

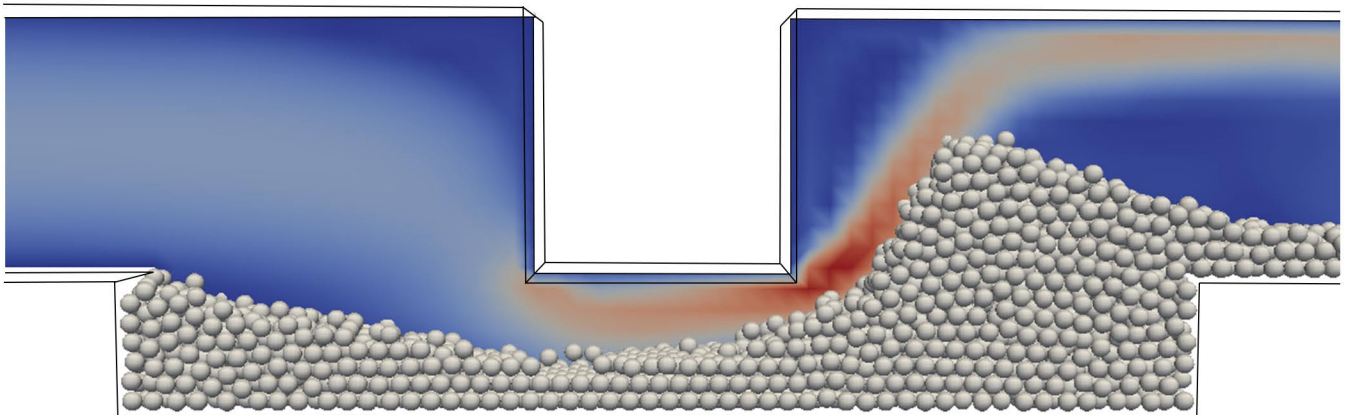


AALBORG UNIVERSITY  
STUDENT REPORT

# Particle accumulation in square duct cavity with laminar flow

A four-way coupled CFD-DEM investigation of effects of particle-to-fluid density ratio and fluid Reynolds number using model validated against experimental setup

Master's Thesis



---

A. O. Agesen, L. D. D. Blæsbjerg & A. R. Ehlers  
Thermal Energy and Process Engineering  
May 2019

*Things we wish our mothers had told us about CFD-DEM.*

Copyright © Aalborg University 2019

The content of this report is freely available, but publication (with reference) may only be pursued by agreement with the authors.



## AALBORG UNIVERSITY

### STUDENT REPORT

Department of Energy Technology  
Aalborg University  
[www.et.aau.dk](http://www.et.aau.dk)

**Title:**

Particle accumulation in square duct cavity with laminar flow

**Theme:**

Master's Thesis

**Project Period:**

4 February 2019 to 31 May 2019

**Project Group:**

TEPE4-1009

**Participant(s):**

Andreas Overgaard Agesen  
Line Dyhr Damsgaard Blæsbjerg  
Anders Riishede Ehlers

**Supervisor(s):**

Jakob Hærvig  
Chungen Yin

**Copies:** 6

**Total Page Numbers:** 176

Report Page Numbers: 144

Appendices: A-I

Supplements: 1-5

**Date of Completion:**

31 May 2019

**Abstract:**

In this project report a numerical model for the investigation of particle accumulation as subject to fluid flow in a simple geometry is developed. The geometry consists of a square duct, where a block is located over a cavity, which is initially filled with particles. The numerical model is developed using four-way coupled Computational Fluid Dynamics-Discrete Element Method (CFD-DEM) modelling framework. The CFD-DEM model is validated against experimental data using video capture and PIV is used for the determination of fluid flow rate.

A parametric study of particle-to-fluid density ratio and fluid Reynolds number indicates, that increasing fluid Reynolds number decreases the normalised friction factor, ratio of remaining particles in cavity and angle of incline, while increasing the density ratio increases all three.





# Executive Summary

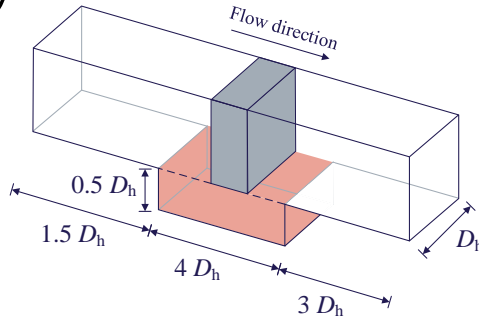
Particulate fluid flow has been subject to great scientific interest; research has been done for fluidised beds, sedimentation and accumulation of particles. Due to the availability of more powerful computers, a coupling between Computational Fluid Dynamics (CFD) and Discrete Element Modelling (DEM) has gained more interest. The coupled CFD-DEM method has however received less attention. This project aims to help fill this gap in literature by investigating how particles in a cavity accumulate, when exposed to a laminar fluid flow, using an experimentally validated four-way coupled CFD-DEM model.

The investigated geometry (Figure 1) consists of a square duct with a cavity in the bottom, which is initially filled with particles. Above the cavity is a block, forcing fluid to enter the cavity. Investigations are conducted with fluid Reynolds number ranging from 600 to 1000 with a particle-to-fluid density ratio ranging from 1.5 to 2.5.

The purpose of the project is to:

- Develop and validate a four-way coupled CFD-DEM modelling framework of particle deposition in a cavity
- Perform a parametric study to investigate the effect of particle-fluid density ratio and fluid Reynolds number

A numerical model is developed using CFDEM, which couples OpenFOAM (CFD) and LIGGGHTS (DEM). The model uses a coupling to transfer momentum between the particles and the particles and fluid. The included fluid-particle force models are determined based on a scaling analysis, which approximates the significance of relevant fluid-particle forces. Sensitivity

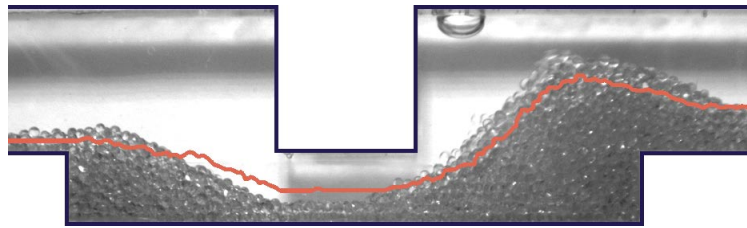


**Figure 1:** Conceptual illustration of geometry. Red area is initially filled with particles.

analyses are conducted for: geometry entrance length, grid independence, particle stiffness and time step.

### **Experimental validation of CFD-DEM model**

Three different experiments are conducted: The first is to validate the drag force sub-model implemented in the CFD-DEM, by experimentally setting a single glass particle in a column of water. The second is to validate the inlet velocity boundary condition implemented in the CFD-DEM, by measuring the velocity profile using PIV. The last is to experimentally investigate the transport and accumulation of the glass particles originally packed in a cavity until the steady-state of the particle positions is reached. The comparison of the drag on a single particle shows a deviation less than 4% in the fall time between the numerical model and the experiment. Figure 2 shows the comparison of the steady state particle positions of experiment and numerical model. The peak particle height is not equal but comparable; the difference is 19%, slope of the incline at the end of the cavity has approximately the same slope angle. As the general tendency is indeed similar the numerical model is considered valid for conducting further investigations.



**Figure 2:** Experimental steady state particle position plotted against the equivalent for the numerical model (illustrated by the red line).

### **Parametric study of particle accumulation**

A parametric study of the effects of changing the particle-to-fluid density ratio and fluid Reynolds number at the inlet is conducted. The parametric study concludes that the higher the Reynolds number and the lower the density ratio the further particles move towards the outlet of the geometry.

# Preface

This is a Master thesis project report written by three M.Sc. students from the *Thermal Energy and Process Engineering* education at Aalborg University. The project is written during spring of 2019 with hand-in date May 31<sup>st</sup>.

## Prerequisites

Knowledge of mathematics, general particle-particle and particle-fluid behaviour and understanding of the forces included in particle collision is needed.

## Reading Guide

The references used in this report is collected in a bibliography at the end of the report. When referring to a specific article/work with more than two authors, the citation is written as last name of the first author followed by "et al." (such as Yang et al. (2015)).

Equations, figures and tables in this report are listed by category, chapter and order of appearance of the given category, as for example Table 3.6, which refers to table number 6 in chapter 3.

After the table of content, the nomenclature is found. This lists abbreviations, dimensionless numbers, greek symbols, latin symbols, subscripts and superscripts used, as well as their respective SI-units.

Numerical values above  $10^4$  are written using comma-seperator for every 1000 (such as 10,000).

## **Applied Software**

The model in this project is developed using a coupled CFD-DEM model made in CFDEM software, which couples OpenFOAM and LIGGGHTS. The software is open source. MATLAB is used for general data processing, while Dantec Dynamics DynamicStudio is used for image processing of the experimental results.

## **Acknowledgements**

A special thanks to Asetek A/S, who have kindly provided a workstation for the calculations required in this project. Also a great thanks to Henrik Sørensen for helping with the PIV-measurements.

# Contents

<b>Executive Summary</b>	<b>v</b>
<b>Preface</b>	<b>vii</b>
<b>Nomenclature</b>	<b>xii</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Reasoning Behind Project . . . . .	1
1.2 State of the Art Review . . . . .	3
1.2.1 Numerical models . . . . .	3
1.2.2 Eulerian-Lagrangian . . . . .	5
1.2.3 Previous studies based on CFD and DEM . . . . .	5
1.3 Problem Statement . . . . .	9
1.3.1 Geometry . . . . .	9
1.3.2 Particles and fluid . . . . .	11
1.3.3 Selection of fluid particle forces . . . . .	11
1.3.4 General numerical considerations . . . . .	13
1.3.5 Mesh and particle sizes . . . . .	14
1.3.6 Assumptions and limitations . . . . .	15
1.4 Report Structure . . . . .	17
<b>2 Numerical Methodology</b>	<b>19</b>
2.1 Overview of the Modelling Methodology . . . . .	19
2.2 Computational Fluid Dynamics . . . . .	20
2.2.1 Governing equations for CFD . . . . .	21
2.2.2 CFD methodology . . . . .	22
2.2.3 Time step . . . . .	23

2.3	Discrete Element Method . . . . .	24
2.3.1	Governing equations for DEM . . . . .	25
2.3.2	DEM methodology . . . . .	26
2.3.3	Collision forces . . . . .	27
2.3.4	DEM time step . . . . .	31
2.4	CFD-DEM Coupling Methodology . . . . .	32
2.5	Governing Equations and Models for CFD-DEM Coupling . .	33
2.5.1	Fluid-particle forces . . . . .	34
2.5.2	Definition of void fraction . . . . .	40
2.5.3	Coupling time steps . . . . .	41
2.6	Chapter Summary . . . . .	41
<b>3</b>	<b>Numerical Model Setup</b>	<b>43</b>
3.1	Geometry and Meshing . . . . .	43
3.2	CFD Setup . . . . .	44
3.2.1	Schemes . . . . .	44
3.2.2	Boundary and initial conditions . . . . .	46
3.3	DEM Setup . . . . .	48
3.3.1	Particle and wall properties . . . . .	48
3.3.2	Initial and boundary conditions . . . . .	48
3.4	Coupling Models . . . . .	49
3.5	Model Sensitivity Study . . . . .	51
3.5.1	Entrance length . . . . .	52
3.5.2	Grid independence . . . . .	53
3.5.3	Time step and stiffness sensitivity . . . . .	57
3.6	Chapter Summary . . . . .	62
<b>4</b>	<b>Experimental Approach</b>	<b>63</b>
4.1	Drag Experiment . . . . .	63
4.2	Experimental Setup . . . . .	65
4.3	Validation of Flow Profile . . . . .	68
4.4	Experiment Procedure . . . . .	72
4.5	Sources of Error . . . . .	73
4.6	Chapter Summary . . . . .	74

<b>5</b>	<b>Sensitivity Study and Model Validation</b>	<b>75</b>
5.1	Initial Conditions . . . . .	76
5.1.1	Initial particle positions . . . . .	76
5.1.2	Simulation flow initialisation sensitivity . . . . .	77
5.2	Smoothing Sensitivity . . . . .	78
5.3	Particle-Particle Rolling Friction Sensitivity . . . . .	80
5.3.1	Rolling friction model . . . . .	80
5.3.2	Rolling friction coefficient . . . . .	82
5.4	Particle-Particle Friction Sensitivity . . . . .	83
5.5	Particle-Wall Friction Coefficient Sensitivity . . . . .	84
5.6	Coefficient of Restitution Sensitivity . . . . .	85
5.7	Final Validation of Numerical Framework . . . . .	86
5.8	Sources of Error . . . . .	88
5.9	Chapter Summary . . . . .	90
<b>6</b>	<b>Model-Based Parametric Study</b>	<b>91</b>
6.1	Data Treatment Methodology . . . . .	93
6.2	Parametric Study Results . . . . .	97
6.2.1	Top position line . . . . .	97
6.2.2	Number of particles . . . . .	99
6.2.3	Investigation of the flow field . . . . .	100
6.3	Discussion . . . . .	102
6.4	Chapter Summary . . . . .	104
<b>7</b>	<b>Conclusion</b>	<b>105</b>
<b>8</b>	<b>Future Work</b>	<b>107</b>
	<b>Bibliography</b>	<b>111</b>
	<b>Appendices</b>	
<b>A</b>	<b>Contact Model</b>	<b>117</b>
<b>B</b>	<b>Koch-Hill-Ladd Method</b>	<b>119</b>
<b>C</b>	<b>Mei Lift</b>	<b>121</b>

<b>D</b>	<b>Generation of the Geometry</b>	<b>123</b>
<b>E</b>	<b>Fully Developed Laminar Flow Profile</b>	<b>127</b>
<b>F</b>	<b>Insertion of Particles</b>	<b>131</b>
<b>G</b>	<b>Coupling Models</b>	<b>133</b>
<b>H</b>	<b>Grid Convergence Index</b>	<b>137</b>
<b>I</b>	<b>Experimental Work</b>	<b>141</b>
	<b>Supplements</b>	
<b>S1</b>	<b>Data Sheet of the Glass Particles</b>	<b>145</b>
<b>S2</b>	<b>Python Block Mesh Script</b>	<b>147</b>
<b>S3</b>	<b>Laminar Flow Profile Script</b>	<b>153</b>
<b>S4</b>	<b>Initiation Script</b>	<b>155</b>
<b>S5</b>	<b>Main CFD-DEM Script</b>	<b>157</b>



# Nomenclature

## Abbreviations

CDT	Constant Directional Torque
CFD	Computational Fluid Dynamics
CFD-DEM	Computational fluid dynamics - discrete element method
DEM	Discrete Element Method
DEV	Deviation
DNS	Direct Numerical Simulation
EPSP	Elastic-Plastic Spring-Dashpot
fps	Frames per second
GCI	Grid convergence index
H.O.T	Higher order terms
IA	Interrogation Area
LBM	Lattice-Boltzmann Method
LES	Large Eddy Simulation
PISO	Pressure Implicit with Splitting of Operators
PIV	Particle Image Velocimetry
PSP	Polyamid Seeding Particles
RAM	Random Access Memory
TFM	Two Fluid Model

## Dimensionless numbers

AR	Aspect ratio	-
----	--------------	---

Co	Courant number	-
Fr	Froude number	-
GCI	Grid convergence index	
Pe	Peclet number	-
Re	Reynolds number	-
St	Stokes number	-

### Greek Symbols

$\alpha$	Void fraction	-
$\chi$	Particle-to-fluid density ratio	-
$\Delta$	Difference	-
$\delta$	Displacement distance	m
$\epsilon$	Dimensionless particle diameter	-
$\eta$	Spacial coordinate	-
$\gamma$	Viscoelastic damping constant	-
$\mathcal{O}$	Order of magnitude	
$\mu$	Dynamic viscosity	kg m <sup>-1</sup> s <sup>-1</sup>
$\nabla$	Divergence	-
$\nu$	Kinematic viscosity	m <sup>2</sup> s <sup>-1</sup>
$\Omega$	Rotation rate	s <sup>-1</sup>
$\omega$	Angular velocity	s <sup>-1</sup>
$\psi$	Spacial coordinate	-
$\rho$	Density	kg m <sup>-3</sup>
$\tau$	Stress tensor	N m <sup>-2</sup>
$\nu$	Poisson's ratio	-
$\varepsilon$	Dimensionless calculation parameter	-
$\Xi$	Coupling interval	-
$\zeta$	Relative error	-
$\zeta$	Spacial coordinate	-

**Latin Symbols**

$A$	Area	$\text{m}^2$
$a$	Width	$\text{m}$
$b$	Height	$\text{m}$
$C$	Coefficient	-
$C$	Constant	-
$D$	Diameter	$\text{m}$
$E$	Discretisation error	-
$E$	Young's Modulus	$\text{Pa}$
$e$	Coefficient of Restitution	-
$F$	Factor	-
$F$	Force	$\text{N}$
$f$	Friction factor	-
$f$	Grid dependent parameter	-
$G$	Shear rate	$\text{s}^{-1}$
$g$	Gravitational acceleration	$\text{m s}^{-2}$
$g$	Parameter (independent of grid size)	-
$H$	Height	$\text{m}$
$h$	Step size	-
$I$	Moment of inertia	$\text{kg m}^2$
$J$	Extension of the Saffmann lift	-
$K$	Implicit momentum exchange	$\text{kg m s}^{-1}$
$k$	Contact spring constant	$\text{N m}^{-1}$
$L$	Length	$\text{m}$
$m$	Mass	$\text{kg}$
$n$	Number of	-
$P$	Pressure	$\text{Pa}$
$p$	Order of convergence	-

$r$	Radius	m
$r$	Refinement ratio	-
$S$	Shear modulus	Pa
$T$	Temperature	K
$t$	Time	s
$U$	Velocity	m s <sup>-1</sup>
$V$	Volume	m <sup>3</sup>
$x$	Spacial coordinate	-
$y$	Spacial coordinate	-
$z$	Spacial coordinate	-
$F$	Force vector	N
$g$	Gravitational acceleration vector	m s <sup>-2</sup>
$T$	Torque vector	N m
$U$	Velocity vector	m s <sup>-1</sup>
$x$	Position vector	m
$dx$	Cell length	m
$Db$	Stokes-Einstein diffusion coefficient	m <sup>2</sup> s <sup>-1</sup>

### Subscripts

$\Omega$	Rotation rate
$\omega$	Angular velocity
0	Initial
a	Added mass
avg	Average
B	Buoyancy
b	Brownian
cell	Cell
CFD	Computational fluid dynamics
col	Collision

crit	Critical
D	Drag
d	Damper / Dashpot
DEM	Discrete element method
eff	Effective
ent	Entrance
eq	Equation
exact	Exact
exp	Experiment
f	Fluid
fp	Fluid-particle
fric	Frictional
G	Shear
g	Gravitational
H	History
h	Hydraulic
hertz	Hertz
I	Inertia
i	Suffix notation
j	Suffix notation
L	Lift
L	Loss
M	Mass coefficient
mag	Magnus
max	Maximum
mod	Model
n	Normal
P	Pressure

p	Particle
pp	Particle-particle
pw	Particle-wall
ray	Rayleigh
rel	Relative
res	Resulting
roll	Rolling
s	Safety
s	Spring
saff	Saffman
smooth	Smoothing
stiff	Stiffness
t	Tangential
tot	Total
U	Velocity
w	Wall
<b>Superscripts</b>	
*	Dimensionless parameter

# Chapter 1

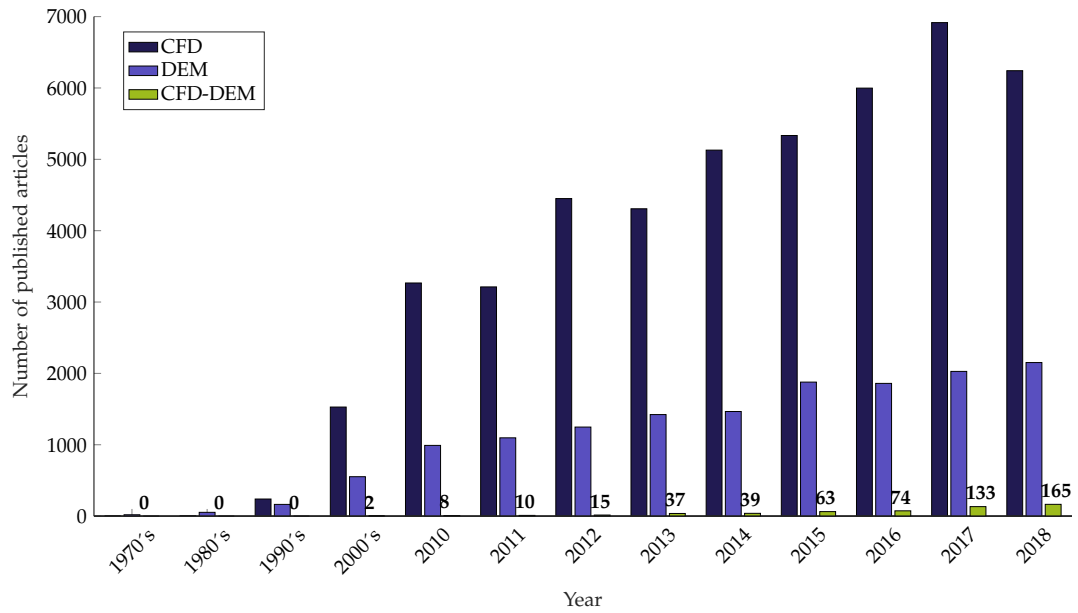
## Introduction

The presence of particles in small volumes can be found in a wide range of flows; sand particles in drinking water pipes (Neilands et al., 2012), corrosion debris in piping systems, flue gas particles in boilers (Pal et al., 2011) or unwanted particles in cooling systems. The particles tend to accumulate in some regions, which depends on the geometry of the piping and the properties of particles and fluid in the specific region. For many cases, this will not cause a problem. Slight accumulation of sand in drinking water pipes will not affect the availability of drinking water at the end user. In applications where heat transfer is involved, the accumulation might however pose a problem, as the particles can reduce heat transfer in the specific areas. This project investigates particle accumulation in a simple geometry consisting of a square duct with a block forcing a laminar fluid flow through a cavity, which is initially filled with particles.

### 1.1 Reasoning Behind Project

Many modern applications involving fluid (and particle) flows have been designed, investigated and evolved based on mathematical models, some simpler than others. Often the particles are not included in models, and for these cases Computational Fluid Dynamics (CFD) has been a widely used procedure. For models regarding only particles, Discrete Element Method (DEM) modelling is often used, however the field has not received as much focus from scientists as CFD. For more sophisticated models involving both fluid and particles, CFD can be used by means of multiphase simulations, where the particles are simulated as a secondary continuous fluid phase.

Another option is to couple the two methods in CFD-DEM models, which can produce good results, however at a high computational cost. The field of CFD-DEM modelling has only recently received a growing attention in the field of science, which is likely based on the fact, that computational power has become more easily available. Figure 1.1 illustrates the number of published articles for CFD, DEM and CFD-DEM from the 1970's until 2018.



**Figure 1.1:** Comparison of publications for CFD, DEM and CFD-DEM over the years. The emphasised numbers are the articles concerning CFD-DEM. The number of articles in the respective decades is an average providing a yearly number of published articles. The data is extracted from Web of Science at the time of writing.

As the development of CFD-DEM has not yet received much attention, this is an interesting field, and even simple investigations can help to further understand the behaviour of particles under influence of a fluid. In addition many CFD and DEM models have been validated separately, but only a limited number of papers have experimentally validated the coupled CFD-DEM modelling framework.



## 1.2 State of the Art Review

This section gives a brief description of different numerical approaches for modelling fluid and particulate applications as well as combining the two in a multiphase flow. This is done to elaborate the reasons for choosing CFD-DEM as a suitable modelling framework. After the numerical presentation, examples of previous work within the field of CFD-DEM are provided.

### 1.2.1 Numerical models

There are several numerical models capable of solving the fluid and particulate phases of a particle-fluid flow. Some of the methodologies are listed in Table 1.1.

**Table 1.1:** Numerical models for modelling the particulate or fluid phase.

Particulate models	Fluid models
Molecular Dynamics	Lattice-Boltzmann Method (LBM)
Event-driven Molecular Dynamics	Pseudo-particle Method
Direct Simulation Monte Carlo	Direct numerical simulation (DNS)
Discrete Element Method	Large Eddy Simulation (LES)

The scope of this project is to model particle accumulation in a fluid with focus on particles of sizes comparable to sand, thus the combination of the dense particulate and fluid phases are of interest.

The DEM method is chosen as the particulate model in this project. The Molecular Dynamics method is only suitable for atom scale, which is not the case for the particles used for this project (Berendsen, 1999). The Event-driven Molecular Dynamics method is most applicable for systems where particles are rarely in contact, this is not valid for particle accumulation, since the particles touch each other in the cavity (Pöschel and Schwager, 2005). Furthermore, a high amount of time steps is required, which makes the method inefficient with regard to computational time. The Direct Simulation Monte Carlo is a statistic method, which is unable to describe the forces acting on the individual particles (Camberos et al., 2008).

For the fluid simulations CFD is used. LBM uses particle clusters and applies a distribution function to these for modelling fluid flow. It is not able to trace the phase-interface in multi-phase problems (Mohamad, 2011). Pseudo-particle Method uses particles to simulate fluid behaviour on a microscopic scale, which still is fundamental and inefficient, thus unfit for this project (Ge and Li, 2001). DNS is a branch of CFD used for high-fidelity fluid analysis, and is able to solve turbulent flow. It is computational costly, and the amount of details provided by the method is not required for this project. LES is used for turbulent flows, where only large eddies are included in calculations. Since the flow is laminar in the squared duct, the LES is not of interest.

The behaviour of particles immersed in a fluid can be simulated using different approaches; Eulerian-Eulerian and Eulerian-Lagrangian, which can use the models presented above. In the next two sections these two approaches is presented.

### **Eulerian-Eulerian**

One Eulerian-Eulerian approach is the Two Fluid Model (TFM), which models the particulate phase as a second fluid and is computational efficient, and for this reason utilised in process modelling. This is a continuum approach and though the method is cost-effective, it must be adapted to fit the specific flow regimes and materials and relies on a series of assumptions and simplifications. Here e.g. the solid phase depends on some empirical correlations (Norouzi et al., 2016).

The TFM models particle behaviour on a mesoscopic level and does not resolve the individual particle behaviour (Norouzi et al., 2016). The model utilises the binary collision model to simulate contact between the particles, which leads to underestimation of the contact forces. The TFM is mostly valid for dilute flows (Sun and Xiao, 2016). An article by Chen and Wang (2014) concluded, that this model is less computational demanding compared to CFD-DEM, but unable to predict well known particle trajectories.

Tebowei et al. (2015) investigated the accumulation of sand particles in petroleum flow using TFM. They investigated the effect of ranging particle diameters as well as pipe geometry, which is seen in Figure 1.2.



**Figure 1.2:** Tebowei et al. (2015) investigated particle accumulation on both straight and V-inclined pipes using TFM.

They found that the particle size and the gradient of the pipe in the V-formation have significant impact on the accumulation of sand particles in the bend.

### 1.2.2 Eulerian-Lagrangian

An alternative to TFM is a coupling model using a fluid model and DEM. The choice of fluid model, such as DNS, LBM or CFD, depends on the application, available computational power and the parameter of interest. The choice of approach is a compromise between computational cost, difficulty and level of details needed. For this project, including the forces acting on the individual particles will give more physical representation of the particle behaviour compared to TFM. A CFD-DEM approach will be used in this project, since this is a computationally economic method able to predict fluid-particle dynamics with good precision.

In the next section previous studies of DEM and the coupling of CFD and DEM are presented to show some of what has been done in this field before.

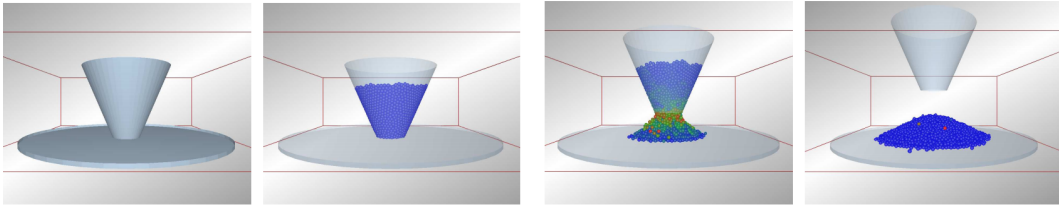
### 1.2.3 Previous studies based on CFD and DEM

Research has been conducted using the CFD and DEM methodologies and the coupled CFD-DEM model. Numerous studies on CFD have been conducted over the years (as seen in Figure 1.1) and the methodology is proved applicable for almost any fluid flow. Thus, no examples of application with only CFD is presented in the following paragraphs. Examples are provided for the DEM and CFD-DEM methodologies.

The DEM method has been used to investigate many different granular processes. Mellmann et al. (2011) suggested the basis for a model, which Mellmann and Teodorov (2011) proved was comparable with empiric tests. The model can be used for the investigation of grain flow in a mixed-flow grain dryer. These simulations were conducted using homogeneous particles to simulate the grain flow.

The DEM approach was used to investigate agglomerate breakage dependent on the initial shape for cuboidal, cylindrical and spherical agglomerates by Liu et al. (2010). Contrary to Mellmann and Teodorov (2011), this investigation included a normal size distribution.

The accumulation of larger granules injected from a funnel, as seen in Figure 1.3, was investigated by Li et al. (2011), who concluded, that the particle-particle rolling friction is the more dominant parameter of the resulting angle of repose. In 2018, Zu et al. (2018) used a very similar procedure to develop a regression model able to predict the repose angle.



**Figure 1.3:** The process of granular accumulation investigation performed by Li et al. (2011).

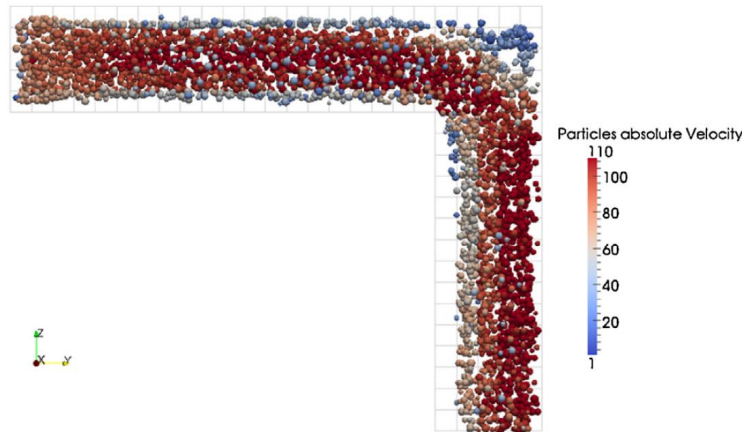
DEM modelling has been limited by computational cost, as the problems of interest often require large number of particles and very small time steps. Recent advances of computing speed and power has made the subject of DEM modelling more interesting. In general, using appropriate time steps is a good way of improving computational efficiency. Usually CFD and DEM simulations does not require the same temporal resolution, why choosing larger time steps for CFD than DEM can improve calculation cost immensely. The difference in time steps can however result in instability of the model. The stability can be improved by implementing smoothing of forces. To further reduce the cost, many researchers, such as Tsuji et al. (1993) and Hærvig et al. (2017), have suggested guidelines

for reducing the cost of computation in DEM simulation by modifying the Young's modulus in order to increase the time step size required to dissolve particle collisions. This method is used in this project.

By combining the two methods, a more sophisticated model can be constructed, which uses the CFD approach to model the fluid behaviour and DEM method to model the granular behaviour.

The subject of particle accumulation in flow has been investigated by others in various ways and scales.

Akbarzadeh and Hrymak (2016) investigated the particle-particle and particle-wall collision using the CFD-DEM methodology. The simulations are performed using a  $90^\circ$  bend in a square duct with a particle-laden laminar flow. The geometry is displayed in Figure 1.4. Similar to the geometry used by Akbarzadeh and Hrymak (2016), the geometry in this project consists of a square duct with four  $90^\circ$  bends. The particles accumulate in the bend due to the high inertia and lack of drag force in the fluid adjacent to the wall.



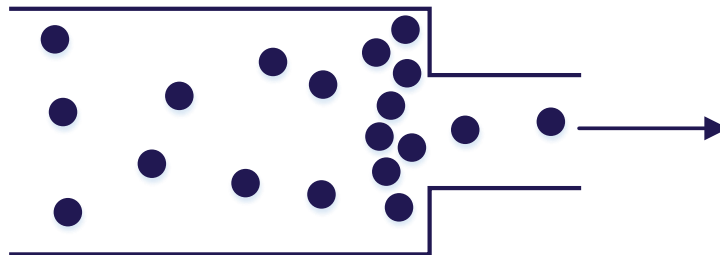
**Figure 1.4:** The investigated geometry in the study conducted by Akbarzadeh and Hrymak (2016) shows particle accumulation in the corner.

The study concludes that the drag force governs the particle motion and determines whether they accumulate in the bend or not. Furthermore, the study investigates the effect of making the right wall moving. The results from the CFD-DEM simulation shows that the moving wall is able to

decrease the particle agglomeration.

Shahzad et al. (2016) and Shahzad et al. (2018) analysed clogging of nanoparticles in microchannels using both DEM and CFD-DEM simulations. They concluded that a two-way coupling using CFD-DEM can simulate the particle-fluid dynamics for microfluid applications more accurately than DEM. The conclusion is based on comparisons with simulation results from Marshall and Li (2014), who, as opposed to earlier work on particle adhesion, investigated the subject by means of a soft-sphere DEM model.

Mondal et al. (2016) have investigated, how particles accumulate at a contraction in a pipe flow. The investigations are conducted for a 2D geometry (the width of the channel is 1.5 the particle diameter) and for solid concentrations between 1% and 50%. The scope of the investigations was to investigate the particle accumulation for different steady flow conditions. The numerical model is validated against existing experimental data on the subject. The validated model is used for a parametric study, where the particle accumulation is investigated. They found a connection between the particle concentration and jamming at the contraction.



**Figure 1.5:** Illustration of the flow in the study by Mondal et al. (2016).

As this section has underlined, some work regarding particle accumulation in square duct geometries has been done, but many aspects of the subject are still to be examined. Based on the previously section, the problem statement, the geometry of interest along with different modelling considerations and finally the assumptions and limitations made in this project is presented.

## 1.3 Problem Statement

As the possibility to study particle accumulation in fluid flow using numerical simulations has become more available, the subject has gained increased attention within a broad array of scientific fields. The investigation of particle accumulation in small scale square ducts with a cavity has seemingly not yet received much attention from scientists, however the results of such an investigation can help to develop the understanding of particle accumulation in fluid flow, which leads to the problem statement of this report:

*The scope of this project is to develop and validate a numerical model able to predict particle transport and accumulation in a cavity within a laminar square duct flow. Furthermore, investigate how the accumulation of particles is affected by flow Reynolds number and particle-to-fluid density ratio.*

In order to analyse the behaviour of particle movement in a geometry as function of a laminar fluid flow, a numerical model is developed with the relevant force models applied. The included forces are based on a scaling analysis of the forces acting on the particles submerged in a moving fluid along with the particle collision forces. Besides the development of the numerical model, an experimental setup is designed, constructed and used for validating the numerical model.

When the model is validated, it can be employed for a parametric study of how the fluid Reynolds number and particle-to-fluid density ratio affects the behaviour of particles in the geometry of interest.

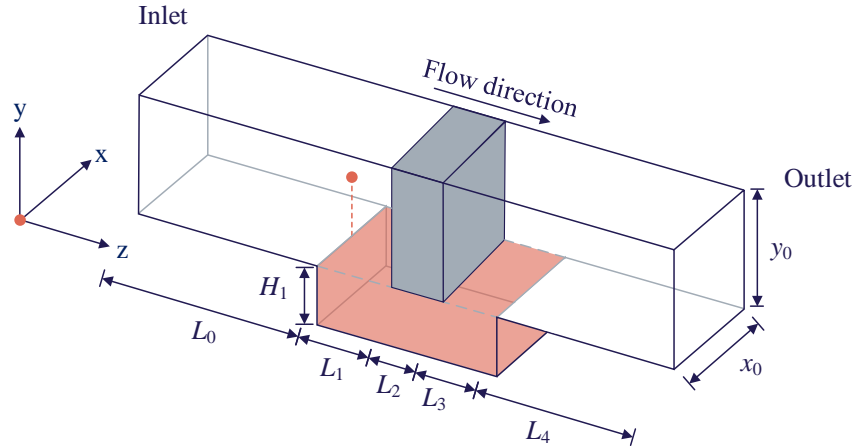
### 1.3.1 Geometry

The geometry used for this investigation consists of a square duct with a cavity in the bottom and a block which forces the fluid through the cavity, as illustrated conceptually in Figure 1.6. The dimensions of both duct, block and cavity are all relative to the hydraulic diameter, which is also valid for any length dimension mentioned throughout this report.

The inlet and outlet region of the geometry are placed with some distance from the cavity itself. The distance between cavity and geometry exit

also allows for investigation of particle behaviour after leaving the cavity.

As seen in Figure 1.6, *origin* is located in the middle of the channel at the edge before reaching the cavity. The definition of all dimensionless distances are listed in Table 1.2.



**Figure 1.6:** Illustration of geometry. All distances are relative to hydraulic diameter. Red area is initially filled with particles. Origin is placed at the red dot - dashed red line elaborates that origin is at  $0.5 x_0$ .

**Table 1.2:** Dimensionless distances of geometry referring to Figure 1.6. The lengths before and after the block is identical, thus  $L_1 = L_3$ .

Description	Notation	Dimension [-]
Channel width	$x_0/D_h$	1
Channel height	$y_0/D_h$	1
Channel length	$z_0/D_h$	8.5
Cavity height	$H_1/D_h$	0.5
Channel length before cavity	$L_0/D_h$	1.5
Cavity length	$(L_1 + L_2 + L_3)/D_h$	4
Block length	$L_2/D_h$	1
Channel length after cavity	$L_4/D_h$	3

The red area in Figure 1.6 is initially filled with particles.



### 1.3.2 Particles and fluid

The particles used in this project are modelled as being perfectly spherical with uniform properties throughout the entire volume. All particles are modelled as being identical, with a diameter of  $0.07 D_h$ , thus the particle size is comparable with the size of coarse sand particles.

Next a scaling analysis is made in order to find the relevant fluid particle forces to include in the numerical model.

### 1.3.3 Selection of fluid particle forces

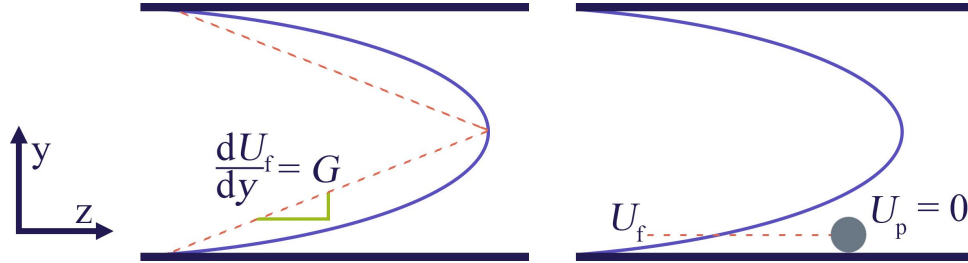
To identify which forces to include in the numerical model, a scaling analysis is conducted as suggested by Marshall and Li (2014). The analysis is based on comparison of a series of parameters, displayed in Table 1.3, and is intended for cases with *sub-micron* particles, which is not the case in this project, and thus this is only used as an indication.

**Table 1.3:** The parameters included in the scaling analysis along with their values (Marshall and Li, 2014).

Parameter	Definition	Value
Dimensionless particle diameter	$\epsilon = D_p/D_h$	0.07
Fluid Reynolds number	$Re_f = \rho_f U_{f,avg} D_h/\mu_f$	2000
Particle Reynolds number	$Re_p = \rho_f D_p (U_f - U_p)/\mu_f$	31
Shear Reynolds number	$Re_G = \rho_f G D_p^2/\mu_f$	39
Density ratio	$\chi = \rho_p/\rho_f$	0.4
Stokes number	$St = \rho_p D_p^2 U_f/(18 \mu_f D_h)$	0.3
Particle Froude number	$Fr = U_{f,avg}/\sqrt{g D_p}$	0.9
Particle Peclet number	$Pe_p = D_p (U_f U_p)/Db$	$9.2 \cdot 10^{10}$

The shear rate ( $G$ ) is given by the derivative of the velocity profile, which for this case is estimated linearly as illustrated in Figure 1.7 left side. The approximation of the particle Reynolds number ( $Re_p$ ) is made at a distance

of the particle radius from the wall, where  $U_p = 0$ , as seen in Figure 1.7 right. The Stokes number (St) and particle peclet number ( $Pe_p$ ) is determined under same assumptions as  $Re_p$ . The Stokes-Einstein relation (Db) is given as  $Db \approx 3.1 \cdot 10^{-16}$  for this case (Marshall and Li, 2014).



**Figure 1.7:** Shear rate is approximated on a linear gradient of the velocity profile. Particle Reynolds number is determined near to the wall.

The scaling analysis defines the order of magnitude of forces based on the parameters relative to the magnitude of particle drag. The analysis is based on a single particle suspended in a fluid flow with the assumption, that the particle Reynolds number is low (in this case 31). The estimate of order of magnitude is listed in Table 1.4, along with the force ratio and the calculated order of magnitude. The calculations are based on a critical fluid Reynolds number of 2000 and laminar flow is assumed.

From Table 1.4 it is seen that, Magnus lift force and the gravitational force has the highest order of magnitude. The Brownian force is approximately  $\mathcal{O}(10^{-6})$  lower than the drag force and therefor it is neglected in the numerical model. It must be kept in mind, that the scaling analysis is made for a single sub-micron particle, and as this case is for multiple larger particles, drag and gravity forces are likely the dominating factors. The importance of drag force was also underlined by Akbarzadeh and Hrymak (2016), as mentioned in Section 1.2.3.

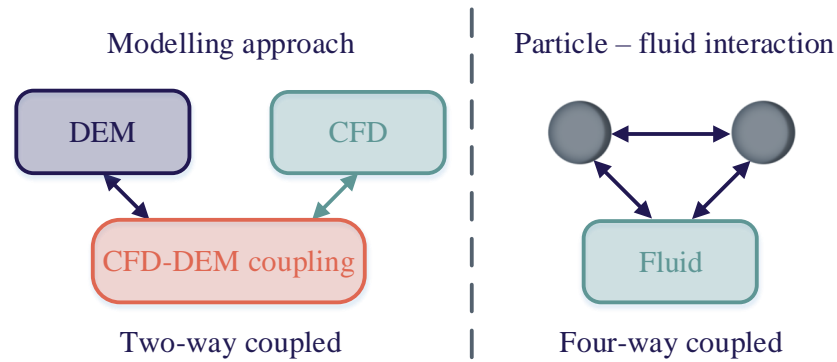
**Table 1.4:** Estimates of order of magnitude ( $\mathcal{O}$ ) for different fluid forces on particles at  $\text{Re}_f = 2000$ .

Force contribution	Force ratio	$\mathcal{O}$	Calculated $\mathcal{O}$	Included
Drag	$F_D/F_D$	1	1	Yes
Saffman lift	$F_{\text{saff}}/F_D$	$\mathcal{O}(\text{Re}_G^{1/2})$	6.2	Yes
Magnus lift	$F_{\text{mag}}/F_D$	$\mathcal{O}(\text{Re}_G)$	39.2	Yes
Added mass	$F_a/F_I$	$\mathcal{O}(\chi^{-1})$	2.5	Yes
Pressure-gradient	$F_p/F_I$	$\mathcal{O}(\chi^{-1})$	2.5	Yes
Basset history	$F_H/F_D$	$\mathcal{O}(\text{Re}_p^{1/2})$	5.6	Yes
Gravitational	$F_g/F_D$	$\mathcal{O}(\epsilon \text{Fr}^2)^{-1}$	17.6	Yes
Brownian	$F_b/F_D$	$\mathcal{O}(\epsilon^{-1} \text{St} \text{Pe}_f^{1/2})$	$3 \cdot 10^{-6}$	No

### 1.3.4 General numerical considerations

For the numerical study a two-way coupled model using both CFD and DEM is developed. This means that the the particles are affected by the fluid flow and that momentum is transferred back to the fluid phase through a momentum exchange term. The particulate model (DEM) transfers information to the coupling which transfers it to the fluid model (CFD). The fluid and particulate models are not directly connected.

Often in two-way coupling the particle volume fraction is small, but the mass ratio might be significant (Benra et al., 2011). This is the case for this project. The two-way coupled modelling approach is applied, since it is more accurate than the one-way coupled (Horwitz and Mani, 2016). The particle-fluid interactions are four-way coupled, since interactions are considered between fluid-particle and particle-particle. This is especially important for cases where fluid-particle volume ratios are high, which is the case in this project. The coupling is illustrated in Figure 1.8.



**Figure 1.8:** Distinction between coupling approaches in modeling and interactions.

The next sections concerns the considerations of mesh size used in the numerical model.

### 1.3.5 Mesh and particle sizes

For CFD, a mesh with quite small cell sizes are common, especially at areas where critical changes in the flow field are expected. When coupling is introduced, the size of cells must consider the size of the particles used in the DEM model. There are two different approaches often used for this, which can be seen in Figure 1.9.

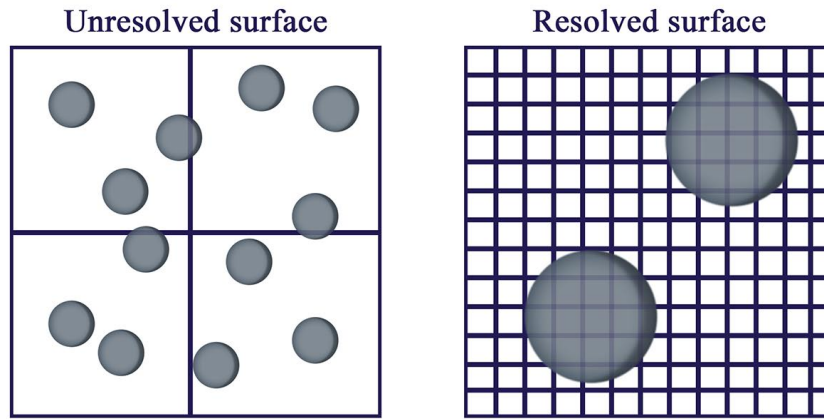
#### Resolved surface method

The resolved surface model requires the size of a cell to be significantly smaller than the particle size. The detailed fluid flow surrounding the particle can be obtained, because the fluid flow is fully resolved. This method has a high computational cost, and is not feasible for systems with many particles.

#### Unresolved surface method

For many applications the finely resolved fluid flow around particles is not important and the calculation of this is therefore not feasible. Instead it can be desired to investigate the fluid-particle dynamics of systems, where a large number of particles are present. For this the unresolved surface

method, which requires particles to be smaller than the fluid cell sizes, can be used. This model neglects the particles impact on the fluid flow within a cell. As the particles can always be contained within a cell, it is possible to use Navier-Stokes equations on the void fraction (fluid phase part) of a cell. The solid phase part can be determined by local averaging of particles (Norouzi et al., 2016). The unresolved surface method is used as computational power available is limited and the resolved flow around every particle is not of high interest for the purpose of this project.



**Figure 1.9:** Unresolved surface is for particles smaller than a cell, whereas resolved surface is used for particles larger than cells.

### 1.3.6 Assumptions and limitations

For the sake of simplification a series of assumptions and limitations are introduced in this project. These are displayed in Table 1.5 and further explained in the two sections below.

**Table 1.5:** The assumptions and limitations in the project.

Assumptions	Limitations
Van der Waals force can be neglected	No heat transfer
Electrostatic forces can be neglected	No Adhesion

**Assumptions**

The electrostatic forces are assumed to be negligible, when the particles are submerged into water. Evaluating the force according to Coulombs law shows, that the electrostatic force is negligible for this project.

The van der Waals force is neglected, since it is assumed not to have influence on the particle behavior. The attractive and repulsive components of van der Waals force have a range of nanometers. The van der Waal forces are dominant for particles, where the distance between particles are greater than the particle radius. Van der Waals force is considerable when working with adhesive particles (Marshall and Li, 2014).

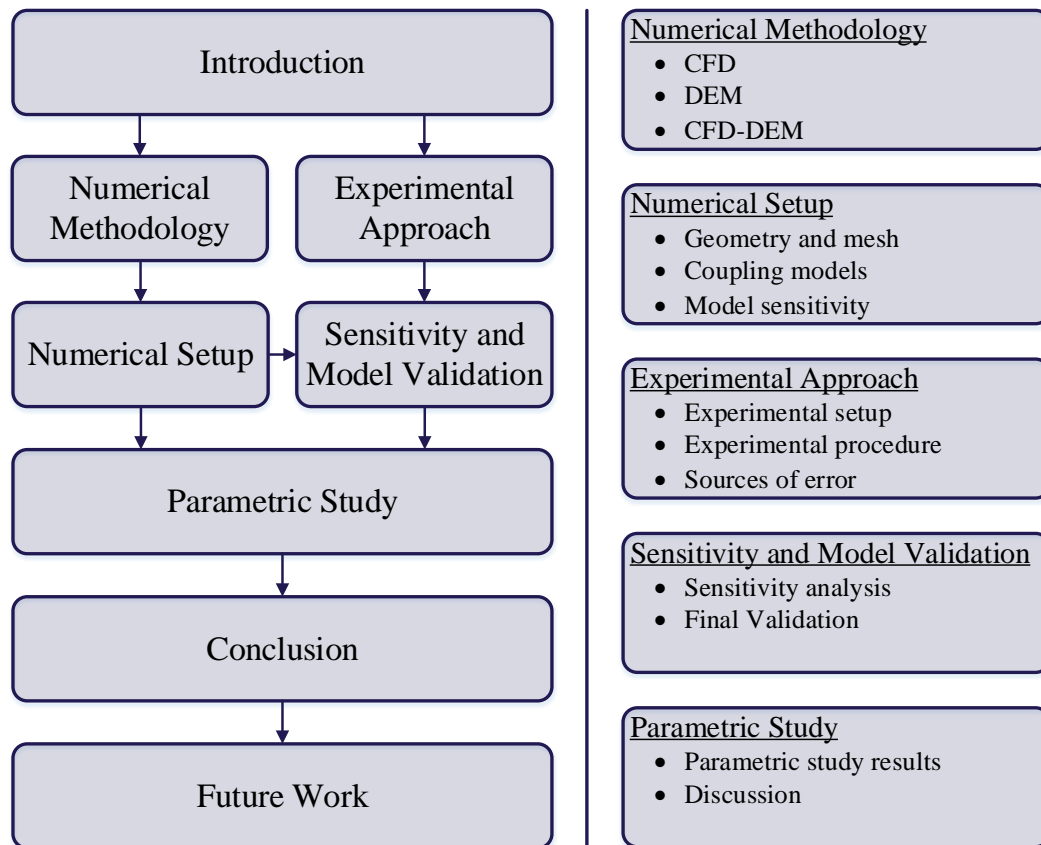
**Limitations**

As mentioned in Section 1.3.2 the particles and fluid have the same temperature, so heat transfer can be neglected.

The adhesive forces are not considered in this project, since due to the size of the particles, which are in the scale of millimetres, the adhesion forces are insignificant.

## 1.4 Report Structure

In this section the report structure is illustrated in Figure 1.10, which shows a simple flowchart of the chapters on the left hand side. On the right hand side of the figure, the main chapters are further described with a few keywords to elaborate the main subjects of the respective chapter.



**Figure 1.10:** Left: Flowchart of the chapters in the report. Right: Keywords of subjects treated in the specific chapter.





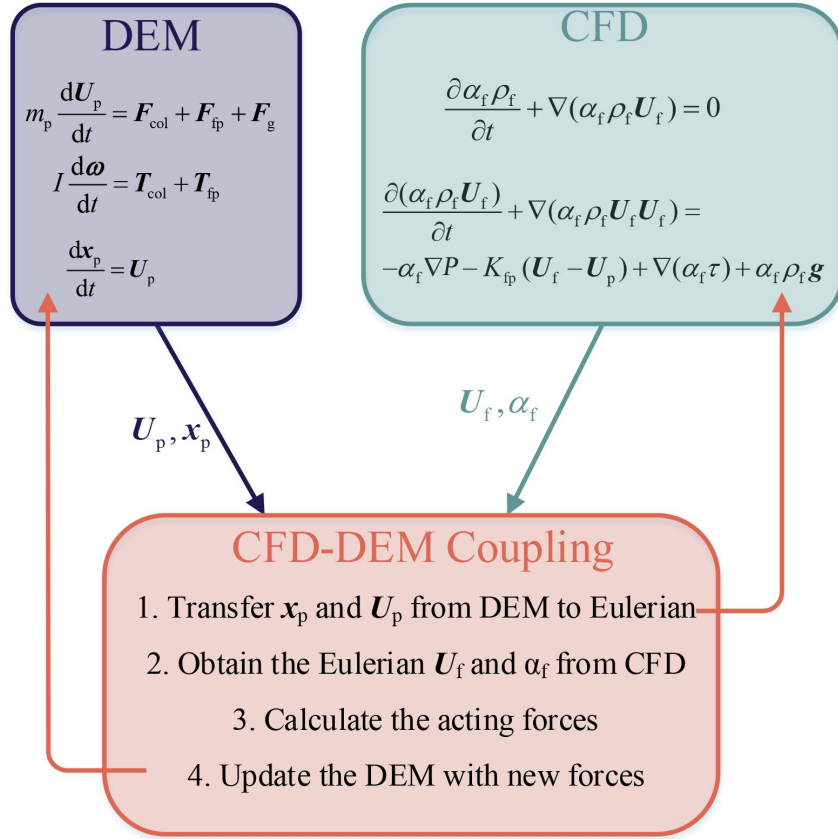
# Chapter 2

## Numerical Methodology

This chapter provides an overview of the methodology used for modelling the fluid and particulate phases, as well as a description of the coupling between them. Each method used is described in a section, starting with an overview of the coupled method in Section 2.1, then follows a description of the fluid phase methodology in Section 2.2, followed by the particulate phase in Section 2.3, and finally the two-way coupling of the methods is described in Section 2.4. The specific procedure for modelling is described in Chapter 3.

### 2.1 Overview of the Modelling Methodology

The overall methodology of a coupled Eulerian-Lagrangian (CFD-DEM) model is illustrated in Figure 2.1. The model couples the fluid phase (CFD) and particulate phase (DEM). When the models are coupled, the DEM model calculates the forces acting on particles due to particle-particle collisions, fluid-particle interaction, such as drag and lift forces, and gravitational forces. The particulate model provides information about the particle locations and velocities to the CFD-DEM coupling. The coupling model transfers the velocity and position to the Eulerian framework from the Lagrangian framework after which the CFD method solves the Navier-Stokes equations and determines the velocity and pressure field. When the Navier-Stokes equations have converged, the fluid-particle forces are updated and used as input for the DEM model calculating new velocities and positions of the particles (Zhao and Shan, 2013).



**Figure 2.1:** Flowchart of the two-way coupled modelling methodology. Where  $\alpha_f$  is the void fraction and  $K_{fp}$  is the fluid-particle momentum exchange. Subscripts f and p denotes fluid and particle respectively.

This is the general CFD-DEM methodology. The CFD model is described in the next section.

## 2.2 Computational Fluid Dynamics

In general, CFD is used to solve pressure and velocity fields based on solving of conservation equations for fluid motion, which means that the mass is conserved along with momentum and energy.

### 2.2.1 Governing equations for CFD

The governing equations in this project are the continuity equation and momentum equation for each of the directions. The continuity equation is given by Equation 2.1 and the momentum equation for the  $x$ -direction is seen in Equation 2.2. The momentum equations for  $y$ - and  $z$ -direction are given accordingly. These equations include the coupling terms given by  $\alpha_f$ , which is the void fraction. By setting  $\alpha_f = 1$  and  $\mathbf{U}_p = \mathbf{U}_f$ , the effects of any particles in the model is ignored, and the general Navier-Stokes equations for CFD are obtained (Norouzi et al., 2016).

$$\frac{\partial \alpha_f \rho_f}{\partial t} + \nabla(\alpha_f \rho_f \mathbf{U}_f) = 0 \quad (2.1)$$

$$\frac{\partial(\alpha_f \rho_f \mathbf{U}_f)}{\partial t} + \nabla(\alpha_f \rho_f \mathbf{U}_f \mathbf{U}_f) = -\alpha_f \nabla P - K_{fp} (\mathbf{U}_f - \mathbf{U}_p) + \nabla(\alpha_f \boldsymbol{\tau}) + \alpha_f \rho_f \mathbf{g} \quad (2.2)$$

Where

$\alpha_f$	Void fraction of fluid
$\rho_f$	Fluid density
$\mathbf{U}_f, \mathbf{U}_p$	Fluid and particle velocity
$P$	Pressure
$K_{fp}$	Implicit momentum exchange between fluid and particle
$\boldsymbol{\tau}$	Liquid stress tensor
$\mathbf{g}$	Gravitational acceleration

These equations consider the fraction of a cell in fluid state and calculates the implicit momentum exchange between solid and fluid by:

$$K_{fp} = \frac{\alpha_f |\Sigma \mathbf{F}|}{V_{\text{cell}} |\mathbf{U}_f - \mathbf{U}_p|} \quad (2.3)$$

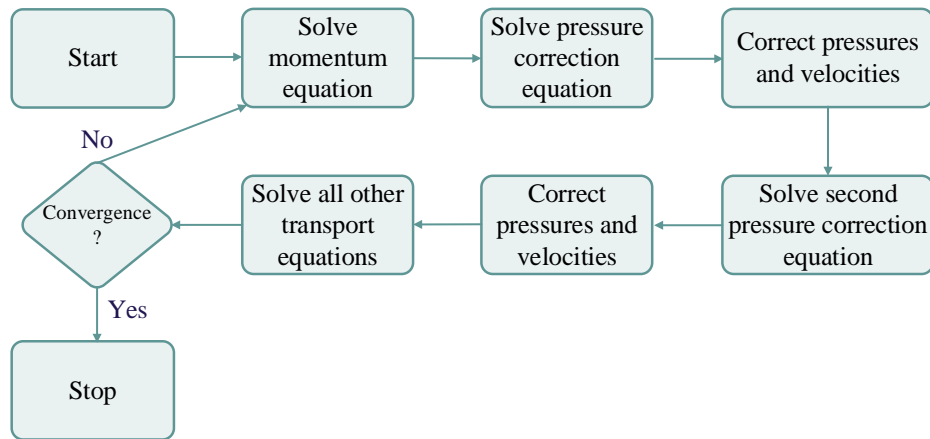
Where

$\mathbf{F}$	Forces
$V_{\text{cell}}$	Cell volume

When the  $K_{fp}$  term is multiplied by the difference in the fluid and particle velocity, the term in Equation 2.2 containing  $K_{fp}$  equals the force per volume for each cell. To solve the governing equations several algorithms exist.

### 2.2.2 CFD methodology

For this project the Pressure Implicit with Splitting of Operators (PISO) algorithm is applied. The PISO solver can solve both steady state and transient problems with incompressible fluids. The PISO algorithm is an extension of the SIMPLE algorithm, where additional predictor step is included meaning the PISO algorithm calculates the cell pressures and velocities, which are corrected using different pressure corrector functions. Figure 2.2 shows a flowchart of the PISO algorithm (Versteeg and Malalasekera, 2007).



**Figure 2.2:** Flowchart of the PISO algorithm.

The first three steps of the algorithm are identical to the SIMPLE algorithm. The algorithm is initiated by a predictor step, where the pressure and velocities are guessed. After determining the initial pressure and velocities the momentum equations are solved. Then the first pressure corrector equation is used to update the initial pressure and velocities. After correcting the pressure and velocities the second corrector equation is solved. Thus, the pressure and velocities are updated again. Then the other transport equations are solved. Afterwards the convergence criteria is checked,

and if convergence is obtained, the algorithm stops. If the convergence criteria is not fulfilled the corrected pressure and velocities are used as input for the next iteration. Calculations are continued until convergence is obtained. The PISO solver applies different schemes for solving different parts of the momentum and continuity equations. The schemes are described in Section 3.2.1 on page 44. In order to ensure a converging solution for the CFD a proper time step must be chosen, which is described in the next section.

### 2.2.3 Time step

The time step for simulation is an important factor. The stability of a CFD model is dependent on the choice of a suitable time step. To determine this, the local Courant number, seen in Equation 2.4, is used:

$$Co = \frac{|\mathbf{u}_{\text{cell}}| \Delta t_{\text{CFD}}}{\Delta x} \quad (2.4)$$

Where

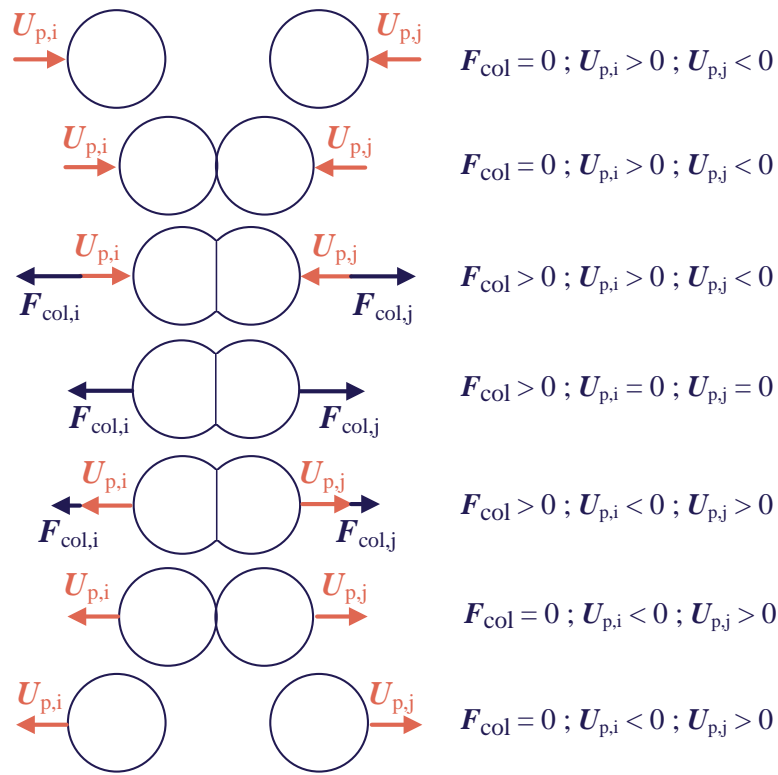
$Co$	Courant number
$\mathbf{u}_{\text{cell}}$	Local cell velocity
$\Delta t_{\text{CFD}}$	CFD time step
$\Delta x$	Cell length

In this project it is ensured, that the Courant number is maintained below a value of 1, thus it is ensured that the fluid does not travel further than a single cell per time step (Versteeg and Malalasekera, 2007). Simulating with a very small time step or an unnecessary high number of cells will significantly increase the computational cost. A grid independence study is conducted in Section 3.5.2 on page 53 along with a calculation of the CFD time step to assess the computational cost. Before determining the cell length in a CFD-DEM model, the DEM model must be considered.

## 2.3 Discrete Element Method

This section introduces the modelling methodology of the discrete element method. The method is presented by the governing equations and a flowchart of the methodology. The forces applied on the particles are calculated by dividing the forces into the categories; collision and gravitational forces.

The concept of DEM is to simulate the collision and deformation of particles, which is illustrated in Figure 2.3.



**Figure 2.3:** The steps during a collision of two particles. The deformation is exaggerated.

The two particles initially have constant velocities of opposite direction, while no collision force is involved. This continues until just before the particles touch. As particles collide and deform, there is a collision force and

velocities continue in the initial direction until the maximum deformation is reached, after which the particle velocities change direction. Just as the collision has ended, no collision force is present, and the velocities are constant. The amount of deformation depends on the particle material, and has been exaggerated in Figure 2.3. In DEM the particle deformation is not included, but instead calculated using an overlap, which is explained in detail in Section 2.3.3. Like in CFD, the DEM method has governing equations, as described next.

### 2.3.1 Governing equations for DEM

The DEM methodology calculates the position and the velocity of particles using Newton's second law of motion to determine forces acting on the individual particles, as given by the governing equations in Equation 2.5 and Equation 2.8 (Marshall and Li, 2014). The total sum of forces on a single particle is:

$$m_p \frac{d\mathbf{U}_p}{dt} = \sum \mathbf{F}_{\text{col}} + \mathbf{F}_g \quad (2.5)$$

Where

$m_p$	Mass of a particle
$\mathbf{F}_{\text{col}}$	Collision force
$\mathbf{F}_g$	Gravitational force

The velocity of the particles, used in Equation 2.5 for calculating the total force, is given by:

$$\frac{d\mathbf{x}_p}{dt} = \mathbf{U}_p \quad (2.6)$$

Where  $\mathbf{x}_p$  is the particle position and  $t$  is the time. The gravitational force in Equation 2.5 is given by:

$$\mathbf{F}_g = m_p \mathbf{g} \quad (2.7)$$

The torque exerted on the a particle is calculated by:

$$I \frac{d\boldsymbol{\omega}}{dt} = \sum \mathbf{T}_{\text{col}} \quad (2.8)$$

Where

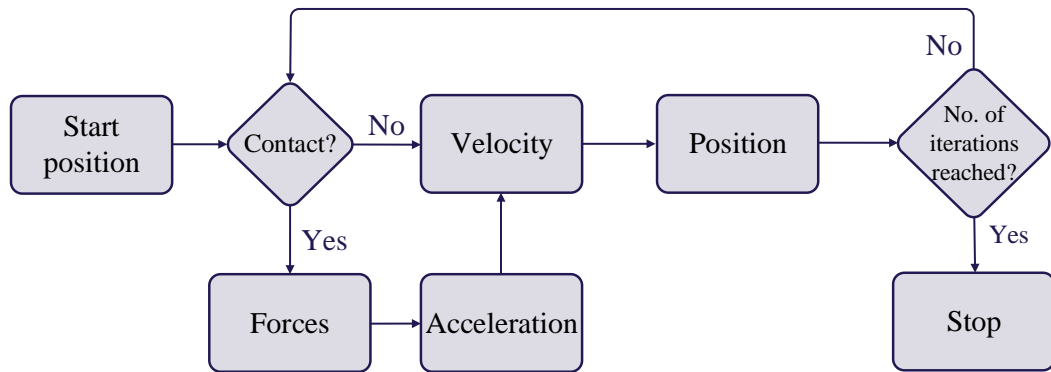
$I$	Moment of inertia
$\omega$	Angular velocity of the particle
$T_{\text{col}}$	Torque

The forces acting on particles are solved in steps for all particles.

The equations are solved by an iterative process which is elaborated in the next section.

### 2.3.2 DEM methodology

The iterative calculation procedure for DEM is illustrated in Figure 2.4. The initial input is the starting position and velocity of a given particle. Based on the position of the particle, any contact between the walls or other particles is determined. For all particles in contact, the interaction force, acceleration and velocity is calculated. Based on the position of previous time step and the velocity of the particle the position of next time step can be determined. This procedure continues until the maximum number of iterations are reached.



**Figure 2.4:** Flowchart of the discrete element method.

The forces are divided into different categories; the collision forces and the gravitational force. The collision forces are described in the following sections.



### 2.3.3 Collision forces

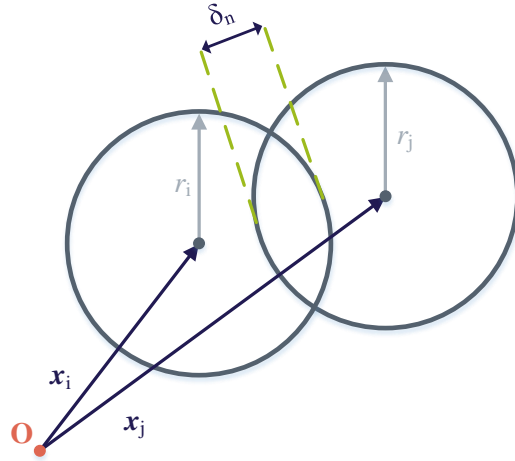
Collision forces are determined using both normal and tangential particle friction forces in a parallel spring-slider-dashpot procedure. The forces depend on the deformation, which is calculated as an overlap, given by Equation 2.9 for the normal displacement, where the  $\delta_n$  is the normal collision overlap of the particles as seen in Figure 2.5. The tangential displacement ( $\delta_t$ ) is estimated from Equation 2.10.

$$\delta_n = r_i + r_j - |\mathbf{x}_i - \mathbf{x}_j| \quad (2.9)$$

$$\delta_t \cong \delta_{t,0} + \mathbf{U}_{\text{rel},t} \Delta t_{\text{DEM}} \quad (2.10)$$

Where

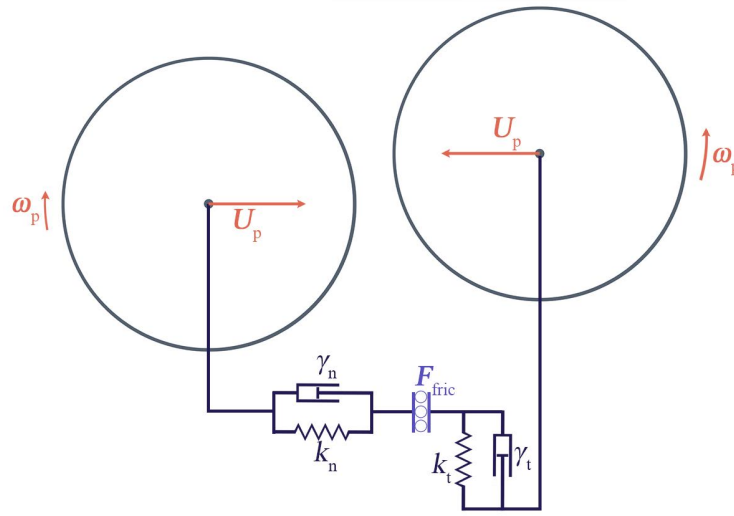
$r$	Particle radii for i and j
$\mathbf{U}_{\text{rel}}$	Relative velocity
$\delta_n, \delta_t$	Inter-particle overlap in normal and tangential direction



**Figure 2.5:** The overlap is defined from the radii and position of the particles.

An example of the spring-slider-dashpot interaction between two particles is illustrated in Figure 2.6. The general collision force for two particles is

given by Equation 2.11 (Renzo and Maio, 2005). The normal collision force is the sum of elastic and dissipative forces in the normal direction, while the tangential collision force describes the sliding, twisting and rolling relating to the tangential surface displacement. The surface friction force is modelled using the tangential slider illustrated in Figure 2.6. When a particle experiences several collisions simultaneously the collision force exerted on the particle is the sum of collision forces exchanged with all the colliding particles.



**Figure 2.6:** Inter-particle force is determined using a parallel spring-slider-dashpot arrangement.

Equation 2.11 for the collision force is given here:

$$\mathbf{F}_{col} = (k_n \delta_n - \gamma_n \mathbf{U}_{rel,n}) + (k_t \delta_t - \gamma_t \mathbf{U}_{rel,t}) \quad (2.11)$$

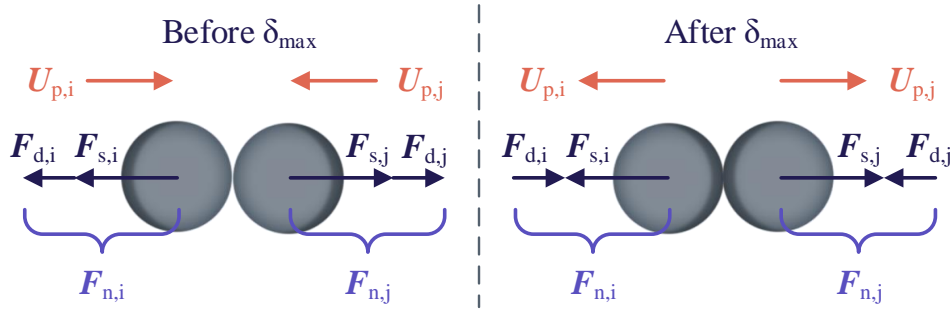
Where

$k_n, k_t$	The normal and tangential contact spring constant
$\gamma_n, \gamma_t$	The normal and tangential visco-elastic damping constant

The spring ( $k$ ) and dashpot ( $\gamma$ ) constants are determined using the equations described in Appendix A.

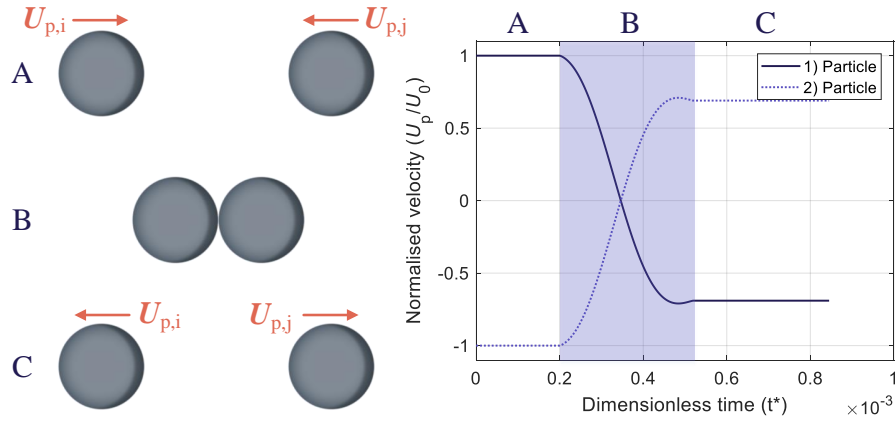
Inter-particle interaction is only present, when the overlapping distance ( $\delta$ ) is positive. Upon interaction, the direction of forces must be kept in

mind; the damper force is always acting in the opposite direction of the particle motion. Whereas the spring force is acting in the opposite direction as  $\delta$  increases, and in the same direction as the particle motion during a decrease of  $\delta$ . In Figure 2.7 an illustration of the direction of the spring and damper forces are given before and after a collision between two particles.



**Figure 2.7:** Before the particle overlap reaches  $\delta_{\max}$  both spring and damper forces ( $F_s$  and  $F_d$ ) act opposite the particle motion. As  $\delta$  decreases from  $\delta_{\max}$  the spring force acts in the particle motion direction, while damper force acts opposing to particle motion.

The overlapping interaction can be modelled using two different approaches: *hard-sphere* or *soft-sphere*. The hard-sphere procedure calculates the inter-particle interaction instantaneously and is incapable of resolving multiple simultaneous cohesive and contact forces. The soft-sphere approach dissolves the interaction in a series of time steps and can handle multiple simultaneous forces. As the purpose of this report is to investigate multiple forces simultaneously, the soft-sphere model is used. The soft-sphere model can be described using a simple 1D case of two glass particles colliding, which is illustrated in Figure 2.8.



**Figure 2.8:** **A:** As particles approach each other, particle velocities are constant. **B:** During collision the velocities decrease to zero, at  $\delta_{\max}$  and change direction due to the spring force. **C:** After collision ends the particle velocities are constant but in the opposite direction.

The dimensionless time in Figure 2.8 is defined as:

$$t^* = t \frac{U_{p,\max}}{D_p} \quad (2.12)$$

Where  $t^*$  is the dimensionless time. The particles in the figure have constant velocity before impact illustrated by situation **A**. Situation **B** shows the particle contact. During the contact the velocities first reduce to 0, where the two lines cross in Figure 2.8. This is where the maximum overlapping distance is reached. After this point, the particle motion changes direction and increases in velocity. Just before the particle collision ends, the velocity peaks and reduces slightly. This is caused by the damper force being stronger than the spring force. For a case with no damper, the resulting absolute velocity of a particle equals the initial absolute velocity, however the direction is opposite. The damper force can be changed by altering the coefficient of restitution, which is described in Equation A.7 on page 118. The damper, which reduces the resulting absolute velocity, depends on the material properties. Situation **C** shows the particles after collision has ended with a new velocity in the opposite direction. The same concept applies for tangential collisions. To ensure a correct model the DEM time step is important.

### 2.3.4 DEM time step

In DEM the time step must be sufficiently small in order to capture a collision between two or more particles. For DEM there are several different methods for estimation of a proper time scale. In this project three different time scales are considered, which refer to a meso- or macroscopic scales. These are the Rayleigh time scale, the contact time scale and the Hertz time criteria (Norouzi et al., 2016). These time scales are explained in this section.

The Rayleigh time is the time, it takes a Rayleigh wave to propagate along the surface of a particle. Thereby representing the time it takes for an elastic wave to transmit through a point at contact.

The Rayleigh time step is given by (Marigo and Stitt, 2015):

$$\Delta t_{\text{ray}} = \frac{\pi r \sqrt{\frac{\rho_p}{S}}}{0.1631 v + 0.8766} \quad (2.13)$$

Where

$\Delta t_{\text{ray}}$	Rayleigh time step
$S$	Shear modulus
$v$	Poisson's ratio

The time step for the calculation of the DEM part must be lower than the calculated  $\Delta t_{\text{ray}}$ .

The contact time scale is determined by the time two particles are in contact (the duration with positive overlapping distance). It can be calculated by setting up a small model with a sufficient high time step to properly resolve the contact.

The last parameter considered for determining the DEM time step is the Hertz criteria. The Hertz criteria defined as:

$$\Delta t_{\text{hertz}} = 2.87 \left( \frac{m_{\text{eff}}^2}{r_{\text{eff}} E_{\text{eff}}^2 \mathbf{U}_{\text{max}}} \right)^{0.2} \quad (2.14)$$

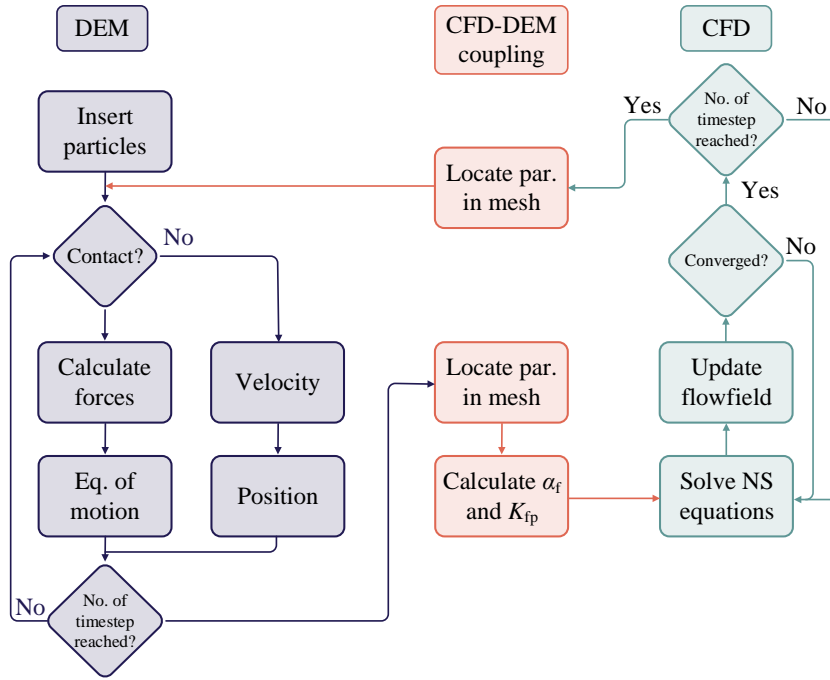
Where

$\Delta t_{\text{hertz}}$	Hertz time step
$m_{\text{eff}}$	Effective mass of a particle
$r_{\text{eff}}$	Effective particle radii
$E_{\text{eff}}$	Effective Young's Modulus (definition see Equation A.8 on page 118)
$\mathbf{U}_{\text{max}}$	Maximum fluid velocity

The Hertz time calculates the duration of a collision between two identical particles. Generally, the time step for DEM simulation is in the order of  $10^{-7}$  and even lower for stiff materials with a high Young's modulus (Norouzi et al., 2016), but the DEM time step can be increased by reducing the Young's modulus, as described in Section 3.5.3, thereby lowering the computational cost, which can be high for a pure DEM model and even higher when coupling CFD and DEM.

## 2.4 CFD-DEM Coupling Methodology

The CFD-DEM coupling combines the CFD and DEM models together by exchanging momentum between the two models. The procedure for running a coupled CFD-DEM simulation, as seen in the flowchart in Figure 2.9, requires initial properties of the particles in the domain. From the initiation, the DEM cycle, as described in Sec 2.3, is calculated for a predefined number of iterations based on the coupling interval, which is described in Section 2.5.3. When the defined number of iterations are reached, the velocity and position of all particles is sent from the Lagrangian domain (DEM domain) to the Eulerian domain (CFD domain) through the CFD-DEM coupling. Next the Navier-Stokes equations are solved and the flow field is updated. If the CFD has converged, the hydraulic forces acting on the particles are transferred to DEM via the CFD-DEM coupling. The cycle is repeated until maximum number of iterations are reached.



**Figure 2.9:** Flowchart of CFD-DEM coupling calculation procedure.

The governing equations for CFD-DEM along with the fluid-particle forces are presented in the next section.

## 2.5 Governing Equations and Models for CFD-DEM Coupling

When coupling the CFD and DEM models, the governing equations for the CFD model, listed in Equation 2.1 and Equation 2.2 on page 21 have an  $\alpha_f \neq 1$ , which indicates that cells can contain both fluid and particles. The governing equations for DEM is also altered when coupled with CFD, since fluid-particle effects is included, resulting in Equation 2.15 and Equation 2.16:

$$m_p \frac{d\mathbf{U}_p}{dt} = \sum \mathbf{F}_{col} + \mathbf{F}_{fp} + \mathbf{F}_g \quad (2.15)$$

$$I \frac{d\boldsymbol{\omega}}{dt} = \sum \mathbf{T}_{col} + \mathbf{T}_{fp} \quad (2.16)$$

Where

$F_{fp}$	Fluid-particle force
$T_{fp}$	Fluid-particle torque

The fluid-particle forces found from the scaling analysis are elaborated in the next section.

### 2.5.1 Fluid-particle forces

There are several fluid-particle forces, which can be included in model. Based on the scaling analysis in Section 1.3.2 on page 11, the following forces were found to have an impact on the particle-fluid interaction:

- Drag
- Magnus lift
- Pressure-gradient
- Saffman lift
- Added mass
- Basset history

The forces are explained throughout the following paragraphs with the respective equations and an illustration of, how the forces affect a particle.

#### Drag force

One of the primary forces in fluid flow is the drag forces exerted on particles in a flow stream. The drag force is generally given by (Mema et al., 2019):

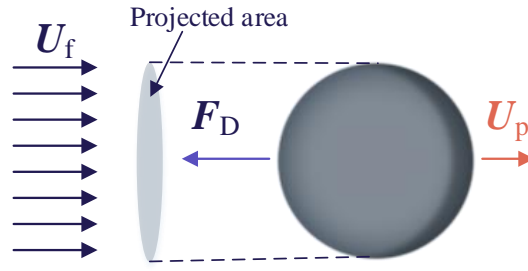
$$F_D = \frac{1}{2} C_D \rho_f A_p \left| \mathbf{u}_f - \mathbf{u}_p \right| \left( \mathbf{u}_f - \mathbf{u}_p \right) \quad (2.17)$$

Where

$C_D$	Drag coefficient
$A_p$	Projected area of a particle

The drag force is illustrated in Figure 2.10, where the projected area is shown. Here it is seen, how the drag force acts opposite to the fluid flow direction and particle velocity.





**Figure 2.10:** Illustration of the drag force and the projected area in a uniform flow field.

There are several models for calculating the drag force exerted on the particles, when interacting with the fluid. The particle flow in this project is dense, meaning the concentration of particles in some regions is high. Thereby, the drag model must account for the interactions between multiple particles, when calculating the drag on the individual particles. In this project the Koch-Hill-Ladd model is used to determine the drag coefficient (Benyahia et al., 2006). The model is further described in Appendix B.

### Lift forces

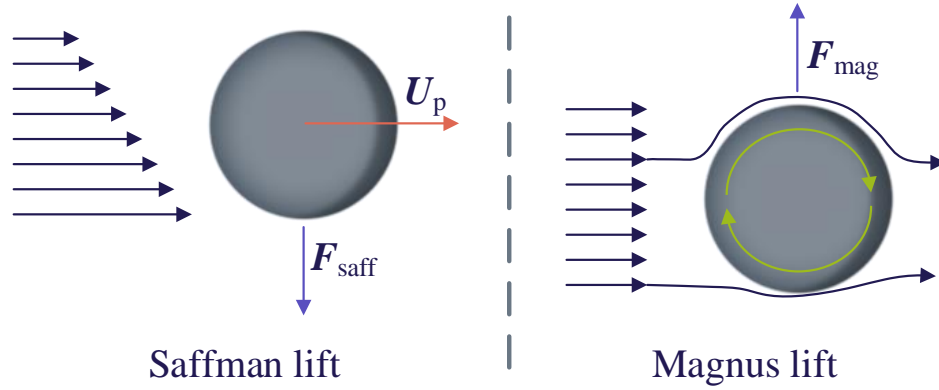
When an object or particle moves in a fluid, forces in the normal direction affects the particle. The dominating forces are the pressure force and the wall shear force due to the no-slip condition at the surface. The sum of forces working in the normal direction to the fluid is called lift force and is generally given by (Mema et al., 2019):

$$F_L = \frac{1}{2} C_L \rho_f A_p \left| \mathbf{U}_f - \mathbf{U}_p \right|^2 \quad (2.18)$$

Where

$C_L$  | Lift coefficient

There are different types of lift forces depending on e.g the flow distribution (Saffman) and rotation of the particle of interest (Magnus). The Saffman lift force describes the lift force applied to a particle by a nonuniform velocity profile. The velocity profile results in different fluid velocities on either side of the particle.



**Figure 2.11:** A illustration of the Saffman and Magnus lift forces.

The higher fluid velocity side induces a low pressure zone, which creates a pressure difference, which causes the Saffman lift. This is illustrated in Figure 2.11, and the Saffman lift force is described in Equation 2.19 (Marshall and Li, 2014).

$$F_{\text{saff}} = 6.46 \mu_f \left( \frac{D_p}{2} \right)^2 (u_f - u_p) \left( \frac{du_f/dy}{\nu_f} \right)^{1/2} \quad (2.19)$$

Where

$F_{\text{saff}}$		Saffman lift force
$\nu_f$		Kinematic viscosity of fluid

The other lift force is the Magnus lift force, which is caused by a rotating particle submerged in a fluid. The rotation results in an uneven velocity distribution around the particle. This is illustrated in Figure 2.11, where it can be seen, that the rotation of the sphere increases the fluid velocity on one side of the sphere, while decreasing the fluid velocity on the other side. The expression for Magnus lift force is seen in Equation 2.20.

$$F_{\text{mag}} = \pi \rho_f \left( \frac{D_p}{2} \right)^3 \boldsymbol{\Omega} \times \mathbf{u}_p \quad (2.20)$$

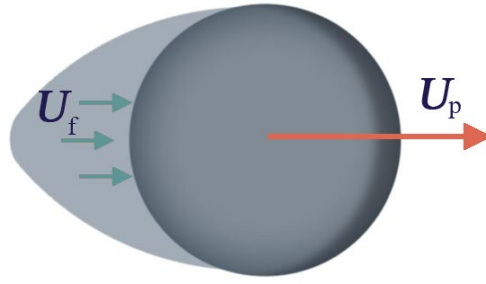
Where

$F_{\text{mag}}$	Magnus lift force
$\Omega$	Rotation rate

Both Saffman and Magnus lift are included in the Mei lift equations by Loth and Dorgan (2009), which are elaborated in Appendix C.

### Added mass

The added mass force describes the extra force required to accelerate a particle through a fluid compared to accelerating a particle through vacuum. The motion of the particle must increase the kinetic energy of the particle-adjacent fluid. The phenomena is illustrated in Figure 2.12 and the added mass force of a sphere is described by Equation 2.21.



**Figure 2.12:** A particle accelerating in a fluid induces kinetic energy to fluid adjacent to the particle.

$$F_a = -C_M \frac{\pi}{6} \rho_f D_p^3 \left( \frac{dU_p}{dt} - \frac{dU_f}{dt} \right) \quad (2.21)$$

Where

$F_a$	Added mass force
$C_M$	Added mass coefficient

The added mass coefficient is 0.5 for a sphere (Marshall and Li, 2014).

### Pressure gradient and buoyancy force

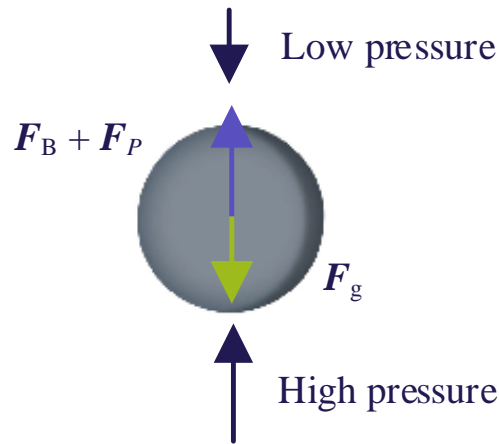
Whenever a particle is submerged into a fluid with regions of different pressure levels, the pressure gradient force must be accounted for in the calculations. Due to the variation in pressure distributions, the fluid is moving from the high pressure regions to the lower pressure regions exerting a force on the particle caused by the pressure gradient. The pressure-gradient force is given by (Marshall and Li, 2014):

$$F_P = -\frac{dU_f}{dt} \rho_f A_p \quad (2.22)$$

Where

$F_P$	Pressure gradient force
-------	-------------------------

From the expression it is seen, that the force acts opposite to the direction of the pressure gradient due to the minus sign. In Figure 2.13 an illustration of how the pressure gradient and buoyancy forces act on a particle is given.



**Figure 2.13:** Illustration of the pressure difference and buoyancy force. The gravitational force acts in the opposite direction of the buoyancy force.

The pressure also affects a body immersed in a fluid by the acting buoyancy force, due to the increase in pressure as function of the depth. A particle submerged in fluid is subject to a low pressure force acting on the particle curvature nearest the surface in the downwards direction and a high pressure force acting on the particle curvature farthest from the surface. The sum of these forces have an upwards direction. The buoyancy force is expressed by (Cengel et al., 2012):

$$F_B = \rho_f g V_p \quad (2.23)$$

The buoyancy force acting on a body equals the weight of the liquid being displaced and acts in the opposite direction of the gravity.

### Basset history force

For a particle accelerated in a large density ratio fluid, the Basset history force must be taken into consideration (Johnson, 1998). The Basset history force is caused by the response of the particle viscous shear force relative to an acceleration in the flow surrounding the particle.

This force can be explained by the Stokes problem, where a flat plate, laying along the  $x$ -axis in a fluid domain, is accelerated by a small velocity gradient  $dU$  in the  $x$ -direction at the time  $t = t + \Delta t$ . Due to no-slip condition the fluid is driven into motion. Thus, the shear stress at the surface of the plate is varying with time, as the flow is accelerated, it leads to a time-varying viscous force, which enhances the force on the particle. The additional history force exerted on a particle at low Reynolds number is expressed by Marshall and Li (2014) in Equation 2.24 and the kernel function  $K(t - (t + \Delta t))$  described in Equation 2.25

$$F_H = \mu_f D_p \int_{-\infty}^t K(t - (t + \Delta t)) \left( \frac{d\mathbf{U}_p}{d(t + \Delta t)} - \frac{d\mathbf{U}_f}{d(t + \Delta t)} \right) d(t + \Delta t) \quad (2.24)$$

Where

$F_H$  | Basset history force

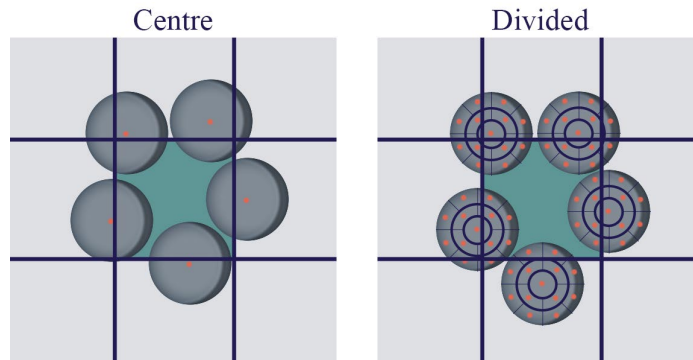
$$K(t - (t + \Delta t)) = \frac{3}{2} \left[ \frac{\pi \rho_f D_p^2}{\mu (t - (t + \Delta t))} \right]^{1/2} \quad (2.25)$$

Thereby the Basset history force accounts for what happens in the boundary layer surrounding a particle due to diffusion of vorticity (Olivieri et al., 2014). Several of these forces are dependent on the fluid velocity, which is affected by the amount of particles present in the individual cells.

The forces are dependent on the particle distribution in each cell. There are several of ways to represent the particles in the fluid cells. The methods are elaborated and elected in the next section.

### 2.5.2 Definition of void fraction

The void fraction, used in the governing equations (Equation 2.1 and 2.2 on page 21), defines the space particles occupy in a cell, and depends on particle positions, which can be determined in two-ways. If the position is defined by the particle centre of mass only, some cells where a large volume fraction of particles is present, but only a small or even no particles have their centre located in the cell, will be viewed by the model as consisting entirely of liquid. One way to avoid this is to divide the spheres into smaller sub-domains with a local centre of mass. This way, the resolution of the void fraction is improved. This is illustrated in Figure 2.14, where the highlighted cell on the left has a fluid void fraction ( $\alpha_f$ ) equal to one, while the highlighted cell on the right side has  $\alpha_f < 1$ .



**Figure 2.14: Left:** All particle centres (red) are outside the highlighted cell. **Right:** Sub-domains are used to determine the fraction of a sphere present in a given cell.

Another coupling parameter which must be kept in mind is the coupling interval, which determines when the CFD calculations are initiated. The coupling time step is presented in the next section.

### 2.5.3 Coupling time steps

The calculation of a CFD-DEM coupled model can be very time consuming and therefore caution to the CFD and DEM time steps must be taken. One crucial point of the CFD-DEM model is, that the time resolution of the two models are not equal; the DEM model requires significantly smaller time steps, determined by the DEM time steps described in Section 2.3, in order to properly resolve the particle collisions. While the required time step for CFD simulations, according to the Courant number in Section 2.2, typically is larger. The time steps are connected by the coupling interval as seen in Equation 2.26:

$$\Xi = \frac{\Delta t_{\text{CFD}}}{\Delta t_{\text{DEM}}} \quad (2.26)$$

Where

$\Xi$  | Coupling interval

Thus, a coupling interval of ten means, that ten DEM iterations are calculated for each CFD iteration.

## 2.6 Chapter Summary

This chapter has described the methodology and theory involved in CFD, DEM and the coupling of the two. Based on the presented theory in this chapter a numerical model is constructed, which is described in Chapter 3 along with sensitivity analyses of the entrance length of the geometry, grid independence study, time step size and the stiffness of the particles.





# Chapter 3

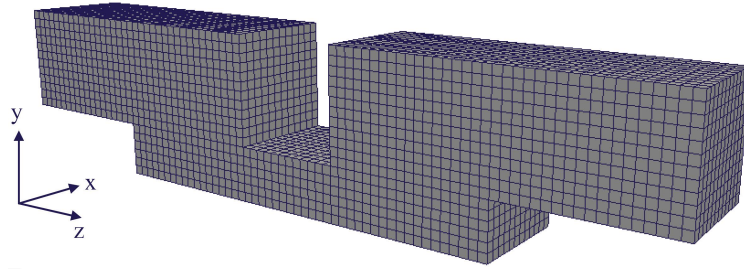
## Numerical Model Setup

A numerical model must be made in order to investigate the particle behaviour as a result of the fluid forces applied. The procedure of deriving a numerical method for this investigation is described in detail throughout this chapter, which includes all parts of the numerical model, such as meshing, CFD and DEM setup, boundary conditions, included forces and coupling procedure. The chapter also includes model sensitivity analyses of the entrance length, grid independence, time step size and stiffness of the particles.

### 3.1 Geometry and Meshing

This section introduces the geometry and meshing of the numerical model. The dimensions of the geometry are presented in Table 1.2 on page 10. The geometry consists of a squared duct with a cavity in the middle. The flow is forced through the cavity by a block. The cavity and block have the same width as the duct.

The meshing of the geometry is conducted using hexahedral cells, as seen in Figure 3.1, and the cells are equally sized.



**Figure 3.1:** View of the mesh topology.

Since the mesh consists entirely of hexahedra, the mesh is orthogonal. A further explanation of the meshing approach can be found in Appendix D. A grid independence study is conducted in Section 3.5.2.

## 3.2 CFD Setup

This section presents the CFD model setup. First the schemes used to solve the governing equations are presented. Afterwards the initial and boundary conditions are described.

### 3.2.1 Schemes

In order to solve the governing equations for the CFD model, the equations are divided into different terms. The terms are presented in Table 3.1. To avoid unrealistic values in the calculation, bounded schemes are preferred.

**Table 3.1:** The governing equations divided into the different terms.

Term	Description
Time	Solves the time terms $\partial/\partial t, \partial^2/\partial t^2$
Interpolation	Linear interpolation
Surface normal gradient	The normal component of the gradient normal to cell face
Gradient	Solves the $\nabla$
Divergence	Solves the $\nabla \bullet$
Laplacian	Solves the $\nabla^2$

The schemes used to solve the CFD model are displayed in Table 3.2.

**Table 3.2:** Schemes used for solving the different terms in the governing equations.

Term	Scheme	Characteristics
Time	Euler	$\mathcal{O}(h)$ , bounded, implicit
Interpolation	Linear	$\mathcal{O}(h^2)$ , lin. interpolation (central diff.)
Surface normal grad.	Uncorrected	No non-orthogonal correction
Gradient	Cell limited Gauss linear	$\mathcal{O}(h^2)$ , Gaussian integration, lim. lin. interpolation
Divergence	Gauss upwind	$\mathcal{O}(h)$ , bounded
Laplacian	Gauss Linear uncorrected	$\mathcal{O}(h)$ , bounded, non-conservative

The specific terms in Table 3.2 are briefly explained through the next sections.

### Time

The time is solved using an implicit backward Euler method, which is first order accurate and bounded. The method is bounded, since the coefficients for the cells have opposite signs at current and previous time steps. This ensures that the scheme is bounded and stable (Moukalled et al., 2016).

### Interpolation

The linear interpolation scheme uses central differencing for interpolation between the cells. Due to the truncation of higher order terms in the Taylor series, this differencing approach is of second order.

### Surface normal gradient

At the cell faces the surface normal gradient is calculated. The surface normal gradient is the normal component of the gradient of the two cells that enclose the face. The scheme is uncorrected, since the mesh consists only of hexahedrals. Thus, the mesh is orthogonal, why non-orthogonality correction is applied.

### Gradient

The gradient is solved by Gaussian integration and linear interpolation between the cell centres and cell faces. The scheme includes a limiter, which ensures that the value of a cell does not exceed values of the neighbouring cells. The limiter requires a specified value between 0 and 1, where 1 ensures boundedness. The method is second order accurate.

**Divergence**

The divergence term is solved by the Gaussian scheme using upwind interpolation, which ensures boundedness, since the coefficients are positive and the coefficient matrix is diagonal dominant. As the upwind interpolation is utilising the backward differencing scheme, it is first-order accurate (Versteeg and Malalasekera, 2007).

**Laplacian**

The Gaussian scheme solves the laplacian term using linear integration. When solving the surface normal gradients for an orthogonal grid the uncorrected scheme can be used. The scheme is first order accurate and bounded.

**3.2.2 Boundary and initial conditions**

In order to solve the Navier-Stokes equations some boundary and initial conditions must be specified, which are presented in this section.

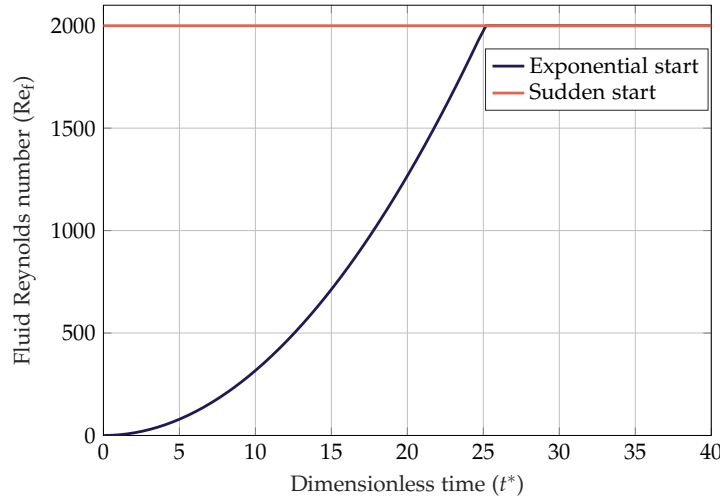
At inlet of the geometry a fully developed laminar flow profile is applied. The flow profile for a square duct is determined analytically using Navier-Stokes equations assuming the fluid is Newtonian with constant properties. Furthermore, it assumes that the pressure gradient in the longitudinal direction is constant. The calculation of the analytically fully developed laminar flow profile is elaborated in Appendix E.

The no-slip condition is applied to all walls in the geometry resulting in a velocity of zero at the face of the walls. Atmospheric pressure is set as an outlet boundary condition. The boundary conditions are listed in Table 3.3.

**Table 3.3:** Boundary conditions used in CFD simulations. See Appendix E for details regarding the inlet boundary condition.

Parameter	Inlet	Outlet	Walls
$U_f$	Fully developed laminar flow profile	zeroGradient	No-slip
$P$	zeroGradient	Atmospheric pressure	zeroGradient
$\alpha_f$	zeroGradient	zeroGradient	zeroGradient
$K_{fp}$	zeroGradient	zeroGradient	zeroGradient
$\rho_f$	zeroGradient	zeroGradient	zeroGradient
$U_p$	zeroGradient	zeroGradient	zeroGradient

At initial condition the fluid velocity in the channel is set to  $U_f = 0$ . Since some time is required for a flow profile to fully develop as a pump is turned on, a slow start up must also be implemented in the numerical model. The inlet boundary condition velocity in the numerical model builds up as an exponential function, which is illustrated in Figure 3.2. The figure also illustrates a sudden start. For a sudden-start initial condition the flow is initiated at the final velocity with a fully developed laminar flow profile. The exponential velocity profile is fully developed at each time step, however the velocity develops exponentially until the desired velocity is reached.

**Figure 3.2:** For the exponential start the fluid reaches the final velocity within a dimensionless time ( $t^* = t U_{f,avg} / D_h$ ) of 25.

The development of the flow profile is experimentally investigated in Section 4.3 on page 68 to find the time, it takes to reach a fully developed laminar flow profile. In Section 5.1 on page 76 the sudden and exponential start up is compared to the experiment to determine the difference between the two start up methods.

The next sections give an overview of material properties, boundary and initial conditions for the DEM approach.

### 3.3 DEM Setup

This section presents the particle and wall properties used for the numerical model. The material properties are divided into particle and wall properties. Finally, the method for inserting and deleting particles is presented.

#### 3.3.1 Particle and wall properties

The particles used in the experiments (see Chapter 4) are made of soda lime glass, and the walls in the geometry is of acrylic. These properties are also used in the numerical model and the properties are displayed in Table 3.4.

**Table 3.4:** The particle and wall properties used for numerical model sensitivity studies.

Parameter	Property	Value	Source
Density ratio	$\chi$	2.5	Supplement S1
Effective Young's modulus (particle)	$E_{\text{eff,pp}}$	$3.3 \cdot 10^{10}$	Supplement S1
Effective Young's modulus (wall)	$E_{\text{eff,pw}}$	$2.9 \cdot 10^9$	Supplement S1
Coefficient of restitution	$e_p$	0.69	Joseph et al. (2001)
Poisson ratio (particle)	$\nu_p$	0.23	He et al. (2018)
Poisson ratio (wall)	$\nu_w$	0.37	He et al. (2018)

#### 3.3.2 Initial and boundary conditions

The particles are inserted before the CFD model is applied, so the cavity is full of particles, which are at rest, at  $t = 0$  for the CFD-DEM simulation.

The particles are inserted randomly with dense packing without any overlapping of particles. The procedure for filling the domain is described in Appendix F.

The particles are bounded by the walls of the channel. Thereby, the particles can only exit the domain at the channel outlet. The particles are deleted, when exiting the channel. In order to keep track of the particles, the total number of particles are tracked at each time step.

As both the CFD and DEM setup has been presented the models for coupling of the two approaches are presented in the following section.

### 3.4 Coupling Models

This section explains the included coupling models, where the coupling force models are based on the scaling analysis seen in Table 1.4 on page 13. The coupling models are also displayed in Appendix G.

#### **Void fraction model**

The void fraction model, described in Section 2.5.2 on page 40, determines how the particles are represented in the CFD domain. For this project the divided model, which divides the particles into 29 equally sized volumes, is used. The values are assigned to the volume centres, thereby the estimation of the void fraction becomes more accurate.

#### **Smoothing model**

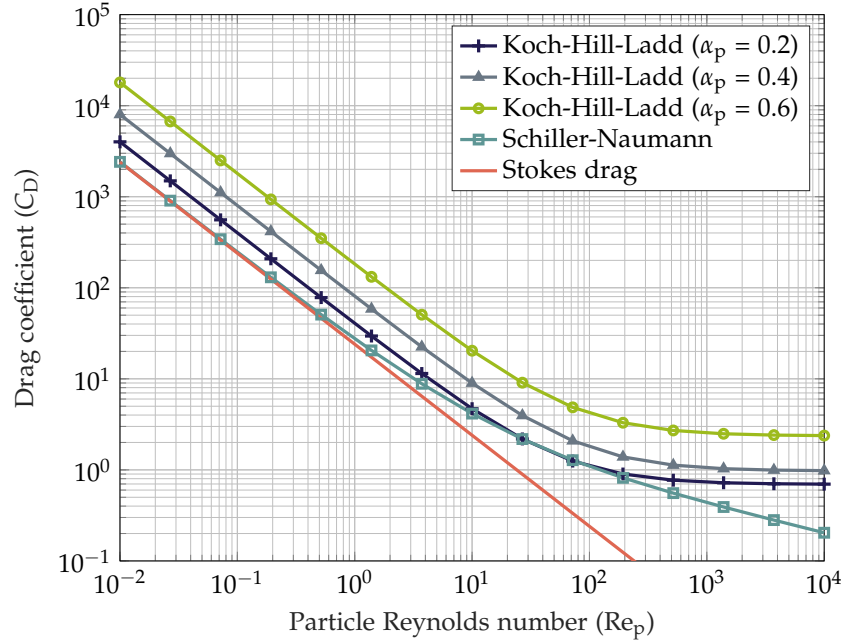
The smoothing model considers the forces within a defined distance from the given cell and helps to avoid single cells experiencing a high Courant number. This is done by means of a diffusion equation, which smears out the original particle fields. By implementing smoothing,  $\Delta t_{\text{CFD}}$  can be increased. The smoothing is applied to the fluid-particle momentum exchange.

#### **Koch-Hill-Ladd drag**

The drag is defined using the Koch-Hill-Ladd drag model, described in Section 2.5.1, which applies to the fluid Reynolds range used in this report. Figure 3.3 shows a comparison between the Koch-Hill-Ladd and the

Schiller-Naumann drag models. The Schiller-Naumann is a modification of the Stokes drag, and can be used in a wider particle Reynolds ( $Re_p$ ) range, but it does not include the void fraction, whereas the Koch-Hill-Ladd drag model takes the void fraction into account (Schiller and Naumann, 1933).

From the figure, it can be seen that the drag coefficient ( $C_D$ ) found by the Koch-Hill-Ladd drag model follows the tendency of the Stokes drag until approximately  $Re_p = 1$  after which it goes towards a constant value, as the turbulence region is reached. The tendency of the Koch-Hill-Ladd drag model applies regardless of void fraction, however changes in void fraction scales the drag coefficient. The Schiller-Naumann drag coefficient is closer to the Stokes value, however as turbulent region is reached, the  $C_D$  does not become constant.



**Figure 3.3:** The drag coefficient as function of the particle Reynolds number for the Koch-Hill Ladd, Schiller-Naumann and Stokes drag model. The Koch-Hill-Ladd drag is plotted for three different particle void fractions. The particle Reynolds number is defined as  $Re_p = \rho_f D_p U_p / \mu_f$ .

In this project the Koch-Hill-Ladd drag model is used, as it also takes the void fraction into account and becomes more constant as the particle Reynolds number approaches 2000.



**Pressure gradient force**

The model calculates the pressure gradient forces on the particles. The model is included based on the scaling analysis.

**Viscous force**

The viscous force applied on the particles is calculated using the viscous force model. Since the Reynolds number is low, the viscous forces play a role in the fluid dynamics.

**Mei lift**

The Mei lift force couples the Saffman and Magnus lift forces and calculates the lift force exerted on each particle.

### 3.5 Model Sensitivity Study

To ensure accuracy of the numerical simulations, several quality checks are performed, which includes the following parameters for both the CFD and DEM part:

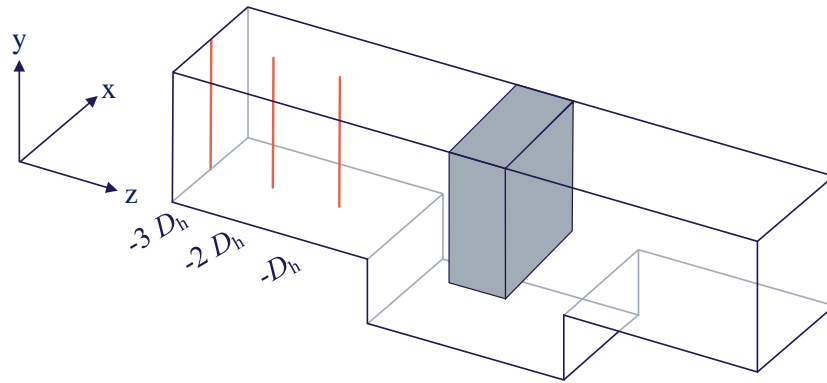
- Entrance length (CFD)
- Grid independence (CFD)
- Time step size
  - Courant number (CFD)
  - Time step size (DEM)
  - Stiffness (DEM)

In this section these five parameters are checked in order to ensure that the entrance length does not affect the flow profile and to find the time step, grid size and stiffness of the particles, which can lower the computational time, while still giving accurate results.

### 3.5.1 Entrance length

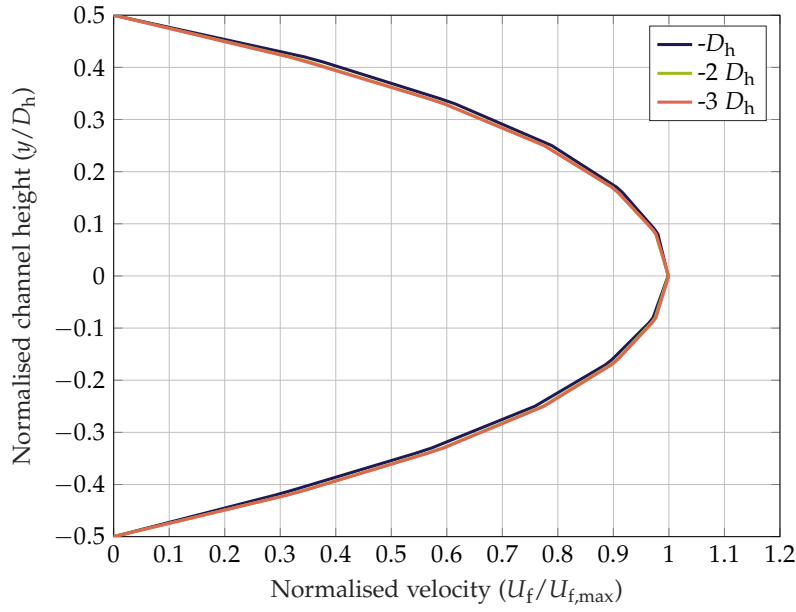
Before checking the grid sensitivity, it is necessary to evaluate the entrance length in order to ensure that the applied laminar flow profile is fully developed. The flow profile is measured at three different lines between inlet and the cavity. Here the inlet has been elongated compared to the geometry presented in Chapter 1 to have a better frame of reference.

The velocity is measured at  $x/D_h = 0$  and covers the entire  $y$ -direction as illustrated by the three lines shown in Figure 3.4.



**Figure 3.4:** The velocity is plotted at three different positions, which are marked with red vertical lines.

As seen in Figure 3.5, the flow profile is constant, as it approaches the cavity. The velocity profile is not affected by the placement of the inlet, when the entrance length ( $L_0/D_h$ ) is greater than or equal to one. The inlet entrance length remains constant for all simulations with a length of  $-1.5 D_h$ , which ensures a small safety margin. As the entrance length has been checked, the grid sensitivity is evaluated in the next section.



**Figure 3.5:** Normalised channel height as function of the normalised velocity at three different positions.

### 3.5.2 Grid independence

It is preferable to run the simulations with a minimum number of cells in order to reduce computational cost, but still achieve reliable results, so a grid sensitivity analysis is conducted to determine the required number of cells in order to ensure grid independence.

The sensitivity to cell size is investigated using the same general case with  $Re_f = 500$ , which is given by  $Re_f = \rho_f U_{f,avg} D_h / \mu_f$ .

A series of CFD simulations are run using different cell sizes. The specific cell size ( $n_{cell}/D_h$ ), the particle diameter to cell size ratio ( $D_p/dx$ ) and the total number of cells ( $n_{cell}$ ) for all simulations are listed in Table 3.5. The different grids are referred to by their specific name, which is denoted in the table.

**Table 3.5:** Grid notation, cells per  $D_h$ , ratio between particle diameter and cell size ( $D_p/dx$ ) and the number of cells for the three different grids.

Grid notation	$n_{\text{cell}}/D_h$	$D_p/dx$	Number of cells
Coarse	8	0.6	5888
Medium	10	0.75	11,500
Fine	12	0.9	19,872

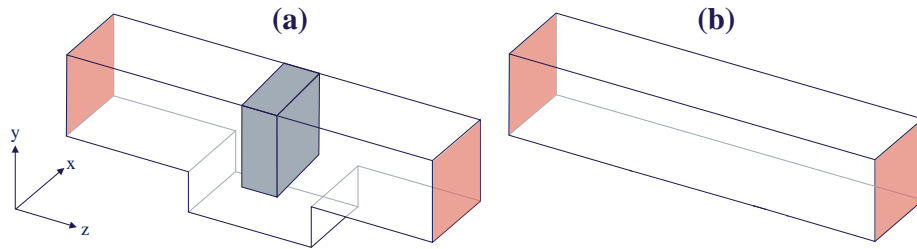
The friction factor is used as the evaluation parameter and is calculated by the general equation for the pressure drop for internal flows, which is defined as (Cengel et al., 2012):

$$\Delta P_L = f \frac{z_0}{D_h} \frac{\rho_f U_{f,\text{avg}}^2}{2} \quad (3.1)$$

Where

$\Delta P_L$	Pressure drop
$f$	Friction factor

For the analysis the friction factor is normalised by the friction factor calculated for a simple squared duct as seen in Figure 3.6b. The pressure is measured at the planes shown in Figure 3.6a-b marked by red, which is located at in- and outlet.

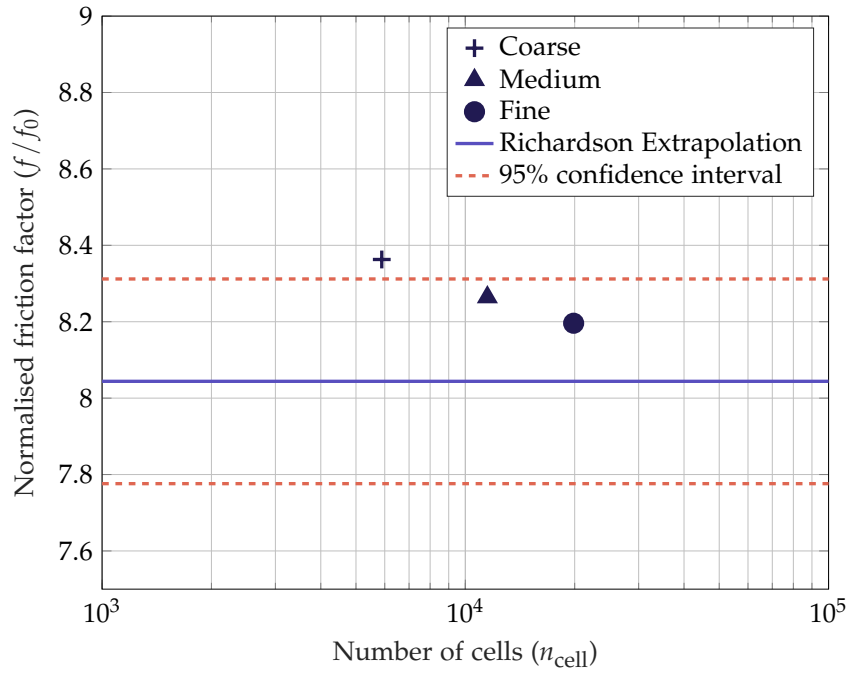


**Figure 3.6:** **a)** Measurement plane for the geometry. **b)** Measurement plane for reference case.

The results of simulations are compared using the Grid Independence Index (GCI), based on the work of Roache (1994), which returns a 95%

confidence interval around the asymptotic solution derived by Richardson interpolation of, in this case, the normalised friction factor. The method and equations are further explained in Appendix H.

In Figure 3.7 the normalised friction factor is plotted as function of the number of cells on a semi-logarithmic scale including the exact solution extrapolated using the Richardson Extrapolation and the 95% confidence interval.



**Figure 3.7:** The normalised friction factor as function of the number of cells for the three simulations for a safety factor of  $F_s = 1.25$  (Roache, 1994).

The figure shows, how refining the grid affect the calculated friction factor. Furthermore, it is seen, that the friction factors are approximately eight times larger than for a plane squared duct.

When the grid is refined, the normalised friction factor approaches the exact solution based on the Richardson extrapolation. As seen in the figure, only the coarse grid is outside the 95% confidence interval.

For calculating the order of convergence ( $p$ ), one refinement ratio is found by averaging the refinement ratios  $r_{12}$  and  $r_{23}$ . Here the subscript 1 refers to the coarse grid and 3 to the fine grid. Since the GCI is calculated

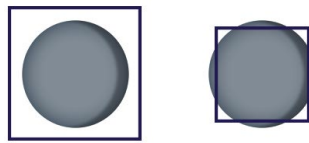
for three different grids, the safety factor ( $F_s$ ) is chosen to be 1.25, which is common practice and based on experience (Roache, 1994).

The asymptotic range is calculated according to Equation H.7 on page 139. This value is used to evaluate, if the obtained results are within the asymptotic range, and thereby can be considered reliable. As seen in the table, this value is close to unity, which indicates that the results are within the asymptotic range, only deviating with -0.8%.

**Table 3.6:** The results obtained from the GCI analysis according to the equations presented in Appendix H.

Notation	Order of converg. ( $p$ )	Asymp. solution $f_{h=0}/f_0$	GCI <sub>12</sub> [%]	GCI <sub>23</sub> [%]	Asymp. range	DEV [%]
$f/f_0$	0.61	8.04	3.3	2.3	0.99	-0.8

In Table 3.6 the GCI is found to be low for both cases. By refining the grid from coarse to medium, the error decreases by 3.3%. A further refinement of the grid does also have impact on the final obtained results, as the GCI decreases by 2.3%. For the fine grid the particle diameter to cell size ratio ( $D_p/dx$ ) ratio is just below unity (0.9). If the grid is additionally refined the particle diameter exceeds the cell length, as seen in Figure 3.8. As grid independence is achieved before exceeding the particle diameter to cell size ratio, finer grids have not been evaluated.



**Figure 3.8:** Illustration of the particle diameter in relation to the cell size of the grid. **Left:**  $D_p/dx < 1$ . **Right:**  $D_p/dx > 1$ .

The grid is considered independent with the medium grid, however as the computational cost difference between a medium and fine grid is within reasonable limits, the studies in this report are conducted for the fine grid

with a cell ratio of  $n_{\text{cell}}/D_h = 12$  having a  $D_p/dx$  ratio of 0.9 with 19,872 cells.

Since the grid size is known, it is possible to calculate the CFD time step. In the next section both the critical CFD and critical DEM time step is analysed along with the stiffness sensitivity.

### 3.5.3 Time step and stiffness sensitivity

The time steps and particle stiffness used in the simulation must be a compromise between simulation cost, stability and particle behaviour. In this section the sensitivity of the time step size for both the CFD and DEM are evaluated along with the effects of changing the stiffness of the particles.

First the CFD time step size is calculated followed by an analysis of the DEM time step with the actual effective Young's modulus for the glass particles. Finally, the stiffness sensitivity of the particles is analysed.

#### CFD time step

The lowest theoretical CFD time step is calculated from the Courant number limitation. If the simulation must stay stable the Courant number must stay below 1. The highest theoretical time step is calculated for  $Re_f = 2000$  with the fine grid using the Courant number equation (Equation 2.4) leading to a dimensionless CFD time step of:

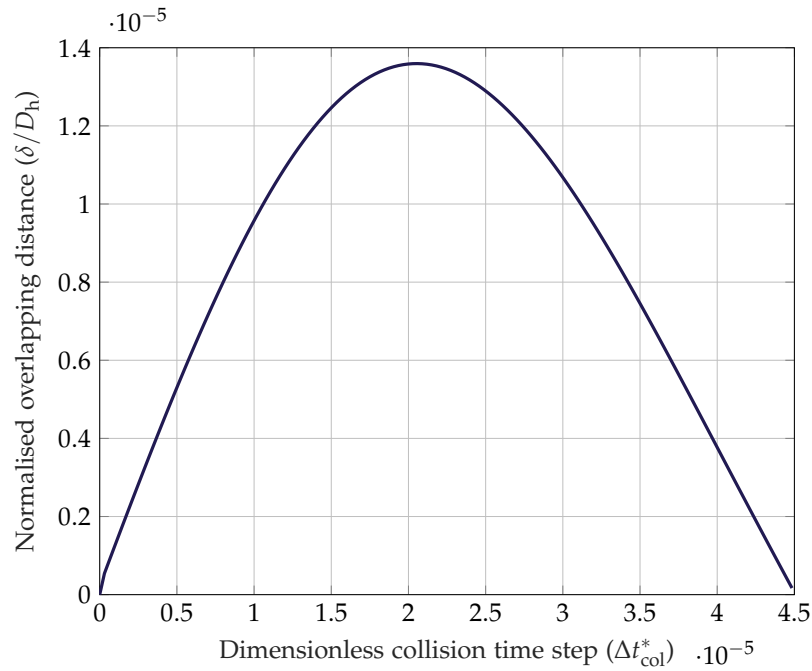
$$\Delta t_{\text{CFD,crit}}^* = 0.085 \quad (3.2)$$

Here the time step is normalised according to Equation 2.12 on page 30, where it is defined as  $t^* = t U_p / D_p$ . The DEM time steps in the next sections is also normalised by Equation 2.12. The CFD time step is maintained below the dimensionless time of 0.085 in order to ensure stability and reliability of the obtained CFD results.

#### DEM time step

The largest theoretical DEM time step is determined by three methods; collision time, Rayleigh time and Hertz criteria. All time steps in this section are evaluated for an effective Young's modulus of  $3.3 \cdot 10^{10}$ .

The collision time is evaluated by investigating the collision of two particles, modelled using the Hertz contact model. The equations are presented in Appendix A. With this model it is possible to determine the duration of collision between two colliding particles. For the contact force calculation the two particles have a diameter of  $0.07 D_p/D_h$ . For the investigation the particles are assumed to have the same velocity as the fluid for a fluid Reynolds number of 1300, which is an extreme case, thus a safety margin is included in the obtained DEM time step. The overlapping distance ( $\delta$ ) is plotted as function of the collision time ( $\Delta t_{\text{col}}^*$ ) in Figure 3.9.



**Figure 3.9:** The normalised overlapping distance as a function of the dimensionless time.

The particles interact when  $\delta/D_h > 0$ , and it is required that  $\Delta t_{\text{DEM}}^* < \Delta t_{\text{col}}^*$  in order to simulate the collision. The collision ends at  $\Delta t_{\text{col}}^* = 4.5 \cdot 10^{-5}$ , so the actual DEM time step must be below this value.

The other method for determining the theoretical DEM time step is the Rayleigh time step, which is calculated using Equation 2.13 on page 31 and for this case the Rayleigh time is calculated to be  $\Delta t_{\text{ray}}^* = 8.1 \cdot 10^{-7}$ , which is the time, it takes for the Rayleigh wave to propagate along the particle



surface.

The Hertz criterion, calculated by Equation 2.14 on page 32, is for a collision of two identical particles using the effective mass, radius and Young's modulus. For two glass particles the Hertz time is  $\Delta t_{\text{hertz}}^* = 8.5 \cdot 10^{-6}$ . All theoretical critical time steps are collected in Table 3.7.

**Table 3.7:** Theoretical dimensionless time steps for the DEM model for  $E_{\text{eff,pp}} = 3.3 \cdot 10^{10}$ .

Method	Notation	Critical time step
Collision time	$\Delta t_{\text{col}}^*$	$4.5 \cdot 10^{-5}$
Rayleigh time	$\Delta t_{\text{ray}}^*$	$8.1 \cdot 10^{-7}$
Hertz criteria	$\Delta t_{\text{hertz}}^*$	$8.5 \cdot 10^{-6}$

From the table is seen, that the Rayleigh time step is the smallest, so the model must fulfil  $\Delta t_{\text{DEM}}^* < 8.1 \cdot 10^{-7}$  to fully resolve the collision between two soda lime glass particles.

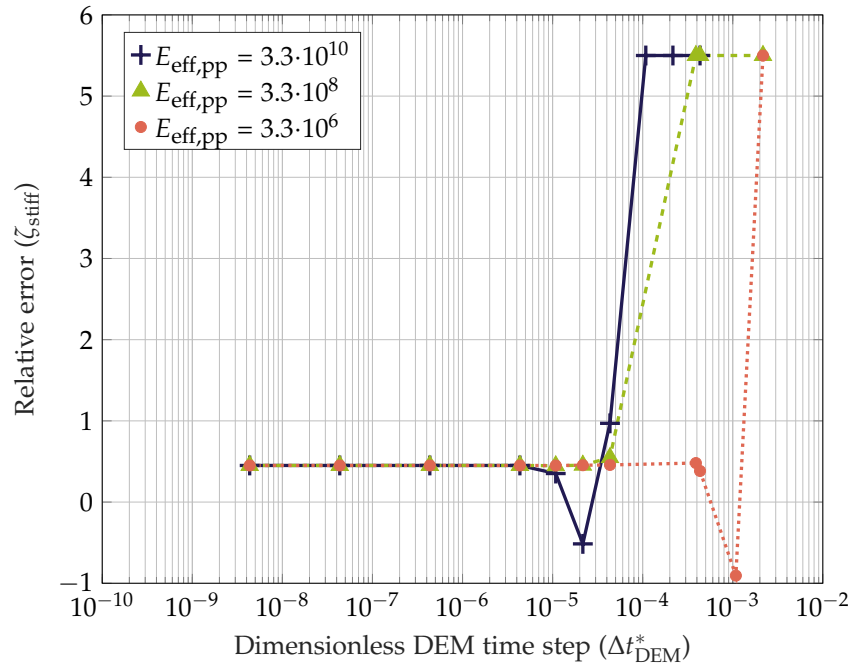
The stiffness sensitivity of the particles are investigated in the next section to increase the time step (and reduce computational cost) and for checking that the resulting velocity is not affected by lowering the effective Young's modulus.

### Stiffness sensitivity

By reducing the effective particle-particle Young's modulus ( $E_{\text{eff,pp}}$ ), the time step can be increased, since the DEM model simulates the particles as more elastic, which increases collision time step ( $\Delta t_{\text{col}}^*$ ), and thereby allows for an increase in  $\Delta t_{\text{DEM}}^*$  (Tsuji et al., 1993). To verify that this procedure can be implemented, a simple investigation is performed, where  $\Delta t_{\text{col}}^*$  for two particles is simulated. The particles collide with a velocity of  $\mathbf{U}_i = -\mathbf{U}_j$ , and the resulting velocities after collision is logged. Based on the initial and resulting velocity, the relative stiffness error ( $\zeta_{\text{stiff}}$ ) is calculated according to Equation 3.3:

$$\zeta_{\text{stiff}} = \frac{U_{0,i} - U_{\text{res},i}}{U_{\text{res},i}} \quad (3.3)$$

In Figure 3.10,  $\zeta_{\text{stiff}}$ , which is the relative error, is plotted as function of the dimensionless DEM time step ( $\Delta t_{\text{DEM}}^*$ ) for three different effective Young's moduli. The graph shows, how  $\Delta t_{\text{DEM}}^*$  can be significantly increased, when  $E_{\text{eff,pp}}$  is lowered.  $\zeta_{\text{stiff}}$  is constant for the three simulations up to  $\Delta t_{\text{DEM}}^* = 4.3 \cdot 10^{-6}$ . A further increase of  $\Delta t_{\text{DEM}}^*$  for the highest effective particle-particle Young's modulus results in a deviation between the initial and resulting velocity, which indicates that the collision is not properly resolved. The same is valid for the two other simulations, but  $\Delta t_{\text{DEM}}^*$  is allowed to be increased further.



**Figure 3.10:** The relative error as a function of the dimensionless DEM time step for three different Young's moduli.

The Rayleigh and Hertz time steps are also calculated for the two lower effective particle Young's moduli as used in Figure 3.10, which is displayed in Table 3.8.

**Table 3.8:** Hertz and Rayleigh time steps for two lower effective particle Young's moduli.

	Hertz		Rayleigh		Collision	
$E_{\text{eff,pp}}$	$3.3 \cdot 10^6$	$3.3 \cdot 10^8$	$3.3 \cdot 10^6$	$3.3 \cdot 10^8$	$3.3 \cdot 10^6$	$3.3 \cdot 10^8$
$\Delta t_{\text{DEM}}^*$	$3.4 \cdot 10^{-4}$	$5.4 \cdot 10^{-5}$	$8.1 \cdot 10^{-5}$	$8.1 \cdot 10^{-6}$	$2.9 \cdot 10^{-5}$	$4.7 \cdot 10^{-6}$

Here the collision time is the smallest time step, why the finally time step must be below this value.

### Final time steps

To ensure a stable simulation that can dissolve the dynamics properly, the time steps are kept below the limit found in Table 3.8. The CFD time step ( $\Delta t_{\text{CFD}}^*$ ) is set to  $5 \cdot 10^{-3}$ . Based on the analysis of the dimensionless DEM time step ( $\Delta t_{\text{DEM}}^*$ ), the time step is chosen to be  $\Delta t_{\text{DEM}}^* = 4.3 \cdot 10^{-6}$ , which is slightly below the one found in Table 3.8, with an effective particle-particle Young's modulus of  $3.3 \cdot 10^8$ , since the effective particle-particle Young's modulus is commonly decreased by a factor 100 for problems, where the fluid forces are more dominant than the collision to reduce computational cost forces (Tsuji et al., 1993). Furthermore, the analysis has shown that the relative error is constant meaning the resulting velocity is similar to the one found for smaller time steps. The DEM time step could be increased just slightly, but a small safety margin is added to ensure stability.

Due to the chosen time steps the resulting coupling interval is set to 1000 meaning a 1000 DEM time step is calculated for each CFD time step. In Table 3.9 the final time steps are given.

**Table 3.9:** The final dimensionless time steps and the coupling interval for the numerical simulations.

$\Delta t_{\text{CFD}}^*$	$\Delta t_{\text{DEM}}^*$	Coupling interval
$5 \cdot 10^{-3}$	$4.3 \cdot 10^{-6}$	1000

The coupling interval is calculated according to Equation 2.26 using the non-dimensionless time steps.

### 3.6 Chapter Summary

In this chapter the different schemes, force- and coupling models have been described, including analyses of the entrance length, grid independence and time step sensitivity. The inlet length of the channel is set to  $-1.5 D_h$ . The grid in the geometry consists of 19,872 cells with a spacing of  $n_{\text{cell}}/D_h = 12$  giving a particle to grid size ratio of  $D_p/dx = 0.9$ . The final time step for CFD, DEM and the coupling interval is summarised in Table 3.9. The time steps are based on an effective particle-particle Young's modulus on  $3.3 \cdot 10^8$ . In Chapter 4 a description of the experimental work is given. The comparison of the experimental and numerical results are found in Chapter 5.

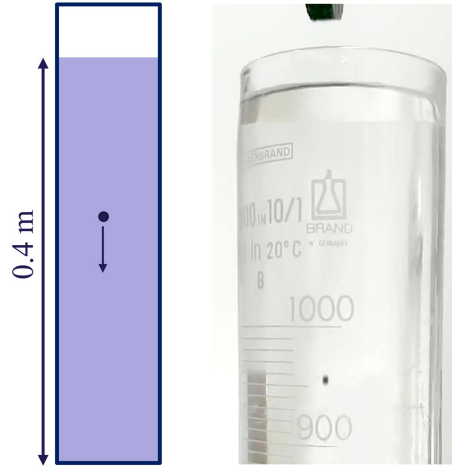
# Chapter 4

## Experimental Approach

In order to verify the numerical model, a series of experiments are conducted. This chapter explains the experimental setup and the procedure of testing. First an experiment for validating the drag force on a single particle is conducted. Hereafter the experimental setup is explained along with the material and fluid properties. Then the inlet flow profile is analysed followed by the experimental procedure for the final test to validate the numerical model described in Chapter 2.

### 4.1 Drag Experiment

In order to validate the drag force calculation of the CFD-DEM simulation an experiment is conducted, where a glass particle is released from rest in a column of water. The particle is captured by video at 30 fps during the entire drop. Figure 4.1 illustrates the experimental setup as well as an image of the particle shortly after release from a pair of pliers, which are used because the glass particle sticks to skin. The details of the test conditions are listed in Table 4.1. The fall time is derived from the number of frames from release to rest at the bottom. As the video is shot with 30 fps, the resulting fall time is rounded off. Assuming that the image, where the particle is released, and the image, where it hits the bottom, is within half a frame, the resulting time is within a precision of one frame; approximately  $\pm 1/30$  s. The resulting fall time is 67 frames (entire fall time is 66 frames plus one frame of error).



**Figure 4.1:** **Left:** The particle descends 0.4 m from just below water surface before reaching bottom. **Right:** Image of particle during descent through water.

**Table 4.1:** Drag experiment specifications

Parameter	Notation	Value
Particle-to-fluid density ratio	$\chi$	2.5
Height	$H$	0.4 m
Water temperature	$T$	20°C
Particle material		Soda-lime glass
Particle diameter	$D_p$	1.4 mm

The experiment is simulated using the CFD-DEM model, though with a geometry identical to the experiment, where the particle is inserted in a column of water at the same height with same properties. The particle position is logged 1000 times per second. Table 4.2 lists the entire fall time for the particles in the experiment and the model. Experimental fall time includes the error of one frame.

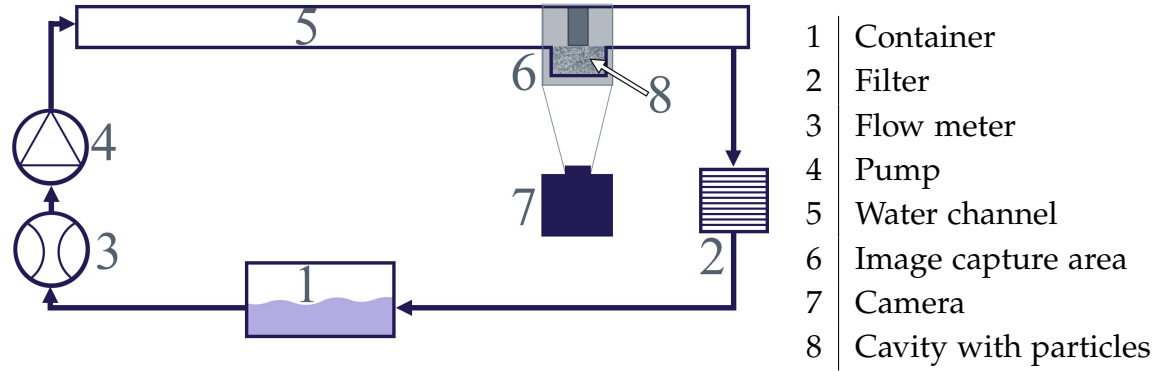
**Table 4.2:** Comparison of results from experiment and numerical model.

$t_{\text{exp}}$ [s]	$t_{\text{mod}}$ [s]	DEV [%]
2.23	2.143	-3.9

The difference in fall time is less than 4% and might be caused by the particle release from the pliers, which is done by hand, where a slight motion in the upwards direction will increase the fall time. The column height is measured on the outside of the glass using a measuring tape, which may induce a slight measurement uncertainty. As the particle is released it started rolling, which applies Magnus lift force on the particle, thereby causing an additional lift and increasing fall distance. Furthermore, the particle is not entirely spherical, this might also affect the resulting fall time, however a deviance of less than 4% indicates, that the drag force on a single particle without collision in the CFD-DEM model are indeed realistic. A more complex experiment is required in order to properly validate the numerical model, where densely particles are present. The experimental setup is presented in the next section.

## 4.2 Experimental Setup

The experimental setup is illustrated in Figure 4.2. Water is pumped from a container to a channel with a cavity in which particles are initially accumulated. A flow meter is mounted before the pump. Between the water channel outlet and the container a filter ensures, that no particles enter the flow meter and pump. The behaviour of the particles as water passes through the cavity is captured with a camera.



**Figure 4.2:** An overview of the experimental setup.

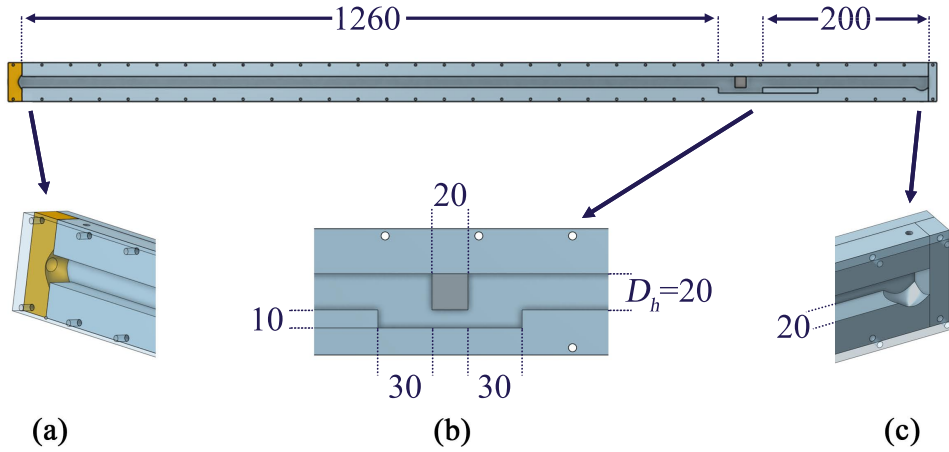
### Considerations and dimensions

The water channel is designed considering the required entrance length in order to ensure a fully developed flow, which is determined according to Cengel et al. (2012):

$$L_{\text{ent}} = 0.05 \cdot D_h \cdot \text{Re}_f \quad (4.1)$$

The entrance length is designed to ensure fully developed flow profile for  $\text{Re}_f < 1260$ , where  $\text{Re}_f = \rho_f U_{f,\text{avg}} D_h / \mu_f$ . The flow profile just before cavity is checked using Particle Image Velocimetry (PIV) during testing, as described in Section 4.3, to ensure this requirement is fulfilled. The dimensions and details of the canal are further described in Figure 4.3. The total water channel length is 1540 mm. **(a):** The flow inlet to the channel is at the same level as the channel and has been countersunk to help reduce flow disturbance and ensure easier flow development. Above inlet is a vertical ventilation hole used to remove air bubbles during filling. **(b):** The total cavity length is 80 mm, cavity height is 10 mm and the depth is 20 mm. **(c):** Outlet is countersunk and has downwards direction to allow for particles to leave the channel. Above outlet is another vertical ventilation hole.





**Figure 4.3:** Overview of water canal dimensions, and inlet and outlet details. Distances are given in mm.

The water channel comprises of two different materials; the sides are made from polycarbonate, while the top, bottom and ends are made from acrylic glass. Air vents comprise of tubes held above the water level. One of the sides can be removed, so that filling and emptying can be done.

#### Experimental setup specifications

The specifications of the different components, as well as the fluid and particle properties are specified in this section. The component specifications for the pump, flow meter and filter used for the experiment are given in Table 4.3.

**Table 4.3:** Specifications of components used in experiments.

Component	Type	Specification
Pump	DC adjustable pump	0.1 to 1.8 l/min
Flow meter	RS PRO Flow meter	0.3 to 10 l/min
Filter	Custom made	300 $\mu\text{m}$

The fluid properties are displayed in Table 4.4. In order to do Particle Image Velocimetry (PIV) measurements the fluid has been seeded with seeding particles with a diameter of 5  $\mu\text{m}$ , which are assumed to follow the streamlines and not affect the fluid properties.

**Table 4.4:** Fluid properties for the fluid used in the experiment assuming the experiments are conducted at the same conditions.

Property	Notation	Value
Temperature	$T$	21°C
Kinematic viscosity	$\nu_f$	$1.002 \cdot 10^{-6} \text{ m}^2/\text{s}$
Density	$\rho_f$	1000 kg/m <sup>3</sup>

The glass particles used for the experiment are made of soda lime glass, and the properties can be seen in Table 4.5.

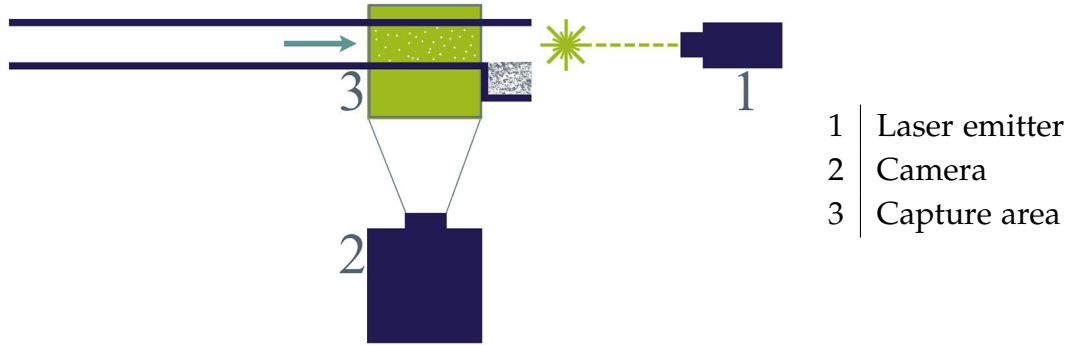
**Table 4.5:** Specifications of the soda lime glass particles used in experiments (ScienceDirect, 2019; Kremer Pigmente, 2019).

Property	Notation	Value
Particle diameter	$D_p$	1.25 to 1.55 mm
Particle density	$\rho_p$	2500 kg/m <sup>3</sup>
Young's modulus	$E_{pp}$	$63 \cdot 10^9 \text{ Pa}$
Friction coefficient	$C_{fric}$	0.9

Before conducting the final experiment for comparison to the coupled CFD-DEM model, the inlet velocity profile is investigated to ensure, that the flow profile is fully developed before reaching the cavity, which is done in the following section.

### 4.3 Validation of Flow Profile

The numerical model has a fully laminar flow profile as inlet condition. It is assumed that the flow profile is fully developed at a distance of  $1.5D_h$  upstream of the cavity. The experimental setup has been designed to ensure that this criteria is fulfilled, which is investigated using PIV. The PIV method, illustrated in Figure 4.4, uses a laser sheet to illuminate the seeding particles in the fluid.

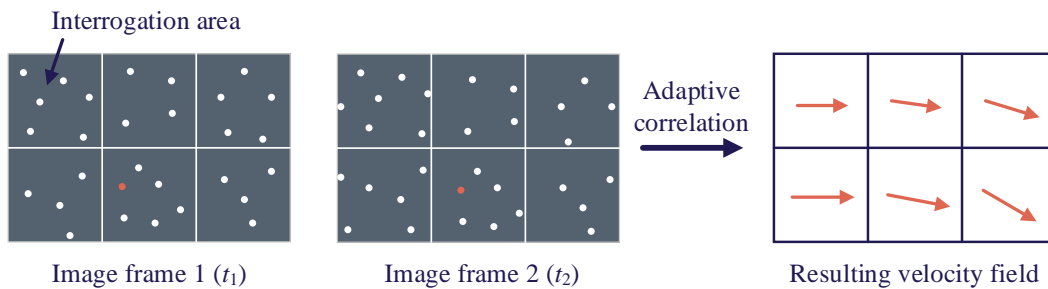


**Figure 4.4:** Overview of PIV setup used for validation of fluid flow profile before cavity.

The particle positions are captured in images with a sample rate of 150 Hz. Sample rate and particle positions are used to define a velocity field. The results are based on a series of at least 600 images, to ensure test reliability. A laser sheet illuminates the seeding particles, which appear on images as light pixels, while pixels where no seeding particle is present in the laser sheet appear dark. The displacement of seeding particles during an interframe time step is determined by adaptive correlation of the illuminated pixels, as seen in Figure 4.5. Further details of the PIV test are given in Appendix I and the PIV settings are listed in Table 4.6.

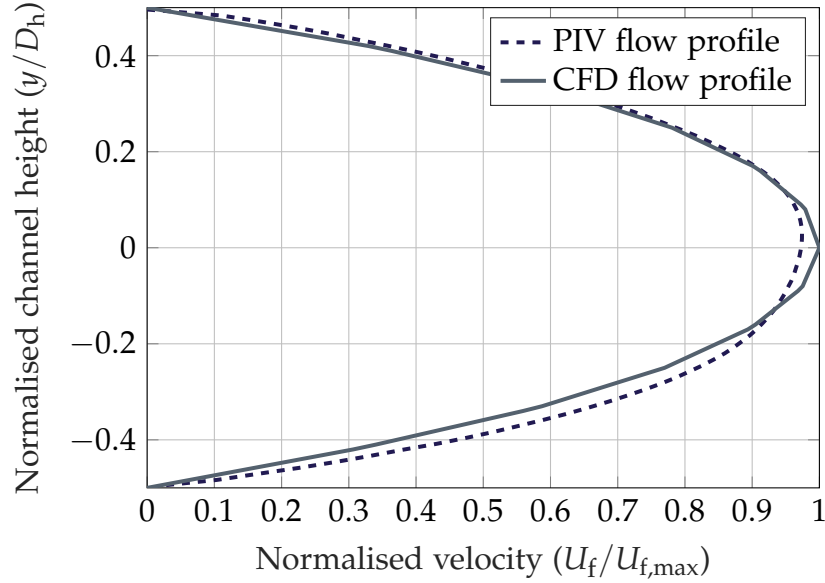
**Table 4.6:** Setting for the PIV measurements.

Setting	Value
Sample rate	150 Hz
Seeding particles	5 $\mu\text{m}$ PSP particles



**Figure 4.5:** A velocity flow field is made from adaptive correlation of two image frames and the time step between the pulses.

The PIV experiment is run with no glass particles in the cavity at  $Re_f = 1440$ , which is above the theoretical limit of the test setup as per Equation 4.1. This is chosen to determine a critical fluid Reynolds number for testing. Comparison between the PIV results and the numerical velocity profile applied at the inlet boundary condition suggest a good agreement, when looking at the velocity gradients, as seen in Figure 4.6.



**Figure 4.6:** Comparison of the inlet flow profile for PIV and CFD for  $Re_f = 1440$ .

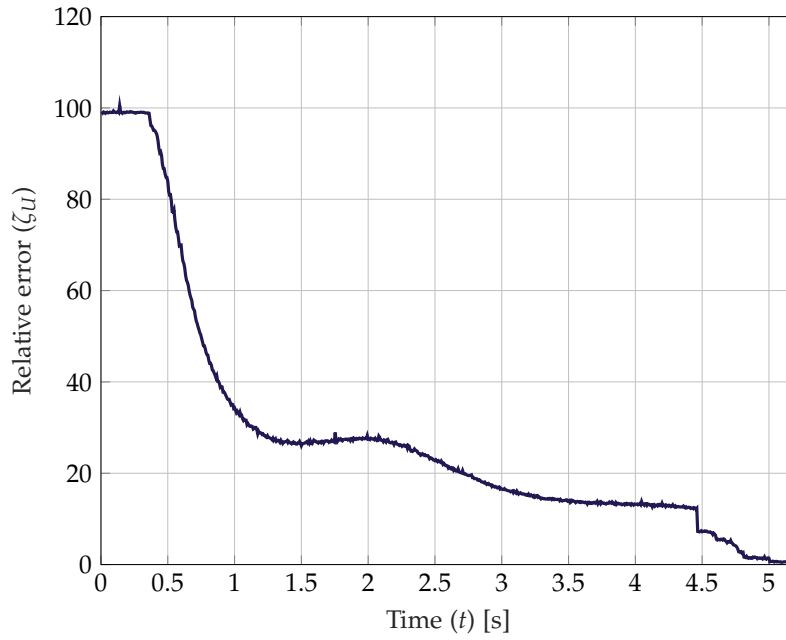
The maximum velocity of the experimental profile, which is not fully symmetric around the horizontal axis, deviates -2.6% from the theoretical. This might be caused by air bubbles near the region of PIV measurement. However as the CFD and PIV velocity profiles are similar. The comparison confirms, that the entry length of the experiment does ensure a fully developed flow profile before the cavity for  $Re_f < 1440$  and that the inlet boundary condition *during* simulations represents correct values.

As the pump is started, the flow is not instantly fully developed, which is also tested by means of PIV. To investigate this the relative error ( $\zeta_U$ ) between the current velocity profile and the final velocity profile is calculated.

The relative error is defined in Equation 4.2:

$$\zeta_U = \sum \left| \frac{\mathbf{U}_{f,avg} - \mathbf{U}_f}{\mathbf{U}_{f,avg}} \right| \quad (4.2)$$

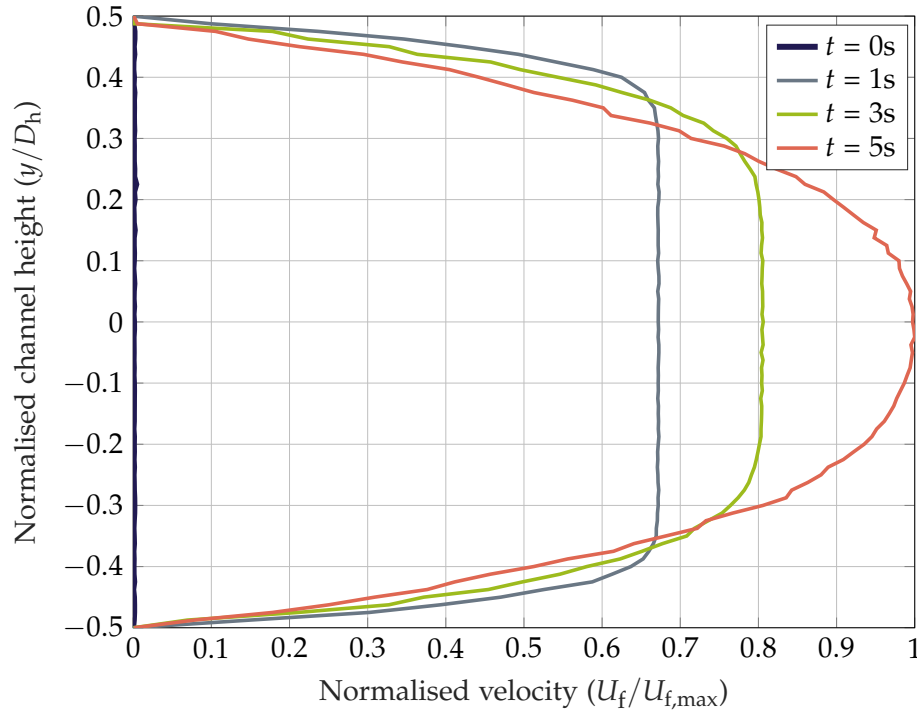
Where the  $\mathbf{U}_{f,avg}$  is the average fluid velocity after five seconds. This error describes the deviance of the sum of velocity vectors in a given time relative to the sum of average velocity vectors. Figure 4.7 shows, that the flow is developed within five seconds.



**Figure 4.7:** The error of the individual flow profiles compared to the final flow profiles plotted for every image sample.

Spikes and disturbances have been filtered from the graph shown in Figure 4.7. They occur from the computer being unable to save the pictures at the time they are taken, which is due to the large number of images (150 fps) that must be saved to the hard drive. These are temporarily stored in the RAM, and as the RAM is filled, some images will be lost, resulting in noise in the data set.

The velocity profile evolves during the first five seconds after the pump is turned on. The development is further illustrated in Figure 4.8, where the velocity profiles during the start up are plotted.



**Figure 4.8:** The evolution of velocity profile during the first five seconds after the pump is turned on. Initially, no flow is present. A fully developed laminar flow profile evolves within five seconds.

These results must be kept in mind during validation of the numerical model. Thus, an exponential start up of the velocity is applied in the CFD setup, as mentioned in Section 3.2 on page 44. The next section deals with the procedure for the final experiment used for comparison of the coupled CFD-DEM model.

## 4.4 Experiment Procedure

As the drag force on a single particle has been validated and the experimental velocity profile has been determined, it is possible to move on to the final experimental test. This test is used to compare the final positions of the glass particles in the cavity with the steady state position of the particles in the numerical model, which is done in Section 5.7. The procedure for testing is done in six steps:

1. Cavity is filled with particles and the channel side is mounted
2. The water channel is slowly filled with water, assuring particles stay at the initial place
3. Air bubbles are removed from the channel
4. The pump is turned on
5. Particles are filmed, as they are moved from cavity by the fluid
6. The test is stopped, when particles are at steady-state

Filling of the cavity is done with a specific number of particles (approximately 7000), which is found by the bulk mass, based on the mass of 100 particles. As the test runs, the volumetric flow rate is adjusted by pump supply voltage. The flow meter is used for the adjustment of the pump to the desired volumetric flow rate, and PIV is used to determine the velocity profile after steady state is reached. The experiment is conducted with a fluid Reynolds number of 1241. The procedure of the experiment involves several steps, that might result in deviances of experimental results even for experiments with the same number of particles subject to the same fluid Reynolds number. These sources of error are discussed in the next section.

## 4.5 Sources of Error

The experimental test procedure may be prone to errors, which can be caused by the following:

- Non-homogeneous particles
- Air bubbles in experimental test
- Uneven distribution of particles (initial conditions)

The soda lime glass particles used in experiments have diameters between 1.25 and 1.55 mm and might not be perfectly spherical. The difference in particle shapes and sizes may impose slightly different results for otherwise identical experiments.

As the test setup consists of a long, horizontal water channel that is slowly filled, air bubbles can easily become trapped in the channel. For larger bubbles, gently tilting the test setup can make the bubbles rise to the air ventilation hole. However smaller bubbles requires more force to move.

Even with some gentle taps on the channel, they are difficult to remove. Applying too much force or tilting too steeply results in the glass particles moving, which is undesired. The presence of these smaller bubbles might affect the flow through the cavity.

The test is reset by draining the channel, removing the side of the test setup not facing the camera, filling the cavity, while the setup is lying down. After filling, the setup is turned to the correct position and particles fall to the bottom of the cavity. After assembly and placement of the test setup the particles might not settle properly in the cavity, resulting in an uneven distribution of particles. At the wall facing the camera more particles might be present compared to the other side, depending on filling and tilting.

## 4.6 Chapter Summary

This chapter has described the experimental work conducted in this project. The drag force in the numerical model has been validated with a deviation below 4% compared to an experiment. Furthermore, the inlet velocity profile is investigated by PIV, which shows a fully developed laminar velocity profile after five seconds measured before the cavity. A deviation of 2.6% between the maximum velocity for the PIV and CFD-DEM is found. Due to the build-up time of the laminar flow profile, the numerical model must be adjusted to simulate the build-up of the velocity profile. This is done in Section 5.1.

The experimental procedure has been presented along with the sources of error introduced conducting the experiments. The numerical model is further compared against the experimental data, in terms of the final position of the particles after reaching steady state. The validation is conducted in Chapter 5.



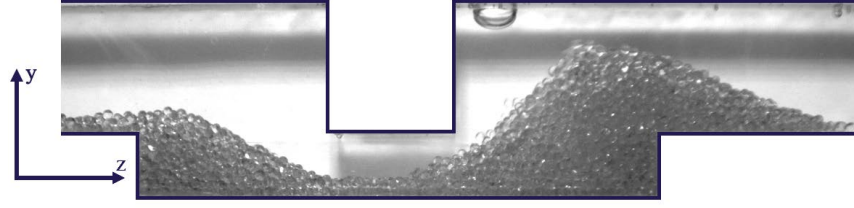
## Chapter 5

# Sensitivity Study and Model Validation

The purpose of this chapter is to investigate the sensitivity of simulation specific choices and quantify the validity of the numerical model by comparison with the experiment. Sensitivity studies are conducted using two identical simulations, with exception of the parameter in question. Results are compared between the simulations and in some cases comparison with experimental results is done. The sensitivity study includes:

- Initial conditions
- Smoothing
- Rolling friction
- Friction
- Coefficient of restitution

The numerical model used for experimental validation is developed based on considerations from the sensitivity study in Section 3.5 on page 51 and for a fluid Reynolds number of  $Re_f = \rho_f U_f D_h / \mu_f = 1241$ , which is identical to the fluid Reynolds number in the experiment. Steady state particle positions of the experiment are seen in Figure 5.1.



**Figure 5.1:** Resulting experimental particle positions (after 12 s) for  $Re_f = 1241$ .

## 5.1 Initial Conditions

This section describes the initial conditions for the validation, the initial position and flow initialisation is investigated.

### 5.1.1 Initial particle positions

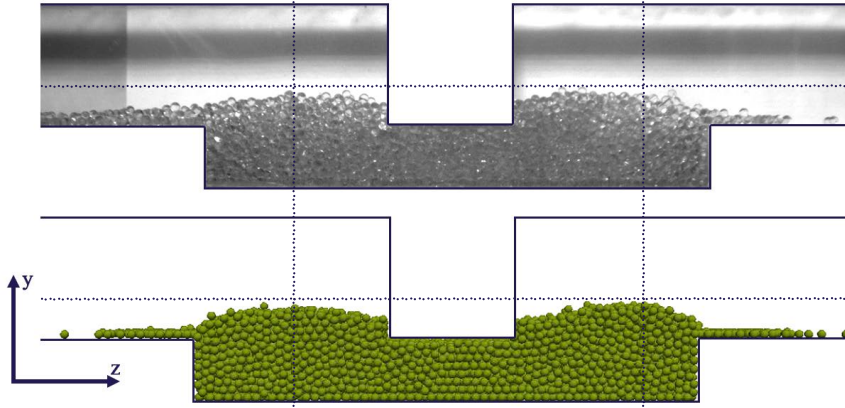
In the experiments, the cavity is filled by removing one side, filling by hand through the side, reassembling the setup and then the experimental setup is turned  $90^\circ$  before the setup is ready for testing. Figure 5.2 shows the filling of the test setup.



**Figure 5.2:** The test setup is filled from the side and tilted  $90^\circ$  before the test is ready.

This filling method makes it challenging to achieve the exact same particle positions for all experiments. To achieve the same initial conditions, the numerical filling of the cavity is adjusted until similar particle positions are seen. The numerical filling is made by inserting particles in different

regions three times, the numerical filling is further elaborated in Appendix F. The particle positions are adjusted by changing number of particles inserted and location of the region, they are inserted in. The results of this is illustrated in Figure 5.3.

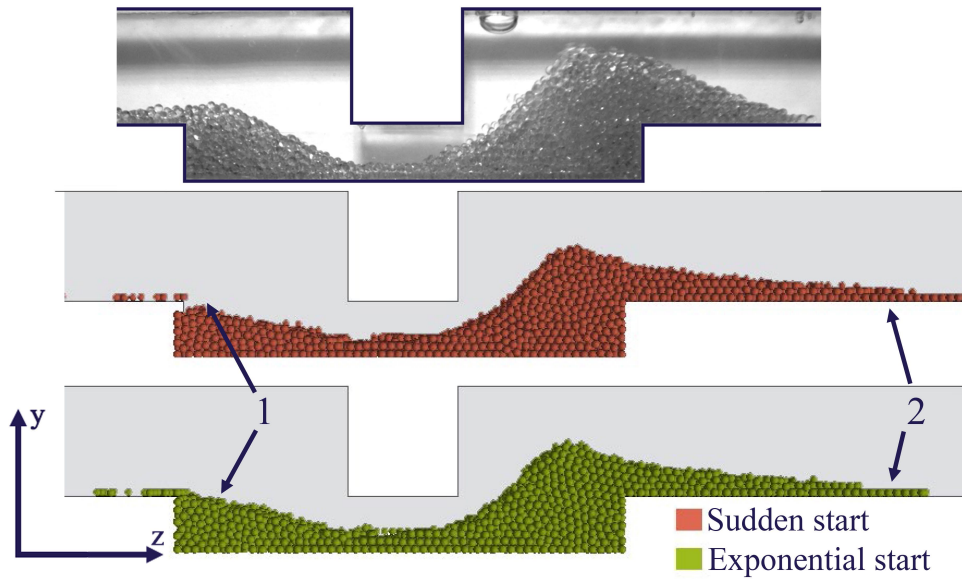


**Figure 5.3:** Initial positions of particles in experiment (top) and simulation (bottom). Dotted lines are inserted for easier comparison.

As the insertion of particles in the simulation is done randomly, exactly identical starting positions is impossible, which can also be seen in Figure 5.3. By close inspection, the amount and starting positions of particles are clearly comparable.

### 5.1.2 Simulation flow initialisation sensitivity

To ensure consistency between simulation and experiment, not only the initial positions, but also the initialisation must ideally be identical, as mentioned in Section 3.2 on page 44 and proven in Section 4.3 on page 68. A sensitivity study of the initialisation of the numerical model is done, where a comparison between the particle positions of simulations with sudden start and with an exponential start (see Section 3.2) is seen in Figure 5.4. Both simulations have a fully developed flow profile at inlet. The simulations are run with  $Re_f = 1241$  based on the initial particle positions described above.



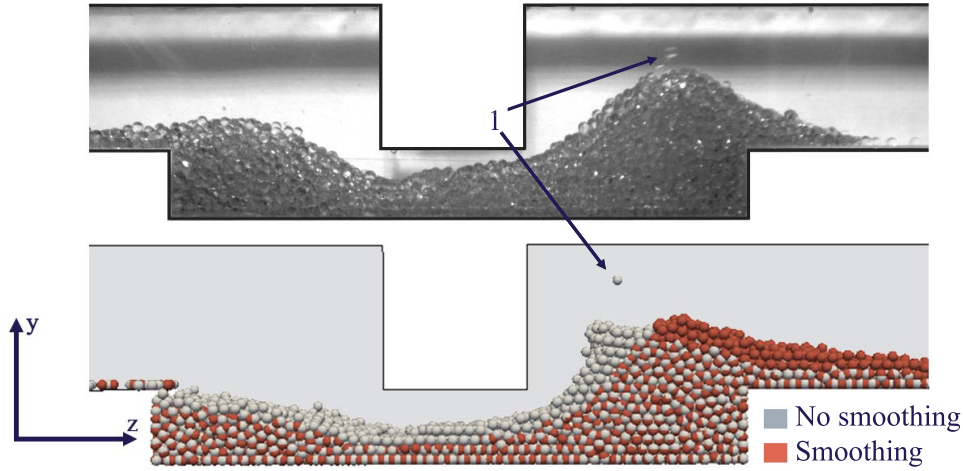
**Figure 5.4:** Comparison of particle positions with sudden and exponential ramp-up start.

Investigating Figure 5.4 there are some differences. At point 1 it is seen that particles with a sudden start have moved further, while an exponential start lets more particles stay in start of the cavity, which is more consistent with the experimental results. At point 2 it is clear, that the sudden start simulation moves particles further, since the flow has a larger velocity at the beginning. Thus, the particles will be drawn by the flow from  $t = 0$  s. Though the general tendency of particle accumulation is the same, the exponential start up is used for validation, as it is more comparable with the experimental starting conditions.

## 5.2 Smoothing Sensitivity

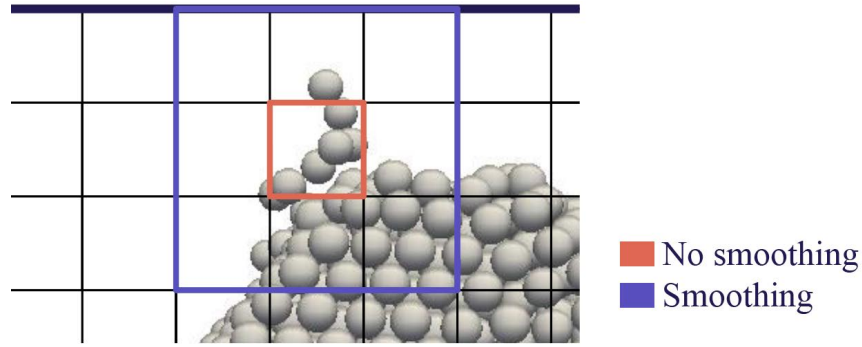
The simulations include smoothing in order to increase stability of the numerical model. In order to investigate this, a simulation identical to the validation simulation, but with no smoothing included, is run. Figure 5.5 shows a comparison between the experiment, a simulation with smoothing and one without at approximately the same time. *Note: the simulation image illustrates two different simulations in the same image. This is done to ease compar-*

ison of the particle positions. There is no interaction between the two simulations.



**Figure 5.5:** Comparison of particle positions after 10 s for simulations with and without smoothing. *Note: Experimental image is from another experiment and is used as it clearly shows a particle lifting from the bulk.*

The figure shows at point **1**, that some particles lift from the particle bulk, which is also seen in the experiment. It is also seen, that the general particle accumulation shows poor resemblance, when no smoothing is included. This indicates, that applying smoothing to the simulations not only improves the stability of the numerical model, but also increases the area of the boundary between dense and dilute particle-fluid flow. This means that the effects of a cell exactly on the boundary will also have an effect on the neighbouring cell, which is in the dense particle area. This is illustrated conceptually in Figure 5.6, where the particle-fluid exchange field of the red cell only affects the red cell itself, if no smoothing is applied. By applying smoothing, the void fraction and implicit momentum exchange in the red cell will also be smoothed out over the blue cells.



**Figure 5.6:** Red cell indicates the forces of a model without smoothing. Blue box shows the cells affected by the smoothing from red cell.

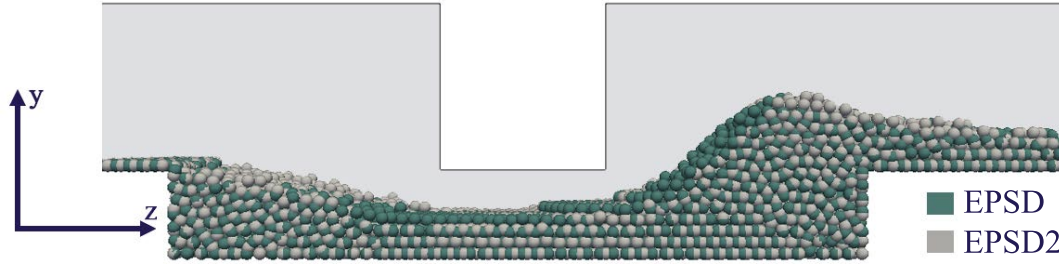
Smoothing is included in the numerical model for the validation, using a normalised smoothing length of  $L_{\text{smooth}}/D_h = 0.25$ .

### 5.3 Particle-Particle Rolling Friction Sensitivity

To determine the effects of the rolling friction between the particles, a series of simulations, with different rolling frictions, are run. Two parameters are used in the evaluation: particle-particle rolling friction coefficient and choice of rolling friction model.

#### 5.3.1 Rolling friction model

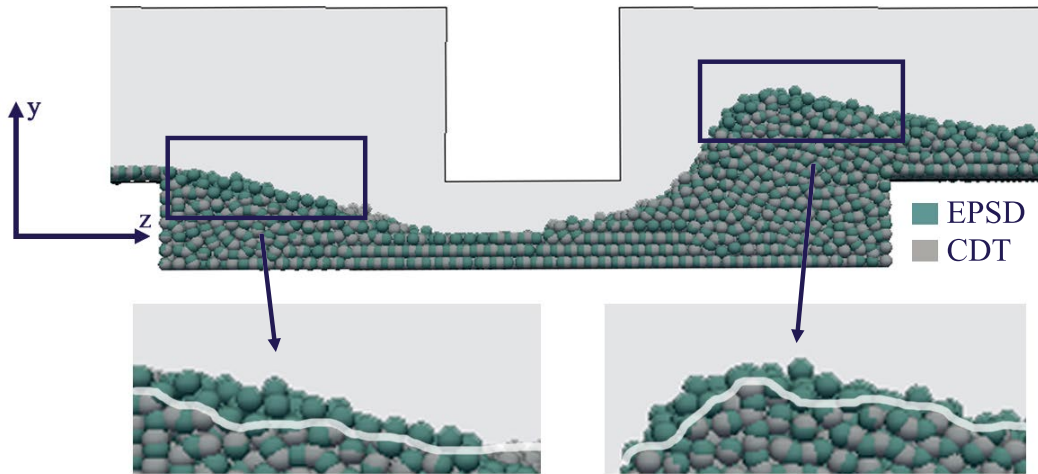
There are different rolling friction models available in the software. Three rolling models are investigated: Constant Directional Torque (CDT), Elastic-Plastic-Spring-Dashpot (EPSD) and EPSD2. The EPSD models use a spring-dashpot method to calculate the rolling friction, while CDT is only dependent on the difference in angular velocity of colliding particles. Both methods include an identical particle-particle coefficient of rolling friction constant ( $C_{pp,roll}$ ). The EPSD model also include a damping coefficient, which in this case is set to 0. The damping coefficient has been investigated by running a few simulations with varying coefficients, where a damping coefficient of 0 shows the best similarity to the experiment. A comparison between the resulting steady state particle positions of the EPSD and EPSD2 models is seen in Figure 5.7.



**Figure 5.7:** Comparison between elastic-plastic spring-dashpot (EPSD) models resulting particle positions.

The image indicates, that the results are similar, which is also expected, since the difference between the models is the lack of a dashpot in EPSD2 and the damper in the EPSD simulation is set to 0. Although a difference is seen between the two models, which is due to a deviation in the estimation of the rolling stiffness. As the slope of particles at outlet for EPSD is more similar to Figure 5.1, thus this model is now compared to the CDT.

Figure 5.8 illustrates a comparison between identical simulations with the CDT and EPSD rolling friction models respectively. The EPSD method is more computationally costly.



**Figure 5.8:** Comparison of EPSD and CDT results. Particles in the EPSD simulation have slightly higher particle placements just as cavity begins and ends, which is more comparable to the experimental results.

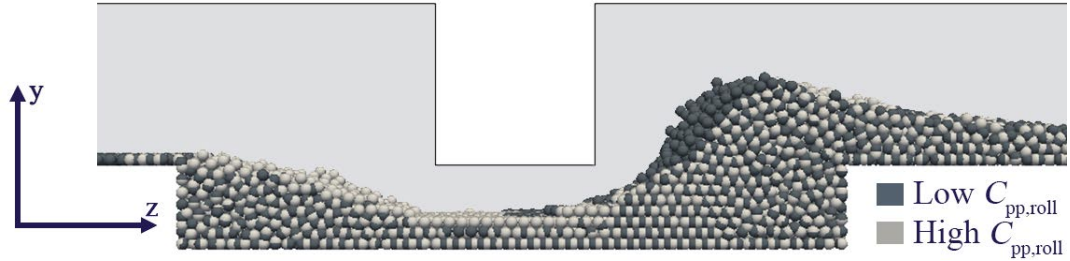
The resulting particle positions are very similar, with only a slight difference at the beginning and end of the cavity, as elaborated in Figure 5.8.



The difference is very subtle, and does not impose a significant effect on the resulting particle positions. The EPSD model does however resemble the experimental results more accurately. Common for the mentioned rolling friction models is the inclusion of a rolling friction coefficient, which is investigated next.

### 5.3.2 Rolling friction coefficient

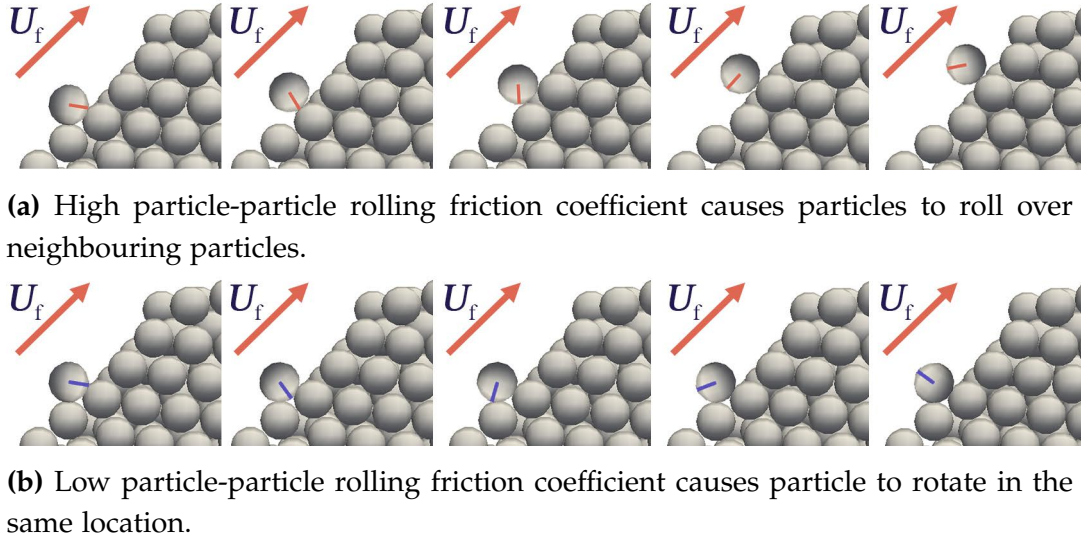
To determine the effects of the particle-particle rolling friction coefficient ( $C_{pp,roll}$ ), two identical models with  $C_{pp,roll} = 0.3$  and  $C_{pp,roll} = 0.03$  are simulated. Comparison of resulting particle accumulation is seen in Figure 5.9.



**Figure 5.9:** Reducing particle-particle rolling friction coefficient ( $C_{pp,roll}$ ) by a factor of 10 results in a steeper incline angle and less downstream sediment transport.

It is seen from Figure 5.9, that the final positions of particles are sensitive to the rolling friction coefficient, which was also found by Li et al. (2011) for a pure DEM case. A reduction in rolling friction coefficient results in more particles leaving the area below the block, which is also the case in the experiment. At the start of the cavity however, more particles are transported downstream, which is not the case for the experiment. Also the angle of incline after the block is visibly steeper with a low  $C_{pp,roll}$ , which is likely caused by the particles rolling as fluid passes them. The momentum exchanged from fluid to particle results in a rolling motion instead of a downstream transport, as illustrated in Figure 5.10.



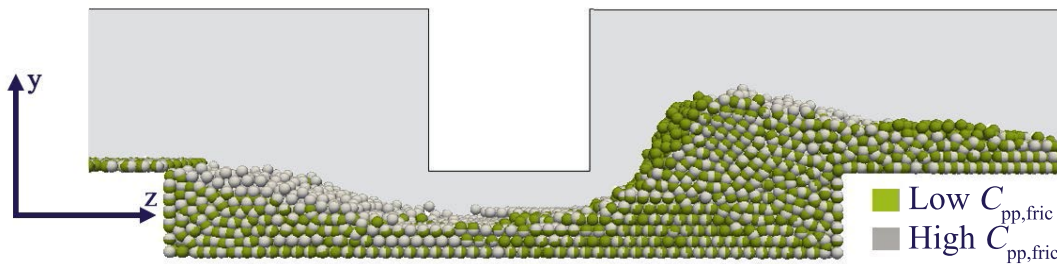


**Figure 5.10:** Effect of the particle-particle rolling friction coefficient.

For the validation simulation the CDT model is used for economic reasons and the particle-particle rolling friction coefficient is set to 0.3.

## 5.4 Particle-Particle Friction Sensitivity

Two identical simulations with the particle-particle friction coefficient changed, are simulated in order to determine the effects of this parameter; one with  $C_{pp,fric} = 0.01$  and one with  $C_{pp,fric} = 0.06$ . Results of simulations are seen in Figure 5.11.

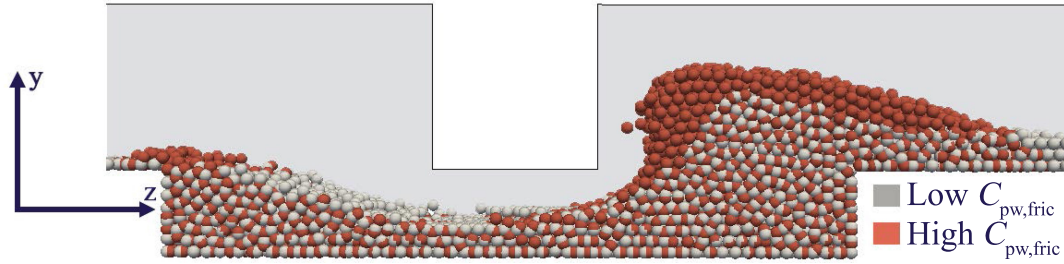


**Figure 5.11:** Reducing the particle-particle friction coefficient results in particles accumulating with a steeper incline after the block.

As the figure indicates, reducing particle-particle friction coefficient results in less particles accumulating in the cavity *before* the block and a steeper incline of particles *after* the block. With lower  $C_{pp,fric}$ , more of the particles will move downstream, resulting in a steeper incline, where the fluid momentum imposed on particles are not high enough, to transport particles further downstream. In addition to particle-particle contact, some particles experience contact with the wall. The particle-particle friction coefficient used for validation is 0.06.

## 5.5 Particle-Wall Friction Coefficient Sensitivity

Particles adjacent to the wall experience both friction between neighbouring particles as well as the wall. The effects of the particle-wall friction coefficient are investigated in two identical simulations with particle-wall friction coefficient ( $C_{pw,fric}$ ) of 0.06 and 0.01 respectively. Comparison of results are seen in Figure 5.12.

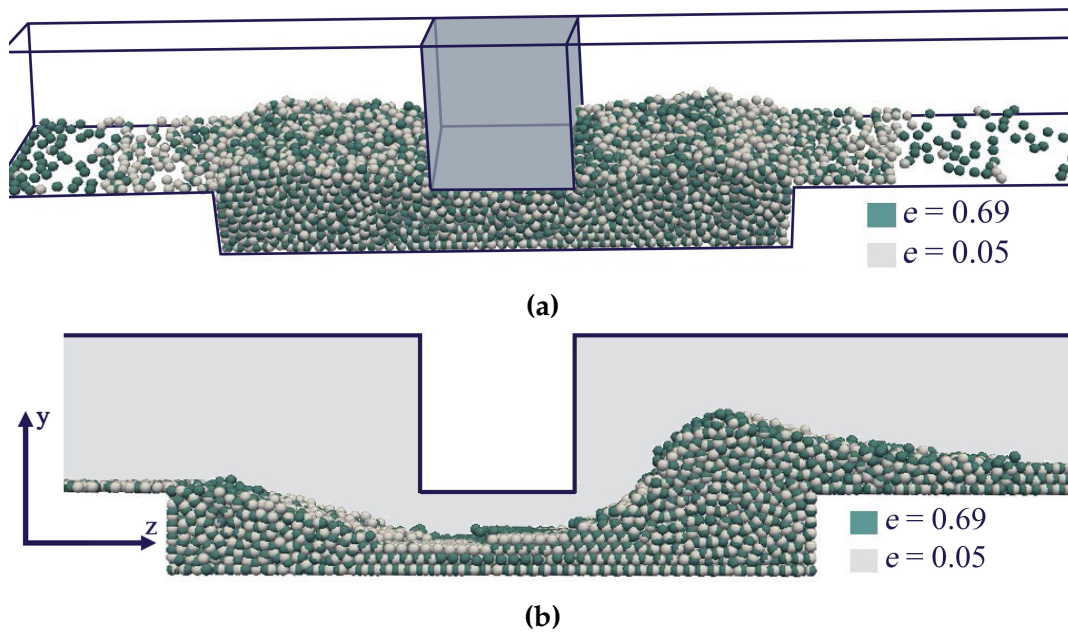


**Figure 5.12:** Increasing particle-wall friction coefficient hinders particles from leaving the cavity.

The figure clearly indicates, that a higher particle-wall friction coefficient increases the amount of accumulated particles within the cavity, as the force required to lift particles from the cavity is higher, than implied by the fluid. The effect does not only cover the wall adjacent particles but also the particles in the middle of the channel in the x-direction. This is caused by the particle-particle friction affecting all particles and not only wall adjacent particles. The particle-wall friction coefficient used for validation is 0.06.

## 5.6 Coefficient of Restitution Sensitivity

The coefficient of restitution ( $e$ ) of glass is relatively high, which means that the glass particles have an initial velocity, which is higher than the resulting velocity of the particles after a collision ( $U_{p,0} > U_{p,res}$ ). While particles with a low  $e$  will be subject to have a much higher initial velocity than resulting velocity after the collision ( $U_{p,0} \gg U_{p,res}$ ). This also means, that simulations with the actual  $e$  of glass takes significantly longer to start, due to the particles bouncing during the filling of the domain. When all particles are inserted, the simulation must run longer for the particles to settle. Using a low  $e$ , the particles settle quicker, and thereby reducing the computational cost. To investigate the effects of this, two identical simulations with exception of the  $e$ , which is set to 0.69 (the actual  $e$ ) and 0.05 respectively, are run. Figure 5.13a illustrates the starting position of particles, while Figure 5.13b illustrates the final positions of particles.



**Figure 5.13:** (a) Initial positions of particles. View is tilted and seen slightly from above. (b) Final particle positions.

Figure 5.13a underlines, that the particles move more during initial filling of the cavity, however the general tendency is similar, with only a few

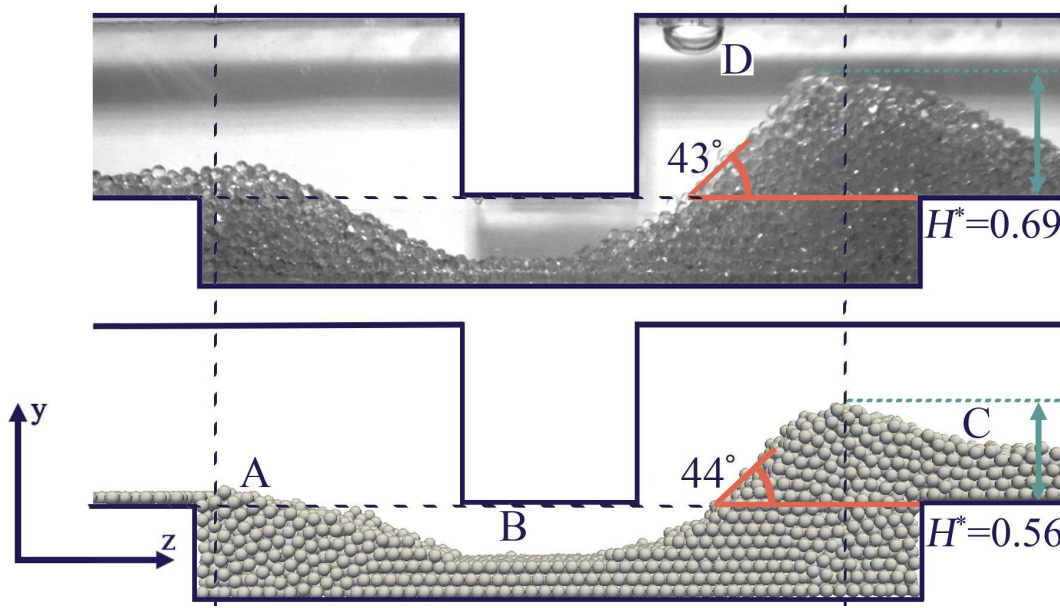
more particles being placed in the water channel with the actual  $e$ . The filling simulation time has a computational cost of more than a factor four with  $e = 0.69$ , compared to a  $e = 0.05$ . In Figure 5.13b it is seen, that the final positions of particles are almost identical despite the difference in  $e$ . The deviance in final position might be caused by the slight difference in starting positions of particles. The reason for this similarity is likely, that the densely packed particles does only experience collisions with neighbouring particles with  $U_{p,0} \approx 0$ , which means the effects of  $e$  can be neglected. For validation the coefficient of restitution is set to 0.05.

## 5.7 Final Validation of Numerical Framework

The final validation is made with considerations based on the comparisons made so far in this chapter. The coefficients used in this simulation are listed in Table 5.1. The simulation is started with exponential inlet condition and is run until steady state is reached. Comparison between experiment and simulation is seen in Figure 5.14, which shows the angle of incline and the specific particle height ( $H^* = H_{p,max}/D_h$ ), where  $H_{p,max}$  is the height from *water channel bottom* to the highest positioned particle.

**Table 5.1:** Values used in the numerical model for the final validation simulation.

Parameter	Value	
Effective particle-particle Young's modulus	$E_{eff,pp}$	$3.3 \cdot 10^8$
Effective particle-wall Young's modulus	$E_{eff,pw}$	$2.9 \cdot 10^7$
Coefficient of Restitution	$e$	0.05
Particle-wall rolling friction coefficient	$C_{pw,roll}$	0.30
Particle-particle rolling friction coefficient	$C_{pp,roll}$	0.30
Particle-wall friction coefficient	$C_{pw,fric}$	0.06
Particle-particle friction coefficient	$C_{pp,fric}$	0.06



**Figure 5.14:** Comparison between experimental (top) and numerical (bottom) steady state particle accumulation. Dotted lines are inserted for easier comparison. A, B, C and D indicate points of interest. The specific particle height is defined as  $H^* = H_{p,max}/D_h$ .

The figure shows, that the simulation does show similar tendencies with, but is not identical to the experimental results. The angle of incline in the simulation ( $44^\circ$ ) is almost identical to the experiment ( $43^\circ$ ), while  $H^*$  of the simulation (0.56) differ by -19% relative to the experiment (0.69).

At location A, the particle height in the experiment is higher than in the simulation and may be caused by the two layers of particles in the channel, upstream from location A, as opposed to the single layer of particles in the experiment. The particle incline at A will affect the direction of fluid entering the cavity, so that a more downward directional flow direction occurs towards location B.

At location B, three horizontal rows of particles are located in the simulation as opposed to only two rows in the experiment. This indicates that the angle of fluid direction might not be steep enough compared to the experiment, which can be caused by the lack of particles in location A.

At location C there is an obvious difference between simulation and experiment, where the particle height in the simulation is lower than in the experiment. The lack of particles in location C is likely caused by the lack

of particles leaving location **B**.

At location **D** a large bubble is present, which might affect the flow of the experiment. This bubble is not included in the simulation.

The numerical model and experimental results show the same tendency, however they are not identical. This can be caused by several reasons.

## 5.8 Sources of Error

Some of the possible sources of error are described in this section and include:

- Material properties
- Start up conditions
- Use of unresolved model
- Drag acting in boundary layer

### Material properties

The material properties used in the validation are chosen based on available values in literature or from the manufacturer, and tweaked based on experience from investigations conducted in this chapter. The values from literature may be given for specific cases and have been used as a starting point, however the simulation with these values does not represent an exact match with the experiment. Tweaking of the values is related to a very high computational cost, as a vast number of simulations are required to determine the correct values for each parameter.

### Start up conditions

The inlet velocity profile during start up of the experiment, which is seen in Figure 4.8 on page 72, develops to a fully laminar flow profile. Though considerations of initial inlet conditions in simulation have been made to imitate the true start up, the development of flow profiles are not identical. Also, the simulations are started with particle positions, that are not exactly identical to those of the experiment, however similar. These factors might cause slight deviance between the experiment and simulation.

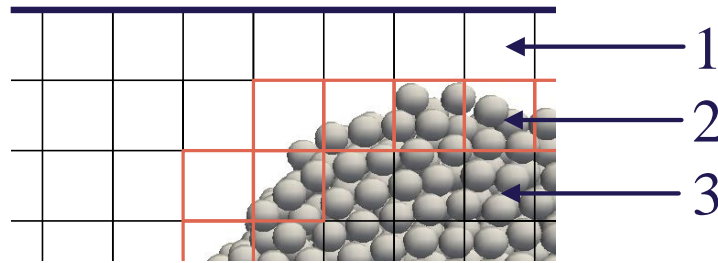
### Use of resolved method

The validation is conducted with a model based on an unresolved approach.

Using this method does not fully dissect the flow around particles, which might result in deviance between model and experiment, however as the simulation includes many particles in a densely packed area, using a resolved method is too computational expensive to be possible within the time limits of this project. This is also underlined by Mondal et al. (2016).

### Boundary layer drag

The drag force model determines the drag coefficient, which is a function of the void fraction and the fluid Reynolds number. In Figure 5.15, a conceptual sketch of a grid and an accumulation of particles is seen.



**Figure 5.15: 1:** The fluid void fraction is  $\alpha_f = 1$  in cells with no particles. **2:** Boundary cells between dilute and dense particle-fluid flow, marked by red. **3:** For dense particle-fluid flow the fluid void fraction is approximately  $\alpha_f \approx C$ .

At cell type **1**, there are no particles present in the cells, why the void fraction equals one, which means that for cells of this type, the drag coefficient is a function of the fluid Reynolds number ( $C_D = f(Re_f)$ ). When particles are present in a cell, the drag coefficient becomes dependent on the fluid Reynolds number and the fluid void fraction ( $C_D = f(Re_f, \alpha_f)$ ). For cell type **3**, the fluid void fraction approximately equals a constant value ( $\alpha_f \approx C$ ) and the fluid Reynolds number approaches zero ( $Re_f \approx 0$ ), which means the drag force applied to cells of this type is both almost uniform for all cells of this type and very low. For cell type **2**, which are cells covering the boundary between dense and dilute particle-fluid flow and highlighted by red in Figure 5.15, there is not a uniform particle distribution. This means, that  $\alpha_f \neq C$  and  $\alpha_f$  in one cell may be significantly different from  $\alpha_f$  in the neighbouring cell. As smoothing is included, this may lead to unrealistic results in the individual cells.



## 5.9 Chapter Summary

This chapter has described the effects of some parameters, such as material properties, starting positions of particles as well as initiation procedure of numerical model, in order to produce a valid numerical model. The investigations of effects of the different parameters provide a basis for tweaking of the numerical model, which is used for validation. Examples of sources of error are presented as well. As the simulation results show good agreement with the results of the experiment, a parametric study is conducted in Chapter 6 using this model.



# Chapter 6

## Model-Based Parametric Study

The purpose of this chapter is to describe, how the fluid-particle dynamics are affected by a change in fluid Reynolds number ( $Re_f = \rho_f U_{f,avg} D_h / \mu_f$ ) and the particle-to-fluid density ratio ( $\chi = \rho_p / \rho_f$ ). The studies are conducted on the geometry and model setup described in Chapter 2 and Chapter 3. The cases are simulated until steady state is achieved and all results are evaluated at steady state. The final particle positions are investigated to determine the correlation between chosen parameter and particle positions. The results of the parametric study are seen in Section 6.2.

### *Fluid Reynolds number*

The fluid Reynolds number can be altered by the fluid density, hydraulic diameter, fluid velocity or the dynamic viscosity. As the geometry and fluid properties are constant, the fluid velocity is changed. The study includes three different flow velocities, all of which are in the laminar flow region. The investigated fluid Reynolds numbers are 600, 800 and 1000.

### *Density ratio*

The fluid properties are kept constant, thus the density of the particles are changed. The analysed particle to fluid density ratios are 1.5, 2.0 and 2.5.

All simulations are identical with exception of the values of the two specific parameters. This means that simulations are run with identical fluid and particle properties, the same time steps, initial- and boundary

conditions. The values for the properties are seen in Table 6.1. It should be noted, that the parametric study is based on the validated model, but conducted with slightly different input properties. The final positions of the particles are only functions of the fluid Reynolds number and density ratio.

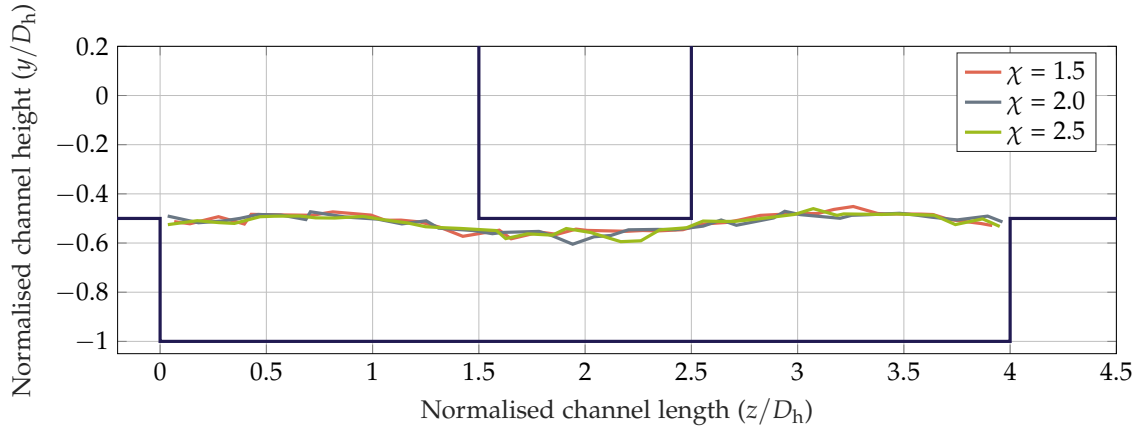
**Table 6.1:** The particle and wall properties used in the numerical model for the parametric study.

Parameter	Property	Value
Dimensionless particle diameter	$D_p/D_h$	0.07
Effective Young's modulus (particle-particle)	$E_{\text{eff},pp}$	$3.3 \cdot 10^8$
Effective Young's modulus (particle-wall)	$E_{\text{eff},pw}$	$3 \cdot 10^6$
Coefficient of restitution	$e$	0.69
Friction coefficient (particle-particle)	$C_{pp,\text{fric}}$	0.4
Friction coefficient (particle-wall)	$C_{pw,\text{fric}}$	0.4
Rolling friction (particle-particle)	$C_{pp,\text{roll}}$	1
Rolling friction (particle-wall)	$C_{pw,\text{roll}}$	0.7

The initial position of the particles will defer between the parametric studies, the differences in initial conditions are presented in the next paragraph.

### Initial position of the particles

Each density ratio is given a specific initialisation, which introduces the possibility of different filling, resulting in different particle positions. However, as seen in the top position line in Figure 6.1, which illustrates the initial particle positions for the three density ratios, the particle positions are similar. This indicates, that the difference in resulting positions will be a result of the change in parameter, and not a result of initial position of the particles.

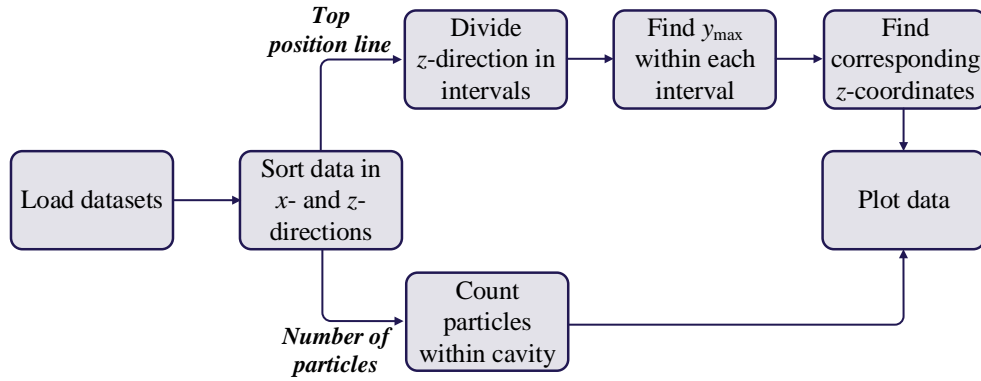


**Figure 6.1:** Comparison of top position line for initial positions with particle-to-fluid density ratios of  $\chi = 1.5$ ,  $\chi = 2.0$  and  $\chi = 2.5$ .

The parametric studies are evaluated at steady state using different approaches. The methodology for the data treatment of the parametric study is presented in the next section.

## 6.1 Data Treatment Methodology

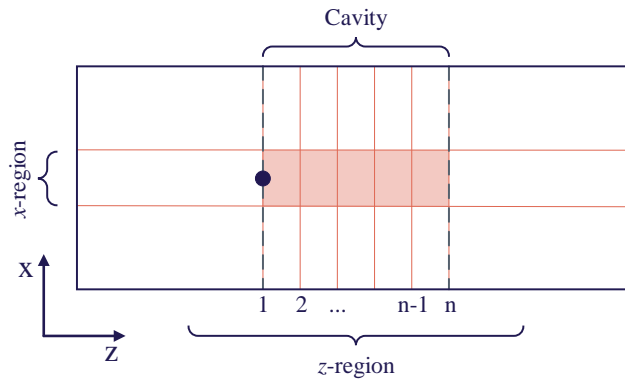
The particle position results are compared in two ways: Top position line of highest particles and number of particles in the cavity. The simulation results are compared at the same time step and data is treated as per the flowchart in Figure 6.2. Additionally the flow field is investigated by the pressure drop across the geometry. The different data treatment methods are explained after the flowchart.



**Figure 6.2:** Flowchart of the approaches for data treatment for analysing the results from the parametric study.

### *Top position line*

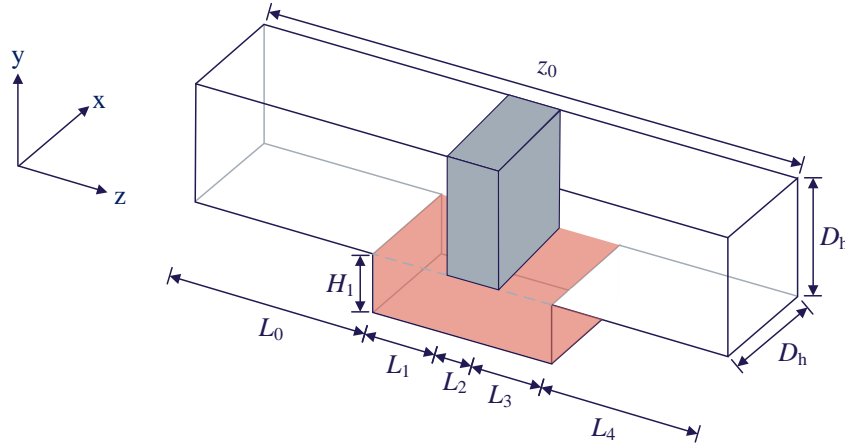
The particle accumulation is compared by means of a line representing  $y_{\max}$  along the  $z$ -axis. This is determined according to the flowchart in Figure 6.2. The  $z$ -region of interest is from the edges of the cavity, as shown in Figure 6.3. The  $x$ -region has a width of  $0.2 D_h$  centred around  $x/D_h = 0$ .



**Figure 6.3:** Definition of the interval in the  $x$ - and  $z$ -directions to find  $y_{\max}$  in each interval. The region of interest is shaded with red and the origin is marked by a blue dot.

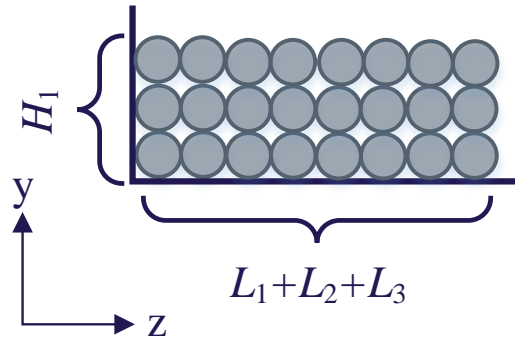
### *Number of particles*

The amount of particles within a region at the end of simulation is used to compare results. The region is located at the cavity and covers the entire  $y$ - and  $x$ -directions, as illustrated by red in Figure 6.4.



**Figure 6.4:** The region used for comparison of the number of particles located in the cavity area.

The number of particles within the region is determined before the simulations are initiated and when steady state is reached. The number of particles in the cavity at steady state is normalised by the number of particles theoretical used for filling the cavity with stacking of particles as shown in Figure 6.5.



**Figure 6.5:** The stacking of particles used for normalising the number of particles in the cavity at steady state.

The theoretical number of particles inside the cavity, when filling to the top of the cavity, is found from:

$$n_{p,0} = \frac{H_1}{D_p} \frac{(L_1 + L_2 + L_3)}{D_p} \frac{x_0}{D_p} \quad (6.1)$$

Where

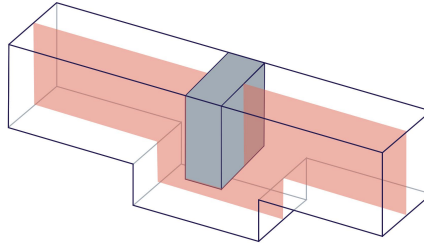
$H_1$	The cavity height
$(L_1 + L_2 + L_3)$	The cavity length
$x_0$	The cavity width

The resulting number of particles is given as the ratio between the number of particles in the cavity region at steady state normalised by the theoretical initial number of particles in the cavity region given by  $n_{p,res}/n_{p,0}$ .

#### *Investigation of the flow field*

It is investigated, how the density ratio affects the flow conditions in the model. For the investigations the friction factor and the fluid velocity are evaluated. The friction factor ( $f = 2 \Delta P_L D_h / (z_0 \rho_f U_f^2)$ ) is used for comparison in the parametric study, where the pressure is measured at in- and outlet.

The change in flow velocity is investigated in the centre plane of the geometry, as shown in Figure 6.6.



**Figure 6.6:** Location of plane used for velocity field evaluation.

The data treatment methodology of the results of the parametric study has been presented and in the following section the results is presented.

## 6.2 Parametric Study Results

In this section the results of the parametric study is presented, which starts by presenting the initial position of the particle followed by the analyses of the top position line and number of particles in the cavity area. The last evaluated parameter is the flow field by means of the friction factor and fluid velocities.

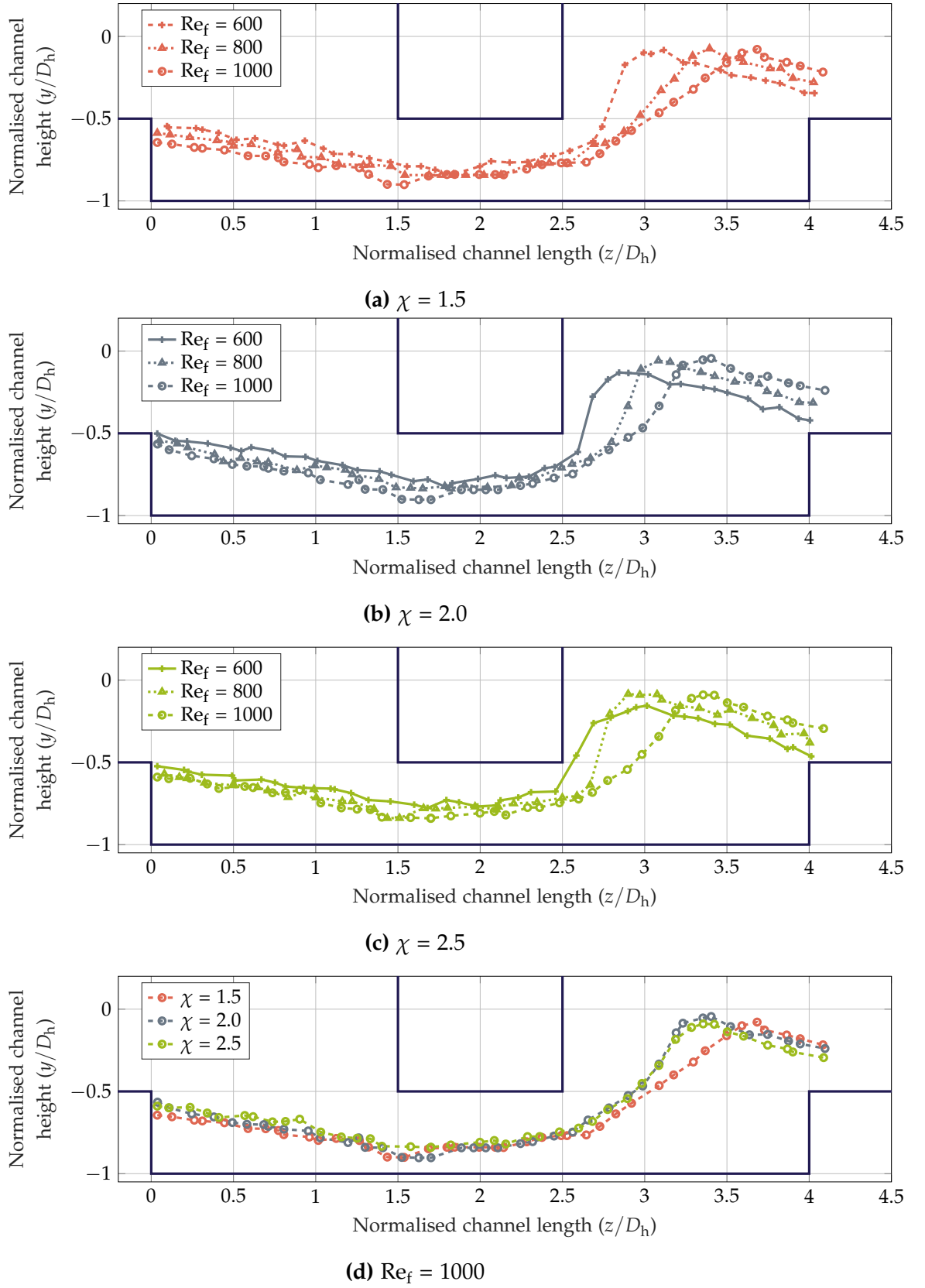
### 6.2.1 Top position line

The final positions of the particles are evaluated by a top position line, as described in Section 6.1. The top position lines are displayed with all three Reynolds numbers for each density ratio;  $\chi = 1.5$  is illustrated in Fig 6.7a,  $\chi = 2.0$  in Fig 6.7b and  $\chi = 2.5$  in Fig 6.7c. Also a top position line of the three density ratios at  $Re_f = 1000$  is illustrated in Figure 6.7d for the sake of easier comparison. The figures illustrate the same general tendency.

*Before the block* ( $0 < z/D_h < 1.5$ ): The particle top position line at the start of the cavity indicate that for lower  $Re_f$ , the top position is higher. This is common for all three density ratios. Increasing  $Re_f$  lowers the top line position slightly, however the difference in top line position is within approximately one  $D_p$ .

*Below the block* ( $1.5 < z/D_h < 2.5$ ): The simulations indicate that for  $\chi = 1.5$  the top position line is almost identical, however for simulations with  $\chi \leq 2$ , increasing  $Re_f$  will result in more particles to move further and the top position line to be lowered.

*After the block* ( $2 < z/D_h < 4$ ): The top position line incline is steeper at lower  $Re_f$ . As the fluid Reynolds number increases the incline decreases and the distance between the particles and the block becomes larger. Similar tendencies are seen for decreasing density ratios. For the lowest density ratio, the incline is less steep. As the density ratio increases the particles come closer to the block due to the higher mass.



**Figure 6.7:** Top position lines for the different density ratios in the parametric study. The density ratio is illustrated by color and the different Reynolds numbers are displayed with different markers.

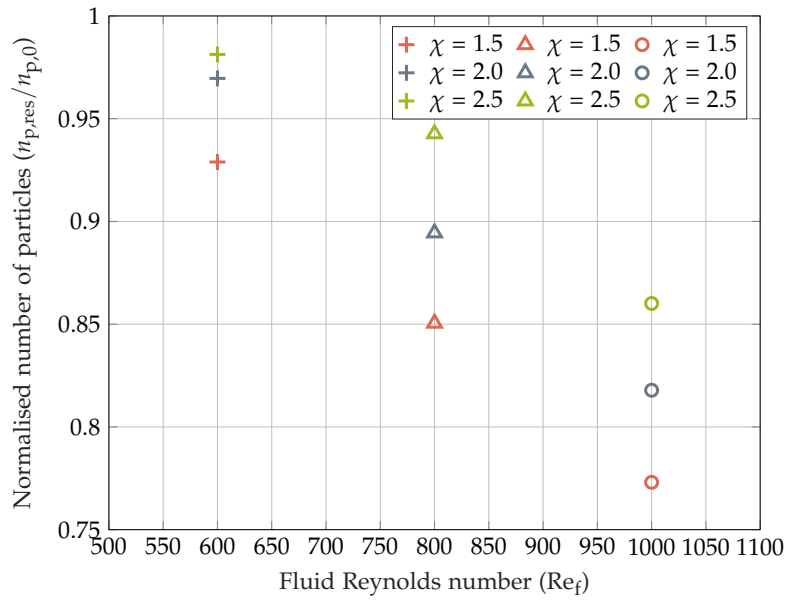


Figure 6.7d illustrates the top position line for  $Re_f = 1000$  with the density ratios  $\chi = 1.5, 2.0$  and  $2.5$ . This figure is presented to give a more clear indication of the effects of a change in the particle-to-fluid density ratio as a function of the fluid Reynolds number.

As the figure shows, a lower particle-to-fluid density ratio evens the top position line more, resulting in lowered particle height before the block and a less steep incline after the block. Another effect is the peak position being moved further in the  $z$ -direction, which is caused by the reduced particle density and thereby a reduced gravitational force.

### 6.2.2 Number of particles

The number of particles within the cavity region is analysed and the parametric study results are shown in Figure 6.8 as a function of  $Re_f$ .



**Figure 6.8:** The number of particles in the cavity region at steady state normalised by the initial number of particles plotted as a function of fluid Reynolds number for the three different density ratios in the cavity region, which is shown in Figure 6.4.

The figure indicates that both the particle-to-fluid density ratio and the Reynolds number affect the particle sediment in the cavity. When the par-

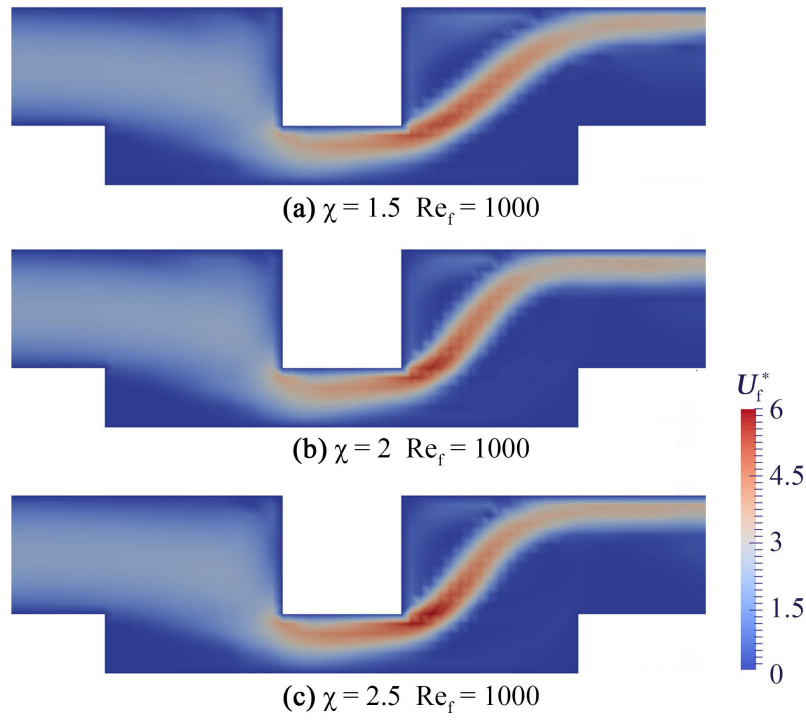
ticles have a low density ratio, the fraction is close to one meaning more particles still are present in the cavity area. A decrease in the density ratio also results in downstream transport of particle sediments. The same is valid for increasing the fluid Reynolds number. As the Reynolds number is increased more particles leave the cavity. The combination of the lowest density ratio and highest Reynolds number results in increased downstream sediment transport, which is caused by the combination of lightest particles and highest fluid velocity.

In the next section the flow field is investigated. This includes investigations of how the changes in density ratio and Reynolds number, which affect the number of particles in the cavity, influences the pressure drop for the geometry.

### 6.2.3 Investigation of the flow field

When steady state is reached, the resulting fluid velocity flow field is analysed and compared. The comparison is done for all three particle-to-fluid density ratios at  $Re_f = 1000$ , in order to determine the effects of particle density on fluid velocity. The flow velocity plane is seen in Figure 6.6 and the CFD comparison is seen in Figure 6.9.

As seen in Figure 6.9, at a lower particle-to-fluid density ratio ( $\chi = \rho_p/\rho_f$ ), the fluid velocity is lower and the high-velocity area is less steep, while higher  $\chi$  values results in a higher maximum fluid velocity as well as a steeper incline. Comparing Figure 6.7d with Figure 6.9, it can be seen, that the flow angle after the block for lower particle-to-fluid density ratios is less steep, which is also the case for the particle incline angle. It is also seen, that locations, where many particles are present, have  $U_f \approx 0$ . In other words, the fluid flow does not penetrate the particle boundary layer, and particles impose wall-like conditions for the fluid. The pressure drop through the geometry for the three cases are expressed in terms of the specific friction factor relative to the friction factor for a straight channel of same length, and is listed in Table 6.2.



**Figure 6.9:** Comparison of fluid velocity as function of particle accumulation for the three density ratios. The dimensionless fluid velocity is defined as  $U_f^* = U_f / U_{f,avg}$ .

**Table 6.2:** The normalised friction factor for varying fluid Reynolds numbers and density ratios. It is normalised by the friction factor  $f_0$ , which is for a straight channel as in the grid independence study.

$\chi = 1.5$		$\chi = 2.0$		$\chi = 2.5$	
$Re_f$	$f/f_0$	$Re_f$	$f/f_0$	$Re_f$	$f/f_0$
600	141.7	600	229.6	600	303.5
800	76.7	800	107.7	800	157.8
1000	56.4	1000	66.4	1000	81.3
(a)		(b)		(c)	

From Table 6.2 it is seen that the normalised friction factor decreases with increasing Reynolds number for all the investigated density ratios.

The friction factor ( $f = 2 \Delta P_L D_h / (z_0 \rho_f U_f^2)$ ) becomes more significant with higher density ratios, since the particles are not easily moved at high density ratio as shown in Figure 6.7d for  $Re_f = 1000$ . The lower amount of particles near the block lead to a lower pressure difference from inlet to outlet in the geometry. For high Reynolds numbers, the lower amount of particles can be seen in Figure 6.8.

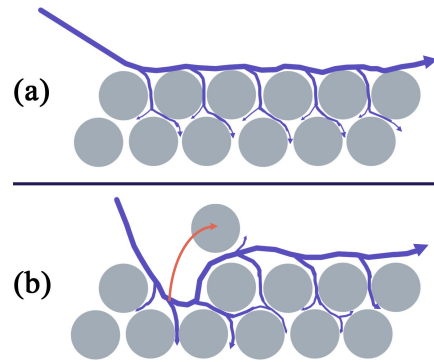
The friction factor is proportional to the pressure difference across the geometry. Looking at e.g. Table 6.2a, it is seen, how the pressure difference decreases with increasing Reynolds number. This is due to an increase in the area between the block and the particles at higher Reynolds numbers as seen in Figure 6.7a. The greater the Reynolds number the more significant the decrease in pressure difference becomes.

The friction factor is inversely proportional to the velocity squared. Thus, the friction factor decreases with increasing the Reynolds number, since the Reynolds number is proportional to the velocity of the fluid.

The results are discussed in the next section, where key parameters in the model are discussed.

### 6.3 Discussion

Based on the results of simulations, it seems that though the particles rotate, the two dominating forces are the gravitational force, keeping particles in place, and the drag force, pulling particles loose from the densely packed positions and letting them roll over the others. The friction of the neighbouring particles, the fluid velocity and angle of attack may be key factors in the release of particles from the pack. As can be seen in Figure 6.9, the locations with low void fraction (e.g. bottom corners in the cavity, where the particle accumulates) has  $U_f \approx 0$ . For lower fluid velocity and angle of attack, the fluid seemingly does not penetrate into the densely packed particles and the particles just act as a wall, which is illustrated conceptually in Figure 6.10a. For an increase in angle of attack, the fluid is able to penetrate the particles and can easier overcome the friction and gravity forces resulting in particles being lifted, as seen in Figure 6.10b. The same is likely the case for higher fluid velocity at the boundary layer as well as particles with less friction or lower density.



**Figure 6.10:** Conceptual sketch. **(a)** Low angle of attack or low fluid velocity struggles to overcome gravity and friction. **(b)** Higher angle of attack or fluid velocity overcomes friction and gravity.

The increase in  $Re_f$  results in a lower pressure drop, as particles are moved further, resulting in a larger opening between particles and block for the fluid. By increasing  $Re_f$  the downstream sediment transport can seemingly be increased. The same can be done by decreasing the particle-to-fluid density ratio. It is possible that the pressure drop for the case with a fluid Reynolds number of 600, can be further decreased by briefly increasing the flow to  $Re_f = 1000$  till steady state and then decreasing to  $Re_f = 600$  again, since the increased downstream sediment transport results in larger opening for the fluid, however this has not been investigated.

## 6.4 Chapter Summary

This chapter has described the parametric study parameters as well as the methods of evaluation used to determine the effects of a given parameter. The effects of change in parameters are summarised in Table 6.3. A discussion of the parametric study results is given and a conclusion of the report is given in Chapter 7.

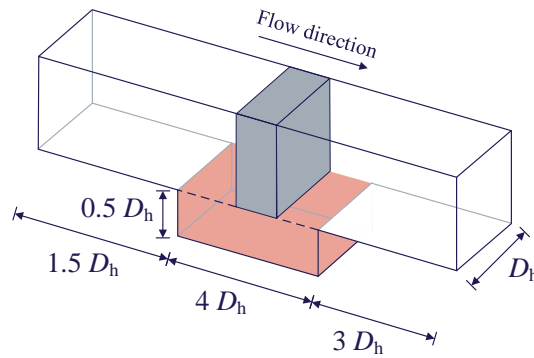
**Table 6.3:** Summary of effects of change in particle-to-fluid density ratio and inlet fluid Reynolds number. Angle of incline describes particle top line incline after the block.

Parameter	Angle of incline	$n_{\text{res}}/n_0$	$f/f_0$
$Re_f$ increase	decrease	decrease	decrease
$\chi$ increase	increase	increase	increase

# Chapter 7

## Conclusion

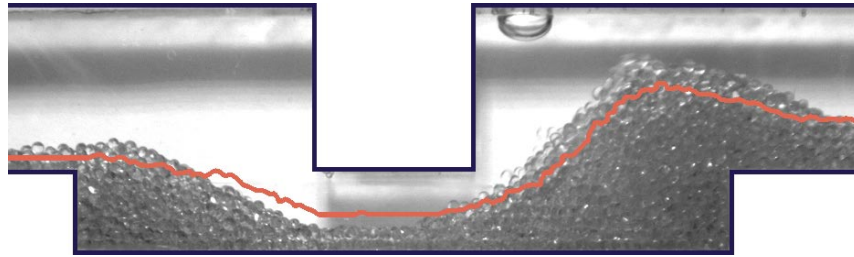
The scope of this project is to develop a numerical model that can simulate the fluid-particle dynamics in a simple geometry, which is seen in Figure 7.1.



**Figure 7.1:** The geometry used for the study. The red area is initially filled with particles.

The numerical model is constructed by using a four-way coupling of CFD and DEM framework. The fluid model (CFD) evaluates the flow through the geometry by solving the Navier-Stokes equations. The particulate model (DEM) evaluates the particle dynamics simulating the forces due to collision between particles and gravity. The coupled model includes the forces interacting between the particulate and fluid phases. The developed model is capable of simulating, how the particle behave, when a fluid passes through the geometry. The sensitivity of the model has been tested by varying entrance length, stiffness, time steps and particle and wall properties.

The model is validated by means of three different experimental tests using PIV, single particle drag and visual data logging, and shows good agreement between the simulation and experiment, as seen in Figure 7.2. The peak particle height of the simulation does not reach the same height as in the experiment, which is likely caused by a slight difference in particle accumulation at the start of the cavity. Another possible factor may be the change in void fraction at the top layer of the particles at each time step being a challenge for the drag model in combination with smoothing used in the numerical model.



**Figure 7.2:** Comparison between experiment and simulation for  $Re_f = 1241$ . Red line indicates top height of simulation particles.

The model is used for a parametric study of the effects of changing the particle-to-fluid density ratio ( $\chi$ ) and fluid Reynolds number ( $Re_f$ ) at inlet. The study is conducted for  $\chi = 1.5$ ,  $\chi = 2.0$  and  $\chi = 2.5$  and  $Re_f = 600$ ,  $Re_f = 800$  and  $Re_f = 1000$ .

This study indicates, that an increase in particle-to-fluid density ratio results in an increased normalised friction factor. This is caused by reduced downstream sediment transport, which results in a steeper angle of incline after the block for higher  $\chi$ , and is more critical at low  $Re_f$ . For  $\chi = 2.0$  and  $\chi = 2.5$  at  $Re_f = 1000$  the top lines in the cavity are almost identical, where a higher  $\chi$  results in a higher particle top position in the start of the cavity, relative to the flow direction, and a lower top position after the particle peak.

An increase in fluid Reynolds number increases downstream sediment transport and decreases the normalised friction factor.

It is concluded that the use of numerical simulations for investigation of fluid-particle behaviour can provide realistic results, and therefore is a useful tool in further understanding of this field.



# Chapter 8

## Future Work

This project has presented a model able to realistically represent fluid-particle behaviour in a simple geometry, as well as investigating the effects of changing two parameters, however this subject has a vast amount of possible subjects that can be explored. Some of the future work subjects are suggested in this chapter.

### Possible implementation

The knowledge provided in this report may be useful for industrial usage, where piping geometries result in particle accumulation, such as sudden contractions. For higher number of particles, the pressure loss may indeed be decreased by occasionally increase the  $Re_f$  to remove the particles from cavities, thus resulting in both lower pumping power required and less risk of clogging. If possible the system may benefit from occasional flushing, which may also be done by flushing with a higher-density fluid, thereby reducing particle-to-fluid density. This may be especially useful for heat exchangers, where particle accumulation can significantly reduce the heat transfer.

### Further parametric studies

This project has only briefly looked into the effects of altering parameters with respect to the particle-fluid dynamics. This can be subject to comprehensive studies as many different parameters can be investigated.

Examples of parametric studies are:

- Other particle sizes
- Rolling and sliding friction coefficients
- Further expansion of flow rates and particle-to-fluid density ratios

### **Other geometries**

The geometry used in this project is simple. Investigations similar to the ones conducted in this project can be done on other geometries. This can be rounded pipes or geometries that mimic typical piping components, such as valves.

### **Non-spherical particles**

The particles used in the numerical model have been perfectly spherical. Thus, the force models are for spherical particles. If the particles are not perfectly spherical or have other shapes, the force models must be altered, since they are based on spherical particles. This introduces more complexity to the model. Such modifications will make the model applicable to a wider range of problems.

### **Fully-resolved simulations**

This project has used the unresolved method to model the particle and fluid inter-phase. In order to dissolve the fluid interaction on the surface of the particles, a fully resolved method can be used. This will affect the computational time, since the mesh around the particles must be much finer, but it will give a better knowledge of what happens at the boundary layer between the fluid and particles.

### **Heat transfer**

In applications where heat transfer is involved such as heat exchangers, the presence of unwanted particles can cause pressure losses in the system, which is unwanted. The geometry can be altered and the model can be extended to include heat transfer. This will provide information about

which flow regimes are beneficial to operate in, in terms of minimal pressure loss and the lowest possible particle accumulation. Such investigations can be beneficial for all systems including heat transfer, where particles can possibly be present and cause problems.

## **Turbulence**

In order for the model to be applicable for a wider range of flow regimes, it can be changed from laminar to a turbulence model. The force models will need to be altered to fit the turbulence. As an example the drag coefficient becomes lower for higher Reynolds number, which results in a lower drag force exerted on the particles. Adding turbulence makes the model more flexible and extends the applicability.



# Bibliography

- Akbarzadeh, V. and A.N. Hrymak (2016). "Coupled CFD–DEM of particle-laden flows in a turning flow with a moving wall". In: *Computers & Chemical Engineering* 86, pp. 184 –191. DOI: 10.1016/j.compchemeng.2015.12.020.
- Benra, F. et al. (2011). "A Comparison of One-Way and Two-Way Coupling Methods for Numerical Analysis of Fluid-Structure Interactions". In: *Journal of Applied Mathematics* 2011, pp. 1–16. DOI: 10.1155/2011/853560.
- Benyahia, S. et al. (2006). "Extension of Hill-Koch-Ladd drag correlation over all ranges of Reynolds number and solids volume fraction". In: *Powder Technology* 162, pp. 166–174. DOI: 10.1016/j.powtec.2005.12.014.
- Berendsen, H. J. C. (1999). *Molecular Dynamics Simulations: The Limits and Beyond*. Springer Berlin Heidelberg, pp. 3–36. ISBN: 978-3-642-58360-5.
- Camberos, J. et al. (2008). "On Direct Simulation Quasi-Monte Carlo Methods". In: *40th Thermophysics Conference*. DOI: 10.2514/6.2008-3915.
- Cengel, Y. et al. (2012). *Fundamentals of Thermal-Fluid Science*. McGraw-Hill Education. ISBN: 978-007-132511-0.
- Chen, X. and J. Wang (2014). "A comparison of two-fluid model, dense discrete particle model and CFD-DEM method for modeling impinging gas–solid flows". In: *Powder Technology* 254, pp. 94 –102. DOI: 10.1016/j.powtec.2013.12.056.
- DCS Computing (2019a). *CFDEM Documentation*. Visited May 30<sup>th</sup> 2019. URL: [www.cfdem.com/media/CFDEM/docu/CFDEMcoupling\\_Manual.html](http://www.cfdem.com/media/CFDEM/docu/CFDEMcoupling_Manual.html).
- (2019b). *LIGGGHTS Documentation*. Visited May 30<sup>th</sup> 2019. URL: [www.cfdem.com/media/DEM/docu/Manual.html](http://www.cfdem.com/media/DEM/docu/Manual.html).

- Ge, Wei and Jinghai Li (2001). "Macro-scale pseudo-particle modeling for particle-fluid systems". In: *Chinese Science Bulletin* 46, pp. 1503–1507. doi: 10.1007/BF02900568.
- He, H. et al. (2018). "Contrasting roles of speed on wear of soda lime silica glass in dry and humid air". In: *Journal of Non-Crystalline Solids* 502, pp. 236–243. doi: 10.1016/j.jnoncrysol.2018.09.014.
- Horwitz, J. and A. Mani (2016). "Accurate calculation of Stokes drag for point-particle tracking in two-way coupled flows". In: *Journal of Computational Physics* 318, pp. 85–109. doi: 10.1016/j.jcp.2016.04.034.
- Hærvig, J. et al. (2017). "On the adhesive JKR contact and rolling models for reduced particle stiffness discrete elements simulations". In: *Powder Technology* 319, pp. 472–482. doi: 10.1016/j.powtec.2017.07.006.
- Johnson, R. W. (1998). *The Handbook of Fluid Dynamics*. Springer Verlag. ISBN: 3-540-64612-4.
- Joseph, G. et al. (2001). "Particle-wall collisions in a viscous". In: *Journal of Fluid Mechanics* 433, pp. 329–346. doi: 10.1017/S0022112001003470.
- Kremer Pigmente (2019). *Glass Bubbles Material Safety Data Sheet*. Visited May 30<sup>th</sup> 2019. URL: [www.kremer-pigmente.com/media/pdf/59835MSDS.pdf](http://www.kremer-pigmente.com/media/pdf/59835MSDS.pdf).
- Li, Q. L. et al. (2011). "Discrete Element Analysis of Friction Coefficient on Granular Accumulation". In: *Mechatronics and Information Technology* 2, pp. 894–899. doi: 10.4028/www.scientific.net/AEF.2-3.894.
- Liu, L. et al. (2010). "Impact breakage of spherical, cuboidal and cylindrical agglomerates". In: *Powder Technology* 199, pp. 189–196. doi: 10.1016/j.powtec.2010.01.007.
- Loth, E. and A.J. Dorgan (2009). "An equation of motion for particles of finite Reynolds number and size". In: *Environmental Fluid Mechanics* 9, pp. 187–206. doi: 10.1007/s10652-009-9123-x.
- Marigo, M. and E. Stitt (2015). "Discrete Element Method (DEM) for Industrial Applications: Comments on Calibration and Validation for the Modelling of Cylindrical Pellets". In: *KONA Powder and Particle* 32, pp. 236–252. doi: 10.14356/kona.2015016.
- Marshall, J. S. and S. Li (2014). *Adhesive Particle Flow - A Discrete-Element Approach*. Cambridge University Press. ISBN: 978-1-107-03207-1.

- Mellmann, J. and T. Teodorov (2011). "Solids transport in mixed-flow dryers". In: *Powder Technology* 205, pp. 117–125. DOI: 10.1016/j.powtec.2010.08.073.
- Mellmann, J. et al. (2011). "Moisture content and residence time distributions in mixed-flow grain dryers". In: *Biosystems Engineering* 109, pp. 297–307. DOI: 10.1016/j.biosystemseng.2011.04.010.
- Mema, I. et al. (2019). "Effect of lift force and hydrodynamic torque on fluidisation of non-spherical particles". In: *Chemical Engineering Science* 195, pp. 642–656. DOI: 10.1016/j.ces.2018.10.009.
- Mohamad, A.A. (2011). *Lattice Boltzmann Method - Fundamentals and Engineering Applications with Computer Codes*. Springer Verlag London. ISBN: 978-0-85729-454-8.
- Mondal, S. et al. (2016). "Coupled CFD-DEM simulation of hydrodynamics bridging at constrictions". In: *International Journal of Multiphase Flow* 84, pp. 245–263. DOI: 10.1016/j.ijmultiphaseflow.2016.05.001.
- Moukalled, F. et al. (2016). *The Finite Volume Method in Computational Fluid Dynamics An Advanced Introduction with OpenFOAM® and Matlab*. Springer. ISBN: 978-3-319-16873-9.
- Neilands, K. et al. (2012). "Accumulation and modeling of particles in drinking water pipe fittings". In: *Drinking Water Engineering and Science* 5, pp. 47–57. DOI: 10.5194/dwes-5-47-2012.
- Norouzi, H. et al. (2016). *Coupled CFD-DEM Modelling Formulation, Implementation and Application to Multiphase Flows*. John Wiley & Sons. ISBN: 9781119005292.
- Olivieri, S. et al. (2014). "The Effect of the Basset History Force on Particle clustering in Homogeneous and Isotropic Turbulence". In: *Physics of Fluids* 26, pp. 1–7. DOI: 10.1063/1.4871480.
- Pal, S. et al. (2011). "Predicting Fly Ash Accumulation in a Fossil Power Plant Flue Gas Duct Using Computational Fluid Dynamics". In: *American Society of Mechanical Engineers, Power Division (Publication) POWER* 2, pp. 597–609. DOI: 10.1115/POWER2011-55077.
- Pöschel, T. and T. Schwager (2005). *Computational Granular Dynamics: Models and Algorithms*. Springer Berlin Heidelberg New York. ISBN: 978-3-540-21485-4.

- Renzo, A. D. and F. P. D. Maio (2005). "An Improved Integral Non-linear Model for the Contact of Particles in Distinct Element Simulations". In: *Chemical Engineering Science* 60, pp. 1303–1312. DOI: 10.1016/j.ces.2004.10.004.
- Roache, P. (1994). "Perspective: A Method for Uniform Reporting of Grid Refinement Studies". In: *Journal of Fluids Engineering* 116, pp. 405–413. DOI: 10.1115/1.2910291.
- Schiller, L. and A. Naumann (1933). "Über die grundlegenden Berechnungen bei der Schwerkraft Aufbereitung". In: *Z. Ver. Deutsch. Ing.* 77, pp. 318–320.
- ScienceDirect (2019). *Soda-Lime Glass*. Visited May 30<sup>th</sup> 2019. URL: [www.sciencedirect.com/topics/chemistry/soda-lime-glass](http://www.sciencedirect.com/topics/chemistry/soda-lime-glass).
- Shahzad, K. et al. (2016). "Numerical Investigation of Hard-gel Microparticle Suspension Dynamics in Microfluidic Channels: Aggregation/fragmentation Phenomena, and Incipient Clogging". In: *Chemical Engineering Journal* 303, pp. 202–216. DOI: 10.1016/j.cej.2016.05.134.
- Shahzad, K. et al. (2018). "Aggregation and Clogging Phenomena of Rigid Microparticles in Microfluidics - Comparison of a Discrete Element Method (DEM) and CFD-DEM Coupling Method". In: *Microfluidics and Nanofluidics* 22, pp. 104–121. DOI: 10.1007/s10404-018-2124-7.
- Spiga, M. and G. Morino (1994). "A Symmetric Solution for Velocity Profile in Laminar Flow through Rectangular Ducts". In: *International Communication in Heat and Mass Transfer* 21, pp. 469–475. DOI: 10.1016/0735-1933(94)90046-9.
- Sun, R. and H. Xiao (2016). "Sediment micromechanics in sheet flows induced by asymmetric waves: A CFD-DEM study". In: *Computers & Geosciences* 96, pp. 35–46. DOI: 10.1016/j.cageo.2016.07.007.
- Tebowei, R. et al. (2015). "Modelling effects of particle size and pipe gradient on sand transport in multiphase pipes". In: *WIT Transactions on Engineering Sciences* 89, pp. 323–334. DOI: 10.2495/MPF150281.
- Tsuji, Y. et al. (1993). "Discrete particle simulation of two-dimensional fluidized bed". In: *Powder Technology* 77, pp. 79–87. DOI: 10.1016/0032-5910(93)85010-7.



- van Buijtenen, M.S. et al. (2011). "Numerical and experimental study on multiple-spout fluidized beds". In: *Chemical Engineering Science* 66, pp. 2368–2376. doi: 10.1016/j.ces.2011.02.055.
- Versteeg, H.K. and W. Malalasekera (2007). *An Introduction to Computational Fluid Dynamics - The Finite Volume Method*. 2nd. Pearson Prentice Hall. ISBN: 978-0-13-127498-3.
- Yang, M. et al. (2015). "SPH-based Fluid Simulation with a New Surface Tension Formulation". In: *2015 International Conference on Virtual Reality and Visualization (ICVRV)*, pp. 295–300. doi: 10.1109/ICVRV.2015.34.
- Zhao, J. and T. Shan (2013). "Coupled CFD-DEM simulation of fluid-particle interaction in geomechanics". In: *Powder Technology* 239, pp. 248–258. doi: 10.1016/j.powtec.2013.02.003.
- Zu, E.X. et al. (2018). "Discrete Element Method of Coke Accumulation: Calibration of the Contact Parameter". In: *IFAC-PapersOnLine* 51. 5th IFAC Workshop on Mining, Mineral and Metal Processing MMM 2018, pp. 241–245. doi: 10.1016/j.ifacol.2018.09.425.



# Appendix A

## Contact Model

In this appendix the equations for the Hertz model for the frictional force implemented in LIGGGHTS are given. The material presented in this appendix is based on DCS Computing (2019b). The normal and tangential spring constants for the Hertz model frictional force are given by:

$$k_n = \frac{4}{3} E_{\text{eff}} \sqrt{r_{\text{eff}} \delta_n} \quad (\text{A.1})$$

$$k_t = 8 G_{\text{eff}} \sqrt{r_{\text{eff}} \delta_n} \quad (\text{A.2})$$

Where

$k$	Contact spring constant
$E_{\text{eff}}$	Effective Young's Modulus
$r_{\text{eff}}$	Effective particle radius
$\delta_n$	Displacement distance in the normal direction
$G_{\text{eff}}$	Effective shear modulus

The viscoelastic damping constant ( $\gamma$ ) for the normal and tangential directions are determined by:

$$\gamma_n = -2 \sqrt{\frac{5}{6}} \beta \sqrt{S_n m_{\text{eff}}} \leq 0 \quad (\text{A.3})$$

$$\gamma_t = -2 \sqrt{\frac{5}{6}} \beta \sqrt{S_t m_{\text{eff}}} \leq 0 \quad (\text{A.4})$$

Where

$\gamma$	Viscoelastic damping coefficient
$\beta$	A function of the coefficient of restitution
$S$	A function of Young's/shear modulus, effective radius and displacement distance

In Equation A.3 and A.4 the  $S$ -terms depend on Young's modulus, the shear modulus and the equivalent radius:

$$S_n = 2E_{\text{eff}}\sqrt{r_{\text{eff}}\delta_n} \quad (\text{A.5})$$

$$S_t = 8G_{\text{eff}}\sqrt{r_{\text{eff}}\delta_t} \quad (\text{A.6})$$

$\beta$  is a function of the coefficient of restitution ( $e$ ) and can be calculated using Equation A.7:

$$\beta = \frac{\ln(e)}{\sqrt{\ln(e)^2 + \pi^2}} \quad (\text{A.7})$$

Where

$e$	Coefficient of restitution
-----	----------------------------

The effective Young's modulus, shear modulus, mass and radius are given by (Marshall and Li, 2014):

$$E_{\text{eff}} = \left( \frac{1 - v_1^2}{E_1} + \frac{1 - v_2^2}{E_2} \right)^{-1} \quad (\text{A.8})$$

$$G_{\text{eff}} = \left( \frac{2 - v_1}{G_1} + \frac{2 - v_2}{G_2} \right)^{-1} \quad (\text{A.9})$$

$$r_{\text{eff}} = \left( \frac{1}{r_1} + \frac{1}{r_2} \right)^{-1} \quad (\text{A.10})$$

$$m_{\text{eff}} = \left( \frac{1}{m_1} + \frac{1}{m_2} \right)^{-1} \quad (\text{A.11})$$

Where

$v$	Poisson's ratio
-----	-----------------

# Appendix B

## Koch-Hill-Ladd Method

This appendix includes the equations of the Koch-Hill-Ladd method. The Koch-Hill-Ladd drag model determines the drag coefficient as (DCS Computing, 2019a):

$$C_D = K_{fp,D} V_p \quad (B.1)$$

Where

$K_{fp,D}$	The inter-phase momentum transfer coefficient if only drag force is applied to the particles.
$V_p$	Particle volume

The inter-phase momentum transfer coefficient due to drag is calculated by (van Buijtenen et al., 2011):

$$K_{fp,D} = \frac{18 \mu_f \alpha_f^2 \alpha_p}{D_p^2} \left( F_0(\alpha_p) + \frac{1}{2} F_3(\alpha_p) Re_p \right) \quad (B.2)$$

Where

$\mu_f$	Dynamic viscosity of the fluid
$\alpha_f, \alpha_p$	Void fraction of fluid and particle
$D_p$	Particle diameter
$F_0, F_3$	Force relations

The force relations ( $F_0$  and  $F_3$ ) from Equation B.2 are defined by (van Buijtenen et al., 2011) as a function of the particle void fraction:

$$F_0(\alpha_p) = \begin{cases} \frac{1 + 3 \sqrt{\frac{\alpha_p}{2}} + \frac{135}{64} \alpha_p \ln(\alpha_p) + 16.14 \alpha_p}{1 + 0.681 \alpha_p - 8.48 \alpha_p^2 + 8.16 \alpha_p^3} & \text{if } \alpha_p < 0.4 \\ \frac{10 \alpha_p}{\alpha_f^3} & \text{if } \alpha_p \geq 0.4 \end{cases} \quad (\text{B.3})$$

$$F_3(\alpha_p) = 0.0673 + 0.212 \alpha_p + \frac{0.0232}{\alpha_f^5} \quad (\text{B.4})$$

# Appendix C

## Mei Lift

The Mei lift combines the shear flow induced Saffman lift and the rotational induced Magnus lift. The equations for Mei lift in the CFD-DEM coupling are presented throughout this appendix. The equations without a source occurs from the source file (DCS Computing, 2019a).

The Mei lift force is calculated by:

$$F_L = 0.125 \pi C_L \frac{||\mathbf{u}_{rel}|| |\mathbf{u}_{rel}^\omega|}{||\boldsymbol{\omega}||} D_p^2 \quad (C.1)$$

Here the lift coefficient, which includes a second order terms, is approximated by (Loth and Dorgan, 2009):

$$C_L = J^* \frac{12.92}{\pi} \sqrt{\frac{\omega^*}{Re_p}} + \Omega_{p,eq}^* C_{L,\Omega}^* \quad (C.2)$$

Where

$C_L$	The lift coefficient
$J^*$	Extension of the Saffman lift eliminating $Re_p \ll Re_\omega^{1/2}$
$\omega^*$	The normalised vorticity with respect to particle diameter and relative velocity
$\Omega_{p,eq}^*$	Equation for predicting the equilibrium spin rate
$C_{L,\Omega}^*$	Lift coefficient for rotational induced lift (Magnus lift)

The extension of the Saffman lift ( $J^*$ ) depends on the parameter  $\varepsilon$ , which is defined by:

$$\varepsilon = \frac{2\sqrt{\nu_f ||\omega||}}{||\mathbf{u}_{rel}||} \quad (\text{C.3})$$

Thus, the extensions are defined by:

$$J^* \equiv \frac{C_{L,mag}}{C_{L,saff}} = \begin{cases} -140 \varepsilon \log_{10}(1/\varepsilon^2) & \text{if } \varepsilon < 0.1 \\ 1 - 0.287/\varepsilon^2 & \text{if } \varepsilon > 20 \end{cases} \quad (\text{C.4})$$

At any other condition, the  $J^*$  is approximated by (Loth and Dorgan, 2009):

$$J^* \equiv \frac{C_{L,mag}}{C_{L,saff}} \simeq 0.3 \left( 1 + \tanh \left[ \frac{5}{2} \log_{10} \left( \sqrt{\frac{\omega^*}{Re_p}} + 0.191 \right) \right] \right) \left( \frac{2}{3} + \tanh \left[ 6 \sqrt{\frac{\omega^*}{Re_p}} - 1.92 \right] \right) \quad (\text{C.5})$$

The equation for calculating the spin rate is given by (Loth and Dorgan, 2009):

$$\Omega_{p,eq}^* = \frac{\omega^*}{2} (1 - 0.0075 Re_\omega (1 - 0.062 \sqrt{Re_p} - 0.001 Re_p)) \quad (\text{C.6})$$

And the Magnus lift contribution is given by (Loth and Dorgan, 2009):

$$C_{L,mag}^* = 1 - \left( 0.675 + 0.15 \left( 1 + \tanh \left[ 0.28(\Omega_p^* - 2) \right] \right) \right) \tanh \left[ 0.18 \sqrt{Re_p} \right] \quad (\text{C.7})$$



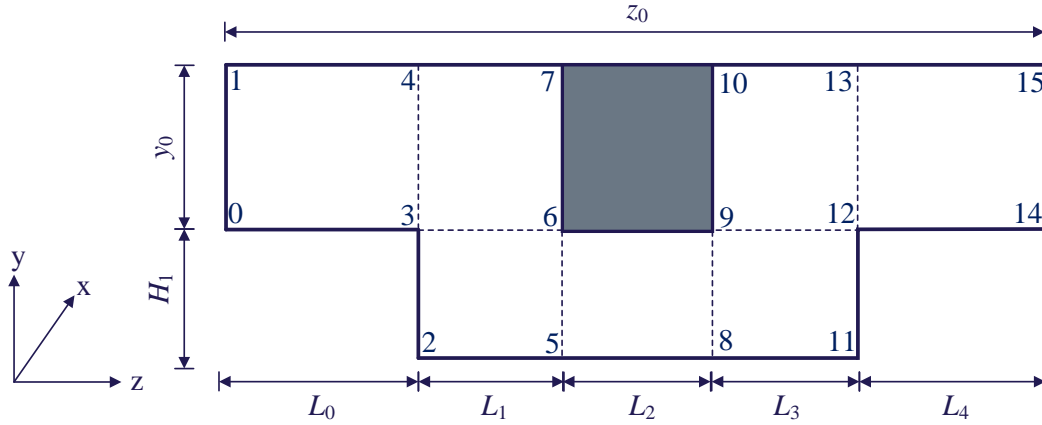
# Appendix D

## Generation of the Geometry

In this appendix an introduction to the generation of the geometry using *blockMesh* is given. First the meshing approach is described, after which the boundary conditions are explained along with the mesh topology.

### The Geometry

The geometry is illustrated in Figure D.1.



**Figure D.1:** Geometry is constructed from a total of 32 vertices. Vertices 0-15 in one  $y/z$ -plane and vertices 16-32 in a parallel plane with distance  $D_h$ .

A python script is used for generating a *blockMesh* file for OpenFOAM.

The geometry consists of 32 nodes called vertices, which are numbered from 0-31. *Origin* for the generation of the geometry is located at vertex 0. The geometry comprises of seven blocks, four for the channel and three for the cavity. Each block is defined by eight vertices and the  $x,y,z$ -directions are defined by the order of the vertices.

For the first block in Figure D.1, which consists of the vertices: 0, 1, 3, 4, 16, 17, 19, 20 the right order of defining the block is (0 16 17 1 3 19 20 4). Figure D.2 shows, how to write the code for defining the first block (seen from left) in the *blockMeshDict* in OpenFOAM.

$$\text{hex } (0 \ 16 \ 17 \ 1 \ 3 \ 19 \ 20 \ 4) \ \underbrace{(12 \ 12 \ 18)}_{\substack{x \quad y \quad z}} \ \text{simpleGrading} \ \underbrace{(1 \ 1 \ 1)}_{\substack{x \quad y \quad z}}$$

**Figure D.2:** How to define a block in OpenFOAM by using the vertices.

The *hex* states, that it is a hexagon. In the second entry the number of cells in the three directions is stated. In the third entry the cell expansion in all three directions is given, meaning the aspect ratio can be controlled. Since squared cells are favoured, the grading is set to 1 in all directions.

After the vertices and blocks have been specified, the boundary faces are defined. A face consists of four vertices, which are located at each corner, e.g. the vertices 0-3-4-1, which defines the outer wall pointing out of the paper for the first block. These vertices are defined *counter-clockwise* around an outwards-pointing normal vector of the given face. The internal faces are not defined, thereby the fluid is allowed to pass through them.

A Python script has been used to generate the geometry and an extended description along with the code can be found in Supplement S2.

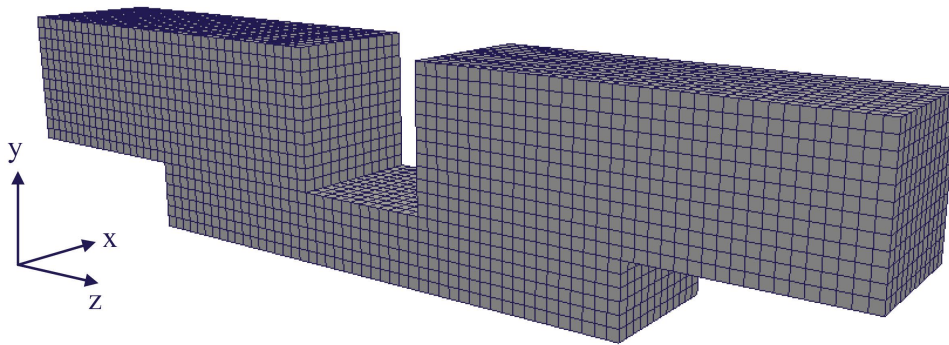
The geometry is meshed for calculating the fluid phase and the meshing is described in the next section.

## Meshing

The geometry is meshed manually and comprises entirely of perpendicular faces and quadratic cells. All cells are identical in size.

In *blockMesh* the number of cells is controlled by specifying the number of cells in the  $x,y,z$ -direction for each of the defined blocks. To ensure an equal cell size the length in a given direction is divided by the length of one cell.

The wall-adjacent cells are of same size as internal cells - no refinement of cells near walls are included. The meshing topology for the final mesh is seen in Figure D.3.



**Figure D.3:** View of the mesh topology.

The geometry consists of several of internal boundaries and the boundary condition are described in the following section.

## Boundaries

Fluid and particles can freely cross internal faces, while the walls are impenetrable and subject to no-slip condition. Fluid enters the geometry at inlet in a fully developed laminar velocity profile. In Appendix E it is described, how to implement a fully developed laminar flow profile in a square duct.

Fluid and particles can exit the geometry at outlet, which is defined as gauge pressure with no velocity gradient.



## Appendix E

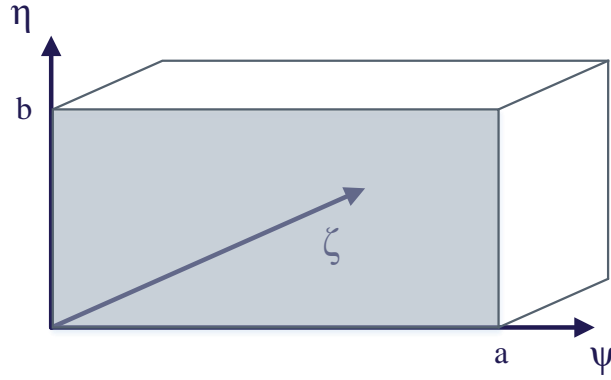
# Fully Developed Laminar Flow Profile

As a fully developed laminar flow profile is required for the flow before reaching the cavity, a long inlet section is needed for the flow to develop. To reduce the computational time, a fully developed laminar flow profile is applied at the inlet of the rectangular duct, so the total entrance length is significantly reduced.

Spiga and Morino (1994) have derived an expression for a laminar flow profile in a rectangular duct, which is based on Navier-Stokes equation for momentum with some assumptions. The fluid is assumed to be Newtonian and has constant fluid properties along with a constant pressure gradient in the longitudinal direction, thereby the Navier-Stokes equation for momentum in dimensionless form is reduced to:

$$\frac{\partial^2 \mathbf{u}_f}{\partial x^2} + \frac{\partial^2 \mathbf{u}_f}{\partial y^2} + 1 = 0 \quad (\text{E.1})$$

In Figure E.1 a sketch of a rectangular duct is seen along with the notation of height, length and depth of the duct. The notation is used for the following derivation.



**Figure E.1:** Sketch of arbitrary rectangular duct with the notation for the derivation.  $\psi$ ,  $\eta$ , and  $\zeta$  are the spacial coordinates.

For the derivation  $x$  and  $y$  are dimensionless coordinates defined as:

$$x = \psi/a \quad (0 \leq x \leq 1) \quad (\text{E.2})$$

$$y = \eta/a \quad (0 \leq y \leq \text{AR}) \quad (\text{E.3})$$

Where

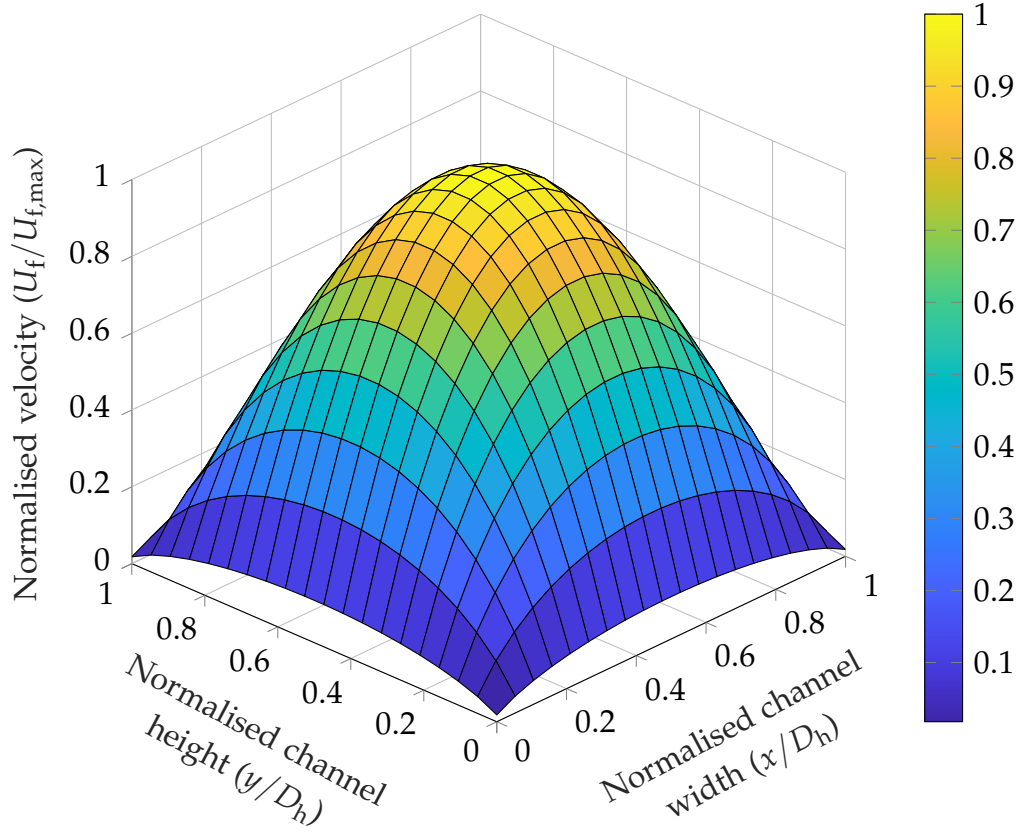
$\psi$	Axis coordinate
$a$	$\psi$ -position
$\eta$	Axis coordinate
AR	Aspect ratio $b/a$
$b$	$\eta$ -position

A no-slip boundary condition is not implemented in this calculation, since the output from this script is the velocity in the cell centres. In OpenFOAM a no-slip boundary condition is applied at the wall face.

By using a Fourier sine transformation to solve the partial differential equation in Equation E.1, the following equation for a symmetric fully developed laminar flow profile:

$$u(x, y) = \frac{16\text{AR}^2}{\pi^4} \sum_{l \text{ odd}}^{\infty} \sum_{o \text{ odd}}^{\infty} \frac{\sin\left(l\pi\frac{\psi}{a}\right) \sin\left(o\pi\frac{\eta}{b}\right)}{\ln\left(\text{AR}^2 l^2 + o^2\right)} \quad (\text{E.4})$$

The flow profile obtained by this equation is symmetric with respect to  $\psi$  and  $\eta$ . An example of the output of the equation is seen in Figure E.2, where the length and width of the duct is given in the  $x$  and  $y$ -direction and the velocity in the  $z$ -direction.



**Figure E.2:** The fully developed laminar flow profile for  $Re = 1300$  for the inlet of the geometry using Equation E.4.

In Supplement S3 the MATLAB script for generating the laminar flow profile is provided, which saves a table with the velocities at cell centres to a text-file in the right format to be read by OpenFOAM.

The inlet condition for the velocity is defined as a *fixedValue* with the value being a nonuniform list vector. Before listing all the values of the flow profile, it is important to specify the number of values to be read.

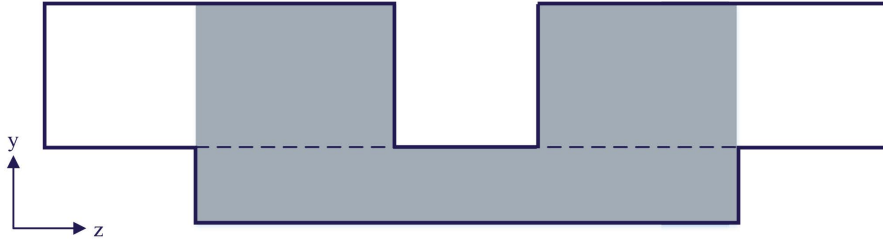




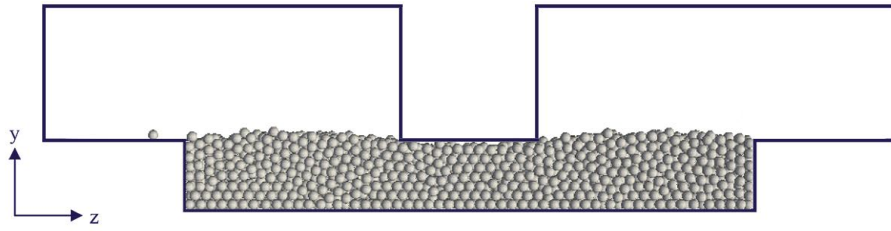
# Appendix F

## Insertion of Particles

The geometry includes a cavity, which is initially filled with particles. The filling region is highlighted by gray in Figure F.1 in which the particles are inserted randomly. The particles are not allowed to overlap with other particles. After insertion the particles are allowed to settle within the cavity during a given number of DEM iterations. If more particles are required, more are inserted. The random filling method does not ensure, that particles are perfectly packed, and some particles might be above the plane of the channel bottom, indicated by a dashed line in Figure F.1. An example of settled spheres in the domain is seen in Figure F.2.



**Figure F.1:** Particles are randomly inserted in the entire area highlighted by gray. After insertion, particles fall into the cavity due to gravity and settle, before any further simulations are conducted. Dashed line indicates plane of channel bottom.



**Figure F.2:** Example of particles after filling of domain. Some particles remain above the cavity border.

After filling of the domain, the DEM model can be coupled with the CFD model, to determine the effect of fluid passing through the channel.

# Appendix G

## Coupling Models

The CFD and DEM methodologies are coupled to a CFD-DEM model using different models, which are described in this appendix. The models are displayed in Table G.1.

*modelType* in the second row of Table G.1 refers to, if the pressure is assigned to both the fluid and particle phase or only the fluid phase. *Model A* defines, that the pressure is assigned to both phases.

*couplingInterval* determines, how often the DEM and CFD models are coupled.

The cells, where the particles are within, are determined by *locateModel*. The engine command defines, how the model searches for the particles. The model uses a tree search method, which locates the particle positions in a tree structure.

The mesh is fixed during simulations by setting *meshMotionModel* to noMesh-Motion.

*IOModel* determines, how files are saved and written in the CFD regime, here the basic model is used, which writes the velocities, positions and radii of the particles to the output file.

**Table G.1:** A list of the coupling models in the CFD-DEM model. The particle density is for a particle-to-fluid density ratio of 2.5.

Model	Type	Properties	Value
modelType	A		
couplingInterval			x
voidFractionModel	divided	alphaMin	0.1
		scaleUpVol	1.0
locateModel	engine	treeSearch	true
meshMotionModel	noMeshMotion		
IOModel	basicIO		
probeModel	off		
dataExchangeModel	twoWayMPI	liggghtsPath	../DEM/in.liggghts_resume
averagingModel	dense		
clockModel	off		
smoothingModel	constDiffSmoothing	lowerLimit	0.001
		upperLimit	$1 \cdot 10^{10}$
		smoothingLength	0.005
forceModels	KochHillDrag	velFieldName	U
		rhoParticle	2500
		voidfractionFieldName	voidfraction
		interpolation	true
		treatForceExplicit	false

*Table continues on next page*

*probeModel* is set to off, thus no probe is elected.

The data is exchanged by a two way MPI, meaning the CFD and the DEM are two way coupled.

*averagingModel* transfers velocities from the Lagrangian to the Eulerian framework. The dense function is added, since the flow is dense in the cavity.

*clockModel* is set to off.

*SmoothingModel* is set to constDiffSmoothing with a radius of 0.5 mm and is solved using a Gauss-Seidel solver. The smoothing model interpolates the coupling forces over a specified distance instead of assigning the forces to

the cell, where the particles center is located.

The momentum coupling is performed by an implicit couple, which is the most stable way of coupling. The momentum coupling utilises the forces calculated using the *forceModels* for calculating the  $K_{fp}$ .

The *turbulenceModelType* is defined in the turbulence properties. Since laminar flow is simulated, this is turned off.

Table continued.

Model	Type	Properties	Value
forceModels	gradPForce	pFieldName	p
		voidfractionFieldName	voidfraction
		velocityFieldName	U
		interpolation	true
	viscForce	treatForceExplicit	false
		velocityFieldName	U
		interpolation	true
		treatForceExplicit	false
	MeiLift	velFieldName	U
		useSecondOrderTerms	
		interpolation	true
		verbose	true
	virtualMassForce	treatForceExplicit	false
		velFieldName	U
		phiFieldName	phi
		treatForceExplicit	false
momCoupleModels	implicitCouple	splitUrelCalculation	true
		velFieldName	p
		granVelFieldName	Us
turbulenceModelType	turbulenceProperties	voidfractionFieldName	voidfraction

If the value column contains a letter, then it refers to a specific file defined in OpenFOAM. True or false indicates whether the property is on or off.



# Appendix H

## Grid Convergence Index

In this appendix the equations and theory for the Grid Convergence Index (GCI) are presented. The theory is based on Roache (1994).

The GCI approach is used for calculating the order of convergence in an objective way, and thereby estimate the asymptotic solution and finally the discretisation error. When applying the GCI, an estimation of the error from the simulated solution to the asymptotic solution, which is based on Richardson extrapolation, is found. Additionally, it shows, how a further refinement of the grid affects the numerical error. If many simulations are to be run, the GCI can also be used for calculating the error introduced by running the simulations on a coarser mesh. Thus, the additional error can be taken into consideration in the later data processing. It is assumed, that the discretisation error can be calculated using the following equation:

$$E = f(h) - f_{\text{exact}} = C h^p + \text{H.O.T} \quad (\text{H.1})$$

Where

$E$	Discretisation error
$f$	Parameter (grid dependent)
$C$	Constant
$h$	Grid size
$p$	Order of convergence
H.O.T	Higher Order Terms

Here the exact solution means a solution for a grid size ( $h$ ) approaching zero. From the numerical model a theoretical order of convergence is

given, but it is affected by various parameters such as boundary conditions, the numerical schemes and the grid size. By knowing three solutions for different grid size, the order of convergence ( $p$ ) from Equation H.1 can be calculated by:

$$p = \frac{\ln \frac{f_1 - f_2}{f_2 - f_3}}{\ln r} \quad (\text{H.2})$$

The parameters  $f$  must be grid dependent in order to estimate the effects of the grid refinement. The subscript (1) defines the parameter for the coarsest grid and (3) for the finest grid. The parameter  $r$  in Equation H.2 denotes the refinement ratio, which can be either an integer or non-integer, and it is defined as:

$$r = \frac{h_{i+1}}{h_i} \quad (\text{H.3})$$

Here the grid size ( $h$ ) can be either the grid size or the number of cells.

Even though the simulations may converge, it is important to check, if results from the simulations are within the asymptotic range of convergence. The asymptotic range indicates, if the simulated result is acceptable. If the result is outside the asymptotic range, the grid must be further refined, or else the GCI will not give useful information.

In the following equation the formula for calculating the asymptotic range of convergence is given by:

$$C = \frac{E}{h^p} \quad (\text{H.4})$$

Where the discretisation error is defined as:

$$E = \frac{\zeta}{r^p - 1} \quad (\text{H.5})$$

With  $\zeta$  being the relative error, which can be calculated using:

$$\zeta = \frac{f_{i+1} - f_i}{f_{i+1}} \quad (\text{H.6})$$

It is not sufficient to use the relative error to estimate, if the grid is fine enough, since it is not including the refinement ratio and the order of convergence, why it might be under- or overestimated.



When analysing three different grids, the asymptotic range of convergence can be calculated by the GCI, as given in Equation H.7:

$$\text{GCI}_{23} = \text{GCI}_{12} \cdot r^p \quad (\text{H.7})$$

By using this method a higher order estimate can be obtained using two lower order estimates. Therefore, by using two second order results from two simulations, a fourth order estimate is achieved.

For deriving the Richardson extrapolation to calculate the asymptotic solution, the Taylor series is written for a parameter  $f$  with a grid size  $h$  from a simulation, which is expressed by:

$$f = f_{h=0} + g_1 h + g_2 h^2 + g_3 h^3 + \dots \quad (\text{H.8})$$

Where the asymptotic solution is indicated by  $f_{h=0}$ .  $g$  is independent of the grid size. The parameters employed for calculating the asymptotic solution,  $f_{h=0}$ , must be results extracted at the same point or plane from the two simulations with different grids and representing the same parameter (e.g. pressure or friction coefficient). Assuming the two values are of second order, the  $g_1$  equals zero and the third- and higher order terms are neglected, which gives the following equations for the exact solution:

$$f_{h=0} \cong f_{i+1} + \frac{f_{i+1} - f_i}{r^p - 1} \quad (\text{H.9})$$

This equation is the generalised expression for a  $p^{\text{th}}$ -order of convergence.

The GCI can now be calculated, and if the error is small, it indicates that the results will not change much by refining the grid. The GCI is calculated using Equation H.10:

$$\text{GCI} = \frac{F_s \cdot |\zeta|}{r^p - 1} 100 \quad (\text{H.10})$$

$F_s$  is a safety factor, which is depending on the number of solutions used in the calculations of GCI. When two solutions are used, it is recommended to use a safety factor of 3.0. If three or more solutions are used, then the safety factor can be lowered to 1.25.



# Appendix I

## Experimental Work

This appendix includes the experimental work. First the Particle Image Velocimetry (PIV) experiment is described, providing information about the used equipment in the test setup and the post processing.

### Particle Image Velocimetry

This section explains the equipment used in the experimental setup. The object of the experiment is to check if the flow in the test setup is fully developed before reaching the cavity. For this investigation PIV is used. This test requires a laser emitter, a camera and seeding particles. Post processing is done using Dantec Dynamics *DynamicStudio* software.

### Laser emitter specifications

The specifications of the laser unit used for PIV is listed in Table I.1. The PIV tests are conducted using a constant laser sheet.

**Table I.1:** Laser unit specifications. Input voltage is given for the laser PSU.

Manufacturer	Model	Input	Output	Wavelength
CNI	MGL-W-532	220 V	0-5 W	532 nm

### Image capture specifications

The images are captured using an Basler acA1920-155um USB 3.0 camera. This camera uses a Sony IMX174 CMOS sensor capable of 164 fps and a resolution of 2.3 MP. A 60mm /1:2.8 Nikkor AF Micro lens is used.

The camera settings used for the image capture are listed in Table I.2. The capture area is marked by the box in Figure I.1, where it is seen, that the capture area covers the entire height of the water channel plus a few millimeters of the top and bottom of the channel.



**Figure I.1:** PIV image capture area is located before the cavity and at centre of the channel in the  $y$ -direction. The area is chosen to avoid interference from assembly bolts. The area covers the entire height of the channel plus a few millimeters of the channel material. The image capture frame is for conceptual understanding and not to actual scale.

**Table I.2:** Settings used for image capture.

Aperture	Sample size	Sample rate
2.8	600+ images	150 Hz

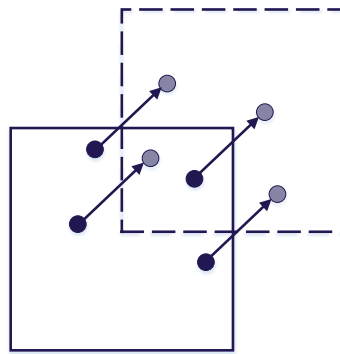
### Seeding particles

The seeding particles used for PIV are PSP-5 polyamid particles that have a mean particle diameter of 5  $\mu\text{m}$ .

After conducting the experiments, the data must be post processed. The PIV post processing methodology is presented in the following section.

## PIV Post Processing

The captured images are post processed using the Adaptive Correlation method in *DynamicStudio*, which uses an iterative method to adapt the interrogation areas (IA's) to fit the velocity vectors. The methodology is illustrated in Figure I.2.



**Figure I.2:** The adaptive correlation methodology.

The adaptive correlation uses large IA's to produce smaller IA's iteratively.

**Table I.3:** Adaptive correlation settings.

Setting	Value
Initial IA	64x256
Final IA	16x64
Iteration steps	2
Minimum peak height ratio	1.25
Local neighbourhood validation	Moving average
Acceptance factor	0.2
IA Offset	Central difference

The minimum peak height ratio determines the difference in peak height between the highest peak and the second highest peak. This value is introduced to ensure a global maximum. In this project the ratio is chosen to 1.25 which means that the highest peak is minimum 25% higher than the second highest. By electing a high ratio, it is assured that it is the global maximum, which is found in adaptive correlation.

The local neighbourhood validation is used for finding spurious vectors by comparing the local cell value to the neighbouring cells. If the data is spurious, the vector will be replaced by an interpolated vector. The interpolated vector is found from the values in the neighbouring cells. The spurious vectors are found from the defined acceptance factor (in this case 0.2). For low acceptance factors all vectors are replaced, whereas for higher factors all vector remain uncorrected.

The central difference scheme is used for determining the displacement of the particles, and chosen since it has the highest accuracy.

After the adaptive correlation is conducted, vector statistics are generated. The vector statistics are generated from the vector maps created by the adapted correlation. The mean velocities and directions of the velocity is presented in the vector statistics. Furthermore, the statistical parameters such as the standard deviation, variance and covariance can be extracted from the vector statistics.

The vector statistics allows for different data representations, in this project the flow will be represented by a velocity profile.

# **Supplement S1**

## **Data Sheet of the Glass Particles**

This supplement includes a data sheet of the glass particles used in the project, which are manufactured by Kremer Pigmente GmbH & Co.

## 59821 - 59822 Glass Beads

These solid glass beads are manufactured of soda-lime glass in a specially developed process and then provided with a gliding cut polish which does not damage the material and is environment friendly.

The development of this type of bead is primarily based **on its use as grinding media in pearl mills** for grinding and dispersing color pigments, paint, ink, agro chemicals, pharmaceuticals, minerals, magnetic coatings and fillers for paper and plastic industry.

### Further applications:

- As an agent used for mechanical plating (electroplating) for small metallic parts,
- Mixed elements in atomizers,
- Reflector beads > 1 mm for road markings - particularly to increase visibility at night by rain,
- Filling media in electronics for protection against explosion stirring media in aerosols,
- Polishing elements in the optical industry.

### Chemical Characterization:

SiO <sub>2</sub>	72 %
Na <sub>2</sub> O	13 %
CaO	9 %
MgO	4 %
Al <sub>2</sub> O <sub>3</sub>	1 %
K <sub>2</sub> O; Fe <sub>2</sub> O <sub>3</sub>	1 %

Deviations of +/- 1 % are possible.

**PbO free < 0.01 %**

### Approximate technical and physical values:

Hardness according to Mohs:	≥ 6
Elasticity module:	63 GPa
Specific weight:	2.5 kg/dm <sup>3</sup>
Bulk weight:	1.5 kg/dm <sup>3</sup>
Abrasion after 100 h grinding:	1.5 – 1.8 %

Size range	
Product No. 59822:	0.00 – 0.05 mm
Product No. 59821:	0.04 – 0.07 mm

Subject to the modification of technical details.



# Supplement S2

## Python Block Mesh Script

This supplement includes the code for generating the *blockMesh* for OpenFOAM. First the lengths, heights and width are written *mm* followed by the number steps, vertices per repetition and cells in the three directions. This is specified from line 14-27.

Then the vertices in the first *x*-plane are defined using the specified lengths from above (line 29-47). From line 49-56 the 16 defined vertices are duplicated with an altered *x*-coordinate matching the width of the channel and the same *y,z*-coordinates.

In line 70-79 the blocks are specified as explained in the sections above and shown in Figure D.2 on page 124. This is done for all seven blocks. The next part of the script from line 87-136 defines the walls, in- and outlet.

The final part of the script generates the text-document with the right text and commands and inserts the listed number of the vertices in the right order. The python script for generating the blockMesh file is shown below:

```
1  #!/usr/bin/env python
2  # -*- coding: utf-8 -*-
3
4  """case.py: outputs a blockMeshDict file to be used for OpenFOAM"""
5  __author__      = "TEPE-1009"
6  __email__      = ""
7
8  from OpenFoamDataWriter import OpenFoamDataWriter
9  import os
10 import numpy as np
11
12 class pyMesh(object): #class definition
13     def __init__(self):
14         self._L0 = 30 #parameter 0 (z-direction)
```

```

15     self._L1 = 30 #parameter 1 (z-direction)
16     self._L2 = 20 #parameter 2 (z-direction)
17     self._L3 = self._L1 #parameter 3 (z-direction)
18     self._L4 = 60 #parameter 4 (z-direction)
19     self._H1 = 10 #parameter 3 (y-direction)
20     self._H2 = 20 #parameter 4 (y-direction)
21     self._W = 20 #Width (x-direction)
22
23     self._nStep = 1 #number or repetitions
24     self._nVertices = 20 #number of vertices per repetition
25     self._nCellsx = 10
26     self._nCellsy = 10
27     self._nCellsz = 10
28
29     def _create_one_level_vertices(self, x, z0): #define initial vertices (knudepunkter)
30         return [
31             (x, 0, z0 + 0), #0
32             (x, self._H2, z0 + 0), #1
33             (x, -self._H1, z0+ self._L0), #2
34             (x, 0, z0+self._L0), #3
35             (x, self._H2, z0+self._L0), #4
36             (x, -self._H1, z0+self._L0 + self._L1), #5
37             (x, 0, z0+self._L0 + self._L1), #6
38             (x, self._H2, z0+self._L0 + self._L1), #7
39             (x, -self._H1, z0+self._L0 + self._L1 + self._L2), #8
40             (x, 0, z0+self._L0 + self._L1 + self._L2), #9
41             (x, self._H2, z0+self._L0 + self._L1 + self._L2), #10
42             (x, -self._H1, z0+self._L0 + self._L1 + self._L2+self._L3), #11
43             (x, 0, z0+self._L0 + self._L1 + self._L2+self._L3), #12
44             (x, self._H2, z0+self._L0 + self._L1 + self._L2+self._L3), #13
45             (x, 0, z0+self._L0 + self._L1 + self._L2+self._L3+self._L4), #14
46             (x, self._H2, z0+self._L0 + self._L1 + self._L2+self._L3+self._L4), #15
47         ]
48
49     def _create_vertices_data(self): #loop to create all vertices by calling "def
        _create_one_level_vertices"
50         points_array = []
51         for step in range(self._nStep):
52             z0 = step*(self._L0+self._L1+self._L2)
53             for x in [0, self._W]:
54                 points_array = points_array + self._create_one_level_vertices(x, z0)
55             self._points_array = points_array
56         return points_array
57
58     def _reuse_points(self, points_list, points_array): #reduce the array of vertices by
        concentratating vertices with exact same position
59         new_points_list = []
60         for test in points_list:
61             val_list = []
62             for counter,value in enumerate(points_array):
63                 if points_array[test] == value:
64                     val_list.append(counter)
65             max_index = max(val_list)
66             new_points_list.append(max_index)

```

```

67     new_points_list = ' '.join(map(str, new_points_list))
68     return new_points_list
69
70 def _create_one_level_blocks(self, step): #create one level of blocks
71     return [
72         "hex {} {} {} {} {} simpleGrading {} {}
           {}".format(self._reuse_points([0+self._nVertices*step,
           16+self._nVertices*step, 17+self._nVertices*step, 1+self._nVertices*step,
           3+self._nVertices*step, 19+self._nVertices*step, 20+self._nVertices*step,
           4+self._nVertices*step], self._points_array), self._nCellsx, self._nCellsy,
           self._nCellsz, 1, 1, 1), #0
73         "hex {} {} {} {} {} simpleGrading {} {}
           {}".format(self._reuse_points([2+self._nVertices*step,
           18+self._nVertices*step, 19+self._nVertices*step,
           3+self._nVertices*step, 5+self._nVertices*step, 21+self._nVertices*step,
           22+self._nVertices*step, 6+self._nVertices*step], self._points_array),
           self._nCellsx, self._nCellsy, self._nCellsz, 1, 1, 1), #1
74         "hex {} {} {} {} {} simpleGrading {} {}
           {}".format(self._reuse_points([3+self._nVertices*step,
           19+self._nVertices*step, 20+self._nVertices*step, 4+self._nVertices*step,
           6+self._nVertices*step, 22+self._nVertices*step, 23+self._nVertices*step,
           7+self._nVertices*step], self._points_array), self._nCellsx, self._nCellsy,
           self._nCellsz, 1, 1, 1), #2
75         "hex {} {} {} {} {} simpleGrading {} {}
           {}".format(self._reuse_points([5+self._nVertices*step,
           21+self._nVertices*step, 22+self._nVertices*step, 6+self._nVertices*step,
           8+self._nVertices*step, 24+self._nVertices*step, 25+self._nVertices*step,
           9+self._nVertices*step], self._points_array), self._nCellsx, self._nCellsy,
           self._nCellsz, 1, 1, 1), #3
76         "hex {} {} {} {} {} simpleGrading {} {}
           {}".format(self._reuse_points([8+self._nVertices*step,
           24+self._nVertices*step, 25+self._nVertices*step, 9+self._nVertices*step,
           11+self._nVertices*step, 27+self._nVertices*step, 28+self._nVertices*step,
           12+self._nVertices*step], self._points_array), self._nCellsx, self._nCellsy,
           self._nCellsz, 1, 1, 1), #5
77         "hex {} {} {} {} {} simpleGrading {} {}
           {}".format(self._reuse_points([9+self._nVertices*step,
           25+self._nVertices*step, 26+self._nVertices*step, 10+self._nVertices*step,
           12+self._nVertices*step, 28+self._nVertices*step, 29+self._nVertices*step,
           13+self._nVertices*step], self._points_array), self._nCellsx, self._nCellsy,
           self._nCellsz, 1, 1, 1), #6
78         "hex {} {} {} {} {} simpleGrading {} {}
           {}".format(self._reuse_points([12+self._nVertices*step,
           28+self._nVertices*step, 29+self._nVertices*step, 13+self._nVertices*step,
           14+self._nVertices*step, 30+self._nVertices*step, 31+self._nVertices*step,
           15+self._nVertices*step], self._points_array), self._nCellsx, self._nCellsy,
           self._nCellsz, 1, 1, 1), #7
79     ]
80
81 def _create_blocks_data(self): #loop to create all vertices by calling "def
    _create_one_level_blocks"
82     blocks_array = []
83     for step in range(self._nStep):
84         blocks_array = blocks_array + self._create_one_level_blocks(step)

```

```

85         return blocks_array
86
87     def _create_patches_data_left(self): #Inlet
88         return [
89             "{}".format(self._reuse_points([0, 16, 17, 1], self._points_array)),
90         ]
91
92     def _create_patches_data_right(self): #Outlet
93         return [
94             "{}".format(self._reuse_points([14+self._nVertices*(self._nStep-1),
95                                             15+self._nVertices*(self._nStep-1), 31+self._nVertices*(self._nStep-1),
96                                             30+self._nVertices*(self._nStep-1)], self._points_array)),
97         ]
98
99     def _create_patches_data_walls(self, step): #Walls
100         return [
101             "{}".format(self._reuse_points([step*self._nVertices+0, step*self._nVertices+1,
102                                             step*self._nVertices+4, step*self._nVertices+3], self._points_array)), #Wall 0
103             "into paper",
104             "{}".format(self._reuse_points([step*self._nVertices+20,
105                                             step*self._nVertices+17, step*self._nVertices+16, step*self._nVertices+19],
106                                             self._points_array)), #Wall 0 out of paper",
107             "{}".format(self._reuse_points([step*self._nVertices+2, step*self._nVertices+3,
108                                             step*self._nVertices+6, step*self._nVertices+5], self._points_array)), #Wall 1
109             "into paper",
110             "{}".format(self._reuse_points([step*self._nVertices+18,
111                                             step*self._nVertices+21, step*self._nVertices+22, step*self._nVertices+19],
112                                             self._points_array)), #Wall 1 out of paper",
113             "{}".format(self._reuse_points([step*self._nVertices+3, step*self._nVertices+4,
114                                             step*self._nVertices+7, step*self._nVertices+6], self._points_array)), #Wall 2
115             "into paper",
116             "{}".format(self._reuse_points([step*self._nVertices+20,
117                                             step*self._nVertices+19, step*self._nVertices+22, step*self._nVertices+23],
118                                             self._points_array)), #Wall 2 out of paper",
119             "{}".format(self._reuse_points([step*self._nVertices+5, step*self._nVertices+6,
120                                             step*self._nVertices+9, step*self._nVertices+8], self._points_array)), #Wall 3
121             "into paper",
122             "{}".format(self._reuse_points([step*self._nVertices+22,
123                                             step*self._nVertices+21, step*self._nVertices+24, step*self._nVertices+25],
124                                             self._points_array)), #Wall 3 out of paper",
125             "{}".format(self._reuse_points([step*self._nVertices+8, step*self._nVertices+9,
126                                             step*self._nVertices+12, step*self._nVertices+11], self._points_array)), #Wall 5
127             "out of paper",
128             "{}".format(self._reuse_points([step*self._nVertices+25,
129                                             step*self._nVertices+24, step*self._nVertices+27, step*self._nVertices+28],
130                                             self._points_array)), #Wall 5 into paper",
131             "{}".format(self._reuse_points([step*self._nVertices+9, step*self._nVertices+10,
132                                             step*self._nVertices+13, step*self._nVertices+12], self._points_array)), #Wall 6
133             "out of paper",
134             "{}".format(self._reuse_points([step*self._nVertices+26,
135                                             step*self._nVertices+25, step*self._nVertices+28, step*self._nVertices+29],
136                                             self._points_array)), #Wall 6 into paper",
137             "{}".format(self._reuse_points([step*self._nVertices+12,
138                                             step*self._nVertices+13, step*self._nVertices+15, step*self._nVertices+14],

```

```

self._points_array)), #Wall 7 out of paper
112 " ({}).format(self._reuse_points([step*self._nVertices+29,
step*self._nVertices+28, step*self._nVertices+30, step*self._nVertices+31],
self._points_array)), #Wall 7 into paper
113 " ({}).format(self._reuse_points([step*self._nVertices+7, step*self._nVertices+6,
step*self._nVertices+22, step*self._nVertices+23], self._points_array)), #Block
114 " ({}).format(self._reuse_points([step*self._nVertices+10,
step*self._nVertices+26, step*self._nVertices+25, step*self._nVertices+9],
self._points_array)), #Block
115 " ({}).format(self._reuse_points([step*self._nVertices+6, step*self._nVertices+9,
step*self._nVertices+25, step*self._nVertices+22], self._points_array)), #Block
bottom
116 ]
117
118 def _create_patches_data_bottom(self, step): #Bottom
119     return [
120         " ({}).format(self._reuse_points([step*self._nVertices+0, step*self._nVertices+3,
step*self._nVertices+19, step*self._nVertices+16], self._points_array)), #Bottom 0
121         " ({}).format(self._reuse_points([step*self._nVertices+2, step*self._nVertices+5,
step*self._nVertices+21, step*self._nVertices+18], self._points_array)), #Bottom 1
122         " ({}).format(self._reuse_points([step*self._nVertices+5, step*self._nVertices+8,
step*self._nVertices+24, step*self._nVertices+21], self._points_array)), #Bottom 3
123         " ({}).format(self._reuse_points([step*self._nVertices+8, step*self._nVertices+11,
step*self._nVertices+27, step*self._nVertices+24], self._points_array)), #Bottom
side left
124         " ({}).format(self._reuse_points([step*self._nVertices+12,
step*self._nVertices+14, step*self._nVertices+30, step*self._nVertices+28],
self._points_array)), #Bottom side right
125         " ({}).format(self._reuse_points([step*self._nVertices+3, step*self._nVertices+2,
step*self._nVertices+18, step*self._nVertices+19], self._points_array)), #Bottom
side right
126         " ({}).format(self._reuse_points([step*self._nVertices+11,
step*self._nVertices+12, step*self._nVertices+28, step*self._nVertices+27],
self._points_array)), #Bottom side right
127     ]
128
129
130 def _create_patches_data_top(self, step): #Top
131     return [
132         " ({}).format(self._reuse_points([step*self._nVertices+1, step*self._nVertices+17,
step*self._nVertices+20, step*self._nVertices+4], self._points_array)), #Top 0
133         " ({}).format(self._reuse_points([step*self._nVertices+4, step*self._nVertices+20,
step*self._nVertices+23, step*self._nVertices+7], self._points_array)), #Top 2
134         " ({}).format(self._reuse_points([step*self._nVertices+10,
step*self._nVertices+26, step*self._nVertices+29, step*self._nVertices+13],
self._points_array)), #Top 6
135         " ({}).format(self._reuse_points([step*self._nVertices+13,
step*self._nVertices+29, step*self._nVertices+31, step*self._nVertices+15],
self._points_array)), #Top 7
136     ]
137
138 def _create_patches_data(self):
139     patches_array = []
140

```

```

141 patches_array = patches_array + ["inlet", "{", "type patch;", "faces", "("]
142 patches_array = patches_array + self._create_patches_data_left()
143 patches_array = patches_array + [");", "}"]
144
145 patches_array = patches_array + ["outlet", "{", "type patch;", "faces", "("]
146 patches_array = patches_array + self._create_patches_data_right()
147 patches_array = patches_array + [");", "}"]
148
149 patches_array = patches_array + ["walls", "{", "type wall;", "faces", "("]
150 for step in range(self._nStep):
151     patches_array = patches_array + self._create_patches_data_walls(step)
152 patches_array = patches_array + [");", "}"]
153
154 patches_array = patches_array + ["bottom", "{", "type wall;", "faces", "("]
155 for step in range(self._nStep):
156     patches_array = patches_array + self._create_patches_data_bottom(step)
157 patches_array = patches_array + [");", "}"]
158
159 patches_array = patches_array + ["top", "{", "type wall;", "faces", "("]
160 for step in range(self._nStep):
161     patches_array = patches_array + self._create_patches_data_top(step)
162 patches_array = patches_array + [");", "}"]
163
164 return patches_array

```

# Supplement S3

## Laminar Flow Profile Script

This supplement includes the script of how to generate the laminar flow profile in the a square duct used as inlet condition for the model. The code is shown for  $Re_f = 1300$ .

```
1  %%% Script for calculating the velocity profile for OpenFOAM
2  %%% d. 16-02-2019
3  %% Specifying the dimensions of the duct
4  % Remember to adjust the Reynolds number before running the script
5  clc;
6  clear all;
7  close all;
8  %%
9  directory = 'C:\Users\anden\Dropbox (Aalborg Universitet)\P10\Scripts\MATLAB2tikz';
10 addpath(genpath(directory));
11 %%
12 a = 0.02; % [m]
13 b = 0.02; % [m]
14 beta = b/a;
15
16 ndiv = 20; % Number of dividings
17 Ndiv = ndiv*2+1; % steps to reach cell centers
18 countx = 0;
19 Ndiv_x = linspace(0,1,Ndiv);
20 x_1 = Ndiv_x(1,2);
21 x_2 = Ndiv_x(1,length(Ndiv_x)-1);
22
23 %% Calculating the theoretical flow profile
24 for x = linspace(x_1,x_2,ndiv) %x
25     countx = countx +1;
26     county = 0;
27     for y = linspace(x_1,x_2,ndiv) %y
28         county = county +1;
29         U = 0;
30         for n = 1:2:ndiv
31             for m = 1:2:ndiv
```

```

32         U = U + (16*(beta^2)/(pi^4))*(sin(n*pi*x)*sin(m*pi*y))/(n*m*((beta^2)*n^2 +
33             m^2));
34     end
35     U_tot(countx, county) = U;
36 end
37
38 end
39
40
41 %% Scaling the velocity
42 % Remember to adjust the Reynolds number before running the script
43 nu = 1.0023e-6;           %Kinematic viscosity of water @ 20 degrees
44 Dh = a;                   % [m]
45 Re = 1300;                % Desired Reynolds number
46
47 Uavg = Re * nu/Dh;
48 U_Re = Uavg * 2;          % Maximum velocity of laminar flow profile
49 Umax = max(max(U_tot));   % Max. value for the 3D profile
50 ratio = U_Re/Umax;        % Ratio
51 Unew = U_tot*ratio;       % New scaled velocity

```



# Supplement S4

## Initiation Script

This supplement includes the initiation script for the CFD-DEM model. The script inserts the particles in the cavity region.

```
1  echo          both
2  log            ../DEM/log.liggghts
3  thermo_log     ../DEM/post/thermo.txt
4
5  #----- LOAD INPUTS -----
6  include in.Variables
7
8  #----- GENERAL SETTING -----
9  atom_style      granular
10 atom_modify     map array
11 communicate     single vel yes
12
13 boundary        f f f
14 newton          off
15 units           si
16 processors       2 1 2
17
18 region          reg block ${xMin} ${xMax} ${yMin} ${yMax} ${zMin} ${zMax} units box
19 create_box      2 reg
20 neighbor        ${radius} bin
21 neigh_modify    delay 0
22
23 #----- MATERIAL PROPERTIES -----
24 fix             m1 all property/global youngsModulus peratomtype ${youngsMod} ${youngsWall}
25 fix             m2 all property/global poissonsRatio peratomtype ${poissonsP} ${poissonW}
26 fix             m3 all property/global coefficientRestitution peratomtypepair 2 ${coeffRestPP}
   ${coeffRestPW} ${coeffRestPW} ${coeffRestWW}
27 fix             m4 all property/global coefficientFriction peratomtypepair 2 ${coeffFricPP}
   ${coeffFricPW} ${coeffFricPW} ${coeffFricWW}
28 fix             m5 all property/global coefficientRollingFriction peratomtypepair 2 ${rollPP}
   ${rollPW} ${rollPW} ${rollWW}
29
```

```

30 #----- CONTACT MODEL -----
31 pair_style gran model hertz tangential history rolling_friction cdt
32 pair_coeff * *
33
34 #----- GRAVITY SETTINGS / TIMESTEP -----
35 timestep    ${dt}
36 fix        gravi all gravity 9.81 vector 0.0 -1.0 0.0
37
38 #----- DEFINITION OF WALLS (SETTING WALL TYPE) -----
39 fix zwalls2 all wall/gran model hertz tangential history rolling_friction cdt primitive type
40           2 zplane ${zMin}
41 fix zwalls3 all wall/gran model hertz tangential history rolling_friction cdt primitive type
42           2 xplane ${xMax}
43 fix zwalls4 all wall/gran model hertz tangential history rolling_friction cdt primitive type
44           2 xplane ${xMin}
45 fix zwalls5 all wall/gran model hertz tangential history rolling_friction cdt primitive type
46           2 yplane ${yMax}
47 fix zwalls6 all wall/gran model hertz tangential history rolling_friction cdt primitive type
48           2 yplane ${yMin}
49
50 #----- CAD GEOMETRY -----
51 fix      cad all mesh/surface file mesh/geometry.stl type 2 scale 1
52 fix      cadWall all wall/gran model hertz tangential history rolling_friction cdt mesh
53           n_meshes 1 meshes cad
54
55 #----- PARTICLE INSERTION -----
56 region    bc block 0 0.02 -0.01 0.02 0.032 0.108 units box
57 fix      pts1 all particletemplate/sphere 15485863 atom_type 1 density constant ${rho}
58           radius constant ${radius}
59 fix      pdd1 all particledistribution/discrete 15485867 1 pts1 1.0
60 fix      ins all insert/pack seed 123457 distributiontemplate pdd1 vel constant 0.0 -1.0
61           0.0 insert_every once overlapcheck yes all_in yes particles_in_region ${nPart} region bc
62
63 #----- INTEGRATION -----
64 # apply nve integration to all particles that are inserted as single particles
65 fix      integr all nve/sphere
66
67 #----- OUTPUT -----
68 dump      dmp all custom/vtk 10000 post/particles_*.vtk id type x y z vx vy vz
69 dump      dumpstress all mesh/gran/VTK 10000 post/mesh_*.vtk stress wear cad
70
71 #----- INSERTION OF FIRST PARTICLES -----
72 run      400000
73 write_restart post/restart/liggghts.restart

```

# Supplement S5

## Main CFD-DEM Script

This supplement includes the main CFD-DEM script, where the fluid and particulate phases are coupled. The script is for the DEM part of the model.

```
1  echo          both
2  log            ../DEM/log.liggghts
3  thermo_log     ../DEM/post/thermo.txt
4
5  #-----Load variables-----
6  include        in.Variables
7
8  #----- INITIAL SETTINGS -----
9  atom_style     granular
10 atom_modify    map array
11 communicate    single vel yes
12 boundary       f f f
13 newton         off
14 units          si
15 soft_particles  yes          #For youngs modulus BELOW 5e6
16 #hard_particles yes          #for youngs modulus ABOVE 1e9
17 processors      2 1 2
18 read_restart   ../DEM/post/restart/liggghts.restart
19 neighbor        ${radius} bin
20 neigh_modify    delay 0
21
22 #----- MATERIAL PROPERTIES -----
23 fix            m1 all property/global youngsModulus peratomtype ${youngsMod} ${youngsWall}
24 fix            m2 all property/global poissonsRatio peratomtype ${poissonsP} ${poissonW}
25 fix            m3 all property/global coefficientRestitution peratomtypepair 2 ${coeffRestPP}
   ${coeffRestPW} ${coeffRestPW} ${coeffRestWW}
26 fix            m4 all property/global coefficientFriction peratomtypepair 2 ${coeffFricPP}
   ${coeffFricPW} ${coeffFricPW} ${coeffFricWW}
27 fix            m5 all property/global coefficientRollingFriction peratomtypepair 2 ${rollPP}
   ${rollPW} ${rollPW} ${rollWW}
28
29 #----- CONTACT MODEL -----
```

```

30 pair_style      gran model hertz tangential history rolling_friction cdt
31 pair_coeff      * *
32
33 #----- GRAVITY SETTINGS / TIMESTEP -----
34 timestep        ${dt}
35 fix             gravi all gravity 9.81 vector 0.0 -1.0 0
36
37 #----- CAD GEOMETRY -----
38 fix             cad all mesh/surface file ../DEM/mesh/new_geometry.stl type 2 scale 1
39 fix             cadWall all wall/gran model hertz tangential history rolling_friction cdt
              mesh n_meshes 1 meshes cad
40
41 #----- CFDEM COUPLING -----
42 fix             cfd all couple/cfd couple_every 1000 mpi
43 fix             cfd2 all couple/cfd/force/implicit CAddRhoFluid 0.4
44
45 #----- INTEGRATION -----
46 fix             integr all nve/sphere
47
48 #----- CALCULATIONS -----
49 compute         rke all erotate/sphere
50 thermo_style    custom step atoms ke c_rke vol
51 thermo         0
52 thermo_modify   lost ignore norm no
53 compute_modify  thermo_temp dynamic yes
54
55 dump            dmp all custom/vtk 100000 post/particles_*.vtk id type x y z vx vy vz
56 dump            dumpstress all mesh/gran/VTK 100000 post/mesh_*.vtk stress wear cad
57 run            1

```