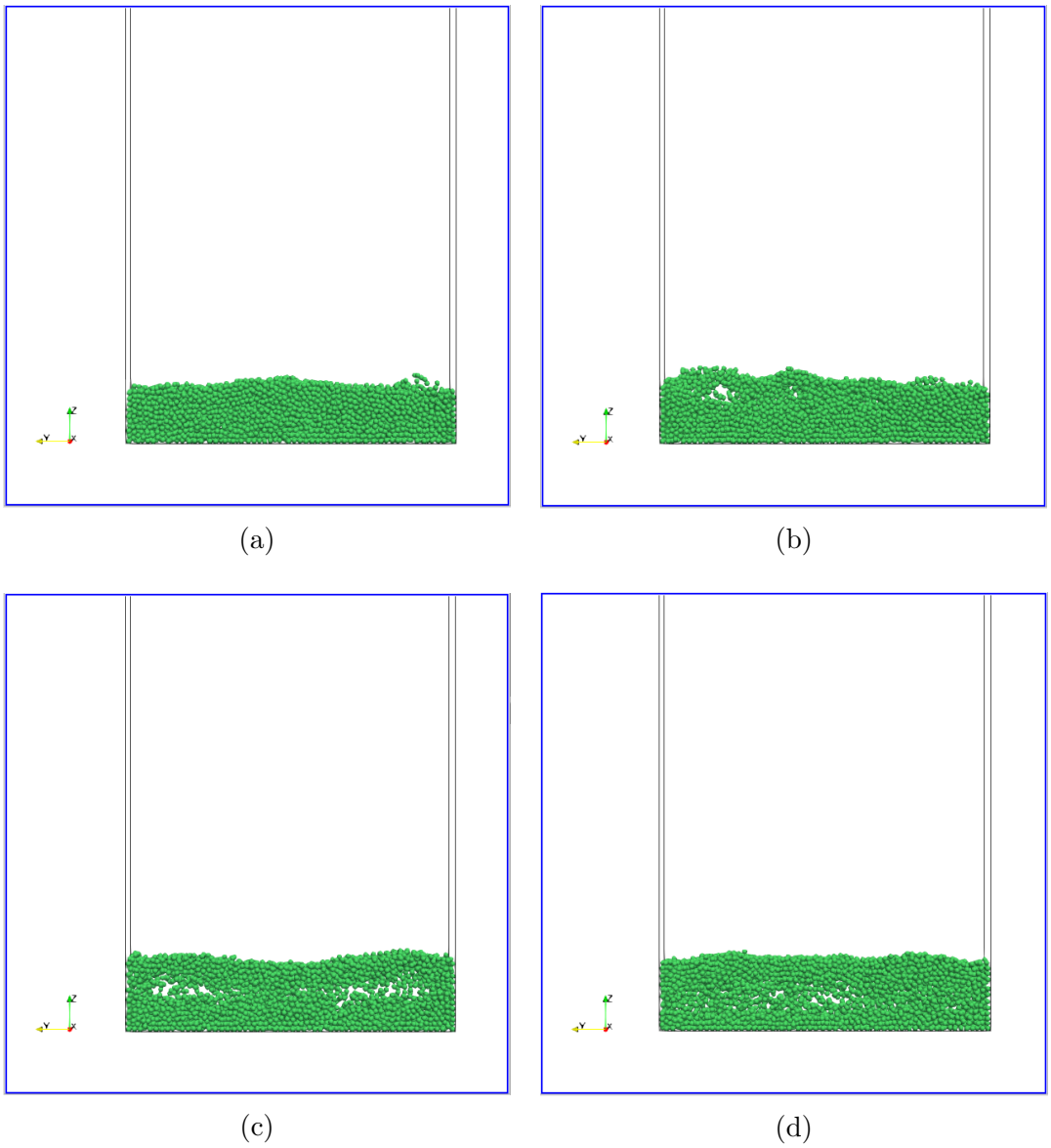


Figure 4.22: Particle movements obtained from *DPConFoam* (a and c) and *denseParticleFoam* (b and d) at 1.5 and 2 s.

In figure 4.23, the oscillations are the maximum pressure drop at each time step (VPD), and the straight lines are the average value of the oscillations (MPD). Also, the average values are recorded in table 3. The mean pressure drop obtained from *DPConFoam* and *denseParticleFoam*[1] are 575.13 Pa and 565.25 Pa, respectively. This indicates that the average pressure obtained from *DPConFoam* is around 1.70% higher

[1]The pressure drop read from denseParticleFoam is kinematic pressure, which means it has to multiply by the density of the fluid to become static pressure.

than the one from *denseParticleFoam*. This small difference could be mainly due to the transformation of governing equations and the implementation of hydrostatic pressure. These changes make the convergence steps of two solvers become different and eventually lead to this disparity.

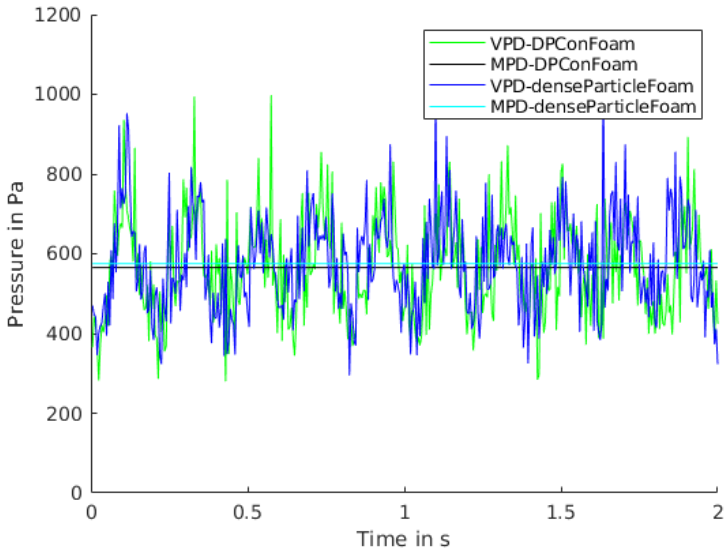


Figure 4.23: Maximum pressure drop at every time step.

Figure 4.24 shows the averaged particle height with respect to time, where the oscillations are the averaged height at each time step (AH), and the straight lines are the mean value of AH (MAH). By reading the plot, the MAH of *DPConFoam* and *denseParticleFoam* are $5.21{\cdot}10^{-3}$ m and $5.20{\cdot}10^{-3}$ m, respectively, and the relative error between two solvers is 0.19%. Also, the bed expansion dynamics trend obtained from the two solvers is almost the same. This shows that the *DPConFoam* has reached an excellent agreement with *denseParticleFoam* on bed expansion dynamics.

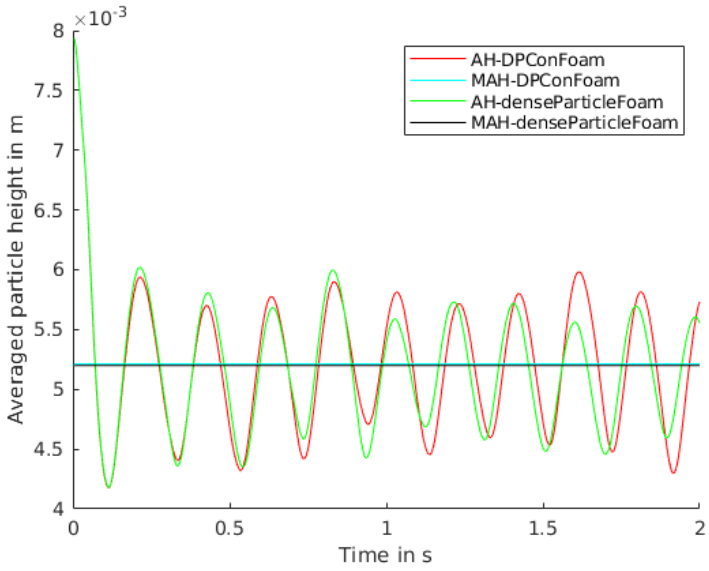


Figure 4.24: Average height of all particles with respect to time.

Variables	**Solver**	**Result**
Pressure drop (Pa)	*DPConFoam*	575.13
	denseParticleFoam	565.25
Average height (m)	*DPConFoam*	$5.21 \cdot 10^{-3}$
	denseParticleFoam	$5.20 \cdot 10^{-3}$

Chapter 5

Conclusions, Recommendations, & Future Work

5.1 Conclusions

This chapter summarizes the work that has been accomplished during this study. Also, some suggestions that might be helpful for future works are proposed at the end of this chapter. This project has the overall goal to develop a two-way coupled multi-fluid CFD-DEM solver, which will be used to simulate sand transport in W-O emulsions, and it was divided into three phases.

In the first phase, a modified solver, *DPConFoam*, was created based on *denseParticleFoam*. It has the same form of continuity equations and variables as *multiphaseInterFoam*. This work demonstrates the numerical validation of *DPConFoam*, which is a modified version of *denseParticleFoam* that uses the strong conservation form of governing equations and the modified pressure term p_{rgh}. First, the particle patterns in the domain are compared. Second, two time-averaged parameters were numerically compared, the mean value of the maximum pressure drop and the mean value of all particles' height. As a result, the relative differences between these two criteria are 1.70% and 0.192%, respectively. This proves that *DPConFoam* can predict the bed expansion dynamics and the rheological behaviour of the flow correctly. Moreover, it can be concluded that *DPConFoam* and *denseParticleFoam* are two numerically equivalent solvers based on the comparisons. In this way, *DPConFoam* can

be used as a base solver to couple with a multi-fluid model.

The second phase focused on developing and implementing a new W-O emulsion viscosity model. In this study, the measured viscosity data were analyzed. It was found that the water cut functions listed in table 4.3 have similar coefficients. Therefore, three representational viscosity functions were created to cover the entire shear rate and water cut range. This decision was made to optimize the efficiency and accuracy of the solver. This set of functions can be found in table 4.4, and they achieved high correlation coefficients to the measured data, which are all above 0.95. Thus, these equations were implemented in OpenFOAM in the form of Heaviside step functions and tested in a horizontal pipe and a simplified flow loop. The simulations were performed by *multiphaseInterFoam* and *multiphaseEmulsionFoam*, so that the influence of the new viscosity model could be compared with the original one.

In the horizontal pipe case, the *multiphaseEmulsionFoam* shows better predictions of the dynamics of the interface and flow. First, small waves were observed in the *multiphaseEmulsionFoam* case but not in the *multiphaseInterFoam* case. As mentioned by Edomwony et al. [62], there should always be waves on the interface, and the size of the wave mainly depends on the velocity of the flow. Second, the velocity gradient shows a clear change at the interface in the *multiphaseEmulsionFoam* case, which matched the velocity profile shown in [61]. In contrast, the *multiphaseInterFoam* case incorrectly shows a smooth transition.

In the simplified flow loop case, the simulation results obtained from both *multiphaseInterFoam* and *multiphaseEmulsionFoam* cases show that the W-O mixture tended to flow at the center of the pipe, which is due to the drag reduction film effect. However, the flow regime obtained from the new *multiphaseEmulsionFoam* is similar to churn flow at the short vertical pipe right after the inlet. Also, the pressure drop in the *multiphaseEmulsionFoam* case is approximately 13.2% larger than the *multiphaseInterFoam* case, as expected. In addition, the higher viscosity values are usually obtained at the W-O interface for the *multiphaseEmulsionFoam* case. In contrast, the

maximum values in the *multiphaseInterFoam* case are found in the pure oil regions. The *multiphaseEmulsionFoam* results correspond to the expected trend of viscosity change, which means the new viscosity function is properly working. Furthermore, *multiphaseEmulsionFoam* requires more computational power, mainly due to the new viscosity model. It takes approximately 20% longer to finish the simulations on the coarse grid.

In the third phase, the two-way coupled multi-fluid CFD-DEM solver theory was developed. The challenge of this is to couple the Eulerian model (VOF method from *multiphaseInterFoam*) with the Lagrangian model (DEM model from *DPConFoam*). To deal with this problem, the continuous phase volume fraction was introduced to the VOF method, which required a transformation of the governing equation, as shown in section 3.4. Also, the surface tension force was considered in the momentum equation of *DPConFoam*. With these changes, the VOF method and *DPConFoam* are able to be coupled and implemented in future works.

5.2 Future Work

To further investigate in this area, the following proposed recommendations can be considered:

- **Two-way coupled multiphase, incompressible CFD-DEM solver**
  Combine the two-way coupled single fluid phase CFD-DEM solver with the VOF method based on the derived equation. With a one way coupled solver, the particles do not affect the flow, whereas, the two-way coupled solver takes the particle forces and volume fraction into account in all governing equation. In this way, the changes in the flow regime caused by accumulation of particles can be observed.

- **Run the flow loop case on a finer mesh to gain mesh independence**
  The current meshes are not able to achieve mesh Independence due to limited

resources. Therefore, it is necessary to refine the fine and perform further mesh independence check.

- **Improve the pressure drop and flow rate measurement**

  One of the experimental plans of this study was to measure the pressure drop before and after the separator. This was cancelled due to damage of the equipment and lack of replacements. In the future, the pressure should be measured for validation purpose. Also, preparing more tubes and learning how to connect the tubes can prevent this happening again. In terms of the flow rate measurement, measuring the flow rate at a certain point before the inlet. In this way, a more accurate inlet boundary condition would be obtained. Also, measuring the flow rate at the outlet can be used to validate the phase 3 solver.

- **Further investigation of the emulsion viscosity**

  Measuring the emulsion viscosity at more water cut points should be performed to get a more accurate function. Also, the emulsion viscosity after the inversion point should be studied. Moreover, learning how to use rheometer can be useful for further investigation. Furthermore, different oils and emulsions would have different properties, especially their viscosity. This implies that new viscosity functions need to be used if the oils are changed.

Bibliography

[1] C Fernandes, D Semyonov, L. Ferrás, and J. M. Nóbrega, "Validation of the cfd-dpm solver dpmfoam in openfoam through analytical, numerical and experimental comparisons," *Granular Matter*, vol. 20, no. 4, pp. 1–18, 2018.

[2] "Energy annual report," *2019-2020 - Open Government*, 2020.

[3] P. K. Kilpatrick, "Water-in-crude oil emulsion stabilization: Review and unanswered questions," *Energy & Fuels*, vol. 26, no. 7, pp. 4017–4026, 2012.

[4] M. A. Farah, R. C. Oliveira, J. N. Caldas, and K. Rajagopal, "Viscosity of water-in-oil emulsions: Variation with temperature and water volume fraction," *Journal of Petroleum Science and Engineering*, vol. 48, no. 3-4, pp. 169–184, 2005.

[5] S. L. Kokal *et al.*, "Crude oil emulsions: A state-of-the-art review," *SPE Production & facilities*, vol. 20, no. 01, pp. 5–13, 2005.

[6] H. P. Ronningsen *et al.*, "Correlations for predicting viscosity of w/o-emulsions based on north sea crude oils," 1995.

[7] J. Zhang, H. Yuan, J. Zhao, and N. Mei, "Viscosity estimation and component identification for an oil-water emulsion with the inversion method," *Applied Thermal Engineering*, vol. 111, pp. 759–767, 2017.

[8] L. Li and B. Li, "Implementation and validation of a volume-of-fluid and discrete-element-method combined solver in openfoam," *Particuology*, vol. 39, pp. 109–115, 2018.

[9] L Jing, C. Kwok, Y. F. Leung, and Y. Sobral, "Extended cfd–dem for free-surface flow with multi-size granules," *International Journal for Numerical and Analytical Methods in Geomechanics*, vol. 40, no. 1, pp. 62–79, 2016.

[10] X. Sun and M. Sakai, "Three-dimensional simulation of gas–solid–liquid flows using the dem–vof method," *Chemical Engineering Science*, vol. 134, pp. 531–548, 2015.

[11] C. Li, Q. Huang, S. Ma, and C. Ji, "An experimental study on the viscosity of water-in-oil emulsions," *Journal of Dispersion Science and Technology*, vol. 37, no. 3, pp. 305–316, 2016.

[12] G. I. Taylor, "The viscosity of a fluid containing small drops of another fluid," *Proceedings of the Royal Society of London. Series A, Containing Papers of a Mathematical and Physical Character*, vol. 138, no. 834, pp. 41–48, 1932.

[13] I. M. Krieger and T. J. Dougherty, "A mechanism for non-newtonian flow in suspensions of rigid spheres," *Transactions of the Society of Rheology*, vol. 3, no. 1, pp. 137–152, 1959.

[14] R. Pal, "A novel method to correlate emulsion viscosity data," *Colloids and Surfaces A: Physicochemical and Engineering Aspects*, vol. 137, no. 1-3, pp. 275–286, 1998.

[15] E. Richardson, "The formation and flow of emulsion," *Journal of colloid science*, vol. 5, no. 4, pp. 404–413, 1950.

[16] S. W. Hasan, M. T. Ghannam, and N. Esmail, "Heavy crude oil viscosity reduction and rheology for pipeline transportation," *Fuel*, vol. 89, no. 5, pp. 1095–1100, 2010.

[17] M. Meriem-Benziane, S. A. Abdul-Wahab, M. Benaicha, and M. Belhadri, "Investigating the rheological properties of light crude oil and the characteristics of its emulsions in order to improve pipeline flow," *Fuel*, vol. 95, pp. 97–107, 2012.

[18] T. S. T. Ariffin, E. Yahya, and H. Husin, "The rheology of light crude oil and water-in-oil-emulsion," *Procedia engineering*, vol. 148, pp. 1149–1155, 2016.

[19] I Yaron and B Gal-Or, "On viscous flow and effective viscosity of concentrated suspensions and emulsions," *Rheologica Acta*, vol. 11, no. 3-4, pp. 241–252, 1972.

[20] S. J. Choi and W. Schowalter, "Rheological properties of nondilute suspensions of deformable particles," *The Physics of Fluids*, vol. 18, no. 4, pp. 420–427, 1975.

[21] L. L. Schramm, "Petroleum emulsions: Basic principles," in ACS Publications, 1992.

[22] N Phan-Thien and D. Pham, "Differential multiphase models for polydispersed suspensions and particulate solids," *Journal of Non-Newtonian Fluid Mechanics*, vol. 72, no. 2-3, pp. 305–318, 1997.

[23] W. J. Rider and D. B. Kothe, "Reconstructing volume tracking," *Journal of computational physics*, vol. 141, no. 2, pp. 112–152, 1998.

[24] C. W. Hirt and B. D. Nichols, "Volume of fluid (vof) method for the dynamics of free boundaries," *Journal of computational physics*, vol. 39, no. 1, pp. 201–225, 1981.

[25] M. Sussman and E. G. Puckett, "A coupled level set and volume-of-fluid method for computing 3d and axisymmetric incompressible two-phase flows," *Journal of computational physics*, vol. 162, no. 2, pp. 301–337, 2000.

[26] M. Sussman, P. Smereka, and S. Osher, "A level set approach for computing solutions to incompressible two-phase flow," *Journal of Computational physics*, vol. 114, no. 1, pp. 146–159, 1994.

[27] V. Boniou, T. Schmitt, and A. Vié, "Comparison of interface capturing methods for the simulation of two-phase flow in a unified low-mach framework," *arXiv preprint arXiv:2106.01864*, 2021.

[28] F. Losasso, R. Fedkiw, and S. Osher, "Spatially adaptive techniques for level set methods and incompressible flow," *Computers & Fluids*, vol. 35, no. 10, pp. 995–1010, 2006.

[29] E. Stenmark, "On multiphase flow models in ansys cfd software," M.S. thesis, 2013.

[30] E. Berberovic, "Investigation of free-surface flow associated with drop impact: Numerical simulations and theoretical modeling," Ph.D. dissertation, Technische Universität, 2010.

[31] T. Holzmann, "Mathematics, numerics, derivations and openfoam®," *Loeben, Germany: Holzmann CFD*, 2016.

[32] S. Márquez Damián, "An extended mixture model for the simultaneous treatment of short and long scale interfaces," 2013.

[33] P.-G. De Gennes, F. Brochard-Wyart, D. Quéré, *et al.*, *Capillarity and wetting phenomena: drops, bubbles, pearls, waves*. Springer, 2004, vol. 315.

[34] J. U. Brackbill, D. B. Kothe, and C. Zemach, "A continuum method for modeling surface tension," *Journal of computational physics*, vol. 100, no. 2, pp. 335–354, 1992.

[35] B. Lafaurie, C. Nardone, R. Scardovelli, S. Zaleski, and G. Zanetti, "Modelling merging and fragmentation in multiphase flows with surfer," *Journal of computational physics*, vol. 113, no. 1, pp. 134–147, 1994.

[36] M. Rudman, "Volume-tracking methods for interfacial flow calculations," *International journal for numerical methods in fluids*, vol. 24, no. 7, pp. 671–691, 1997.

[37] J. Zhao and T. Shan, "Coupled cfd–dem simulation of fluid–particle interaction in geomechanics," *Powder technology*, vol. 239, pp. 248–258, 2013.

[38] M. Chiesa, V. Mathiesen, J. A. Melheim, and B. Halvorsen, "Numerical simulation of particulate flow by the eulerian–lagrangian and the eulerian–eulerian approach with application to a fluidized bed," *Computers & chemical engineering*, vol. 29, no. 2, pp. 291–304, 2005.

[39] P. A. Cundall and O. D. Strack, "A discrete numerical model for granular assemblies," *geotechnique*, vol. 29, no. 1, pp. 47–65, 1979.

[40] Y. Tsuji, T. Kawaguchi, and T. Tanaka, "Discrete particle simulation of two-dimensional fluidized bed," *Powder technology*, vol. 77, no. 1, pp. 79–87, 1993.

[41] H. Zhu, Z. Zhou, R. Yang, and A. Yu, "Discrete particle simulation of particulate systems: Theoretical developments," *Chemical Engineering Science*, vol. 62, no. 13, pp. 3378–3396, 2007.

[42] A. Bérard, G. S. Patience, and B. Blais, "Experimental methods in chemical engineering: Unresolved cfd-dem," *The Canadian Journal of Chemical Engineering*, vol. 98, no. 2, pp. 424–440, 2020.

[43] S. Razavi, "Numerical simulation of sand retention mechanisms," 2021.

[44] B. Blais, D. Vidal, F. Bertrand, G. S. Patience, and J. Chaouki, "Experimental methods in chemical engineering: Discrete element method—dem," *The Canadian Journal of Chemical Engineering*, vol. 97, no. 7, pp. 1964–1973, 2019.

[45] J. Marshall and K Sala, "Comparison of methods for computing the concentration field of a particulate flow," *International Journal of Multiphase Flow*, vol. 56, pp. 4–14, 2013.

[46] G. B. Macpherson, N. Nordin, and H. G. Weller, "Particle tracking in unstructured, arbitrary polyhedral meshes for use in cfd and molecular dynamics," *Communications in Numerical Methods in Engineering*, vol. 25, no. 3, pp. 263–273, 2009.

[47] Y. Tsuji, T. Tanaka, and T Ishida, "Lagrangian numerical simulation of plug flow of cohesionless particles in a horizontal pipe," *Powder technology*, vol. 71, no. 3, pp. 239–250, 1992.

[48] H. C. Hamaker, "The london—van der waals attraction between spherical particles," *physica*, vol. 4, no. 10, pp. 1058–1072, 1937.

[49] H Krupp and G Sperling, "Theory of adhesion of small particles," *Journal of Applied Physics*, vol. 37, no. 11, pp. 4176–4180, 1966.

[50] R. Fisher, "On the capillary forces in an ideal soil; correction of formulae given by wb haines," *The Journal of Agricultural Science*, vol. 16, no. 3, pp. 492–505, 1926.

[51] T Gillespie and W. Settineri, "The effect of capillary liquid on the force of adhesion between spherical solid particles," *Journal of Colloid and Interface Science*, vol. 24, no. 2, pp. 199–202, 1967.

[52] G. Lian, C. Thornton, and M. J. Adams, "A theoretical study of the liquid bridge forces between two rigid spherical bodies," *Journal of colloid and interface science*, vol. 161, no. 1, pp. 138–147, 1993.

[53] Y. Xiong, M. Zhang, and Z. Yuan, "Three-dimensional numerical simulation method for gas–solid injector," *Powder technology*, vol. 160, no. 3, pp. 180–189, 2005.

[54] Y. A. Çengel and J. M. Cimbala, *Fluid Mechanics: Fundamentals and Applications*. McGraw-Hill Higher Education, 2010.

[55] T. B. Anderson and R. Jackson, "Fluid mechanical description of fluidized beds. equations of motion," *Industrial & Engineering Chemistry Fundamentals*, vol. 6, no. 4, pp. 527–539, 1967.

[56] D. Gidaspow, *Multiphase flow and fluidization: continuum and kinetic theory descriptions*. Academic press, 1994.

[57] Y. Feng and A. Yu, "Assessment of model formulations in the discrete particle simulation of gas- solid flow," *Industrial & engineering chemistry research*, vol. 43, no. 26, pp. 8378–8390, 2004.

[58] B. Xu and A. Yu, "Numerical simulation of the gas-solid flow in a fluidized bed by combining discrete particle method with computational fluid dynamics," *Chemical Engineering Science*, vol. 52, no. 16, pp. 2785–2809, 1997.

[59] M. Vångö, S. Pirker, and T. Lichtenegger, "Unresolved cfd–dem modeling of multiphase flow in densely packed particle beds," *Applied Mathematical Modelling*, vol. 56, pp. 501–516, 2018.

[60] K. Chu, B. Wang, A. Yu, and A. Vince, "Cfd-dem modelling of multiphase flow in dense medium cyclones," *Powder Technology*, vol. 193, no. 3, pp. 235–247, 2009.

[61] Z. Huang, M. Senra, R. Kapoor, and H. S. Fogler, "Wax deposition modeling of oil/water stratified channel flow," *AIChE journal*, vol. 57, no. 4, pp. 841–851, 2011.

[62] L. C. Edomwonyi-Otu and P. Angeli, "Pressure drop and holdup predictions in horizontal oil–water flows for curved and wavy interfaces," *Chemical Engineering Research and Design*, vol. 93, pp. 55–65, 2015.

[63] *User guide: Inlet outlet.*

[64] J. H. Ferziger, M. Perić, and R. L. Street, *Computational methods for fluid dynamics*. Springer, 2002, vol. 3.

[65] *User guide: Hydrostatic pressure effects.*

[66] L. C. Edomwonyi-Otu, "Drag reduction in oil-water flows," Ph.D. dissertation, UCL (University College London), 2015.

[67] J. G. Flores, X. T. Chen, C. Sarica, and J. P. Brill, "Characterization of oil-water flow patterns in vertical and deviated wells," *SPE Production & Facilities*, vol. 14, no. 02, pp. 102–109, 1999.

Appendix A: Oil Information for chapter 3

before discharge to surface water.
Local guidelines on emission limits for volatile substances must be observed for the discharge of exhaust air containing vapour.

9. PHYSICAL AND CHEMICAL PROPERTIES

Appearance	:	liquid
Colour	:	clear
Odour	:	Slight hydrocarbon
Odour Threshold	:	Data not available
pH	:	Not applicable
pour point	:	-42 °C / -44 °FMethod: ISO 3016
Initial boiling point and boiling range	:	> 280 °C / 536 °Festimated value(s)
Flash point	:	210 °C / 410 °F Method: ISO 2592
Evaporation rate	:	Data not available
Flammability (solid, gas)	:	Data not available
Upper explosion limit	:	Typical 10 %(V)
Lower explosion limit	:	Typical 1 %(V)
Vapour pressure	:	< 0.5 Pa (20 °C / 68 °F) estimated value(s)
Relative vapour density	:	> 1estimated value(s)
Relative density	:	0.835 (15 °C / 59 °F)
Density	:	835 kg/m3 (15.0 °C / 59.0 °F) Method: ISO 12185
Solubility(ies)		
Water solubility	:	negligible
Solubility in other solvents	:	Data not available
Partition coefficient: n-octanol/water	:	Pow: > 6(based on information on similar products)
Auto-ignition temperature	:	> 320 °C / 608 °F
Viscosity		

Viscosity, dynamic	:	Data not available
Viscosity, kinematic	:	22 mm2/s (40.0 °C / 104.0 °F) Method: ASTM D445 4.8 mm2/s (100 °C / 212 °F) Method: ASTM D445 700 mm2/s (-20 °C / -4 °F) Method: ASTM D445
Explosive properties	:	Not classified
Oxidizing properties	:	Data not available
Conductivity	:	This material is not expected to be a static accumulator.
Decomposition temperature	:	Data not available

10. STABILITY AND REACTIVITY

Reactivity	:	The product does not pose any further reactivity hazards in addition to those listed in the following sub-paragraph.
Chemical stability	:	Stable.
Possibility of hazardous reactions	:	Reacts with strong oxidising agents.
Conditions to avoid	:	Extremes of temperature and direct sunlight.
Incompatible materials	:	Strong oxidising agents.
Hazardous decomposition products	:	Hazardous decomposition products are not expected to form during normal storage.

11. TOXICOLOGICAL INFORMATION

Basis for assessment	:	Information given is based on data on the components and the toxicology of similar products.Unless indicated otherwise, the data presented is representative of the product as a whole, rather than for individual component(s).
Symptoms of Overexposure	:	Oil acne/folliculitis signs and symptoms may include formation of black pustules and spots on the skin of exposed areas. Ingestion may result in nausea, vomiting and/or diarrhoea.Local necrosis is evidenced by delayed onset of pain and tissue damage a few hours following injection.
Information on likely routes of	:	Skin and eye contact are the primary routes of exposure

Appendix B: Code for the New Viscosity Function

```
Foam::tmp<Foam::volScalarField>
Foam::multiphaseEmulsionMix::mu() const
{
    PtrDictionary<phase>::const_iterator iter = phases_.begin();
    tmp<volScalarField> tmu = iter()*iter().rho()*iter().nu();
    volScalarField& mu = tmu.ref();

    const dictionary& alphaControls = mesh_.solverDict("alpha");
    bool emulsionMu(alphaControls.lookup<bool>("emulsion"));
    volScalarField nonemulsion = iter()*dimensionedScalar(dimless,0);

    PtrDictionary<phase>::const_iterator iter2 = iter;

    if(emulsionMu == false)
    {

      for (++iter; iter != phases_.end(); ++iter)
      {mu += iter()*iter().rho()*iter().nu();}
    }

    else if (iter().name() == "water" && emulsionMu)
    {

      //MG: Zero mu
      mu -=mu;
      for (++iter; iter != phases_.end(); ++iter)
      {
         if(iter().name() == "oil")
         {
            continue;
         }
         else
         {
           mu += iter()*iter().rho()*iter().nu();
           nonemulsion += iter();
         }
      }
     }

     else if(iter().name() == "oil" && emulsionMu)
     {
```

```
 mu -=mu;
 for (++iter; iter != phases_.end(); ++iter)
 {
    if (iter().name() == "water")
    {
      iter2 = iter;
    }
    else
    {
      mu += iter()*iter().rho()*iter().nu();
      nonemulsion += iter();
    }
  }
}

else
{
  nonemulsion = iter();
  for (++iter; iter != phases_.end(); ++iter)
  {
     if (iter().name() == "water")
     {
       iter2 = iter;
     }

     else if(iter().name() == "oil")
     {
       continue;
     }

     else
     {
       mu += iter()*iter().rho()*iter().nu();
       nonemulsion += iter();
     }
    }
}

if (emulsionMu)
{
  volScalarField alpha_w = iter2()/(1-nonemulsion+VSMALL);
  alpha_w = min(max(alpha_w, scalar(0)), scalar(1));

  mu += (1-nonemulsion)*
      (
neg(dimensionedScalar(dimTime, 1.0)*strainRate()-40-VSMALL)*
(
    neg(alpha_w - 0.7+VSMALL)*0.04919*exp(4.986*alpha_w)
   +pos(alpha_w-0.7+VSMALL)*49216934.2723*exp(-24.6195*alpha_w)
)
+pos(dimensionedScalar(dimTime, 1.0)*strainRate()-40-VSMALL)*
neg(dimensionedScalar(dimTime, 1.0)*strainRate()-100)*
(
    neg(alpha_w-0.7+VSMALL)*0.04237*exp(4.010*alpha_w)
   +pos(alpha_w-0.7+VSMALL)*3074612.3957*exp(-21.8464*alpha_w)
```

```
        )
        +pos(dimensionedScalar(dimTime, 1.0)*strainRate()-100)*
        (
            neg(alpha_w - 0.7+VSMALL)*0.04071*exp(3.666*alpha_w)
           +pos(alpha_w - 0.7+VSMALL)*1204812.7792*exp(-20.9096*alpha_w)
        )
    )*dimensionedScalar(dimViscosity,1)*dimensionedScalar(dimDensity,1);
        }
      return tmu;
}
```

Appendix C: Code of the Momentum Equation in *DPConFoam*

```
alphaRhoc = rhoc*alphac;
alphaRhocf = fvc::interpolate(alphaRhoc);
alphaRhoPhic = alphaRhocf*phic;



fvVectorMatrix UcEqn
(

    fvm::ddt(alphaRhoc, Uc) + fvm::div(alphaRhoPhic, Uc)
  + turbulence->divDevTau(alphaRhoc,Uc)
  ==
    cloudSU
);

UcEqn.relax();

fvOptions.constrain(UcEqn);

volScalarField rAUc(1.0/UcEqn.A());
volScalarField rASpUc(1.0/(UcEqn.A() - cloudSUp));
surfaceScalarField rASpUcf("Dp", fvc::interpolate(rASpUc));

surfaceScalarField phicSUSu
(
    fvc::flux(rASpUc*cloudSUu)
);


surfaceScalarField phicSUSp
(
    fvc::interpolate(rASpUc*cloudSUp)
);


if (pimple.momentumPredictor())
{
    solve
    (
```

```
        UcEqn
     ==
        fvc::reconstruct
        (
            (phicSUSu + phicSUSp*phic)/rASpUcf
          - (
          ghf*fvc::snGrad(alphaRhoc)
          + fvc::snGrad(p_rgh)
          )*mesh.magSf()
        )
      + (fvm::Sp(cloudSUp, Uc) - cloudSUp*Uc)
    );

    fvOptions.correct(Uc);

}
```

Appendix D: Improved the accuracy of the sand separation efficiency measurement

During the internship in Sand Separation Technology Inc., the sand separation efficiency of G4 model was measured with an improved and more accurate method. Compared with the original method, sand bags that are able to collect sand particles larger than 1.0 micrometer were used. Besides, in the previous experiments, the volume of sand was used to determine the separation efficiency. The accuracy of the separation was improved be checking the mass of the carryovers and recovered sand instead of the volume. After collecting all the sand from the flow loop, they were placed on a tray for one day to become dry. The measurements were repeated three times to ensure the results are precise. With these improvements, the sand separation efficiency was measured with 80% water cut emulsion and with pure water at two flow rates. The averaged results and their standard deviations are listed in table D.1.

Table D.1 reveals that the sand separation efficiencies obtained from the improved method have acceptable standard deviation. The maximum is 0.0124 and the minimum is 0.0004, which happened in the case with 35 gpm of emulsion and pure water, respectively. On the one hand, it can be concluded that the presence of emulsion will significantly lower the efficiency of the separator, especially at lower flow rate. In terms of the 25 gpm cases, the separation efficiency decreased 21.3%, whereas the 35 gpm cases only decreased 10.65%.

On the other hand, at different flow rate, the separation efficiency of the emulsion

Table D.1: Viscosity equation test cases

Case	Flow rate (gpm)	Fluid	Recovered (g)
1	25	Water	1610.7
2	25	Emulsion	1250.7
3	35	Water	1633.7
4	35	Emulsion	1422.0
Case	**Carryover (g)**	**Separation efficiency (%)± Standard deviation**	
1	84.3	95.09±0.10	
2	444.3	73.79±0.61	
3	118.7	93.14±0.04	
4	301.7	82.49±1.24	

cases show a difference of 8.7%. In contrast, the water cases only have a slight difference of 1.95%,. This may be caused by the shear rate. As mentioned before, the emulsion viscosity tends to increase dramatically at low shear rate. A slower inlet flow rate may lead to a lower flow velocity inside of the separator, so that the local shear rate is also decreased. Therefore, the emulsion will carry more sand to the outlet, which results in a large decrease in separation efficiency.

Appendix E: Problems with the Current Version of *MIDPFoam*

After the *DPConFoam* was developed, a two-way coupled multi-fluid CFD-DEM solver was tried to be implemented into OpenFOAM, and it is named as *MIDPFoam*. This solver is aimed to combine the modified VOF method with the *DPConFoam*. However, two problems were found from the simulations, which are as follows:

1. **Mass conservation problem**

   The current version of *MIDPFoam* cannot guarantee mass conservation of each fluid phase. For example, when simulating the particle sedimentation problem (the first case in Sun et al.'s paper [10]), the volume fraction of the water increased and of the air decreased when the particles fell into the water from the air. In this case, the simulation did not diverge because the system would automatically correct the overall volume fraction to one. Besides, this problem could lead to divergence when solving a single fluid phase with particles since no other fluid phase makes up the overall volume fraction to one.

2. **Boundary condition problem**

   The second problem was observed when replicating the dam break case (the second case in Sun et al.'s paper [10]). The particles near the bottom walls tended to have a very small velocity, which means those particles did not have sufficient momentum to go forward. As a result, the phenomenons observed in the experiments and Sun's simulations cannot be correctly simulated by *MIDP-Foam*. If the boundary conditions of the bottom wall were changed to slip

walls, the particles show a similar pattern to the experimental results. However, this boundary condition would lead to a wrong flow regime. Therefore, ensuring the particles and fluids can have boundary conditions independently set in *MIDPFoam* could be a possible solution to correctly simulating flows with the transport of particles.