An extended multiphase drift-flux model with a discretized form of population balance equations for flocculation and settling process of cohesive sediment in deep-sea mining plumes

Ziyang Huang

$$\frac{\partial \alpha_k}{\partial t} + \nabla \cdot (\alpha_k \boldsymbol{u_k}) = (\sum_{j=1}^{k-2} 2^{j-k+1} A_{k-1,j} \beta_{k-1,j} N_{k-1} N_j + \frac{1}{2} A_{k-1,k-1} \beta_{k-1,k-1} N_{k-1}^2 - \sum_{j=1}^{k-1} 2^{j-k} A_{k,j} \beta_{k,j} N_k N_j - \sum_{j=k}^{k_{max}} A_{k,j} \beta_{k,j} N_k N_j - S_k N_k + \sum_{j=k+1}^{k_{max}} \Gamma_{k,j} S_j N_j) v_k$$

TU Delft

AN EXTENDED MULTIPHASE DRIFT-FLUX MODEL WITH A DISCRETIZED FORM OF POPULATION BALANCE EQUATIONS FOR FLOCCULATION AND SETTLING PROCESS OF COHESIVE SEDIMENT IN DEEP-SEA MINING PLUMES

by

Ziyang HUANG

to obtain the degree of Master of Science in Offshore & Dredging Engineering
at the Delft University of Technology

Thesis Committee

Dr. ir. R.L.J. Helmons	Technische Universiteit Delft
Dr. ir. G.H. Keetels	Technische Universiteit Delft
Ir. M. Elerian	Technische Universiteit Delft
Dr. ir. W.-P. Breugem	Technische Universiteit Delft
Dr. ir. C. Chassagne	Technische Universiteit Delft

TU Delft Delft University of Technology

ABSTRACT

To meet energy demand towards a low-carbon future, the global market demand is growing for metals such as cobalt and nickel which are major elements in batteries. Polymetallic nodules, which are formed on abyssal plains at depths ranging from 4 to 6 kilometres and are distributed in high abundance on the top of the seabed, contains several times more cobalt and nickel than the entire global terrestrial reserves. This has raised the interest to exploit these resources from the deep ocean. The seafloor mining tool (SMT) can move along the soft sea bottom and can collect polymetallic nodules at the surface of sea bottom. While doing so, it will also entrain sediments and water, the excess of water and sediment entrained is discharged at the back of the SMT, forming a sediment plume. The sediment plume dispersion has strong adverse impacts on deep-sea environment: (1) the blanketing effect might clog the feed channels to the seabed organisms and benthic habitat located on the topsoil; (2) toxic chemicals are contained in the sediment plumes, which could have permanent damage on habitat. Thus, it is essential to study the behaviour of this sediment plume in order to limit plume dispersion and thus to reduce its environmental impact.

Experimental research is a powerful technique to study the plume behaviour and its environmental impacts. However, a major problem is that experiments sometimes take a long time due to complex experimental set-up. Compared to experiments, numerical analysis can save time and costs when solving complex problems. Furthermore, numerical modelling can provide deeper understanding and flexibility for boundary conditions and sediment types, which is applicable on both model and prototype scale. Previous numerical studies have noted the significant role of flocculation in limiting the dispersion of plumes generated in deep-sea mining operations, but flocculation process has not been modelled explicitly. This study aims to establish a numerical model to study the flocculation process and its effect on sediment transport.

Previous flocculation-fluid dynamics modelling has applied a Euler-Euler method with additional population balance equations (Golzarijalal et al., 2017). The disadvantage of the approach is that many equations needs to be solved, which greatly increases the computational costs. To avoid excessive computational costs, the sediment transport is described by a multiphase drift-flux model in this study. The flocculation process is modelled by a discretized form of population balance equations which describe the temporal and spatial variation of the number density of particles for each sediment fraction due to aggregation and breakup. The author has found that, by multiplying the particle volume, the population balance can be efficiently incorporated in the phase continuity equations in the drift-flux model. The flocculation population dynamics of particle aggregation and breakup can thus be characterized by the phase transition terms on the right-hand side of the phase continuity equations. Hence, no additional equation needs to be introduced and solved.

To identify whether the numerical model is mathematically correct, verification is car-

ried out to check conservation relationships (e.g., mass conservation) and iterative convergence of numerical results. Then, an initial numerical investigation has shown the results can qualitatively show the three settling stages (i.e., flocculent settling, hindered settling and compression settling) found in the experimental studies. Afterwards, the collision efficiency, which is treated as an empirical constant in this study, is calibrated using the experimental data from the settling column tests conducted by Enthoven (2021) for illite of different concentrations (i.e., 20, 40, 100 g/L) in freshwater without salt. The results of calibration show a good fit to the experimental data. The difference between the numerical and the experimental settling curves for 100 g/L illite is only 2.42%. Another advantage of the numerical simulation is that it can provide the particle size distribution over time, which is not measured in the experiments.

The major novelty of this study is the coupling of the drift-flux model and the population balance equation. Flocculation is described with the added phase transition source terms in the drift-flux model and no additional equation is introduced, which inherits the characteristics of population balance and the merits of drift-flux model in reducing computational costs. In the future work, it is advised to calibrate other flocculation parameters related to breakup process and to validate the numerical model. The flocculation modeling technique as proposed in this study can be incorporated as a module into an extended drift-flux model to predict the dispersion of deep-sea mining plumes.

ACKNOWLEDGEMENTS

I would not have been able to complete this challenging project without the guidance, support from many people.
First, I want to express my thanks to my mentor Dr.Ir. R.L.J. Helmons who provided me this wonderful topic. This thesis would not have been accomplished without his dedicated guidance and support throughout my graduation project.
Secondly, I would like to extend my sincere thanks to my chariman Dr.ir. G.H. Keetels, who is my first teacher introducing me to the amazing world of CFD and offered me the position of student assistant. Being a student assistant provides me the opportunity for strengthening my knowledge in CFD and enriching my experience.
I would also like to express my deep appreciation to my supervisor Ir. M. Elerian from whom who I have learned a lot of practical experience with OpenFOAM. Thank you so much for your time, assistance, and patience in guiding me from a beginner to an intermediate user of OpenFOAM.
Additionally, I would like to thank Boyao Wang for sharing his experience in CFD and programming. I am also thankful to D.H.B. Enthoven for providing his experimental data without which I would have not calibrated my model.
Finally, I would like to thank my parents who encouraged me and supported me throughout the time of my study.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

List of Abbreviations

2D Two Dimensional

3D Three Dimensional

CCZ Clarion-Clipperton Fracture Zone

CFD Computational Fluid Dynamics

CFL Courant–Friedrichs–Lewy

CV Control Volume

COM Center Of Mass

DEM Discrete Element Method

DNS Direct Numerical Simulation

EIA Environmental Impact Assessment

FVM Finite Volume Method

ISA International Seabed Authority

LES Large Eddy Simulations

LHS Left Hand Side

MIDAS Managing Impacts of Deep-seA reSource exploitation

MSV Mining Support Vessel

OpenFOAM Open-source Field Operation And Manipulation

PBE Population Balance Equation

PDE Partial Differential Equation

PSD Particle Size Distribution

PISO Pressure-Implicit with Splitting of Operators

RANS Reynolds-Averaged Navier Stokes

RHS Right Hand Side

SIMPLE Semi-Implicit Method for Pressure Linked Equations

SMT Seafloor Minging Tool

TVD Total Variational Diminishing

VTS Vertical Transport System

List of Symbols

Greek symbols

α_s	The total volume concentration of sediment	–
β	Collision frequency function	s^{-1}
ϵ	Turbulent kinetic energy dissipation rate	$Jkg^{-1}s^{-1}$
Γ	Breakup distribution function	–
κ	Von Kármán constant	–
κ_c	The probability of breakup induced by inter-particle collision	–
Λ_k	Phase transition term for phase k	s^{-1}
μ	Dynamic viscosity	Pas
ν	Kinetic viscosity	Nsm^{-2}
ν_t	Turbulence eddy viscosity	Nsm^{-2}
$\overline{\alpha}(H_i)$	The layer-averaged sediment concentration as a function of height	–
Ψ	Shape factor	–
ρ_f	Density of liquid	kgm^{-3}
ρ_k	Density of phase k	kgm^{-3}
ρ_m	Density of mixture	kgm^{-3}
ρ_p	Density of sediment particles	kgm^{-3}

Latin symbols

$\boldsymbol{T_k^T}$	Turbulent stress	$kgm^{-2}s^{-2}$
$\boldsymbol{T_k}$	Viscous stress	$kgm^{-2}s^{-2}$
$\boldsymbol{u_f}$	Velocity of the carrier fluid	ms^{-1}
$\boldsymbol{u_k}$	Velocity of phase k	ms^{-1}
$\boldsymbol{u_m}$	Velocity of mixture	ms^{-1}
$\boldsymbol{u_{km}}$	Diffusion velocity of phase k	ms^{-1}

$\boldsymbol{u_{kr}}$	Relative velocity of phase k	$\mathrm{m\,s^{-1}}$
A	Collision efficiency	–
b	An empirical constant in breakup frequency function	–
C_1	An empirical value in the settling-velocity equation of Ferguson and Church	–
C_2	An empirical value in the settling-velocity equation of Ferguson and Church	–
C_d	Drag coefficient	–
c_k	Mass fraction of phase k	–
d	Particle diameter	m
d_p	Diameter of sediment particles	m
E_b	An empirical constant in breakup frequency function	–
F_b	Buoyancy	N
F_d	Frictional drag force	N
F_g	Gravitational force	N
F_y	Floc strength	Pa
G	Shear rate	$\mathrm{s^{-1}}$
g	Gravitational acceleration	$9.81\,\mathrm{m\,s^{-2}}$
G_k	Buoyancy production or consumption term	$\mathrm{J\,kg^{-1}\,s^{-1}}$
K	Boltzmann constant	$1.380\,648\,52 \times 10^{-23}\,\mathrm{J\,K^{-1}}$
k	Turbulent kinetic energy	$\mathrm{J\,kg^{-1}}$
N	Number density of particles	$\mathrm{m^{-3}}$
n	Richardson and Zaki index	–
n_f	Fractal dimension	–
p	Pressure	Pa
P_k	Turbulent kinetic energy production term	$\mathrm{J\,kg^{-1}\,s^{-1}}$
R	Submerged specific weight of the soil	–
Re_p	Particle Reynolds number	–
S	Breakup frequency function	$\mathrm{s^{-1}}$
v_r	Relative velocity between sediment particles and liquid	$\mathrm{m\,s^{-1}}$

v_t	Terminal velocity of sediment particles	$\mathrm{m\,s^{-1}}$
V_{hs}	Hindered settling function	–
v_{hs}	Hindered settling velocity	$\mathrm{m\,s^{-1}}$
w	Settling velocity	$\mathrm{m\,s^{-1}}$

1

INTRODUCTION

1.1. GENERAL BACKGROUND

The HMS Challenger Expedition (1872-1876) founded the modern oceanography. The significant discoveries included:

- Discovery of mountains in the sea, including the well-known Mid-Atlantic Ridge in the middle of the Atlantic Ocean.
- Discovery of curious marine life at sea bottoms.
- Physical measurement and charting of ocean currents, salinity and temperature.
- Collection of ocean bottom deposits, including polymetallic nodules.

Nonetheless, the commercial value of those deposits was first unravelled by Mero (1965) who also predicted that deep-sea would become an important source of metals in the future. Since World War II, persistent demand for those mineral resources has increased steadily. Global consumption of raw materials are high and are expected to increase continuously due to demographic and economic growth, which has imposed an unprecedented strain on the Earth's natural resources and ecosystems. In response to this severe trend, a sustainable development strategy including the transition to a low-carbon and green-technology future is required (The World Bank Group, 2017). According to the World Bank report (2017), to meet energy demand towards a low-carbon future and to accelerate the energy transition, a growing market could be expected for a wide range of mineral and metals (e.g., aluminum, cobalt, copper, manganese, nickel, silver and steel). As shown in Figure 1.1, polymetallic nodule deposits in deep ocean contains several times more cobalt and nickel than the entire global terrestrial reserves, decent amount of copper and a lot of manganese, which has raised the interest for industries to exploit these mineral resources from the deep ocean seabed (Hein et al., 2013). These polymetallic nodules are formed on abyssal plains at depths ranging from 4 to 6 kilometres, where they are distributed in high abundance (10-25 kg/m^2) on the top of the

seabed. Due to continued research, many new technologies in mining and in extracting metal from ores have been developed.

	Clarion-Clipperton Zone Nodules [b]	Global Terrestrial Reserve Base [c]	Global Terrestrial Reserves [c]	Prime Crust Zone [b]
Manganese	5,992	5,200	630	1714
Copper	226	1,000+	690	7.4
Titanium	67	899	414	88
TREO [d]	15	150	110	16
Nickel	274	150	80	32
Vanadium	9.4	38	14	4.8
Molybdenum	12	19	10	3.5
Lithium	2.8	14	13	0.02
Cobalt	44	13	7.5	50
Tungsten	1.3	6.3	3.1	0.67
Niobium	0.46	3.0	3.0	0.40
Arsenic	1.4	1.6	1.0	2.9
Thorium	0.32	1.2	1.2	0.09
Bismuth	0.18	0.7	0.3	0.32
Yttrium	2.0	0.5	0.5	1.7
PGM [e]	0.003	0.08	0.07	0.004
Tellurium	0.08	0.05	0.02	0.45
Thallium	4.2	0.0007	0.0004	1.2

Figure 1.1: Contained metal tonnages in deep ocean and on land ($\times 10^6$ tonnes). (Hein et al., 2013)

1.2. DEEP-SEA MINING PROCESS

In the future, the mining system illustrated in Figure 1.2 could be used for deep-sea mining operations. The mining system mainly consists of 3 components: Seafloor Mining Tool (SMT), Vertical Transport System (VTS) and Mining Support Vessel (MSV).

SEAFLOOR MINING TOOL

SMT is a type of remote-controlled underwater vehicle mounted with a hydraulic or mechanical or hybrid nodule collector system. The self-propelled SMT can move along the soft sea bottom and can collect polymetallic nodules at the surface of sea bottom (Chung, 2003). The prototype of SMT designed by Royal IHC is shown in Figure 1.2. It can travel on the seafloor with a speed ranging from 0.2 to 0.5 m/s and can collect polymetallic nodules hydraulically (Blue Nodules, 2016). Nodules are separated from the collected sand-sediment-nodule mixture and are then transported via the VTS to the MSV at the water surface. The excess of sediment and water is discharged at the back of the SMT. This outflow results in a sediment plume (see Figure 1.3).

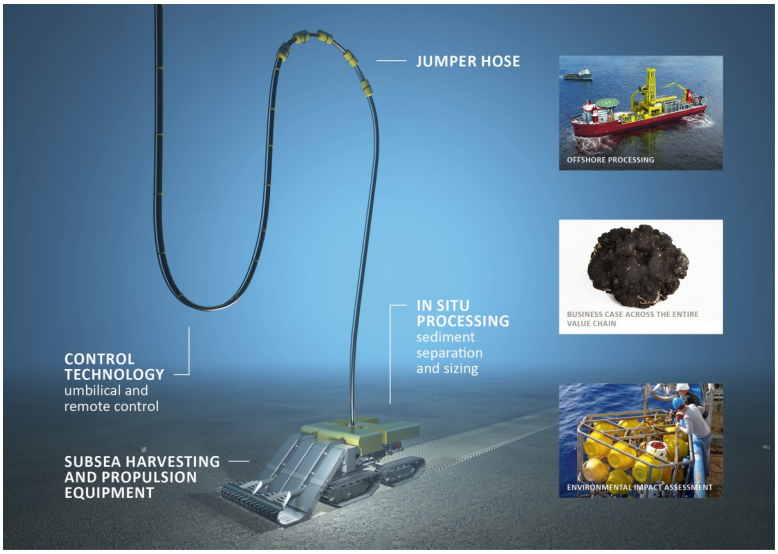


Figure 1.2: Deep-sea mining machine (Image from Blue Nodules, 2016)

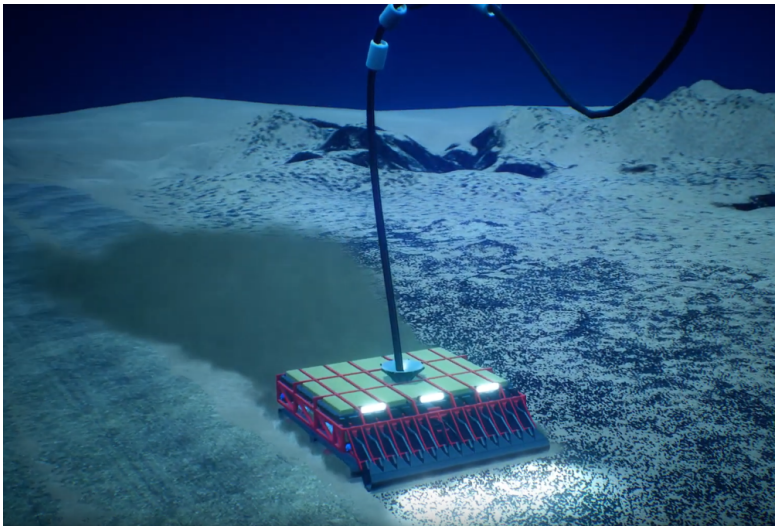


Figure 1.3: Sediment plumes formed at the back of SMT (Blue Nodules, 2020)

VERTICAL TRANSPORT SYSTEM

The VTS is used to transport the collected nodules from the SMT to the MSV. A typical VTS mainly consists of a rigid riser with a length up to several kilometres and a flexible riser with a length of a few hundred metres (Chung, 2003; Volkmann, 2018). The nodules can be lifted by either airlift or pump. According to Volkmann(2018), the designed VTS in the Blue Mining project is able to lift 400 tons of solids per hour from the SMT to the

MSV over a vertical distance of 5 kilometres.

MINING SUPPORT VESSEL

The MSV is an infrastructure at sea surface designed to support mining operations at the sea bottom. Its main function is to provide facilities for power supply, storage, accommodation, workshops and mining consumables (Volkmann, 2018). After the mixture of nodules and sediment is transported to the MSV, on board processing will separate nodules from remaining sediment and water. The remaining mixture of sediments, waste and other effluents can be either released into mid water column or near the seabed, which needs to be defined in the mining exploitation regulations.

1.3. DEEP-SEA MINING PLUMES

1.3.1. CLASSIFICATION

The flow created by mining operations can be recognized as negative buoyant plumes or jets (Protogene, 2018). The term "plume" refers to the flow produced by continuous sources of only buoyancy. The term "jet" is used to describe the flow forced by initial momentum and by neutral buoyancy. Based on the sources of momentum, the flow can be classified into several types, given in Table 1.1.

	Continuous Source	Intermittent Source
Momentum only	Jet	Puff
Buoyancy only	Plume	Thermal plume
Both momentum and buoyancy	Buoyant jet or forced plume	Buoyant puff

Table 1.1: Classification of flow created by mining operations by the source of momentum. The origin of the table is from Lee and Chu (2003) and has then been adapted by Byishimo (2018).

The deep-sea mining plumes can also be classified by their origins, which is given in Table 1.2.

Type	Origin	Level of damage to environment
Plumes caused by maneuvring the SMT	The movement and collection operation of the SMT	The least harmful among the three
Sediment plumes	The mixture discharged through the diffuser at the back of the SMT, which is the excess of water and sediment entrained in the collection process	Harmful
Tailing discharge plumes	Sediment released through the discharge pipe from the MSV	Harmful

Table 1.2: Classification of deep-sea mining plumes

The sediment plumes discharged by the SMT and the tailing plumes released by the MSV, which are the source of turbidity plumes, have strong adverse impacts on deep-sea environment (Spearman et al., 2020). The spatial scale of the dispersion of these turbidity plumes ranges from a few hundred meters to several kilometres from the mining zone (Jankowski and Zielke, 2001).

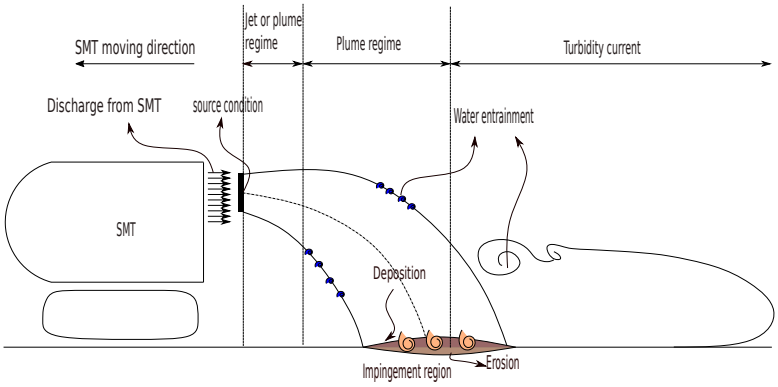


Figure 1.4: The development of sediment plumes discharged by the SMT (Elerian, n.d.)

Based on the horizontal length scale, the development of sediment plumes discharged by the SMT can be classified into 4 stages shown in Figure 1.4.

- Discharging zone: In this region, the water-sediment mixture is discharged from the SMT. The discharge rate and initial momentum are controlled by the SMT.

- Jet regime: As the sediment plume moves away from the discharged point, it widens due to entrainment of ambient water. The buoyancy will become dominant.

- Plume regime: The sediment plume interacts with the seafloor in the impingement region, which causes deposition and erosion.

- Turbidity current regime: After the impingement, a turbidity current is generated and spreads further along the seafloor. Turbidity currents are the suspension flows driven by excess density caused by the dense suspended particles (Srivatsan et al., 2004). Gravity acting on the density difference between the mixture and the ambient fluid drives turbidity currents to move forward. The motion of a turbidity current generates turbulence which is the primary mechanism to keep the sediment in suspension. Turbidity currents are non-conservative in nature. They can freely exchange particles with a sediment bed through erosion and deposition, and can also exchange water with ambient fluid by entrainment and detrainment (Stone, 1999). To maintain the existence of a turbidity current, the sediment in suspension needs to be sustained. Otherwise, the turbidity current will die out.

1

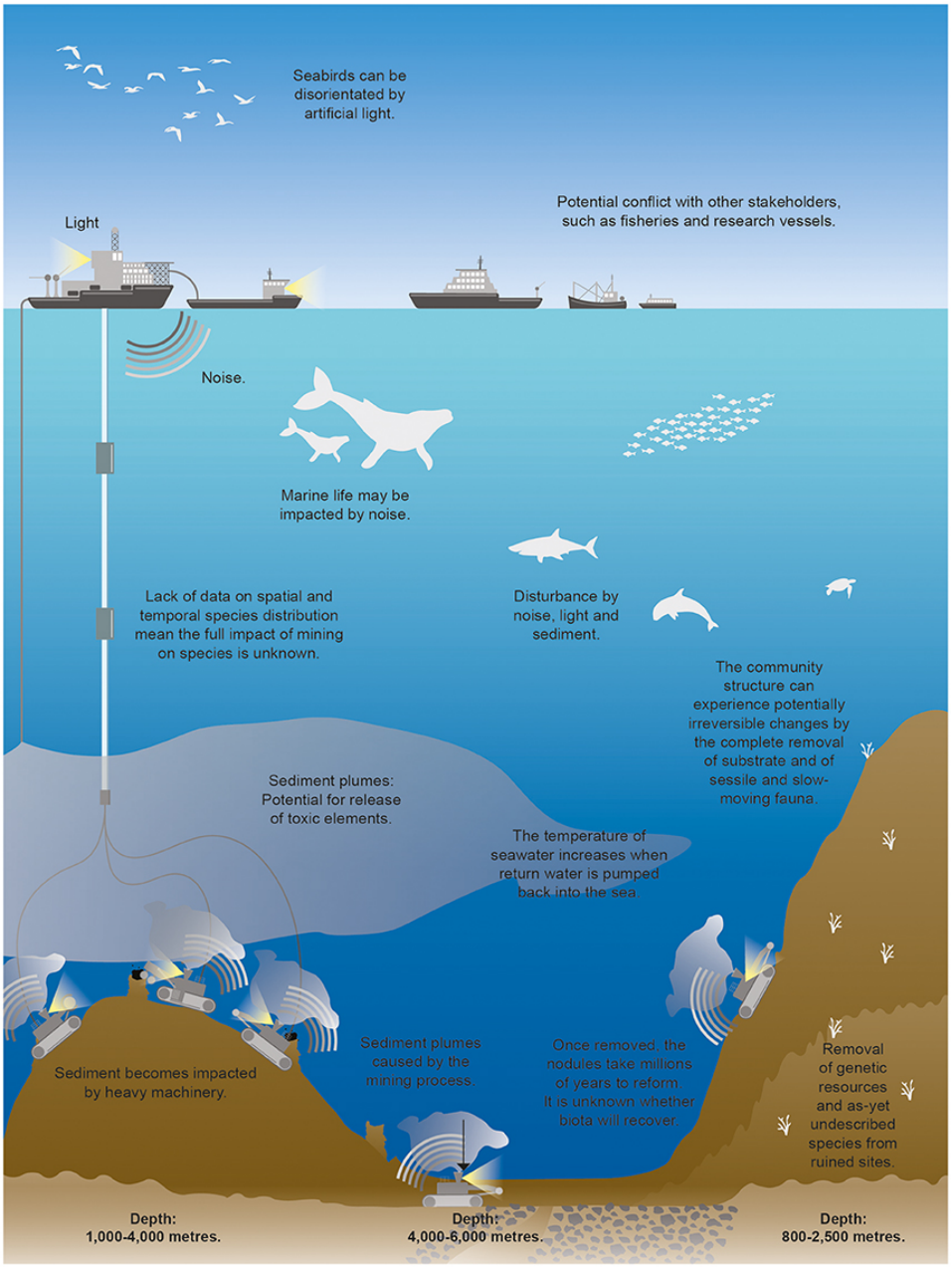


Figure 1.5: Potential impacts from deep-sea mining operations (Miller et al., 2018)

1.3.2. ENVIRONMENTAL IMPACT

With the perspective of exploitation of deep-sea mineral resources, the knowledge about marine environment and the understanding of potential environmental impact of deep-sea mining operations have also been deepened. Miller et al. (2018) has summarized the widespread environmental impacts expected to occur from mining operations on marine ecosystem and biodiversity, see Figure 1.5.

As the severity of environmental pollution of deep-sea mining operations become apparent, the importance of protection of marine systems has been recognized gradually (Roberts et al., 2017; Washburn et al., 2019). Dover (2011) proposed the International Seabed Authority (ISA) to tighten the rules and the regulations on deep-sea mining to mitigate environmental impacts. As an important process, Environmental Impact Assessment (EIA) was implemented to manage industrial projects in the aspect of environmental and social risks (Durden et al., 2018). Other than EIA, collaborative scientific projects were also carried out to assess the environmental impacts caused by deep-sea mining operations. From 2013 to 2016, the MIDAS (Managing Impacts of Deep-seA reSource exploitation) project was conducted to identify the environmental impacts of extracting deep-sea mineral and energy resources, and to develop practical solutions for commercial mining activities (MIDAS, 2016). The study has classified the environmental impacts of deep-sea mining activities in two categories: direct and indirect impacts. Direct impacts are the direct consequences of the removal of resources, which mainly includes: (1) direct mortality of benthic organisms; (2) habitat destruction; (3) habitat fragmentation; (4) habitat modification. Indirect impacts on marine environment mainly comprise: (5) generation of near-seabed sediment plumes; (6) release of toxic chemicals contained in the discharged sediment plumes. The loss of seabed habitats could be permanent for nodule regions and the recovery rates after disturbance can reach several decades (MIDAS, 2016). Other potential impacts comprise the disturbanceS caused by light and noise, and the increase of the seawater temperature due to the heat from mining operations and from the return of dewatering waste (Miller et al., 2018).

1.3.3. PLUME BEHAVIOUR

The plume behaviour is affected by a number of factors that vary with a broad range of temporal and spatial scales. In-situ measurements, laboratory experiments and modelling studies were conducted to study the formation of sediment plumes and the behaviour of particle sedimentation in the near-field. Spearman et al. (2020) have observed that turbidity plumes generated in the field experiments disperse more rapidly than predicted in the simulations. This difference is attributed to the occurrence of flocculation. Spearman et al. (2020) have also found that the spatial dispersion of sediment plumes can be significantly reduced by flocculation. The results of the measurements and numerical modelling have shown that the combined action of flocculation, background turbidity and internal tides can restrict benthic plumes.

To enhance the understanding of the behaviour of the sediment plume generated by deep-sea mining operation, Gillard et al. (2018) performed laboratory tests and numerical simulations using real sediments from Clarion Clipperton Zone (CCZ) with different concentrations (30, 105, 175 and 500 mg/L) and different turbulent shear rates (2.4, 5.7 and 10.4 s^{-1}). The purpose is to reduce the environmental impact by optimizing dis-

charge conditions in terms of sediment concentration and shear rate. The results have shown that the flocculation process is most rapid using 500 mg/L abyssal sediment under a shear rate of 2.4 s^{-1}. The simulation results have also evidenced that, under this condition, sediment deposits relatively fast. As a result, blanking effect caused by plume dispersion has been limited to a smaller area.

1.4. RESEARCH DESCRIPTION

1.4.1. PROBLEM DEFINITION

As discussed in Section 1.3.2, the environmental effects caused by these sediment plumes mainly include:

- A new covering bed will be formed after the deposition of sediment plumes, which causes the alteration of seabed. This blanketing effect closes the feed channels to the seabed organisms and benthic habitat located on the topsoil (i.e., the first few centimetres of the seafloor).
- Toxic chemicals are contained in the sediment plumes. Their damage on habitat could be permanent and the recovery rate is slow (MIDAS, 2016).

According to Boschen et al. (2013), reducing the size of sediment plumes is an effective measures to limit the environmental impacts caused by plumes. To restrict the spread of plumes, the design of SMT including its collector, separator and diffuser needs to be improved to discharge sediment at a higher concentration, which can be achieved by optimizing the mixture flow and reducing the clean water intake (Helmons, 2019).
Experimental research is a powerful technique to study the plume behaviour and its environmental impacts. However, a major problem is that experiments sometimes take a long time due to complex experimental set-up. Compared to experiments, numerical analysis can save time when solving complex problems. Furthermore, numerical modelling can provide deeper understanding and flexibility for boundary conditions and sediment types, which is applicable on both model and prototype scale. Previous numerical studies have noted the significant role of flocculation in limiting plume dispersion, but flocculation process has not been modelled explicitly and its effects were not assessed quantitatively. Spearman et al. (2020) have found that the settling velocities based on the particle size distribution are much lower than the values measured from their study which shows the actual behaviour (i.e., flocculation) of particles suspended in the water column. Therefore, it is essential to develop a numerical model to quantitatively study the flocculation process of cohesive sediment and its effect on settlement.

1.4.2. RESEARCH OBJECTIVES

The first objective of this study is to build a numerical model to describe both sediment transport and flocculation process. The second objective is to verify the numerical model. The verification can be done by checking conservation relationships (e.g., mass conservation) and iterative convergence. The third objective is to calibrate the numerical model. This task is carried out through comparing the numerical results with the experimental work of Enthoven (2021) to find the optimal parameter which gives the best fit to the experimental data.

1.4.3. RESEARCH QUESTIONS

The following questions will be addressed in this research:

1. How can flocculation process be modelled ?
2. Which factors influence the flocculation process ?
3. How to solve the coupled flocculation-fluid dynamics model ?
4. How does flocculation process influence the settling behavior ?

1.5. APPROACH

Computational fluid dynamics (CFD) is used to numerically investigate the effects of flocculation on sediment transport in this study. Sediment suspension is a multiphase mixture consisting of water and soil particles. The drift-flux model is used for the description of the multiphase flow (Goeree, 2018). Compared to Euler-Euler method, only one momentum equation for the mixture needs to be solved in the drift-flux model, which can greatly reduce computational cost as the number of phases rises. The phase continuity equations and the mixture momentum equation are discretized using Finite Volume Method (FVM). Collocated grids are used and the field variables are stored at the centre of each cell.

The code is developed by the author in OpenFOAM (Open Source Field Operation and Manipulation) v2006. OpenFOAM is an open source software for developing numerical solvers and pre-/post-processing tools for CFD. The programming language is C++. The drift-flux model named *driftFluxFoam* is available in OpenFOAM, but it only considers two phases. To model a multiphase system with more than two phases, a multiphase drift-flux solver developed by Delft University of Technology is used in this study, which has not been public yet. To model the flocculation process, the author has implemented a discretized form of population balance equations to describe how sediment populations change over time. Then, the population balance can be efficiently incorporated in the phase continuity equations in the drift-flux model. The proposed flocculation modeling technique incorporated as a module into an extended drift-flux model is then used to study the effect of flocculation on settling process in the verification and calibration procedure.

1.6. OUTLINE

The thesis is composed of seven themed chapters, including this introductory chapter. This chapter mainly includes: the background of this study, some key terms, the problems, the research objectives and questions, the general approach and the thesis structure. In Chapter 2, the theory of sedimentation and the knowledge of flocculation process of cohesive sediment are explained. First, the composition and the properties of cohesive sediment are elaborated. Subsequently, the physics of flocculation and the parameters of flocculation models are introduced. Then, four types of settling process are classified. Besides, the Richardson-Zaki relationship used to calculate settling velocity is elaborated in this chapter. In Chapter 3, the modelling strategies used in this thesis are explained. The sediment transport is described by a multiphase drift-flux model. A

discretized form of population balance equation (PBE) is used to model the population dynamics. Then, the population balance is efficiently incorporated in the phase continuity equations in the drift-flux model. The turbulence of the multiphase system is modelled by a *Buoyant k-ϵ* model. In Chapter 4, the FVM discretization procedure for phase continuity equations and mixture momentum equation is given. Furthermore, the numerical schemes and the solution algorithm are elaborated. In Chapter 5, the numerical verification is done by examining conservation relationships and by checking iterative convergence of simulation results. In Chapter 6, the numerical model is calibrated against the experiments conducted by Enthoven (2021). Finally, the conclusions of this thesis and the recommendations for future research are given in Chapter 7.

2

THEORY

2.1. COHESIVE SEDIMENTS

2.1.1. COMPOSITION

Sediments can be classified as cohesive and non-cohesive according to their size. Cohesive sediments are defined as fine sediments with a size less than 63 µm (Lumborg and Vested, 2008). These fine-grained sediments contain clay, silt and fine-grained sand, mixed with organic compounds (Shrestha and Blumberg, 2005). Clay particles are typically smaller than 2 µm, comprised mainly of illite, kaolinite and montmorillonite (Berlamont et al., 1993). The size of fine clay particles can be less than 0.1 µm, which are colloids in nature and will remain in suspensions due to Brownian motion (Grabowski et al., 2011). The electromagnetic properties cause clay particles to stick to each other by organic bounds, forming larger mud flocs. The high proportion of clays are largely responsible for the nature of cohesive sediments. Silt particles are from 2 µm to 63 µm, while the size of sand grains ranges from 63 µm up to 2 mm. Silt and sand are dominantly composed of quartz, which have no effect on cohesion (Maggi, 2005). Silt are commonly found in sediment suspensions and bed, while sand-sized particles are typically located at high-energy regions such as fluvial environments (Grabowski et al., 2011). Organic components in sediment consist of biological organisms, fecal pellets, detritus, nutrients and polymers (Grabowski et al., 2011; Maggi, 2005). Their existence can enhance the physico-chemical bonds amongst sediment particles and thus alter the stability of aggregates (Mehta, 1991; van Ledden et al., 2004).

2.1.2. COHESIVENESS

The cohesion of cohesive sediments results from electro-chemical inter-particle forces. Clay-sized particles are negatively charged at surface surrounded by a dense and high-concentration cloud of cations in Stern layer (see Figure 2.1). As shown in Figure 2.1, Guoy layer is adjacent to Stern layer. The union of Stern layer and Guoy layer is called double layer. In the double layer, the decrease in ζ electrical charge potential at the interface between water and sediment particles is proportional to the concentration of

ions. The type of minerals and the balance between attractive electrical forces and diffusion within the medium will influence the thickness of this electrical double layer (van Leussen, 1994). The interaction of the double layers of two approaching particles is the governing mechanism for particle aggregation. The inter-particle distance determines whether the resultant electrical force is repulsive or attractive: repulsive forces will separate particles, while attractive forces tends to bring two particles closer. If two approaching particles are sufficiently close and the repulsive force can be overcome, inter-particle collision is possible, which may further induce particle aggregation (Maggi, 2005).

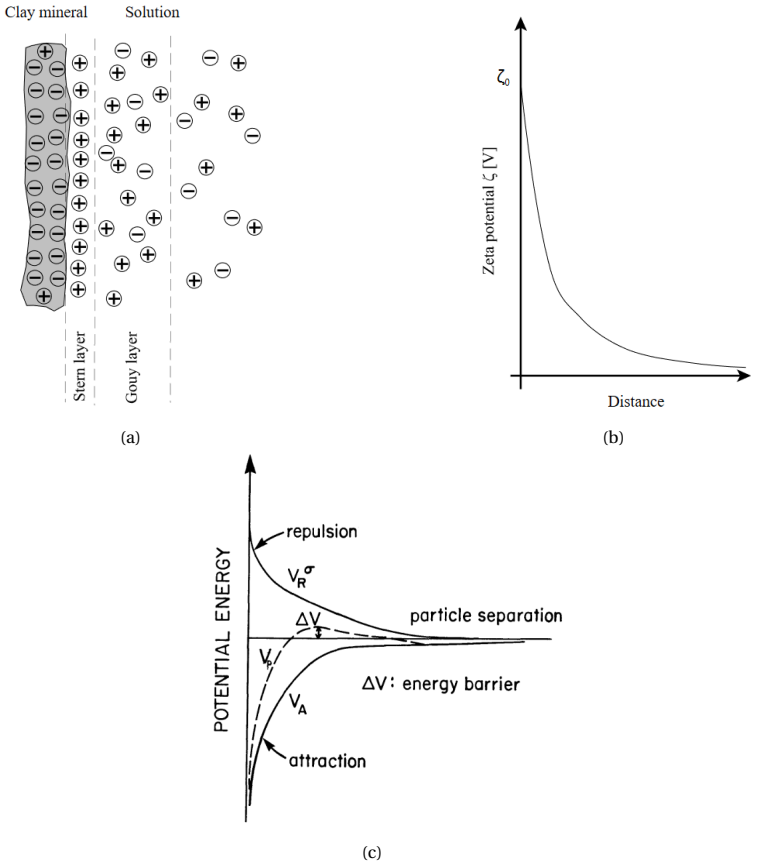


Figure 2.1: (a) An illustration of the structure of charges surrounding clay particles.(Maggi, 2005). (b) ζ potential as a function of the distance to the surface of clay particles (Maggi, 2005). (c) Potential energy as a function of the distance between two approaching particles' surfaces. V_R and V_A are the potential energy for repulsion and attraction respectively. ΔV is the energy barrier (Valioulis, 1983).

2.2. SETTLING BEHAVIOUR

2.2.1. CLASSIFICATION

The settling behaviour of particles is dependent on their nature (e.g, size, cohesive properties) and concentration, which can be classified into four types:

- Discrete settling
- Flocculent settling
- Hindered settling
- Compression settling

A schematic representation of four types of settling behaviour in a settling column is given in Figure 2.2.

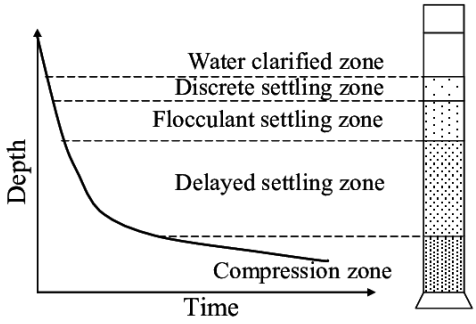


Figure 2.2: Four types of settling behaviour for sludge (Franco et al., 2007)

2.2.2. DISCRETE SETTLING

Discrete settling refers to the settling process in which individual particles fall freely without interactions with other particles and without the tendency to flocculate. In the case, a spherical particle settling in a liquid is under the influence of the gravitational force F_g, buoyancy F_b and the frictional drag force F_d. Once released, the particle will accelerate until it reaches the terminal settling velocity when the three forces are in equilibrium:

$$F_g - F_b = F_d \tag{2.1}$$

The frictional drag force can be calculated by:

$$F_d = \frac{1}{2} C_d \rho_f v_t^2 A \tag{2.2}$$

where C_d is the drag coefficient, A is the sectional or projected area of a particle normal to the settling direction and it equals to $\frac{\pi}{4} d_p^2$ in case of a perfect round particle, and d_p is

the diameter of a spherical solid particle. The resultant force F_r of the gravitational force and buoyancy can be calculated by:

$$F_r = F_g - F_b = (\rho_p - \rho_f) g \frac{\pi d_p^3}{6} \tag{2.3}$$

where ρ_p is the density of the solid particle and ρ_f is the density of liquid. Combining Eq.(2.1), Eq.(2.2) and Eq.(2.3), the force equilibrium can be rewritten as:

$$\frac{1}{2} C_d \rho_f v_t^2 A = (\rho_p - \rho_f) g V \tag{2.4}$$

Then, the terminal velocity of a spherical particle can be derived as:

$$v_t = \sqrt{\frac{4(\rho_p - \rho_f) d_p g}{3 C_d \rho_f}} \tag{2.5}$$

The drag coefficient C_d can be written as:

$$C_d = \frac{4(\rho_p - \rho_f) g d_p}{3 \rho_f v_t^2} \tag{2.6}$$

For non-spherical particles, a shape factor defined in Eq.(2.7) is introduced (Wadell, 1932).

$$\Psi = \frac{A_{reduced}}{A} \tag{2.7}$$

where $A_{reduced}$ is the reduced surface area of an equivalent spherical particle and A is the actual surface area of a particle. Ψ equals to 1 for a spherical particle.
Thus, a general formula for the terminal settling velocity considering particle shape can be derived as:

$$v_t = \sqrt{\frac{4(\rho_p - \rho_f) \Psi d_p g}{3 C_d \rho_f}} \tag{2.8}$$

Ferguson and Church (2004) derived the terminal settling velocity from dimensional analysis. The involved parameters include soild particle diameter d_p, fluid viscosity ν_f, fluid density ρ_f and solid particle density ρ_p.
The particle Reynolds number Re_p is defined as:

$$Re_p = \frac{\rho_f |v_r| d_p}{\nu_f} \tag{2.9}$$

where v_r is the relative velocity between the particle and the liquid and it is given by:

$$v_r = v_p - v_f \tag{2.10}$$

where v_p is the velocity of particle and v_f is the velocity of the liquid.
For small particles with a low particle Reynolds number ($Re_p < 1$), particle settling is

mainly resisted by the viscous drag force caused by laminar flow. The equation converges on Stoke's law and it can be expressed as:

$$v_t = \frac{Rgd_p^2}{C_1 \nu_f} \tag{2.11}$$

where R is the submerged specific weight of the particle which can be calculated by Eq.(2.12). R takes the value of 1.65 for quartz in fresh water. C_1 is a constant with a theoretical value of 18. For natural grains, a higher value of C_1 can be expected which can reach 24 (Ferguson and Church, 2004).

$$R = \frac{\rho_p - \rho_f}{\rho_p} \tag{2.12}$$

For large particles with Re_p ranging from 10^3 to 10^5, the settling is mainly resisted by the turbulent drag force of the wake behind each grain. The terminal settling velocity is given by:

$$v_t = \sqrt{\frac{4Rgd_p}{3C_2}} \tag{2.13}$$

C_2 is the drag coefficient C_d, which can be calculated by Eq.(2.6). $C_2 \approx 0.4$ for smooth spheres and $C_2 \approx 1 \sim 1.2$ for natural grains (Ferguson and Church, 2004; Goeree, 2018). Combining Eq.(2.11) and Eq.(2.13), an explicit equation for the terminal velocity as a function of particle diameter d_p was developed (Ferguson and Church, 2004):

$$v_t = \frac{Rgd_p^2}{C_1 \nu_f + (0.75 C_2 Rgd_p^3)^{\frac{1}{2}}} \tag{2.14}$$

For small Re_p, the terminal velocity obtained by Eq.(2.14) approximates that obtained by Eq.(2.11) for Stokes flow. Similarly, for large Re_p, the result obtained by Eq.(2.14) is asymptotic to that calculated by Eq.(2.13). The settling velocity as a function of particle diameter with three different combinations of C_1 and C_2 values is illustrated in Figure 2.3. The discrete settling behaviour is dominant when sediment particles are of low concentration and have little tendency to flocculate. In the case, interference among particles are weak and the overall settling behaviour shows the characteristics of discrete settling zone in Figure 2.2.

2.2.3. FLOCCULENT SETTLING

The term 'flocculent settling' refers to settling in which suspended particles tend to flocculate, forming larger flocs with higher settling velocities (Haan et al., 1994). The theory of flocculation is given in Section 2.3. In the flocculent settling stage, aggregation occurs and floc size grows while settling, which leads to an increasing settling velocity. Compared to discrete settling, a non-constant settling velocity is observed in the flocculent settling zone. As settling proceeds, particles accumulate at the bottom and they are more likely to interact, which form flocs more efficiently (Sperling, 2007).

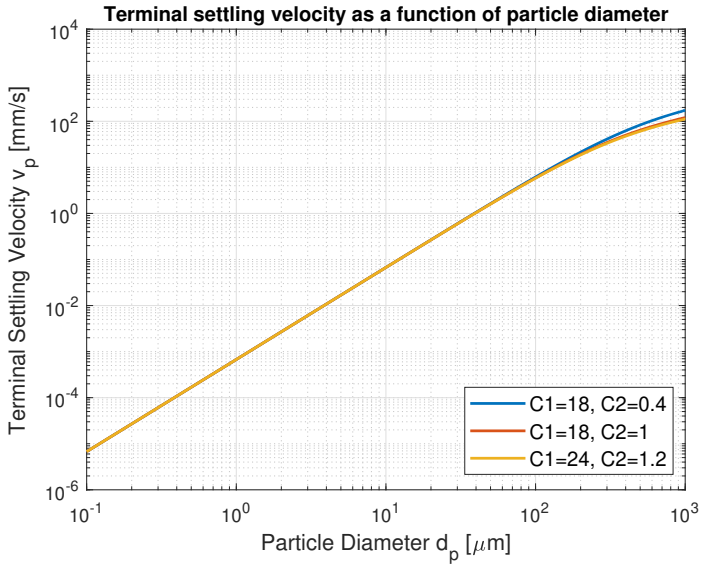


Figure 2.3: Relationship between the terminal settling velocity and particle diameter determined by Eq.(2.14). Settling velocity is calculated using three combinations of C_1 and C_2. Hindered effect is not considered here.

2.2.4. HINDERED SETTLING

The settling velocity of a group of particles are reduced due to the impedance of neighbouring particles which leads to an increase in drag force. This phenomena is named as hindered settling. The hindered settling velocity can be calculated using an empirical equation which is a function of the total volume concentration of solids (Richardson and Zaki, 1954):

$$v_{hs,k} = v_{t,k} V_{hs}(\alpha_s) \tag{2.15}$$

where:

$v_{hs,k}$: the hindered settling velocity of particles belonging to phase k.

$v_{t,k}$: the terminal settling velocity of particles belonging to phase k, which can be calculated by Eq.(2.14).

$V_{hs}(\alpha_s)$: the hindered settling function. The value can be calculated by Eq.(2.18).

α_s: the total volume concentration of sediment.

When the hindered settling velocity is relative to liquid, the hindered settling function becomes (Mirza and Richardson, 1979):

$$V_{hs}(\alpha_s) = (1-\alpha_s)^{n-1} \tag{2.16}$$

where n is the hindered settling exponent or Richardson and Zaki index. Rowe (1987) proposed an empirical equation to approximate n:

$$n = \frac{4.7 + 0.41 Re_p^{0.75}}{1 + 0.175 Re_p^{0.75}} \tag{2.17}$$

where Re_p can be calculated by Eq.(2.9).
Basson et al. (2009) has suggested that the value of hindered settling function should be zero when α_s exceeds the loose sediment packing limit $\alpha_{s,max}$. For clay materials, $\alpha_{s,max}$ approximates 0.45 (Goeree, 2018). Therefore, an modified hindered settling function can be rewritten as:

$$V_{hs} = \begin{cases} (1-\alpha_s)^{n-1} & \alpha_s < \alpha_{s,max} \\ 0 & \alpha_s \geq \alpha_{s,max} \end{cases} \tag{2.18}$$

The hindered effect is not obvious when sediment concentration is low. As settlement proceeds and sediment particles accumulate at the middle and the lower parts of the settling column and thus the concentration of sediment α_s increases gradually, the hindered effect becomes dominant in the delayed settling zone. As a result, the settling velocity is reduced due to rising sediment concentration according to Eq.(2.15) and Eq.(2.16). The hindered effect described by Eq.(2.18) is more significant in delayed settling zone (hindered settling stage) and compression settling zone (compression settling stage) shown in Figure 2.2, where sediment concentration is relatively large.

2.2.5. COMPRESSION SETTLING

Compression settling occurs at the bottom of basin, where a layer of solids with a definite structure is formed (Haan et al., 1994). At the bottom of the settling column, highly concentrated particles are in contact and they can only settle further by compression under the weight of overlying solids. In this stage, the settling velocity decreases gradually, see the declining slope of the settling curve in Figure 2.2. This is due to that water is slowly squeezed out and void space is occupied as solids are compressed.
Modelling compression settling requires to use an extra soil mechanical model accounting for soil consolidation. The CFD framework and the soil mechanical model also need to be coupled. In addition, the time-scale of compression settling stage is much larger than other types of settling and the duration of this stage is theoretically infinite (Ahern, 2017). Thus, it is not practical to completely simulate the compression settling stage as it takes too long to simulate. Consequently, compression settling is rarely modelled explicitly in CFD (Ahern, 2017). In this research, soil compaction will not be modelled since this study focuses on the effects of flocculation on settling behaviour.

2.3. FLOCCULATION

2.3.1. DEFINITION

Compared to non-cohesive sediments composed of sand and gravel, fine-grained particles can bond together to form flocs of larger diameter due to cohesion and this process is defined as aggregation, while breakup is the process in which individual particles separate from each other. It is important to understand the flocculation process of cohesive sediments in sediment plumes generated by deep-sea mining operations, since it

influences the PSD (Particle Size Distribution) which represents the relative fraction of particles according to their size. A dynamic PSD will be present during flocculation process due to particle aggregation and breakup, which will cause the variation of settling velocities. Consequently, plume dispersion behaviour is influenced by flocculation.
In the study of Dyer (1989), flocs are classified into microflocs and macroflocs by their sizes. The size of microflocs ranges from 10 to 20 µm while marcroflocs are of the order of milimeters. Microflocs are generally of higher density than macroflocs. Thereby, the strength of the microflocs is larger than that of macroflocs since particles are densely packed in the microflocs. When the size of macroflocs approaches the turbulent Kolmogorov microscale, they will break up (Dyer, 1989).

2.3.2. AGGREGATION

In the theory of collision induced aggregation, collisions are essential for particles to adhere together, forming larger flocs. Collisions could occur among multiple particles, but the probability of collisions involving only two particles is dominant. Therefore, it is reasonable to only consider collisions that occur between two approaching particles (Smoluchowski, 1916; Thomas et al., 1999).

COLLISION FREQUENCY

The frequency of collisions can be characterized by a frequency function β_{ij} for two colliding particles of size class i and j. Collision process is mainly governed by three mechanisms: Brownian motion, turbulent shear and differential settling.

Brwonian Motion

Brownian motion refers to random movement of particles in suspension motivated by thermal energy. It is dominant for particles smaller than 1 µm and its effect on flocculation process becomes negligible when particles are larger than 2 µm (Eisma, 1986; van Leussen, 1994). For Brownian motion, the collision frequency function β_{ij} is related to the absolute temperature T, the dynamic viscosity of the fluid μ and the diameters of two colliding particles d_i and d_j., which can be expressed as (Lick et al., 1993):

$$\beta_{i,j} = \frac{2KT(d_i + d_j)^2}{3\mu d_i d_j} \tag{2.19}$$

where K is the Boltzmann constant, which equals to $1.38064852 \times 10^{-23}\,\mathrm{J\,K^{-1}}$.

Turbulent Shear

The relative motion between particles induced by fluid shear can also cause collisions. Adequate turbulence can increase the collision frequency thus enhancing particle aggregation, while too large turbulent shear stress induces breakup of flocs. The frequency function for turbulent shear induced collision can be expressed as (Camp, 1943):

$$\beta_{i,j} = \frac{G}{6}(d_i + d_j)^3 \tag{2.20}$$

where G is the turbulent shear rate in the fluid, which can be approximated by (Camp, 1943; Saffman and Turner, 1956):

$$G = (\frac{\epsilon}{\nu})^{\frac{1}{2}} \tag{2.21}$$

where ϵ is the turbulent kinetic energy dissipation rate and ν is the kinetic viscosity of liquid.

Differential Settling

Differential settling is the process that large and fast particles capture small and slow particles forming larger flocs in a suspension. In this case, the collision frequency function is related to the difference between the settling velocities w_{si} and w_{sj} of two particles belonging to size class i and j. It can be expressed by (Lick et al., 1993):

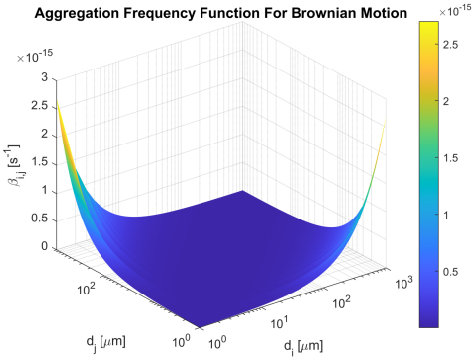
$$\beta_{i,j} = \frac{\pi}{4}(d_i + d_j)^2 |w_{si} - w_{sj}| \tag{2.22}$$

Total collision frequency function

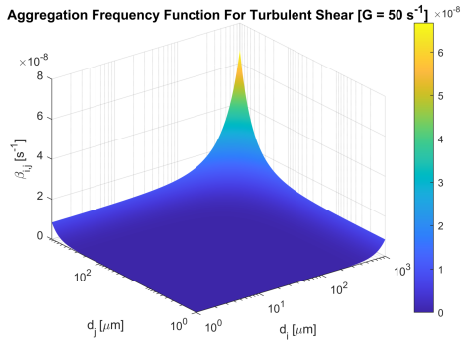
The total collision frequency function $\beta_{i,j}^{Total}$ can be calculated by summing up the collision frequency functions under the three mechanisms:

$$\beta_{i,j}^{Total} = \frac{2KT(d_i + d_j)^2}{3\mu d_i d_j} + \frac{G}{6}(d_i + d_j)^3 + \frac{\pi}{4}(d_i + d_j)^2 |w_{si} - w_{sj}| \tag{2.23}$$

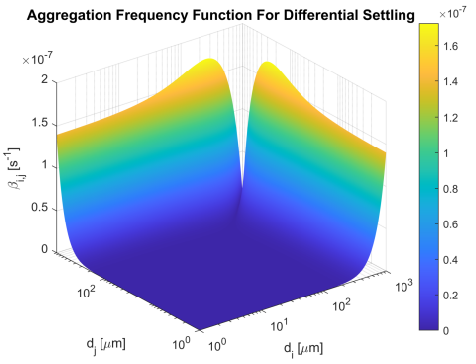
A schematic representation of $\beta_{i,j}$ under the three mechanisms and the total collision frequency $\beta_{i,j}^{Total}$ is shown in Figure 2.4. Figure 2.4 (a), (b) and (c) represents the collision frequencies as a function of the sizes of two colliding particles under the mechanism of Brownian motion, turbulent shear and differential settling respectively. The total collision frequency function is shown in Figure 2.4 (d). A comparison of the values of collision frequency functions under the three mechanisms is shown in Figure 2.4 (e). It can be observed that differential settling governs collision frequency as the sizes of two colliding particles become increasingly different.



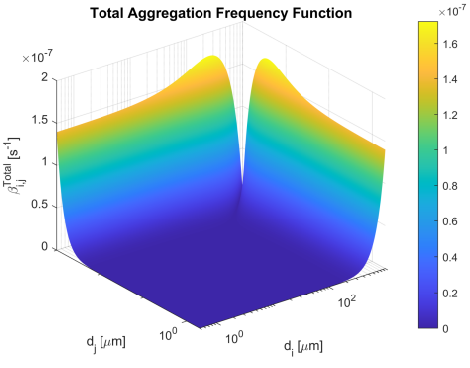
(a) $\beta_{i,j}$ for Brownian motion



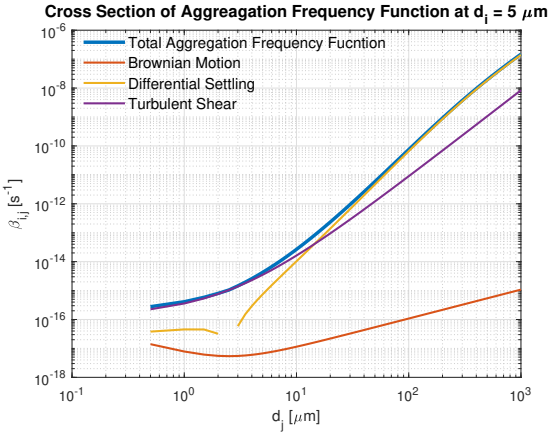
(b) $\beta_{i,j}$ for turbulent shear $[G = 50s^{-1}]$



(c) $\beta_{i,j}$ for differential settling



(d) Total collision frequency function β_{ij}^{Total}



(e) A cross section of total collision frequency function at $d_i = 5\mu m$

Figure 2.4: Collision frequency functions under different mechanisms. Used parameters are: (a) T =293K (20°) (b) $G = 50s^{-1}$ (c) Settling velocity w is determined by Eq.(2.14). $C_1 = 18$, $C_2 = 0.44$, R=1.65, $g = 9.81 m/s^2$, $\mu = 10^{-3} Pa \cdot s$, $\nu = 10^{-6} m^2/s$. For simplicity, hindered settling effect is not considered here. (d) Total collision frequency function given by Eq.(2.23). (e) A cross section of total aggregation frequency function at $d_i = 5\mu m$. The collision frequency is zero for differential settling interrupts at $d_i = d_j = 5\mu m$ because $\beta_{ij} = \frac{\pi}{4}(d_i + d_j)^2 |w_{si} - w_{sj}| = 0$. Similar figures can be found in the research conducted by Maggi (2005).

COLLISION EFFICIENCY

Not all particle collisions lead to aggregation. The probability of aggregation of two particles belonging to size class i and j after collision can be described by a collision efficiency function $A_{i,j}$. For collision induced by Brownian motion, Valioulis (1983) has pointed out that:

- Hydrodynamic and van der Waals forces (i.e., attraction and repulsion between particles) influence the collision probability. As van der Waals forces become of shorter range, hydrodynamics forces are dominant for the collision process of two approaching particles.
- Collision efficiency increases with increasing Hamaker constant, which is defined in Eq.(2.24) and represents the strength of the van der Waals force (Hamaker, 1937). However, the enhancement decreases when two colliding particles are of increasingly different size.
- Hydrodynamic forces reduce collision efficiency and this effect becomes more pronounced for particles which are similar in size.

$$\text{Hamaker Constant} = \pi^2 C N_1 N_2 \tag{2.24}$$

where C is the so-called London coefficient in the particle–particle interaction. N_1 and N_2 are the number densities of the size classes of two interacting particles.
For shear induced inter-particle collision, Adler (1981) analyzed the collision efficiency for two unequal particles in a simple shear flow, considering the influence of hydrodynamic interactions rigorously. Adler found that aggregation between particles of similar size is favoured over aggregation between particles of different size in most cases. The detailed expression for collision efficiency can be found in his study.
For differential settling induced aggregation, Shafrir and Neiburger (1963) theoretically computed collision efficiencies as a function of the radii for water droplets in air. To verify these theoretical values, Levin et al. (1973) experimentally determined the collision efficiencies for droplets with radius less than 120 µm falling through cloud droplets. Due to experimental set-ups, the difference between the experimental and the theoretical results is large. To bridge the difference, an modified analytical expression which is consistent with experimental results can be found in the study of Valioulis (1983):

$$\begin{aligned}
A_{i,j} &= A_0 + A_1 + A_2 + A_3 + A_4 \\
r_1 &= \min(r_i, r_j), r_2 = \max(r_i, r_j) \\
A_0 &= 0.95 - (0.7 - 0.005 r_2)^4 (7.92 - 0.12 r_2 + 0.001 r_2^2) \\
A_1 &= -(\frac{r_1}{r_2} - 0.5)^2 \\
A_2 &= -1.5 exp[-(0.0015 r_2^2 + 8)\frac{r_1}{r_2}] \\
A_3 &= -(1 - 0.007 r_2) exp[-0.651 r_2 (1 - \frac{r_1}{r_2})] \\
A_4 &= \begin{cases} 0 & r_2 < 20 \mu m \\ exp[-30(1 - \frac{r_1}{r_2})] & r_2 \geq 20 \mu m \end{cases}
\end{aligned} \tag{2.25}$$

The collision efficiency obtained by Eq.(2.25) is plotted in Figure 2.5. When r_1 is fixed, a reduction in collision efficiency is observed with an increase in the relative particle size $\frac{r_1}{r_2}$. When the relative particle size $\frac{r_1}{r_2}$ is fixed, collision efficiency is larger for a higher value of r_1. This reveals that aggregation between larger particles of different sizes is favoured. It should be noted that Eq.(2.25) was developed for droplets and has not been examined for soil particles so far.

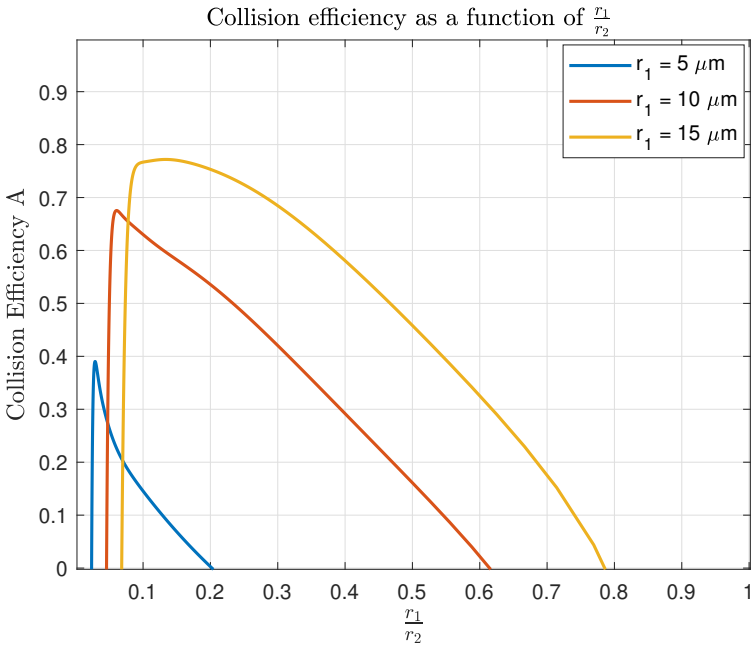


Figure 2.5: Collision efficiencies computed from Eq.(2.25) (Valioulis, 1983)

Nonetheless, collision efficiencies under the three mechanisms are often not distinguished and they are written as an empirical constant, which is estimated by data fitting technique (Golzarijalal et al., 2017; Maggi, 2005; Shen et al., 2018). A summary of the collision efficiencies proposed by various researchers is listed in the Table 2.1.

Author	Expressions	Comment
Pruppacher and Klett (1996); Maggi(2005)	$A_{i,j} = \frac{d_i^2}{2(d_i+d_j)^2}$	Collision efficiency increases with the increasing ratio $\frac{d_i}{d_j}$ (Maggi, 2005)
Spicer et al.(1996); Maggi(2005);Verney et al.(2011); Shen et al.(2018);	$A_{i,j}$ =constant	Collision efficiency is constant and is independent of particles size
Vajihinejad and Soares (2018)	$A_{i,j} = (A_{max} - A_{min})e^{-k_d t} + A_{min}$	Collision efficiency due to chain relaxation of polymers.

Table 2.1: Collision efficiencies used in literature

2.3.3. BREAKUP

Breakup occurs when the imposed external force on flocs exceed the floc strength. Turbulent shear (Spicer et al., 1996) and particle collisions (Serra and Casamitjana, 1998) can both induce breakup of flocs. Breakup process can be described by a breakup frequency function $S_{i,j}$ and by a breakup distribution function $\Gamma_{i,j}$. The breakup frequency function $S_{i,j}$ describes the amount of break-up of flocs per unit time and the breakup distribution function $\Gamma_{i,j}$ represents daughter floc size distribution due to breakup of parent flocs.

BREAKUP FREQUENCY FUNCTION

Breakup induced by turbulent shear

For turbulent shear induced floc breakup, the general form of breakup frequency function S_i can be expressed as:

$$S_i = E_b \cdot G^b \cdot d_i \tag{2.26}$$

where E_b and b are empirical values and are determined by data fitting. As shown in Eq. (2.26), the breakup frequency monotonically increases with shear rate G and particle diameter d_i.

In the study of Shen et al. (2018), a similar form of breakup function proposed by Winterwerp (1998) was used.

$$S_i = E_b G(\frac{d_i - d_P}{d_P})^{3-n_f} \cdot (\frac{\mu G}{Fy/d_i^2})^{\frac{1}{2}} \tag{2.27}$$

where d_P is the size of primary or smallest particles. F_y is the floc strength and a value of 10^{-10} Pa has been used in many studies (Maggi, 2005; Shen et al., 2018). n_f is the fractal dimension which describes fractality of flocs. A detailed elaboration of n_f can be found in the study of Maggi (2005). The typical value of n_f ranges from 2.3 to 3 (Maggi, 2005; Shen et al., 2018).

Breakup induced by particle collisions

Breakup of flocs can also be induced by particle collisions. Under this mechanism, the breakup frequency function is non-linear (Maggi, 2005; Serra and Casamitjana, 1998):

$$S_{i,j} = \sum_{j=1}^{j_{max}} \kappa_c \beta_{i,j} N_j \tag{2.28}$$

where N_j is the number density of particles belonging to size class j and κ_c is the probability of breakup induced by collision (Serra and Casamitjana, 1998). Maggi (2005) has pointed out that the knowledge about κ_c is limited and its value is negligible at low volume concentration. In addition, collisions at low shear rate can hardly cause breakup due to insufficient energy (Serra and Casamitjana, 1998). Consequently, this term is usually neglected.

Total breakup frequency function

The total breakup frequency function $S_{i,j}^{Total}$ can be written as a linear superposition of the two breakup function:

$$S_{i,j}^{Total} = E_b \cdot G^b \cdot d_i + \sum_{j=1}^{j_{max}} \kappa \beta_{i,j} N_j \tag{2.29}$$

The second term is usually neglected when breakup of flocs caused by particle collisions is not considered or is negligible.

BREAKUP DISTRIBUTION FUNCTION

It is difficult to experimentally determine the size distribution of daughter flocs produced by breakup of a parent floc. Only theoretical breakup distribution functions are available and data fitting technique has been used to find the optimal breakup distribution function that could best fit experimental data (Maggi, 2005; Shen et al., 2018). Some theoretical breakup distribution functions proposed in the previous literature are listed in Table 2.2. The binary breakup distribution means that a parent floc breaks up into two identical ones with a particle volume equal to half of the parent floc.

2.4. FLOCCULATION MODEL

Aggregation and breakup of flocs result in the change of number density, size and settling velocity of particles. Population balance equations (PBEs) have been used to describe flocculation process in many scientific research (Maggi, 2005; Quezada, Ramos, et al., 2020; Runkana et al., 2006; Shen et al., 2018; Vanni, 2000; Verney et al., 2011). PBEs can describe the time and spatial variation of the number density of particles due to aggregation and breakup process. Three popular methods are widely used to solve PBEs: discretization method(Kumar and Ramkrishna, 1996), Monte Carlo method (Hashemian and Armaou, 2016) and the method of moments(Marchisio and Fox, 2005).

Author	Expressions	Comment
Spicer et al.(1996)	$\Gamma_{i,j} = 2$, if $\frac{v_j}{v_i} = 2$ $\Gamma_{i,j} = 0$, otherwise	Binary breakup distribution;v_i is the volume of a particel belonging to size class i
Spicer et al.(1996)	$\Gamma_{i,j} = 1$, if $\frac{v_j}{v_i} = 2$ $\Gamma_{i,j} = 2$, if $\frac{v_j}{v_i} = 4$ $\Gamma_{i,j} = 0$, otherwise	Ternary breakup distribution
Spicer et al.(1996)	$\Gamma_{i,j} = \frac{v_j}{v_i} \int_{b_{i=1}}^{b_i} \frac{1}{\sqrt{2\pi}\sigma_f} exp(-\frac{(v-v_{fa})^2}{2\sigma_f^2}) dv$	Gaussian breakup distribution

Table 2.2: Some theoretical breakup distribution functions proposed in previous literature

2.4.1. POPULATION BALANCE EQUATION

A general form of PBE can be written as (Maggi, 2005):

$$\frac{\partial N_k}{\partial t} + \nabla \cdot (N_k \boldsymbol{u_k}) = G_{aggr,k} - L_{aggr,k} + G_{brk,k} - L_{brk,k} + G_{other,k} - L_{other,k} \tag{2.30}$$

where

- N_k: The number density of particles belonging to size class k. It is defined as the number of particles per unit volume.
- $G_{aggr,k}$: Gain of the number density of particles belonging to size class k from aggregation of smaller particles.
- $L_{aggr,k}$: Loss of the number density of particles belonging to size class k due to aggregation with other particles.
- $G_{brk,k}$: Gain of the number density of particles belonging to size class k due to breakup of larger particles.
- $L_{brk,k}$: Loss of the number density of particles belonging to size class k due to breakup of particles belonging to size class k.
- $G_{other,k}$,$L_{other,k}$: Gain/Loss of number density of particles belonging to size class k due to other activities (e.g., chemical and biological reactions). The two terms are not always present in Eq. (2.30). If production or destruction of particles caused by chemical or biological activities are not considered, the terms are neglected.

Other similar forms of PBEs can be found in previous research works (Golzarijalal et al., 2017; Maggi, 2005; Shen et al., 2018; Verney et al., 2011).

3

MODELLING STRATEGIES

3.1. SEDIMENT TRANSPORT

Sediment suspensions is a multiphase system which consists of one carrier fluid (i.e., sea water) and multiple dispersed phases (i.e., sediment). Each phase is a continnum characterized by a specific density, viscosity and particle size. Therefore, particles of the same size belong to one dispersed phase or size class and they occupy certain volume in space which can be described by a volume concentration.
The dynamics of this multiphase system are usually solved by three different methods: CFD-DEM (Computational Fluid Dynamics/Discrete Element Method), Euler-Euler method, and drift-flux modelling.

CFD-DEM
The most computationally intensive one is called CFD-DEM (Computational Fluid Dynamics/Discrete Element Method) (Tsuji et al., 1993). The motion of each sediment particle is described by Newton's law in DEM. The carrier flow is modelled as a continuum and its motion is solved by CFD. A major disadvantage of this method is that DEM is computationally intensive. The maximum number of sediment particles and simulation time are restricted by computational power. Therefore, this approach is not used in this study.

Euler-Euler method
In Euler-Euler method, each phase is modelled as a continnum, the motion of which is described by a phase continuity equation and a momentum equation (Gidaspow, 1994). The phase continuity equation and momentum equation need to be numerically solved for each phase separately. The phase transition terms and inter-phase forces also need to be considered for each phase. The approach is adequate for a two or three-phase system. As the number of phases increase, the approach is infeasible since the number of equations becomes prohibitively large and inter-phase forces become very complex, leading to a substantial increase in computational cost and complexity. In this study, the

simulated sediment suspensions will consist of multiple phases. Therefore, Euler-Euler method is not used in this study.

Drift-flux modelling
A multiphase drift-flux model, which can be viewed as a simplified version of Euler-Euler method, is used in this study. The phase continuity equations need to be solved for each phase, while the momentum equation only needs to be solved for the mixture (Goeree, 2018). Computational power is therefore saved. The details of drift-flux model are given in Section 3.1.1.

3.1.1. DRIFT-FLUX MODELLING

DEFINITIONS

Following quantities are defined which will be used in the following equations of drift-flux model. Similar definitions can be found in the thesis of Goeree (2018).
The volume concentration of sediment or liquid phase k is defined as the ratio of its volume v_k and the total volume of all phases v_{total}. According to this definition, the volume concentration of phase k can be expressed as:

$$\alpha_k = \frac{v_k}{v_{total}} \tag{3.1}$$

The total volume v_{total} is the summation of the volume of each phase:

$$v_{total} = \sum_{k=1}^{N_{tot}} v_k \tag{3.2}$$

where N_{tot} is the total number of phases, including dispersed phases and carrier fluid. The summation of volume concentration of each phase should be equal to one, which means:

$$\sum_{k=1}^{N_{tot}} \alpha_k = 1 \tag{3.3}$$

The density of the mixture can be calculated by:

$$\rho_m = \sum_{k=1}^{N_{tot}} \alpha_k \rho_k \tag{3.4}$$

where ρ_k is the particle density of phase k and $k = 1$ denotes the liquid phase.
If the velocity of phase k is denoted as $\boldsymbol{u_k}$, the mixture velocity $\boldsymbol{u_m}$ can be calculated by the mass weighted averaging:

$$\boldsymbol{u_m} = \sum_{k=1}^{N_{tot}} c_k \boldsymbol{u_k} \tag{3.5}$$

where c_k denotes the mass fraction, which is given by:

$$c_k = \frac{\rho_k \alpha_k}{\rho_m} \tag{3.6}$$

The relative velocity of phase k is defined as the difference between the velocity of phase k $\boldsymbol{u_k}$ and the velocity of the carrier fluid $\boldsymbol{u_f}$:

$$\boldsymbol{u_{kr}} = \boldsymbol{u_k} - \boldsymbol{u_f} \tag{3.7}$$

For the liquid phase, $\boldsymbol{u_k}$ equals to $\boldsymbol{u_f}$, which means the relative velocity $\boldsymbol{u_{kr}}$ equals to 0. The diffusion velocity $\boldsymbol{u_{km}}$ is defined as the difference between the velocity of phase k $\boldsymbol{u_k}$ and the mixture velocity $\boldsymbol{u_m}$:

$$\boldsymbol{u_{km}} = \boldsymbol{u_k} - \boldsymbol{u_m} \tag{3.8}$$

The combination Eq.(3.5), Eq.(3.6) and Eq.(3.8) gives:

$$\begin{aligned} \sum_{k=1}^{N_{tot}} \alpha_k \rho_k \boldsymbol{u_k} = \rho_m \boldsymbol{u_m} = (\sum_{k=1}^{N_{tot}} \alpha_k \rho_k) \boldsymbol{u_m} \\ \sum_{k=1}^{N_{tot}} \alpha_k \rho_k (\boldsymbol{u_k} - \boldsymbol{u_m}) = \sum_{k=1}^{N_{tot}} \alpha_k \rho_k \boldsymbol{u_{km}} = 0 \end{aligned} \tag{3.9}$$

Combining Eq.(3.5), Eq.(3.7) and Eq.(3.8), the diffusion velocity can be rewritten as:

$$\begin{aligned} \boldsymbol{u_{km}} &= \boldsymbol{u_k} - \boldsymbol{u_m} \\ &= \boldsymbol{u_k} - \sum_{k=1}^{N_{tot}} c_k \boldsymbol{u_k} \\ &= (\boldsymbol{u_{kr}} + \boldsymbol{u_f}) - \sum_{k=1}^{N_{tot}} c_k (\boldsymbol{u_{kr}} + \boldsymbol{u_f}) \\ &= \boldsymbol{u_{kr}} - \sum_{k=1}^{N_{tot}} c_k \boldsymbol{u_{kr}} \end{aligned} \tag{3.10}$$

PHASE CONTINUITY EQUATION

The phase continuity equation for phase k can be written as:

$$\frac{\partial \alpha_k \rho_k}{\partial t} + \nabla \cdot (\alpha_k \rho_k \boldsymbol{u_k}) = \Lambda_k' \tag{3.11}$$

Since ρ_k is a constant for each individual phase, Eq.(3.11) can be rewritten as:

$$\frac{\partial \alpha_k}{\partial t} + \nabla \cdot (\alpha_k \boldsymbol{u_k}) = \Lambda_k \tag{3.12}$$

where the source term $\Lambda_k = \frac{\Lambda_k'}{\rho_k}$ represents the transition from other phases into phase k in terms of volume concentration per unit time.

MOMENTUM EQUATION

The momentum equation for phase k is given by:

$$\frac{\partial \alpha_k \rho_k \boldsymbol{u_k}}{\partial t} + \nabla \cdot (\alpha_k \rho_k \boldsymbol{u_k} \boldsymbol{u_k}) = -\nabla \alpha_k p_k + \nabla \cdot (\alpha_k \boldsymbol{T_k} + \alpha_k \boldsymbol{T_k}^T) + \alpha_k \rho_k \boldsymbol{g} + \alpha_k \boldsymbol{m_k} \tag{3.13}$$

p_k is the pressure of phase k. The stress tensor $\boldsymbol{T_k}$ represents the viscous stress while the stress tensor $\boldsymbol{T_k^T}$ represents the turbulent stress. $\boldsymbol{g}$ is the gravitational acceleration vector. The source term $\boldsymbol{m_k}$ represents the interacting force between phase k and other phases.
The mixture momentum equation can be obtained by summing momentum equations for all phases.

$$\begin{aligned}&\frac{\partial \sum_{k=1}^{N_{tot}} \alpha_k \rho_k \boldsymbol{u_k}}{\partial t} + \nabla \cdot \sum_{k=1}^{N_{tot}} (\alpha_k \rho_k \boldsymbol{u_k} \boldsymbol{u_k}) \\ &= -\nabla \sum_{k=1}^{N_{tot}} \alpha_k p_k + \nabla \cdot \sum_{k=1}^{N_{tot}} (\alpha_k \boldsymbol{T_k} + \alpha_k \boldsymbol{T_k}^T) + \sum_{k=1}^{N_{tot}} \alpha_k \rho_k \boldsymbol{g} + \sum_{k=1}^{N_{tot}} \alpha_k \boldsymbol{m_k}\end{aligned} \tag{3.14}$$

Substitute of Eq.(3.4) and Eq.(3.5) into Eq.(3.14) gives the mixture momentum equation:

$$\frac{\partial \rho_m \boldsymbol{u_m}}{\partial t} + \nabla \cdot (\rho_m \boldsymbol{u_m} \boldsymbol{u_m}) = -\nabla p_m + \nabla \cdot (\boldsymbol{T_m} + \boldsymbol{T_m}^T - \sum_{k=1}^{N_{tot}} \alpha_k \rho_k \boldsymbol{u_{km}} \boldsymbol{u_{km}}) + \rho_m \boldsymbol{g} + \boldsymbol{m_m} \tag{3.15}$$

where the source term $\boldsymbol{m_m} = \sum_{k=1}^{N} \alpha_k \boldsymbol{m_k}$ equals to zero as all internal forces cancel out with each other. Thus, Eq.(3.15) can be rewritten as:

$$\frac{\partial \boldsymbol{u_m}}{\partial t} + \nabla \cdot (\boldsymbol{u_m} \boldsymbol{u_m}) = -\frac{1}{\rho_m} \nabla p_m + \frac{1}{\rho_m} \nabla \cdot (\boldsymbol{T_m} + \boldsymbol{T_m}^T) - \nabla \cdot (\sum_{k=1}^{N_{tot}} c_k \boldsymbol{u_{km}} \boldsymbol{u_{km}}) + \boldsymbol{g} \tag{3.16}$$

3.2. FLOCCULATION MODELLING

3.2.1. THE DISCRETIZED FORM OF POPULATION BALANCE EQUATION

In this study, a discretized form of PBEs is used (Hounslow et al., 1988; Spicer and Pratsinis, 1996). Different from the original PBEs present in the literature, an advection term $\nabla \cdot (\alpha_k u_k)$ is included in this study to account for the bulk motion of sediment. In this discretized method, the continuous particle size distribution is divided into discrete number of size classes. Each size class is characterized by a specific diameter. Particles of the same size belong to one size class. The division of different size classes is based on:

$$\frac{v_{i+1}}{v_i} = 2 \tag{3.17}$$

If particles are spherical, Eq.(3.17) can be rewritten as:

$$\frac{d_{i+1}}{d_i} = 2^{\frac{1}{3}} \tag{3.18}$$

Based on this rule, the discretized form of PBE can be written as:

$$\begin{aligned}\frac{dN_k}{dt} + \nabla \cdot (N_k \boldsymbol{u_k}) = &\sum_{j=1}^{k-2} 2^{j-k+1} A_{k-1,j} \beta_{k-1,j} N_{k-1} N_j + \frac{1}{2} A_{k-1,k-1} \beta_{k-1,k-1} N_{k-1}^2 \\ &- \sum_{j=1}^{k-1} 2^{j-k} A_{k,j} \beta_{k,j} N_k N_j - \sum_{j=k}^{k_{max}} A_{k,j} \beta_{k,j} N_k N_j - S_k N_k + \sum_{j=k+1}^{k_{max}} \Gamma_{k,j} S_j N_j\end{aligned} \tag{3.19}$$

HA, β, S and Γ have already been defined in Section 2.3.

- $A_{k,j}$: the efficiency of collision between two particles belonging to size class k and j.
- $\beta_{k,j}$: the frequency of collision between two particles belonging to size class k and j.
- S_k: the breakup frequency of particles belonging to size class k.
- $\Gamma_{k,j}$: the breakup distribution function, which represents the percentage of particles belonging to size class k produced by the breakup of particles of size class j.

The meaning of each term in Eq.(3.19) is listed below.

- $\frac{dN_k}{dt}$: the change of the number density N_k of particles belonging to size class k over time.
- $\nabla \cdot (N_k \boldsymbol{u_k})$: the spatial advection of N_k.
- $\sum_{j=1}^{k-2} 2^{j-k+1} A_{k-1,j} \beta_{k-1,j} N_{k-1} N_j$: the formation of particles belonging to size class k by aggregation of smaller particles of unequal size.
- $\frac{1}{2} A_{k-1,k-1} \beta_{k-1,k-1} N_{k-1}^2$: the formation of particles belonging to size class k by aggregation of equal-sized particles belonging to size class k-1.
- $-\sum_{j=1}^{k-1} 2^{j-k} A_{k,j} \beta_{k,j} N_k N_j$: the loss of particles belonging size class k due to their aggregation with smaller particles.
- $-\sum_{j=k}^{k_{max}} A_{k,j} \beta_{k,j} N_k N_j$: the loss of particles belonging size class k due to their aggregation with equal-sized or larger particles. k_{max} is the total number of dispersed phases.
- $-S_k N_k$: the loss of particles belonging to size class k due to breakup of particles belonging to size class k.
- $\sum_{j=k+1}^{k_{max}} \Gamma_{k,j} S_j N_j$: the formation of particles belonging to size class k due to breakup of larger particles.

Besides, three additional assumptions are made to guarantee the mass conservation:

- Primary or smallest particles cannot break up any more.
- Largest particles cannot aggregate any more.
- Soil particles will not be transformed into water particles.

3.3. CFD-PBE COUPLING

To model the effect of flocculation, no additional equation is needed to couple fluid dynamics and population dynamics. The number density can be related the volume concentration by particle volume which is constant over time.

$$N_k = \frac{\alpha_k}{v_k} \tag{3.20}$$

Using Eq.(3.20), Eq.(3.19) can be reformulated in terms of volume concentration:

$$\begin{aligned}&\frac{\partial \alpha_k}{\partial t} + \nabla \cdot (\alpha_k \boldsymbol{u_k}) \\ &= (\sum_{j=1}^{k-2} 2^{j-k+1} A_{k-1,j} \beta_{k-1,j} N_{k-1} N_j + \frac{1}{2} A_{k-1,k-1} \beta_{k-1,k-1} N_{k-1}^2 - \sum_{j=1}^{k-1} 2^{j-k} A_{k,j} \beta_{k,j} N_k N_j \\ &- \sum_{j=k}^{k_{max}} A_{k,j} \beta_{k,j} N_k N_j - S_k N_k + \sum_{j=k+1}^{k_{max}} \Gamma_{k,j} S_j N_j) v_k\end{aligned} \tag{3.21}$$

Eq.(3.12) and Eq.(3.21) are intrinsically the same equation. The phase transition term Λ_k in Eq.(3.12) equals to the RHS of Eq.(3.21) which represents the transition into the dispersed phase k from other dispersed phases due to aggregation and breakup. Thus, we can obtain that:

$$\begin{aligned}\Lambda_k = &(\sum_{j=1}^{k-2} 2^{j-k+1} A_{k-1,j} \beta_{k-1,j} N_{k-1} N_j + \frac{1}{2} A_{k-1,k-1} \beta_{k-1,k-1} N_{k-1}^2 - \sum_{j=1}^{k-1} 2^{j-k} A_{k,j} \beta_{k,j} N_k N_j \\ &- \sum_{j=k}^{k_{max}} A_{k,j} \beta_{k,j} N_k N_j - S_k N_k + \sum_{j=k+1}^{k_{max}} \Gamma_{k,j} S_j N_j) v_k\end{aligned} \tag{3.22}$$

3.4. TURBULENCE MODELLING

Turbulence is the motion of fluid where the velocity and pressure field develop chaotically. DNS (Direct Numerical Simulation) can solve the Navier-Stokes equations numerically without any turbulence model. However, very small time step and grid size are required to capture the turbulence, which leads to very large computational cost. To save computational power, an adequate turbulence model is needed to describe time and space dependent turbulent behaviour of multiphase flows. Two turbulence models are widely used in numerical simulations: Reynolds averaged Naiver-Stokes (RANS) and Large Eddy Simulations (LES). Compared to LES, RANS requires less computational cost. Besides, RANS has been successfully implemented to simulate flocculation process in turbulent flows (Klassen, 2017; Lee Byung Joon, 2014). In this study, a buoyant $k-\epsilon$ RANS turbulence model is used.

3.4.1. RANS EQUATIONS

In RANS model, an instantaneous quantity $f(\boldsymbol{x}, t)$ is decomposed into one time-averaged part $\bar{f}(\boldsymbol{x}, t)$ and one fluctuating part $f^{'}(\boldsymbol{x}, t)$.

$$f(\boldsymbol{x}, t) = \bar{f}(\boldsymbol{x}, t) + f'(\boldsymbol{x}, t) \tag{3.23}$$

The time-averaged part can be expressed as:

$$\bar{f}(\boldsymbol{x}, t) = \lim_{N \to \infty} \sum_{n=1}^{N} f_n(\boldsymbol{x}, t) \tag{3.24}$$

where $\boldsymbol{x}$ is the position vector and t is the time. The time-averaged values of fluctuating quantities are zero.

$$\overline{f'}(\boldsymbol{x}, t) = 0 \tag{3.25}$$

By applying Eq.(3.23) into Eq.(3.11), the following ReynoldS-averaged phase continuity equation for phase k can be obtained:

$$\frac{\partial \overline{\alpha_k}}{\partial t} + \nabla \cdot (\overline{\alpha_k}\, \overline{\boldsymbol{u_k}}) = -\nabla \cdot (\overline{\alpha'_k \boldsymbol{u'_k}}) + \overline{\Lambda_k} \tag{3.26}$$

where $-\nabla \cdot (\overline{\alpha'_k \boldsymbol{u'_k}})$ represents turbulent diffusion. It can be approximated by:

$$-\nabla \cdot (\overline{\alpha'_k \boldsymbol{u'_k}}) \approx -\nu_t \nabla^2 \overline{\alpha_k} \tag{3.27}$$

ν_t is the turbulent viscosity which can be calculated by Eq.(3.36). According to Eq.(3.23) and Eq.(3.25), $\overline{\Lambda_k + \Lambda'_k} = \overline{\Lambda_k}$, where the time-averaged value of the instantaneous term is zero. Thus, the time-averaged phase transition term $\overline{\Lambda_k}$ caused by flocculation can be expressed as:

$$\overline{\Lambda_k} = \overline{\Lambda_{k,1}} + \overline{\Lambda_{k,2}} + \overline{\Lambda_{k,3}} + \overline{\Lambda_{k,4}} + \overline{\Lambda_{k,5}} + \overline{\Lambda_{k,6}} \tag{3.28}$$

where

$$
\begin{aligned}
\overline{\Lambda_{k,1}} &= \sum_{j=1}^{k-2} 2^{j-k+1} A_{k-1,j} \beta_{k-1,j} \overline{(\overline{N_{k-1}} + N'_{k-1})(\overline{N_j} + N'_j)} \upsilon_k \\
&= \sum_{j=1}^{k-2} 2^{j-k+1} A_{k-1,j} \beta_{k-1,j} \overline{N_{k-1}}\, \overline{N_j} \upsilon_k + \cancel{\sum_{j=1}^{k-2} 2^{j-k+1} A_{k-1,j} \beta_{k-1,j} \overline{N'_{k-1} N'_j} \upsilon_k} \\
\overline{\Lambda_{k,2}} &= \frac{1}{2} A_{k-1,k-1} \beta_{k-1,k-1} \overline{(\overline{N_{k-1}} + N'_{k-1})(\overline{N_{k-1}} + N'_{k-1})} \upsilon_k \\
&= \frac{1}{2} A_{k-1,k-1} \beta_{k-1,k-1} \overline{N_{k-1}}\, \overline{N_{k-1}} \upsilon_k + \cancel{\frac{1}{2} A_{k-1,k-1} \beta_{k-1,k-1} \overline{N'_{k-1} N'_{k-1}} \upsilon_k} \\
\overline{\Lambda_{k,3}} &= -\sum_{j=1}^{k-1} 2^{j-k} A_{k,j} \beta_{k,j} \overline{(\overline{N_k} + N'_k)(\overline{N_j} + N'_j)} \upsilon_k \\
&= -\sum_{j=1}^{k-1} 2^{j-k} A_{k,j} \beta_{k,j} \overline{N_k}\, \overline{N_j} \upsilon_k - \cancel{\sum_{j=1}^{k-1} 2^{j-k} A_{k,j} \beta_{k,j} \overline{N'_k N'_j} \upsilon_k} \\
\overline{\Lambda_{k,4}} &= -\sum_{j=k}^{k_{max}} A_{k,j} \beta_{k,j} \overline{(\overline{N_k} + N'_k)(\overline{N_j} + N'_j)} \upsilon_k \\
&= -\sum_{j=k}^{k_{max}} A_{k,j} \beta_{k,j} \beta_{k,j} \overline{N_k}\, \overline{N_j} \upsilon_k - \cancel{\sum_{j=k}^{k_{max}} A_{k,j} \beta_{k,j} \overline{N'_k N'_j} \upsilon_k} \\
\overline{\Lambda_{k,5}} &= -S_k \overline{(\overline{N_k} + N'_k)} \upsilon_k = -S_k \overline{N_k} \upsilon_k \\
\overline{\Lambda_{k,6}} &= \sum_{j=k+1}^{k_{max}} \Gamma_{k,j} S_j \overline{(\overline{N_j} + N'_j)} \upsilon_k \\
&= \sum_{j=k+1}^{k_{max}} \Gamma_{k,j} S_j \overline{N_j} \upsilon_k
\end{aligned}
\tag{3.29}
$$

$\overline{N'N'}$ represents the local number density fluctuations. This term can be assumed to be small and can thus be neglected. Besides, the author has not found a way to approximate this term.
Similarly, the time-averaged mixture momentum equation is given by:

$$
\frac{\partial \overline{\boldsymbol{u_m}}}{\partial t} + \nabla \cdot (\overline{\boldsymbol{u_m}}\, \overline{\boldsymbol{u_m}}) = -\frac{\nabla \overline{p_m}}{\overline{\rho_m}} + \frac{1}{\overline{\rho_m}} \nabla \cdot (\overline{\boldsymbol{T_m}} + \overline{\boldsymbol{T_m}^T}) - \nabla \cdot (\sum_{k=1}^{N_{tot}} c_k \overline{\boldsymbol{u_{km}} \boldsymbol{u_{km}}}) + \overline{\boldsymbol{g}} \tag{3.30}
$$

The time-averaged viscous stress term is expressed as:

$$
\overline{\boldsymbol{T_m}} = \mu \nabla \overline{\boldsymbol{u_m}} \tag{3.31}
$$

The turbulent stress $\overline{\boldsymbol{T_m^T}}$ is:

$$
\overline{\boldsymbol{T_m^T}} = -\overline{\rho_m \boldsymbol{u}' \boldsymbol{u}'} \tag{3.32}
$$

The term $\overline{\boldsymbol{u_{km}} \boldsymbol{u_{km}}}$ is approximated by Eq.(3.33). The fluctuation is also neglected.

$$
\overline{\boldsymbol{u_{km}} \boldsymbol{u_{km}}} \approx \overline{\boldsymbol{u_{km}}}\, \overline{\boldsymbol{u_{km}}} \tag{3.33}
$$

3.4.2. BUOYANT K-ϵ MODEL

In this multiphase problem, the presence of dispersed phases will influence the turbulence. It has been found that unequal distribution of the dispersed phases in the settling tank can cause density stratification which will produce turbulent kinetic energy k (Brennan, 2001). To model this effect, a buoyancy modification term is added to the standard k-ϵ model developed by Rodi (2017). This modified turbulence model is known as buoyant k-ϵ model, in which an additional buoyancy generation term G_k is added to model the influence of density stratification on turbulent kinetic energy k.
The turbulent kinetic energy is defined as:

$$k = \frac{1}{2}|\boldsymbol{u}'|^2 = \frac{1}{2}(u'^2 + v'^2 + w'^2) \tag{3.34}$$

The Reynolds stress in Eq.(3.32) is calculated by:

$$-\overline{u_i' u_j'} = \nu_t\left(\frac{\partial \overline{u_i}}{\partial x_j} + \frac{\partial \overline{u_j}}{\partial x_i}\right) - \frac{2}{3}k\delta_{ij} \tag{3.35}$$

where ν_t is the eddy viscosity which is given by Eq.(3.36) and δ_{ij} is the Kronecker δ.

$$\nu_t = C_\nu \frac{k^2}{\epsilon} \tag{3.36}$$

where C_ν is a constant which equals to 0.09. This value is obtained by data fitting technique for numerous turbulent flows (Versteeg and Malalasekera, 1995).
Buoyant k-ϵ is a two-equation model, which uses transport equations of turbulent kinetic energy k and of turbulent dissipation rate ϵ to achieve the closure. The turbulent kinetic energy transport equation is given by:

$$\frac{\partial k}{\partial t} + \nabla\cdot(k\boldsymbol{u}) = \nabla\cdot\left(\nu + \frac{\nu_t}{\sigma_k}\nabla k\right) + P_k + G_k - \rho\epsilon \tag{3.37}$$

where P_k is the turbulent kinetic energy production term, which is given by:

$$P_k = \nu_t\left(\frac{\partial \overline{u_i}}{\partial x_j} + \frac{\partial \overline{u_j}}{\partial x_i}\right)\frac{\partial \overline{u_i}}{\partial x_j} \tag{3.38}$$

G_k is the buoyancy production or consumption term:

$$G_k = \frac{g_i}{\rho_r}\frac{\nu_t}{\sigma_t}\frac{\partial \rho}{\partial x_i} \tag{3.39}$$

The turbulent dissipation rate transport equation is given by:

$$\frac{\partial \epsilon}{\partial t} + \nabla\cdot(\epsilon\boldsymbol{u}) = \nabla\cdot\left(\nu + \frac{\nu_t}{\sigma_k}\nabla\epsilon\right) + C_{1\epsilon}(P_k + G_k - C_{3\epsilon}G_k) - C_{2\epsilon}\frac{\epsilon^2}{k} \tag{3.40}$$

The coefficients used in Eq.(3.38) and Eq.(3.40) is given in Table 3.1.
The turbulent boundary layer exists near the wall. Very fine mesh size is required to resolve this near-wall region. Therefore, substantial computational power is required.

$C_{1\epsilon}$	$C_{2\epsilon}$	$C_{3\epsilon}$	σ_k	σ_ϵ	σ_t
1.44	1.92	0.8	1.0	1.3	1.0

Table 3.1: Coefficients used in Eq.(3.38) and Eq.(3.40) (Versteeg and Malalasekera, 1995).

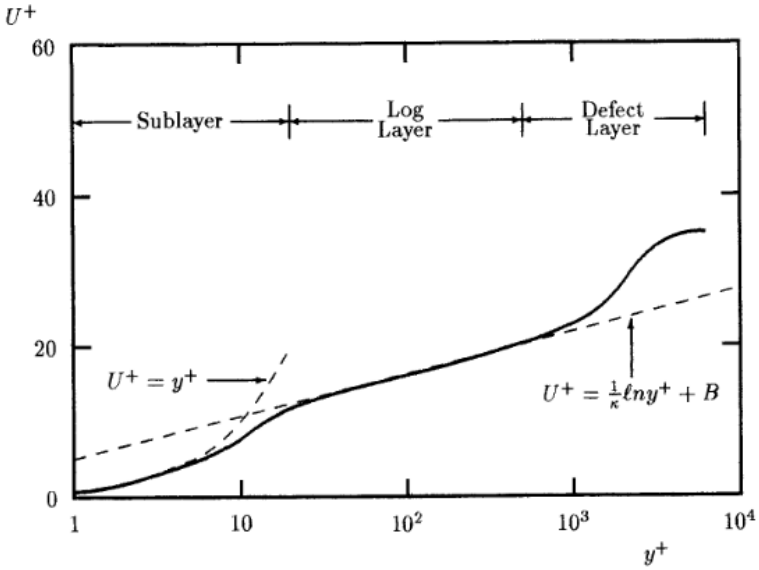


Figure 3.1: Typical velocity profile within turbulent boundary layer (Wilcox, 2006). The meaning of U^+, y^+, κ and B is given in Eq.(3.41).

An alternative approach is to use wall functions, where the Law of the Wall formulation is applied to capture the physics between the wall and the adjacent cell centre. The Law of the Wall introduced by Kármán (1931) states the proportional relationship between the dimensionless velocity U^+ at certain point near the wall and the dimensionless wall distance y^+ which is defined in Eq.(3.41) given by Gersten (2009). Using wall functions leads to a significant reduction of computational costs. The wall-adjacent cell centre needs to be located at $30 < y^+ < 300$, which is in the log layer where Law of Wall holds (see Figure 3.1).

$$\begin{aligned} U^+ &= \frac{1}{\kappa} y^+ + B^+ \\ U^+ &= \frac{U}{U_\tau} \\ U_\tau &= \sqrt{\frac{\tau_w}{\rho}} \\ y^+ &= \frac{y U_\tau}{\nu} \end{aligned} \tag{3.41}$$

where:

U^+: the dimensionless velocity

κ: the Von Kármán constant

C^+: a constant

U: the velocity parallel to the wall

U_τ: the friction velocity

τ_w: the wall shear stress

y: the absolute distance from the wall

4

NUMERICAL IMPLEMENTATION

4.1. FINITE VOLUME METHOD

4.1.1. DISCRETIZATION

Finite volume method (FVM) is used for discretization of partial differential equations (PDEs). The computational domain is divided into control volumes (CV) which are called cells. In the multi-phase drift-flux solver, variables are saved at each cell centre as collocated grids are used. In each cell, PDEs are integrated over the volume.

DISCRETIZATION OF PHASE CONTINUITY EQUATION

In a fixed 2D cell Ω_{ij}, the volume integral of the phase continuity equation is:

$$\frac{\partial \int_{\Omega_{ij}} \alpha_k d\Omega}{\partial t} + \int_{\Omega_{ij}} \nabla \cdot (\alpha_k \boldsymbol{u_k}) d\Omega = \int_{\Omega_{ij}} \Lambda_k d\Omega \tag{4.1}$$

The advection and diffusion terms can be converted to surface fluxes using the divergence theorem. The surface integral can be approximated by mid-point rule (see Appendix C of Goeree, 2018). Thus, the second term on the LHS of Eq.(4.1) becomes:

$$\int_{\Omega_{ij}} \nabla \cdot (\alpha_k \boldsymbol{u_k}) d\Omega = \oint_{\partial\Omega_{ij}} \alpha_k \boldsymbol{u_k} \cdot \hat{\boldsymbol{n}} dS \approx \sum (\alpha_k \boldsymbol{u_k})_f \cdot \hat{\boldsymbol{n}}_f A_f \tag{4.2}$$

where the subscript f denotes the quantities at cell faces. If notations shown in Figure 4.1 are used, Eq.(4.2) can be further written as:

$$\begin{aligned} &\sum (\alpha_k \boldsymbol{u_k})_f \cdot \hat{\boldsymbol{n}}_f A_f \\ &= (\alpha_k u_k A)_e - (\alpha_k u_k A)_w + (\alpha_k u_k A)_n - (\alpha_k u_k A)_s \\ &= \alpha_{k,e} u_{k,e} (y_n - y_s) - \alpha_{k,w} u_{k,w} (y_n - y_s) + \alpha_{k,n} u_{k,n} (x_e - x_w) - \alpha_{k,s} u_{k,s} (x_e - x_w) \end{aligned} \tag{4.3}$$

These fluxes at the faces of non-boundary cells can be interpolated using variables defined at cell centres, while the fluxes at boundaries of the computation domain are prescribed by corresponding boundary conditions.

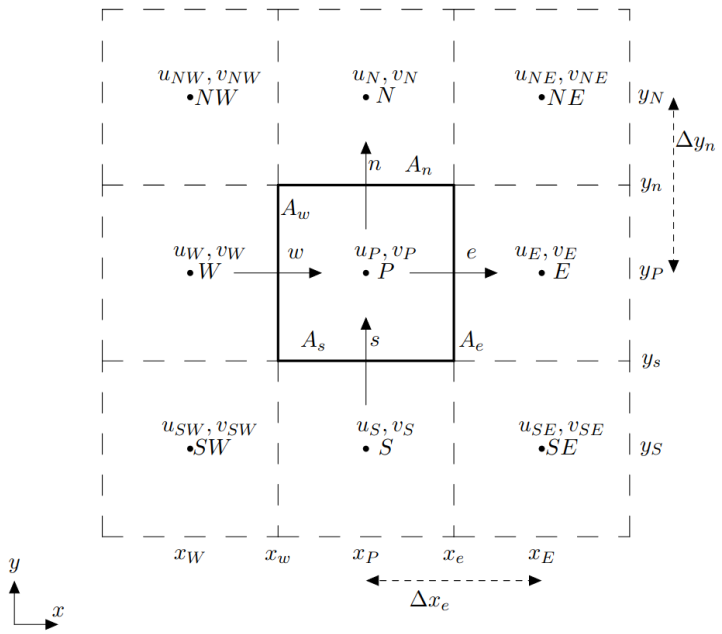


Figure 4.1: A 2D Cartesian grid using compass notations. P, E, W, N and S represent the central, eastern, western, northern and southern cell centre respectively. The lower case e, w, n and s represent interpolated values. (Image from Goeree, 2018)

DISCRETIZATION OF THE MIXTURE MOMENTUM EQUATION

The volume integral of the mixture momentum equation is:

$$\frac{\partial \int_{\Omega_{ij}} \boldsymbol{u_m} d\Omega}{\partial t} + \int_{\Omega_{ij}} \nabla \cdot (\boldsymbol{u_m u_m}) d\Omega = -\int_{\Omega_{ij}} \nabla \frac{p_m}{\rho_m} d\Omega + \int_{\Omega_{ij}} \frac{1}{\rho_m} \nabla \cdot (\boldsymbol{T_m} + \boldsymbol{T_m}^T) d\Omega - \int_{\Omega_{ij}} \nabla \cdot (\sum_{k=1}^{N_{tot}} c_k \boldsymbol{u_{km} u_{km}}) d\Omega + \int_{\Omega_{ij}} \boldsymbol{g} d\Omega \tag{4.4}$$

The volume integral of advection terms can be handled in the same fashion shown in Eq.(4.2) and Eq.(4.3) and is not repeated here.

4.1.2. INTERPOLATION

To interpolate quantities defined at cell faces, three numerical interpolation schemes are usually used: upwind shceme, central scheme and Total Variational Diminishing (TVD) schemes.

UPWIND SCHEME

First-order upwind scheme is the simplest form of upwind schemes. The notations shown in Figure 4.1 are used to explain this scheme. If a physical quantity ϕ is transported from the west towards right $u > 0$, the western cell face is called upwind side and the eastern

is called downwind side. If $u < 0$ which means that ϕ is transported in an opposite direction, the western cell face is called downwind side and the eastern is called upwind side. Taking ϕ_e as an example, the first-order upwind approximation can be written as:

$$\phi_e = \begin{cases} \phi_P & \text{if } u_e > 0 \\ \phi_E & \text{if } u_e < 0 \end{cases} \tag{4.5}$$

The expression is similar in 3D cases. This first-order upwind scheme is simple and does not induce numerical oscillations. However, it is only first-order accurate and its stability is guaranteed only if the Courant–Friedrichs–Lewy (CFL) condition is satisfied (Hirsch, 2007). Besides, when flow direction does not align with grid lines, severe false diffusion appears, which will cause large errors. To solve this problem, grid refinement is required.

CENTRAL SCHEME

Considering ϕ_e at the eastern face of a uniform grid, its value can also be approximated by a central scheme:

$$\phi_e = \frac{\phi_E + \phi_P}{2} \tag{4.6}$$

For non-uniform grids, ϕ_e is approximated by:

$$\phi_e = \frac{x_e - x_P}{x_E - x_P}\phi_E + \frac{x_E - x_e}{x_E - x_P}\phi_P \tag{4.7}$$

The central scheme is second-order accurate. Its accuracy can be guaranteed when Peclet number Pe is smaller than 2 (Liu and Tadmor, 1998). Otherwise, numerical oscillations and divergence will occur (Hundsdorfer and Verwer, 2013).

TVD SCHEMES

Unphysical negative sediment concentrations may appear due to numerical oscillations. These negative values should be avoided when using the multi-phase drift-flux solver to simulate sediment transport since they will lead to the divergence of the numerical solutions. To prevent negative concentrations, so-called flux limiters are used in the TVD schemes. In this thesis, a classical second-order symmetric van Leer limiter, is applied to the advection term of concentration to avoid negative concentrations (Van Leer, 1974). By using this limiter, the sediment concentration can be limited within the range from 0 to 1 (LeVeque et al., 2002).
The van Leer limiter function can be written as:

$$\Phi_{vl}(\theta) = \frac{\theta + |\theta|}{1 + |\theta|} \tag{4.8}$$

where Φ_{vl} is the van Leer flux limiter function and θ is the smoothness indicator for two successive cells, which is given by:

$$\theta = \frac{u_i - u_{i-1}}{u_{i+1} - u_i} \tag{4.9}$$

The van Leer flux limiter as a function of the smoothness indicator θ is illustrated in Figure 4.2.

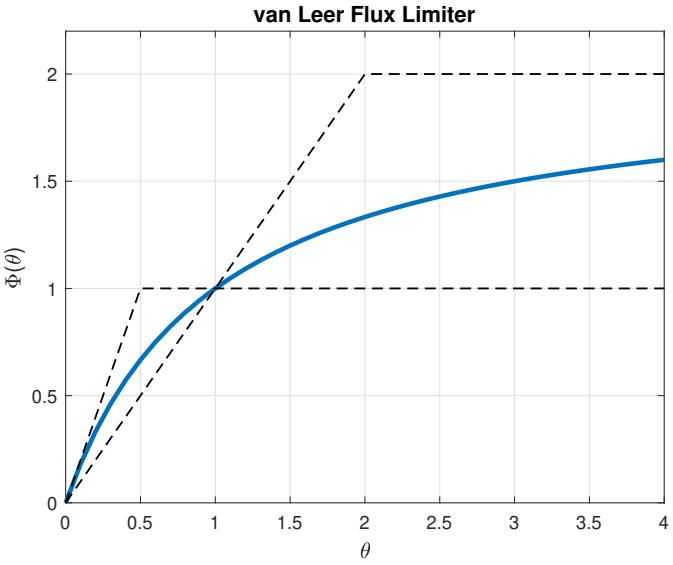


Figure 4.2: van Leer flux limiter (Van Leer, 1974)

4.2. NUMERICAL ALGORITHM

The flow chart of the numerical algorithm used in this study is shown in Figure 4.3. The PIMPLE algorithm is used to solve pressure-velocity coupling. PIMPLE is a combination of PISO (Pressure Implicit with Splitting of Operator) and SIMPLE (Semi-Implicit Method for Pressure-Linked Equations).
The steps of the algorithm are described below:

1. Set the initial concentration field for each phase, the pressure field, and the velocity field of the mixture.
2. Compute the diffusion velocity $\boldsymbol{U_{km}}$. First, the terminal settling velocity is calculated using Eq.(2.14). Subsequently, the hindered settling velocity, which is also the relative velocity $\boldsymbol{U_{kr}}$, is determined by combining Eq.(2.15), Eq.(2.18) and Eq.(2.17). Finally, the diffusion velocity $\boldsymbol{U_{km}}$ can be determined by Eq.(3.10).
3. Determine the phase transition term Λ_k caused by flocculation.
4. Solve the phase continuity equations for each phase.
5. Compute the density and the viscosity of the mixture.
6. Use PIMPLE loop to check the convergence. Repeat this outer loop (step 6-8) until the velocity field is converged or the maximum number of outer correctors is exceeded.
7. Use the momentum predictor to predict the velocity field of the mixture in U equation.

8. Correct the pressure field and the velocity field by inner correctors. Repeat this step until the maximum number of inner correctors is exceeded.
9. Solve the turbulence model.
10. Update the mixture density.
11. Go to next time step and repeat step 2 to 9 until the end of simulation is reached.

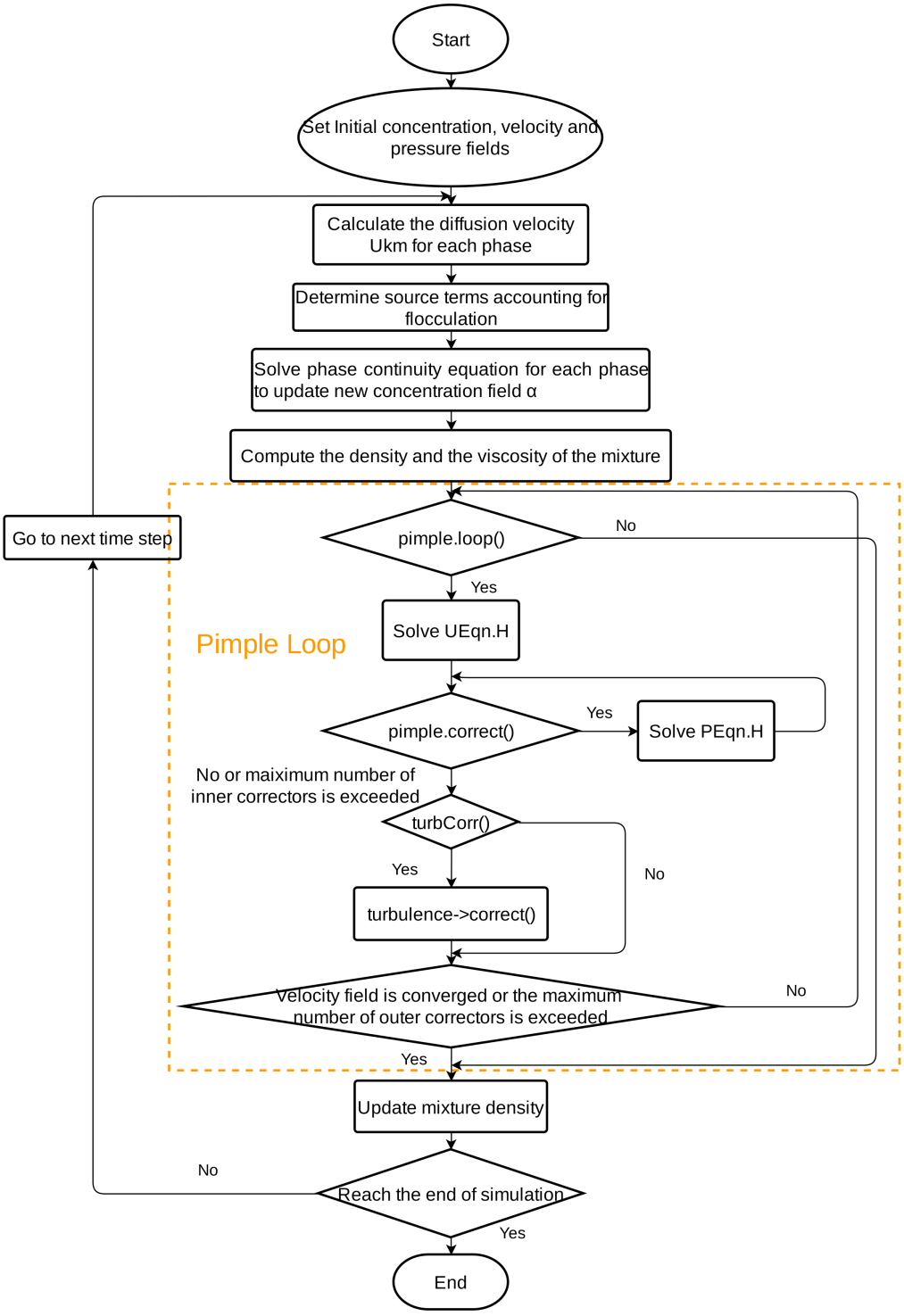


Figure 4.3: Flow chart of the algorithm. A detailed explanation of the PIMPLE algorithm is given by Holzmann (2019).

5

NUMERICAL VERIFICATION

In this chapter, numerical verification is performed to identify whether the implementation of the conceptual flocculating settling model is mathematically correct. In following sections, numerical solutions will be compared to the results from mathematical analysis for different cases in order to examine mass conservation and species conservation. Besides, iterative convergence of numerical results will also be checked.

5.1. DEFINITION

Verification is defined as the process to evaluate how accurately a model implementation represents developers' conceptual model (AIAA Standards, 1998). It examines whether numerical solutions are mathematically correct.
The verification activities in this study include:

- Check conservation relationships for numerical solutions. For example, the mass of sediment should be conserved.
- Iterative convergence study is conducted to check whether residuals reach a specified tolerance. To obtain a tightly converged numerical solution, the residuals should decrease as the simulation progresses.

5.2. SIMULATION SETUP

5.2.1. MESH

The dimension of the 2D settling column used in the simulation is shown in Figure 5.1. The diameter and the height of the settling column are 0.11 m and 0.365 m respectively. This large dimension is used to minimize wall effects (Sanz, 2018). Uniform 2D rectilinear grids, which are 5.2 mm in height and 5.5 mm in width, are used in the following simulations.

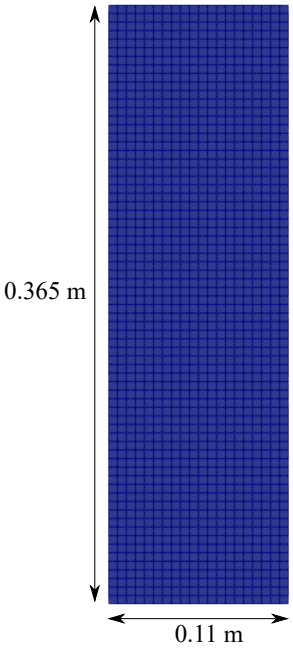


Figure 5.1: Dimension and mesh setup of the 2D simulation domain

5.2.2. INITIAL AND BOUNDARY CONDITIONS

Initial conditions vary for different verification purposes and they will be provided in following sections.

An illustration of the boundary conditions is shown in Figure 5.2. The simulation domain is closed and no fluid enters or leaves through boundaries. For velocity field, no slip boundary condition is used at walls. For concentration field and pressure field, their gradient are zero at walls.

5.3. EXAMINATION OF CONSERVATION

5.3.1. MASS CONSERVATION

The phase continuity equation of phase k can be written as:

$$\begin{aligned}\frac{\partial \alpha_k}{\partial t} + \nabla \cdot \alpha_k \boldsymbol{u_k} = &\sum_{j=1}^{k-2} 2^{j-k+1} A\beta_{k-1,j} N_{k-1} N_j + \frac{1}{2} A\beta_{k-1,k-1} N_{k-1}^2 - \sum_{j=1}^{k-1} 2^{j-k} A\beta_{k,j} N_k N_j \\ &- \sum_{j=k}^{k_{max}} A\beta_{k,j} N_j N_k - S_k N_k + \sum_{j=k+1}^{k_{max}} \Gamma_{k,j} S_j N_j) v_k \\ &= \Lambda_k \end{aligned} \tag{5.1}$$

In Eq.(5.1), the source term Λ_k represents the transition from other phases into phase k.

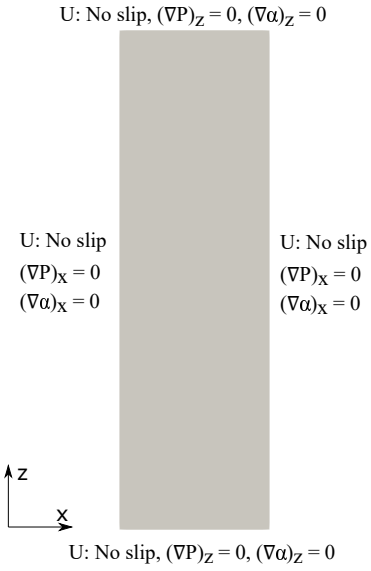


Figure 5.2: Boundary conditions used for verification process. No slip wall boundary condition for velocity field and zero gradient condition for concentration and pressure field.

Since aggregation and break-up only cause inter-phase transitions among the dispersed phases, the summation of Λ_k is zero.

$$\sum_{k=1}^{k_{max}} \Lambda_k = 0 \tag{5.2}$$

Eq.(5.2) can be mathematically proven. If three phases are considered and binary breakup distribution function is used, the phase transition term for each phase can be written as:

$$\begin{aligned}
\Lambda_1 &= -A\beta_{1,1}N_1N_1v_1 - A\beta_{1,2}N_1N_2v_1 + 2S_2N_2v_1 \\
\Lambda_2 &= \frac{1}{2}A\beta_{1,1}N_1^2v_2 - \frac{1}{2}A\beta_{2,1}N_2N_1v_2 - A\beta_{2,2}N_2N_1v_2 - S_2N_2v_2 + 2S_3N_3v_2 \\
\Lambda_3 &= \frac{1}{2}A\beta_{1,2}N_2N_1v_3 + \frac{1}{2}A\beta_{2,2}N_2^2v_3 - S_3N_3v_3
\end{aligned} \tag{5.3}$$

The relationship of particle volume of different size classes satisfy $\frac{v_{k+1}}{v_k} = 2$. The collision frequency function is symmetric, which means $\beta_{k,j} = \beta_{j,k}$. Consequently, the terms of the matching colour in Eq.(5.3) will cancel each other out when taking summation of the three equations. This gives the conservation relationship of total volume concentration of solid phases.

$$\begin{aligned}
&\frac{\partial \sum_{k=1}^{k_{max}} \alpha_k}{\partial t} + \nabla \cdot (\sum_{k=1}^{k_{max}} \alpha_k \boldsymbol{u_k}) = \sum_{k=1}^{k_{max}} \Lambda_k = 0 \\
&\frac{\partial \alpha_s}{\partial t} + \nabla \cdot (\sum_{k=1}^{k_{max}} \alpha_k \boldsymbol{u_k}) = 0
\end{aligned} \tag{5.4}$$

By integrating Eq. (5.4) over the simulation domain, it can be found that the total volume concentration of solid phases is time invariant.

$$\frac{\partial \iiint_V \alpha_s dV}{\partial t} + \iiint_V \nabla \cdot (\sum_{k=1}^{k_{max}} \alpha_k \boldsymbol{u_k}) dV = \frac{\partial \iiint_V \alpha_s dV}{\partial t} + \cancel{\oiint_S \sum_{k=1}^{k_{max}} \alpha_k \boldsymbol{u_k} \cdot \hat{n} d\boldsymbol{S}} = 0$$
$$\iiint_V \alpha_s dV = \text{constant} \tag{5.5}$$

To examine the conservation relationship given by Eq.(5.5), a numerical simulation is performed using three size classes given in Table 5.1 and flocculation parameters given in Table 5.2. For a general purpose, all the three mechanisms governing collision frequency are implemented, although Brownian motion can be neglected for these large particles. In addition, in this verification stage, numerical parameters have not been calibrated yet. Considering the purpose of the verification process is only to check that numerical results are mathematically correct, numerical parameters proposed in previous literature can be used. Hence, the breakup frequency function used in the study of Golzarijalal et al. (2017) is employed here, although it may produce physically inaccurate results.

Conditions / Size Class	Initial Volume Concentration α_i	Diameter d_i [μm]	Particle Settling Velocity [mm/s]
1	0.3%, homogeneous	75	4.94
2	0.3%, homogeneous	94.5	7.59
3	0.3%, homogeneous	119.1	11.54

Table 5.1: Size classes and initial conditions used to check the mass conservation. The settling velocity is calculated by Eq.(2.14). The concentration-dependent hindered settling effect will be considered in the simulation.

The result is shown in Figure 5.3 which plots the average volume concentration of each phase and the average volume concentration of all three solid phases over time. The average volume concentration of three solid phases is a constant equal to 0.09 over time, which has proven Eq.(5.5). The total mass fraction of solid phases is calculated as:

$$m_s = \rho_s \iiint_V \alpha_s dV \tag{5.6}$$

Since the sediment density ρ_s is a constant, the total mass of solid phases m_s is also conserved if the volume concentration is conserved. The slopes of the curves ($\frac{\partial \alpha}{\partial t}$) in Figure 5.3 are dependent on corresponding source terms. The volume-averaged source terms $\frac{1}{V}\int\int\int\Lambda_1 dV$, $\frac{1}{V}\int\int\int\Lambda_2 dV$ and $\frac{1}{V}\int\int\int\Lambda_3 dV$ and their summation $\frac{1}{V}\int\int\int(\Lambda_1 + \Lambda_2 + \Lambda_3) dV$ are shown in Figure 5.4.

Density of water $[kg/m^3]$	1000
Density of sediment particles $[kg/m^3]$	2650
Temperature [K]	293
Collision efficiency $A_{i,j}$	1
Collision frequency function $\Lambda_{i,j}$	$\beta_{i,j} = \frac{2KT(d_i+d_j)^2}{3\mu d_i d_j} + \frac{G}{6}(d_i+d_j)^3 + \frac{\pi}{4}(d_i+d_j)^2\lvert w_{si} - w_{sj}\rvert$
Breakup frequency function S_i	$S_i = 0.00527G^{1.51369}v_i^{\frac{1}{3}}$ (Golzarijalal et al., 2017)
Breakup distribution function $\Gamma_{i,j}$	Binary, $\Gamma_{i,j} = \begin{cases} 2 & \text{j=i+1} \\ 0 & \text{otherwise} \end{cases}$

Table 5.2: Parameters of the flocculation model used to check mass conservation

It is observed that:

- $\frac{1}{V}\int\int\int(\Lambda_1+\Lambda_2+\Lambda_3)dV$ is 0 throughout the simulation time, which has verified Eq.(5.2).
- $\frac{1}{V}\int\int\int\Lambda_1 dV$ and $\frac{1}{V}\int\int\int\Lambda_2 dV$ are always negative over time. Thus, a decreasing trend in the average volume concentration of size class 1 and 2 is observed (see Figure 5.3). Similarly, a monotonically increasing trend is shown in $\frac{1}{V}\int\int\int\alpha_3 dV$ since $\frac{1}{V}\int\int\int\Lambda_3 dV$ is positive during the simulation.
- The change in volume concentrations of three size classes decreases with time due to the decrease in the values of the three flocculation source terms. After approximately 50 seconds, the three flocculation source terms approach zero, which means that the steady state of flocculation is to be reached. In steady state, the volume concentrations stay as constant.

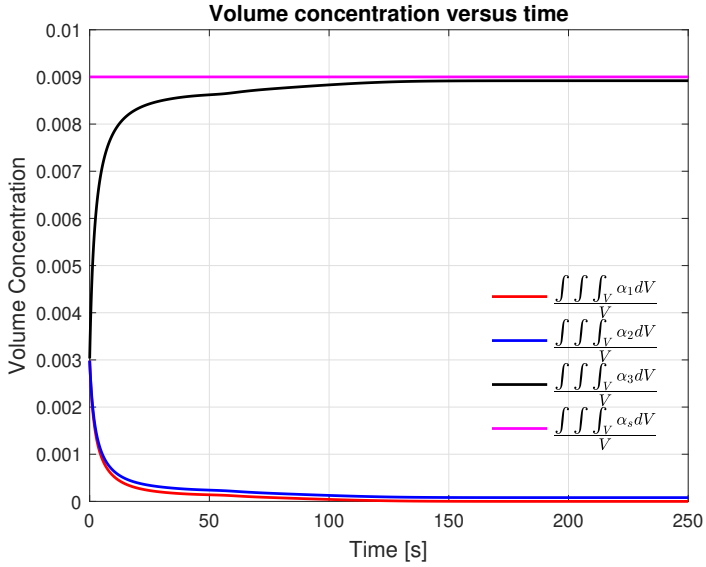


Figure 5.3: Time evolution of volume-averaged volume concentration of each size class

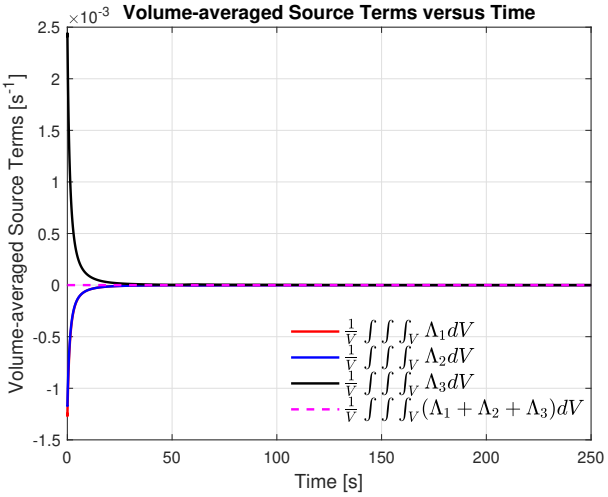


Figure 5.4: Time evolution of the volume-averaged source terms related to flocculation

5.3.2. CASE: NO AGGREGATION OCCURS

The collision efficiency A is a significant parameter which influences aggregation process of particles. If the collision efficiency is set to be zero, the terms accounting for aggregation will be zero and only breakup occurs, see Eq.(5.7). Consequently, no particle aggregation occurs in the simulation.

$$\begin{aligned}
&\frac{\partial \alpha_1}{\partial t} + \nabla \cdot (\alpha_1 \boldsymbol{u_1}) = \Gamma_{1,2} S_2 N_2 v_1 \\
&\frac{\partial \alpha_2}{\partial t} + \nabla \cdot (\alpha_1 \boldsymbol{u_2}) = -S_2 N_2 v_2 \\
&\frac{\partial \alpha_3}{\partial t} + \nabla \cdot (\alpha_3 \boldsymbol{u_3}) = 0
\end{aligned} \tag{5.7}$$

According to Table 5.3, integrating Eq.(5.7) throughout the whole simulation domain gives:

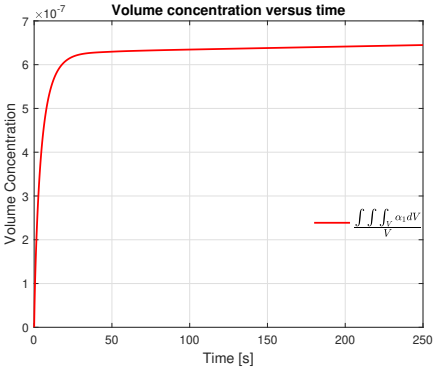
$$\begin{aligned}
&\frac{\partial \iiint_V \alpha_1 dV}{\partial t} = \iiint_V \Gamma_{1,2} S_2 N_2 v_1 dV > 0 \\
&\frac{\partial \iiint_V \alpha_2 dV}{\partial t} = -\iiint_V \Gamma_{1,2} S_2 N_2 v_2 dV < 0 \\
&\iiint_V \alpha_3 dV = 0
\end{aligned} \tag{5.8}$$

The simulation is performed with the conditions given in Table 5.3 and Table 5.4.

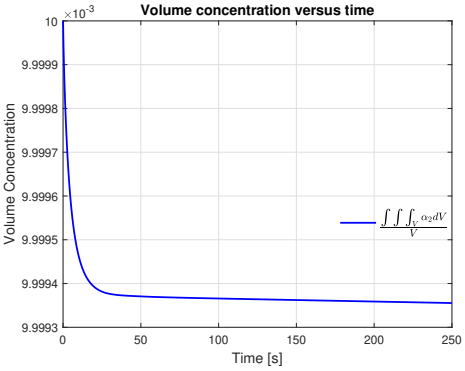
Condition / Size Class	Initial Volume Concentration α_i	Diameter d_i [µm]	Particle Settling Velocity [mm/s]
1	0, homogeneous	75	4.94
2	1%, homogeneous	94.5	7.59
3	0, homogeneous	119.1	11.54

Table 5.3: Size classes and initial conditions for the case that no aggregation occurs. The settling velocity is calculated by Eq.(2.14). The concentration-dependent hindered settling effect will be considered in the simulation.

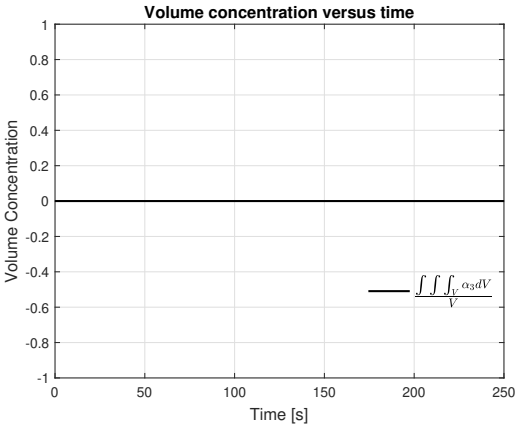
As illustrated in Figure 5.5 (c), the volume concentration of size class 3 remains zero since no aggregation occurs. The increase of the volume concentration of size class 1 is due to breakup of flocs belonging to size class 2 (see Figure 5.5 (a)), which also causes a reduction in the volume concentration of size class 2 (see Figure 5.5 (b)). Similarly, the slope of the curves in Figure 5.6 can also be explained through the flocculation source terms. Detailed explanations have already been given in Section 5.3.1 and are therefore not repeated here.



(a) Time evolution of the volume-averaged volume concentration of size class 1



(b) Time evolution of the volume-averaged volume concentration of size class 2



(c) Time evolution of the volume-averaged volume concentration of size class 3

Figure 5.5: Time evolution of the volume-averaged volume concentrations of three size classes in the case that no particle aggregation occurs

Density of water $[kg/m^3]$	1000
Density of sediment particles $[kg/m^3]$	2650
Temperature [K]	293
Collision efficiency $A_{i,j}$	0
Collision frequency function $\Lambda_{i,j}$	$\beta_{i,j} = \frac{2KT(d_i+d_j)^2}{3\mu d_i d_j} + \frac{G}{6}(d_i+d_j)^3 + \frac{\pi}{4}(d_i+d_j)^2\lvert w_{si} - w_{sj}\rvert$
Breakup frequency function S_i	$S_i = 0.00527G^{1.51369}v_i^{\frac{1}{3}}$ (Golzarijalal et al., 2017)
Breakup distribution function $\Gamma_{i,j}$	Binary, $\Gamma_{i,j} = \begin{cases} 2 & \text{j=i+1} \\ 0 & \text{otherwise} \end{cases}$

Table 5.4: Parameters of the flocculation model for the case that no aggregation occurs

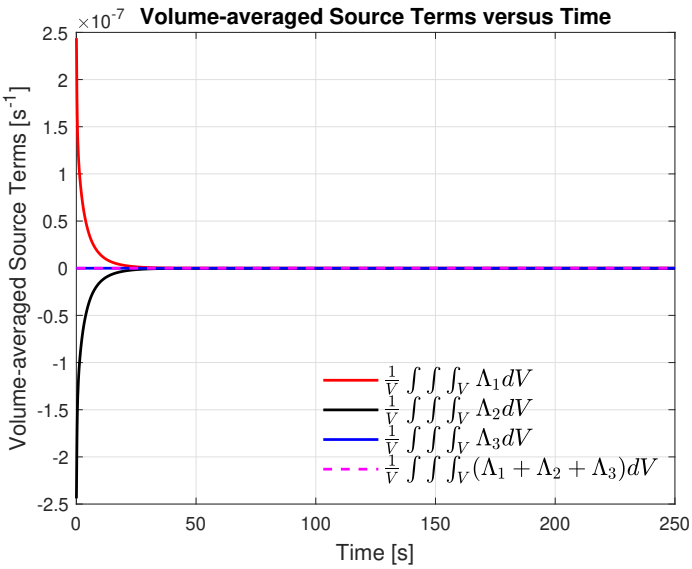


Figure 5.6: Time evolution of the volume-averaged source terms related to flocculation

5.3.3. CASE: NO BREAKUP OF FLOCS OCCURS

It is not physical that no breakup of flocs occurs since turbulent shear rate always exists. But this assumption can be used to check whether the conservation relationship is mathematically correct.

Condition / Size Class	Initial Volume Concentration α_i	Diameter d_i [μm]	Particle Settling Velocity [mm/s]
1	0%, homogeneous	75	4.94
2	1%, homogeneous	94.5	7.59
3	0%, homogeneous	119.1	11.54

Table 5.5: Size classes and initial conditions for the case that no breakup of flocs occurs. The settling velocity is calculated by Eq.(2.14). The concentration-dependent hindered settling effect will be considered in the simulation.

Density of water $[kg/m^3]$	1000
Density of sediment particles $[kg/m^3]$	2650
Temperature [K]	293
Collision efficiency $A_{i,j}$	1
Collision frequency function $\Lambda_{i,j}$	$\beta_{i,j} = \frac{2KT(d_i+d_j)^2}{3\mu d_i d_j} + \frac{G}{6}(d_i+d_j)^3 + \frac{\pi}{4}(d_i+d_j)^2\lvert w_{si}-w_{sj}\rvert$
Breakup frequency function S_i	0
Breakup distribution function $\Gamma_{i,j}$	Binary, $\Gamma_{i,j} = \begin{cases} 2 & \text{j=i+1} \\ 0 & \text{otherwise} \end{cases}$

Table 5.6: Parameters of the flocculation model for the case that no breakup of flocs occurs

Due to the absence of breakup of flocs, the phase continuity equations can be written as:

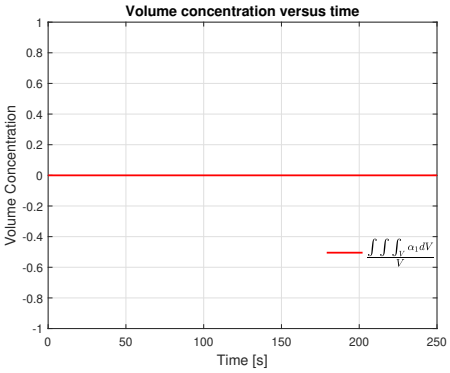
$$\begin{aligned}
\frac{\partial \alpha_1}{\partial t} + \nabla \cdot (\alpha_1 \boldsymbol{u_1}) &= -A\beta_{1,1} N_1 N_1 v_1 - A\beta_{1,2} N_1 N_2 v_1 \\
\frac{\partial \alpha_2}{\partial t} + \nabla \cdot (\alpha_1 \boldsymbol{u_2}) &= \frac{1}{2} A\beta_{1,1} N_1^2 v_2 - \frac{1}{2} A\beta_{2,1} N_2 N_1 v_2 - A\beta_{2,2} N_2 N_1 v_2 \\
\frac{\partial \alpha_3}{\partial t} + \nabla \cdot (\alpha_3 \boldsymbol{u_3}) &= \frac{1}{2} A\beta_{1,2} N_2 N_1 v_3 + \frac{1}{2} A\beta_{2,2} N_2^2 v_3
\end{aligned} \tag{5.9}$$

Based on the conditions given in Table 5.5, integrating Eq.(5.9) over the whole simulation domain gives:

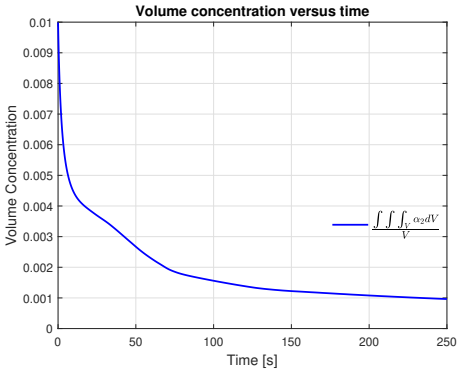
$$\begin{aligned}
\iiint_V \alpha_1 dV &= 0 \\
\frac{\partial \iiint_V \alpha_2 dV}{\partial t} &= -\iiint_V A\beta_{2,2} N_2 N_2 v_2 dV < 0 \\
\frac{\partial \iiint_V \alpha_3 dV}{\partial t} &= \iiint_V \frac{1}{2} A\beta_{2,2} N_2^2 v_3 dV > 0
\end{aligned} \tag{5.10}$$

According to Eq.(5.10), it should be observed that the volume concentration of size class 1 equals to zero over time. This has been shown in Figure 5.5 (a). In addition, the volume concentration of size class 2 should decrease monotonically while the volume concentration of size class 3 should increase monotonically. Figure 5.5 (c) presents the increase in the volume concentration of size class 3 due to aggregation between particles belonging to size class 2, which also causes the reduction of the volume concentration of size class 2, see Figure 5.5 (b). The conservation is thus verified.

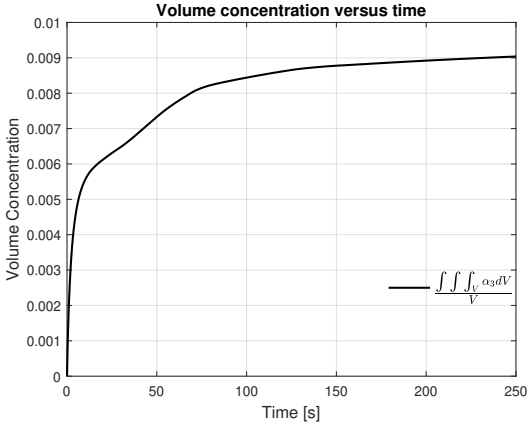
Different from the monotonically declining magnitudes of the slopes observed in Figure 5.3 and Figure 5.5, the magnitudes of the slopes of the volume concentrations of size class 2 and 3 increase slightly in this case from $t = 30s$ to $t = 50s$, see Figure 5.7. However, the author has not found the origin for this increase from the visual observation. Besides, the increase is not attributed to the use of variable time steps since the same phenomena is observed when using a fixed time step. From mathematical point of view (see Eq.(5.10)), the increase can be explained by the increase in the magnitudes of the source terms from $t = 30s$ to $t = 50s$ shown in Figure 5.8 (b).



(a) Time evolution of volume-averaged volume concentration of size class 1

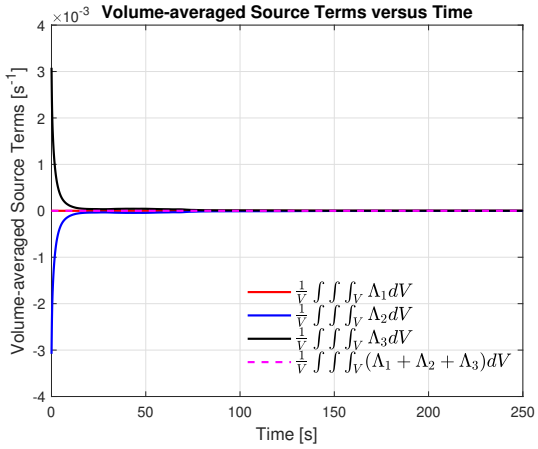


(b) Time evolution of volume-averaged volume concentration of size class 2

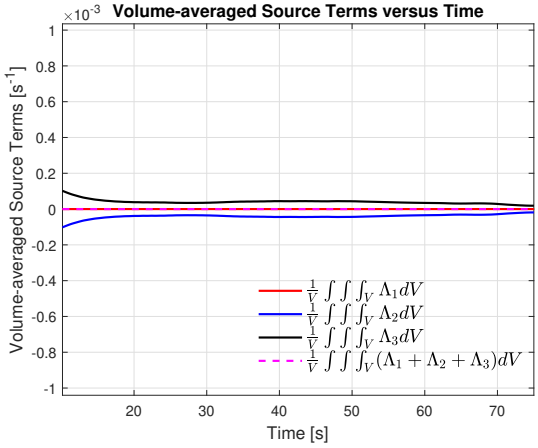


(c) Time evolution of volume-averaged volume concentration of size class 3

Figure 5.7: Time evolution of the average volume concentration in the case that no breakup of flocs occurs



(a) Time evolution of the volume-averaged source terms related to flocculation



(b) The variation of the volume-averaged flocculation source terms from $t = 10s$ to $t = 80s$

Figure 5.8: Time evolution of the volume-averaged source terms related to flocculation for the case that no breakup of flocs occurs

5.4. ITERATIVE CONVERGENCE

In CFD, residuals are defined as local imbalances of solutions in the iterative numerical process. The iterative convergence of numerical solutions can be quantified by residuals. When the residual is smaller than a specified tolerance, converged solutions will be obtained. The residual convergence is assessed for the mass conservation case as discussed before. The result is shown in Figure 5.9 which plots the final residuals of different quantities over time. The final residual can be calculated by subtracting the left hand side of an equation by the right hand side after last iteration.

Quantity	The maximum value of final residuals	Tolerance
α_1	1.8797×10^{-11}	10^{-10}
α_2	1.1674×10^{-11}	10^{-10}
α_3	2.1903×10^{-11}	10^{-10}
U_x	9.7023×10^{-12}	10^{-7}
U_z	1.1939×10^{-11}	10^{-7}
p_{rgh}	7.6451×10^{-9}	10^{-8}
ϵ	9.6048×10^{-8}	10^{-7}
k	9.9420×10^{-8}	10^{-7}

Table 5.7: Maximum final residual values and corresponding tolerance

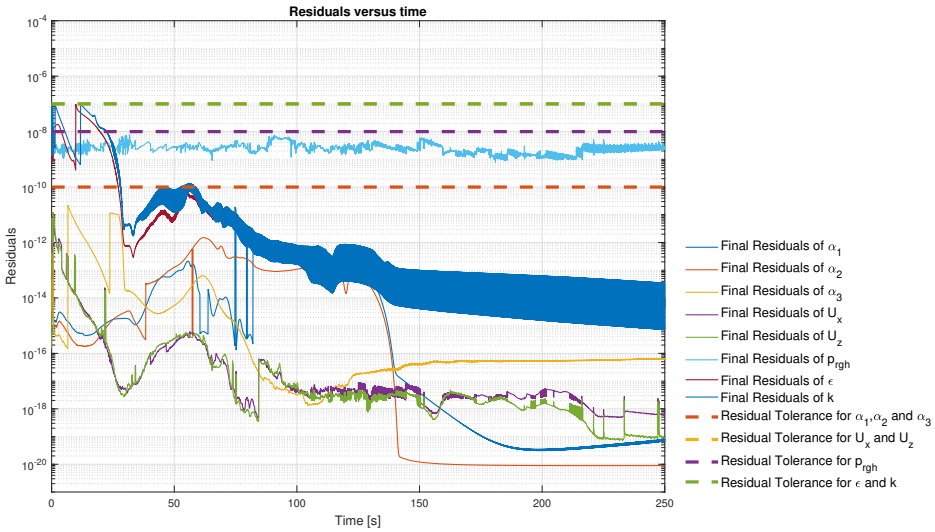


Figure 5.9: Iterative convergence of the volume concentration of three size classes α_1, α_2 and α_3, velocity U_x and U_z , dynamic pressure p_{rgh}, turbulent dissipation rate ϵ, turbulent kinetic energy k.

Figure 5.9 and Table 5.7 show that all final residual values are less than the corresponding tolerances set by the author. Thus, iterative convergence is achieved. If a lower tolerance

value is used, the residual values will decrease accordingly, but it will take longer computation time as more iterations are required to reach the tolerance.

6

NUMERICAL CALIBRATION

Calibration is defined as the process of adjusting model parameters in order to improve the agreement between numerical results and corresponding experimental data (AIAA Standards, 1998). Calibration is essential when the knowledge of a complex physical process is limited. As a large number of numerical parameters may be involved in the flocculation model, sensitivity analysis is necessary to determine the most influential parameters (Hajdukiewicz and Magdalena, 2013). The calibration of parameters is done by minimizing the errors between numerical results and experimental data. In this study, simulations are carried out for settling column tests. Experimental data from the settling column experiments conducted by Enthoven (2021) will be used for calibration process. The difference between calibration and validation is that validation takes place after the calibration of the parameters. Validation aims to check whether numerical results well predict the physical process using the calibrated parameters. Calibration and validation are iterative process to improve the performance of a model.

6.1. DESCRIPTION OF SETTLING COLUMN TESTS

The materials, experimental set-up, experimental procedure and the results of the settling column experiments conducted by Enthoven (2021) are briefly described here. More details are given in Enthoven's thesis.

6.1.1. SEDIMENT PROPERTIES

Data from the experiment using illite will be used in this study since it is hard to obtain a homogeneous distribution of bentonite which behaves inconsistently in the experiments (Enthoven, 2021). The density and size of illite is given in Table 6.1. The size distribution of illite particles shown in Figure 6.1 is measured with by the Mastersizer 2000.

	Density $[kg/m^3]$	d_{10} $[\mu m]$	d_{50} $[\mu m]$	d_{90} $[\mu m]$
Illite	2750	1.39	5.32	18.61

Table 6.1: Properties of illite (Enthoven, 2021)

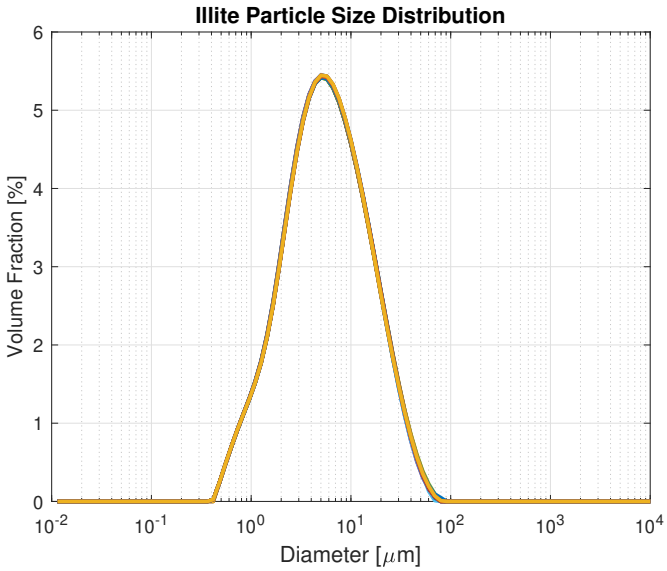


Figure 6.1: Particle size distribution of illite (Enthoven, 2021)

6.1.2. EXPERIMENTAL SETUP

SETTLING COLUMN

The settling column is a graduated cylinder with 0.08 metres in diameter and 0.35 meters in height. The top of the settling column is open to atmosphere.

EXPERIMENTAL PROCEDURE

The experimental procedure is described below (Enthoven, 2021):

1. Add fresh water to a predetermined amount of illite in the settling column to obtain one litre mixture.

2. Mix the water and the illite using a VOS-12020 overhead stirrer which rotates at 500 rpm for 20 minutes in order to obtain a homogeneously distributed mixture. The value of the applied shear rate is not calculated since no calibrated relationship between shear rate and rotational speed is found and the experiment does not aim to investigate how shear rate influences flocculation process. This high-speed mixing is only used to distribute the mixture homogeneously. In addition, finest sediment particles will be obtained with this sufficient mixing.

3. Remove the mixer.

4. Measure temperature and salinity of the mixture.
5. Start GoPro camera with sampling time-laps of 30 seconds.
6. Put the settling column in front of GoPro camera for recording.

MEASUREMENT

The height of the mudline, which is the interface between the mixture and the pure water part,is measured per 30 seconds using image analysis. Different colors are detected at different locations in the images taken by Gopro since varied amount of light passes through different sediment concentrations. As each color corresponds to a unique pixel value in a greyscale image, the relationship between sediment concentrations and pixel values can be described by a calibration curve (Enthoven, 2021). Thus, pixel information can be converted to the sediment concentration field according to the calibration curve. Then, a threshold pixel value is selected manually for image binarization, after which the mudline height can be determined. As shown in Figure A.1 in Appendix A, the interface between the black and the white region in the binary image represents the mudline which can be measured directly.

6.1.3. EXPERIMENT RESULTS

The settling column tests are conducted using illite of different concentrations to investigate the effects of sediment concentration on flocculation and settling behaviour. The time variation of the mudline height in the settling column experiments using fresh water without salt is plotted in Figure A.2 in Appendix A. Irregularities appear in certain experimental curves (i.e. 30 g/L, 50 g/L), which may due to imperfections in measurement and image processing. Experiments for illite with an initial concentration of 20 g/L, 40 g/L and 100 g/L will be used in this study for comparison. A summary of the three experiments is shown in Table 6.2. In the three experiments, fresh water is used and no salt ($CaCl_2$) is added. The duration of the experiments is approximately 500 minutes.

#	Initial mass concentration of illite [g/L]	Initial volume concentration of illite [-]	Temperature [°C]	Water density $[kg/m^3]$
1	20	0.00727	20.1	1000
2	40	0.0145	20.3	1000
3	100	0.0364	18	1000

Table 6.2: Details of experiments (Enthoven, 2021)

The result is shown in Figure 6.2. For sediment with a initial concentration of 20g/L, the mudline height has kept unchanged for first 1,000 seconds. Then, the mudline height decreases with a slight increasing slope. After approximately 7,000 seconds, the hindered effect governs the settlement of illite particles where the slope of the settling curve gradually declines. Subsequent consolidation phase starts at 10,000 seconds, after which the height decreases with a fairly slow rate.

The effects of mass concentration on settlement has also been investigated in Enthoven's tests. As the initial mass concentration of illite increases, the settlement of mudline becomes slower. This is due to that the hindered effect is intensified with an increasing sediment concentration (Enthoven, 2021). Besides, the start time of hindered settling stage is delayed. In the settling test of illite of 100 g/L, the hindered settlement starts at almost the end of the test and the compression settling stage is not present.

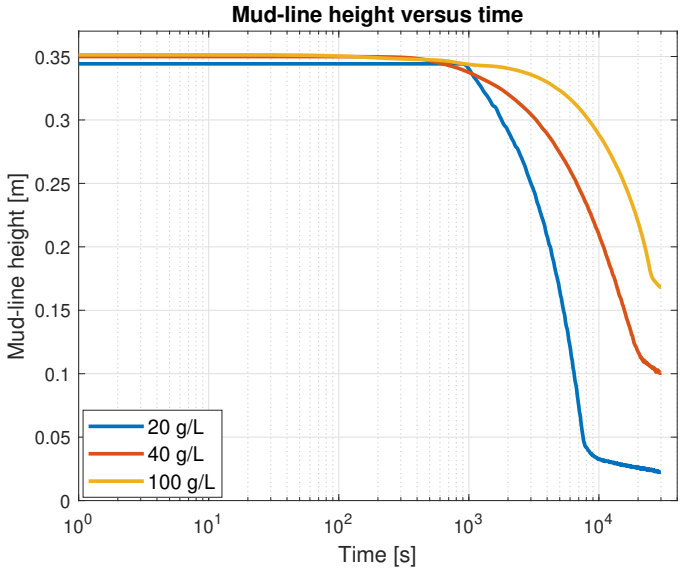


Figure 6.2: Mudline height as a function of sediment concentration (Enthoven, 2021). The mudline is defined as the height above which the sediment concentration is below a specific value set by Enthoven. The mudline height is obtained by image analysis.

6.2. NUMERICAL SIMULATION

6.2.1. SET-UP

As the duration of simulation is 30,000 seconds, 2D model is used to reduce the heavy computational cost. The simulation domain is a plane passing through the vertical axis of the cylindrical column. The width and the height of the computational domain are 0.08 m and 0.35 m respectively. Uniform grids of 4 mm in width and 5 mm in height are used in the following simulations. Figure 6.3 shows the simulation domain and the mesh.

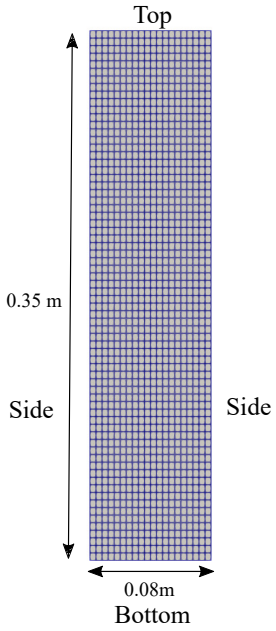


Figure 6.3: A schematic representation of simulation domain

6.2.2. BOUNDARY CONDITIONS

A summary of boundary set-up is shown in Table 6.3.

Parameter	Side	Bottom	Top
$\boldsymbol{U}$	*Noslip*	*Noslip*	*Slip*
α	*zeroGradient*	*zeroGradient*	*zeroGradient*
k	*kqRWallFunction*	*kqRWallFunction*	*zeroGradient*
ϵ	*epsilonWallFunction*	*epsilonWallFunction*	*zeroGradient*
ν_t	*nutkWallFunction*	*nutkWallFunction*	*zeroGradient*

Table 6.3: Boundary conditions

VELOCITY FIELD

No-Slip wall condition

No-slip wall condition describes that the relative velocity between the viscous fluid is zero. It is a Dirichlet boundary condition. The two sides and the bottom of the simulation domain are fixed walls. Thus, the velocity at these walls are assumed to be zero.

$$\boldsymbol{u}_{wall} = \mathbf{0} \tag{6.1}$$

Slip wall condition

The top of the settling column is open to atmosphere, which is a free surface. At this free

surface, the fluid is free to move along the air-liquid interface but it is not able to penetrate the boundary to leave the domain. This can be described by the slip wall condition.

CONCENTRATION FIELD

Inlet Condition

inletOutlet is used to describe inflow conditions. When inflow is toward the inside of the simulation domain, the inflow is prescribed by a Dirichlet boundary condition. If the flow is in the reverse direction toward the outside of the simulation domain, *inletOutlet* is the same as a Neumann boundary condition named *zeroGradient*.

Outlet Condition

outletInlet works in an opposite way as *inletOutlet*. When the outflow is toward the outside of the simulation domain, the outflow is prescribed by a Dirichlet boundary condition. If the outflow is toward the inside of the domain, it works in a same way as *zeroGradient*.

Zero gradient condition

At both sides and bottom of the settling column, *zeroGradient* is used as the concentration boundary condition. It means that the gradient of the concentration equals to zero in the normal direction to the wall. The concentration at boundaries is extrapolated from the value at the nearest cell centre.

Physically, the top of the open settling column is both the inlet and the outlet of the simulation domain. However, setting the top patch as the inlet boundary or the outlet boundary will induce numerical errors in the multi-phase drift-flux solver. To get rid of this, *zeroGradient* condition is applied to the top patch which is open to atmosphere. In the following simulations, illite particles and water are homogeneously distributed in the settling column at the initial time. Once simulations start, particles begin to settle under the density difference. Since the flow is incompressible, it makes no difference to set a settling column which is either closed or open to atmosphere at the top.

PRESSURE FIELD

The boundary condition of the dynamic pressure field p_{rgh} is described by *fixedFluxPressure*. *fixedFluxPressure* condition is usually used for the situations where pressure gradient is zero. Compared to the *zeroGradient* condition, *fixedFluxPressure* can adjust the pressure gradient according to body forces (e.g., gravitational force), which gives a better convergence of the results. A reference pressure value is specified in a reference cell in the simulation domain. According to the author's observation, this set-up can speed up simulations.

k, ϵ, ν_t

Wall functions described in Section 3.4.2 is applied for k, ϵ, ν_t at both sides and bottom of the domain. For the same reason, *zeroGradient* condition is applied at the top.

6.2.3. INITIAL INVESTIGATION

As shown in Eq.(3.21), a number of parameters influence the flocculation process in the numerical simulation. Before quantitatively evaluating the performance of the numerical model, simulations are executed to give an initial investigation of the effects of flocculation on particle settling behaviour in the settling column.

NUMERICAL SET-UP

Size Classes

Nine size classes are used in the simulations. The division of size classes is based on $\frac{v_{i+1}}{v_i} = 2$. Considering particle size distribution of illite shown in Figure 6.1, the primary diameter is chosen to be 3 µm. The diameter and particle settling velocity of each size class is given in Table 6.4.

The numerical simulation starts from the state after the mixing step in the experiments. It is therefore assumed that only smallest particles of 3 µm exist at the initial time in the simulations since larger particles will break into the smallest ones with a sufficient mixing for 20 minutes. Based on this assumption, the size distributions of illite particles before mixing and after mixing are shown in Figure 6.4. The continuous black line is the PSD measured before the mixing step in the experiments. The numerical simulations start after the mixing step, and the discrete diamond marker shows the PSDs of the nine size classes, which is the initial PSD assumed in the numerical simulations.

i	Diameter d_i [μm]	Particle settling velocity [mm/s]
1	3	0.0086
2	3.78	0.0136
3	4.76	0.0216
4	6.00	0.0343
5	7.56	0.0544
6	9.52	0.0862
7	12	0.137
8	15.12	0.2163
9	19.05	0.3423

Table 6.4: Particle diameter and settling velocity calculated by Eq.(2.14)

Simulation Matrix

Table 6.5 summarises the parameters used for the initial investigation.

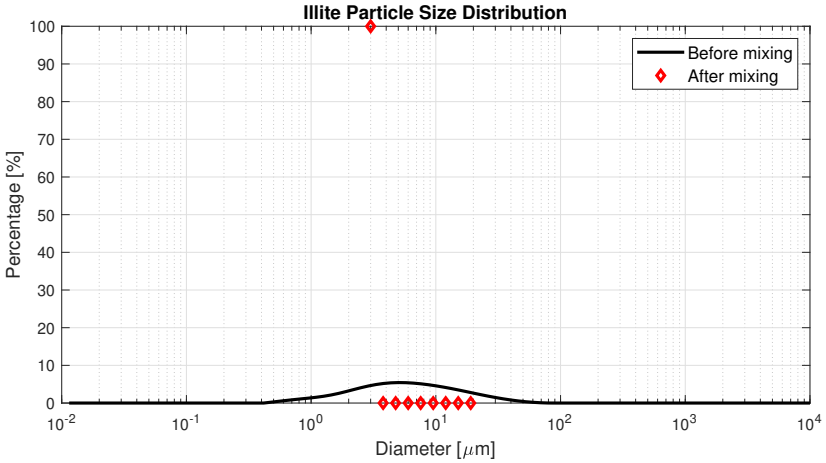


Figure 6.4: Illite particle size distribution before and after initial mixing

Initial mass concentration $[g/L]$	20	40	100
Density of sediment particles$[kg/m^3]$	2750		
Initial volume concentration	0.00727	0.0145	0.0364
	Homogeneously distributed		
Collision efficiency $A_{i,j}$	0.15		
Collision frequency function $\beta_{i,j}$	$\beta_{i,j} = \frac{2KT(d_i+d_j)^2}{3\mu d_i d_j} + \frac{G}{6}(d_i+d_j)^3 + \frac{\pi}{4}(d_i+d_j)^2\lvert w_{si} - w_{sj}\rvert$		
Breakup frequency function S_i	$S_i = E_b G(\frac{d_i - d_P}{d_P})^{3-n_f} \cdot (\frac{\mu G}{Fy/d_i^2})^{0.5}$ $E_b = 2\cdot 10^{-4}$, $F_y = 10^{-10}$ Pa, $n_f = 2.3$, $d_P = d_1$		
Breakup distribution function $\Gamma_{i,j}$	Binary, $\Gamma_{i,j} = \begin{cases} 2 & \text{j=i+1} \\ 0 & \text{otherwise} \end{cases}$		

Table 6.5: Settings of the initial numerical investigations

RESULTS

Mudline

Figure 6.5 shows the layer-averaged sediment concentration over height at t = 100s, 500s, 1000s, 2000s, 3000s, 5000s, 7000s and 10000s. Two interfaces are present in the figure and the upper one is the mudline. The layer-averaged sediment concentration is calculated by:

$$\overline{\alpha}(H_i) = \frac{\sum_{H[cellI]=H_i} \alpha[cellI]V[cellI]}{\sum_{H[cellI]=H_i} V[cellI]} \tag{6.2}$$

where $\overline{\alpha}(H_i)$ is the average concentration of sediment at height $H = H_i$. $\alpha[cellI]$ is the sediment concentration in cell I and $V[cellI]$ is the volume of cell I. In first 500 seconds, the settlement of the mudline is the fastest for the sediment with an initial concentration of 100g/L. As time evolves, the hindered effect becomes dominant. After approximately 2000 seconds, the height of the mudline for sediment with a higher concentration is greater at the same instance of time, which is an evidence for hindered settling behaviour. After 3000 seconds, the change of mudline has become less obvious, which makes it hard to determine the mudline visually. Another phenomena observed in the simulations is the gradual accumulation of soil at the column bottom, see Figure 6.5.

Centre of mass of sediment particles

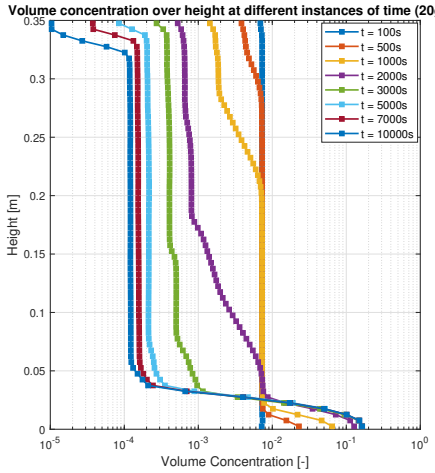
The height of the centre of mass (COM) of sediment is calculated by:

$$H_M = \frac{\sum_{k=1}^{k_{max}} \alpha_k \rho_k z}{\alpha_s \rho_s} \tag{6.3}$$

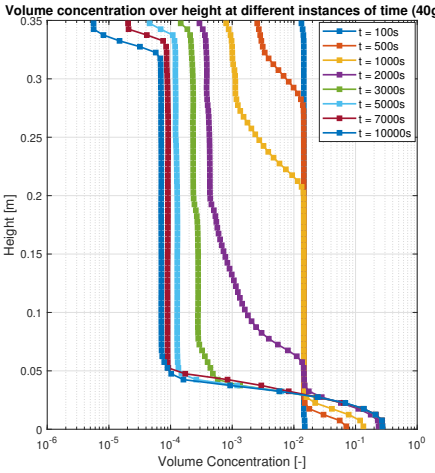
where α_k is the volume concentration of the dispersed phase k and ρ_k is the density of particles belonging to phase k. α_s is the total volume concentration of sediment and ρ_s is the density of sediment. In the simulations, ρ_k and ρ_s are the same, which equal to 2750 kg/m^3.

Figure 6.6 (c) shows that the settlement of sediment can be divided in three stages: flocculent settling, hindered settling and compression settling. The three stages were also found in the study of Haan et al. (1994) and Enthoven (2021), see Figure 6.6 (a) and (b). The height of the COM as a function of sediment concentration is plotted in Figure 6.7. It shows that sediment of a higher concentration settles faster in flocculent settling stage. As time evolves, the hindered effect becomes more dominant since sediment concentration increases due to accumulation of sediment at the bottom. Therefore, the sediment with an initial concentration of 40g/L settles faster than the sediment of 100 g/L after approximately 1375 seconds (the intersection point of orange and red curve shown in Figure 6.7).

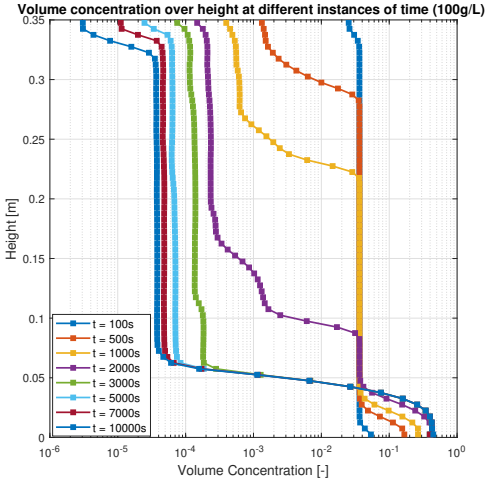
Compared to mudline height, the height of the COM of sediment can be calculated more easily in post-processing process using Eq.(6.3). In the following analysis, the height of the COM of sediment will be used to quantify the settling behaviour of illite.



(a)

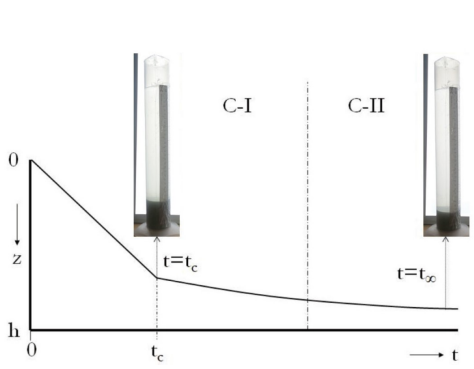


(b)

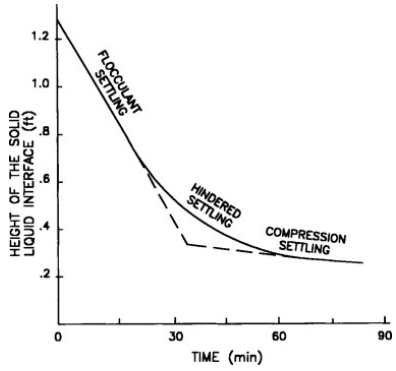


(c)

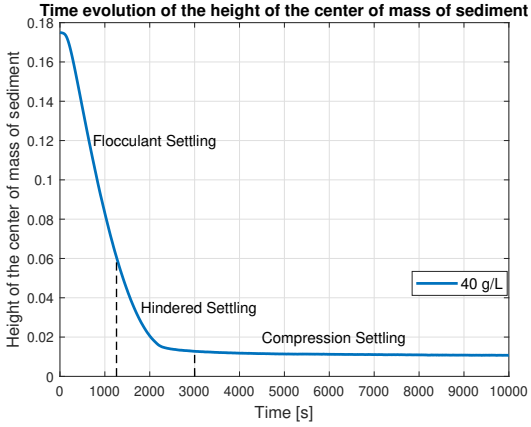
Figure 6.5: Layer-averaged volume fraction of sediment with an initial concentration of (a) 20g/L, (b) 40g/L and (c) 100g/L



(a)



(b)



(c)

Figure 6.6: (a) The schematized settling curve in the study of Enthoven (2021). The consolidation phase is divided in two parts $C-I$ and $C-II$. $t = t_c$ is the start of consolidation phase and $t = t_\infty$ is the end (Enthoven, 2021). (b) The settling tests conducted by Haan et al. (1994). (c) The numerical simulation result for the settlement of sediment with an initial mass concentration 40 g/L in this thesis. The y-axis shows the height of the COM of sediment.

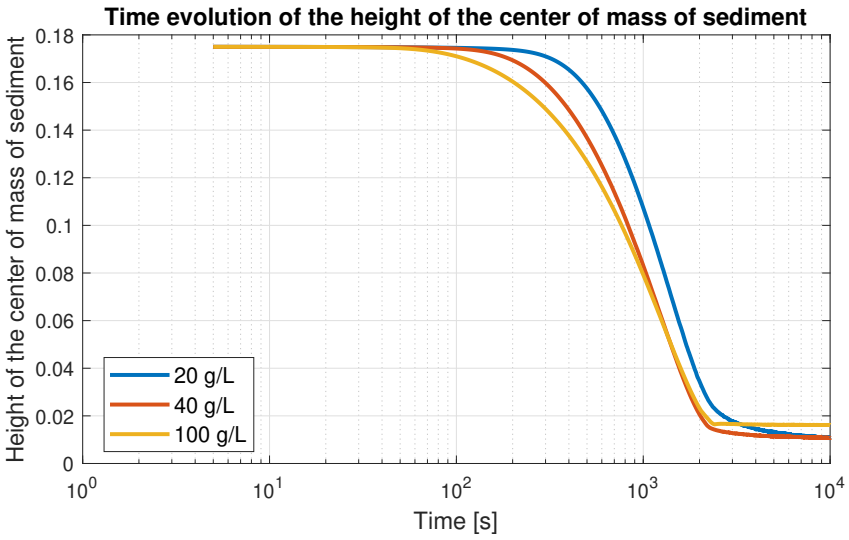


Figure 6.7: Time evolution of the height of the COM of sediment

Particle size distribution

The PSDs at t = 0s, 100s, 500s, 1000s, 2000s, 5000s and 10000s are shown in Figure 6.8. The results have shown that the phase transition into the 9-th size class which has the largest diameter is faster for sediment with a higher initial concentration. Within the flocculent settling stage starting from 0 to approximately 1000s, small particles aggregate into larger ones which have greater settling velocities, which has enhanced the sediment settling behaviour. This has explained why a slightly increasing slope has been observed within flocculent settling stage in Figure 6.6 (c). After 1000 seconds, above 90% of particles have aggregated into the largest ones and the PSD keeps nearly invariant, see Figure 6.8 (f), (g) and (h). Due to an increasing sediment concentration at the column bottom, the hindered settling behaviour becomes gradually apparent. As bottom sediment accumulates further more, soil undergoes compaction.

The formation of larger flocs is mainly caused by differential settling since the shear rate G = 0 s^{-1} as no source of mixing is present after removing the mixer. This has also been noted in the settling column tests conducted by Maggi (2005).

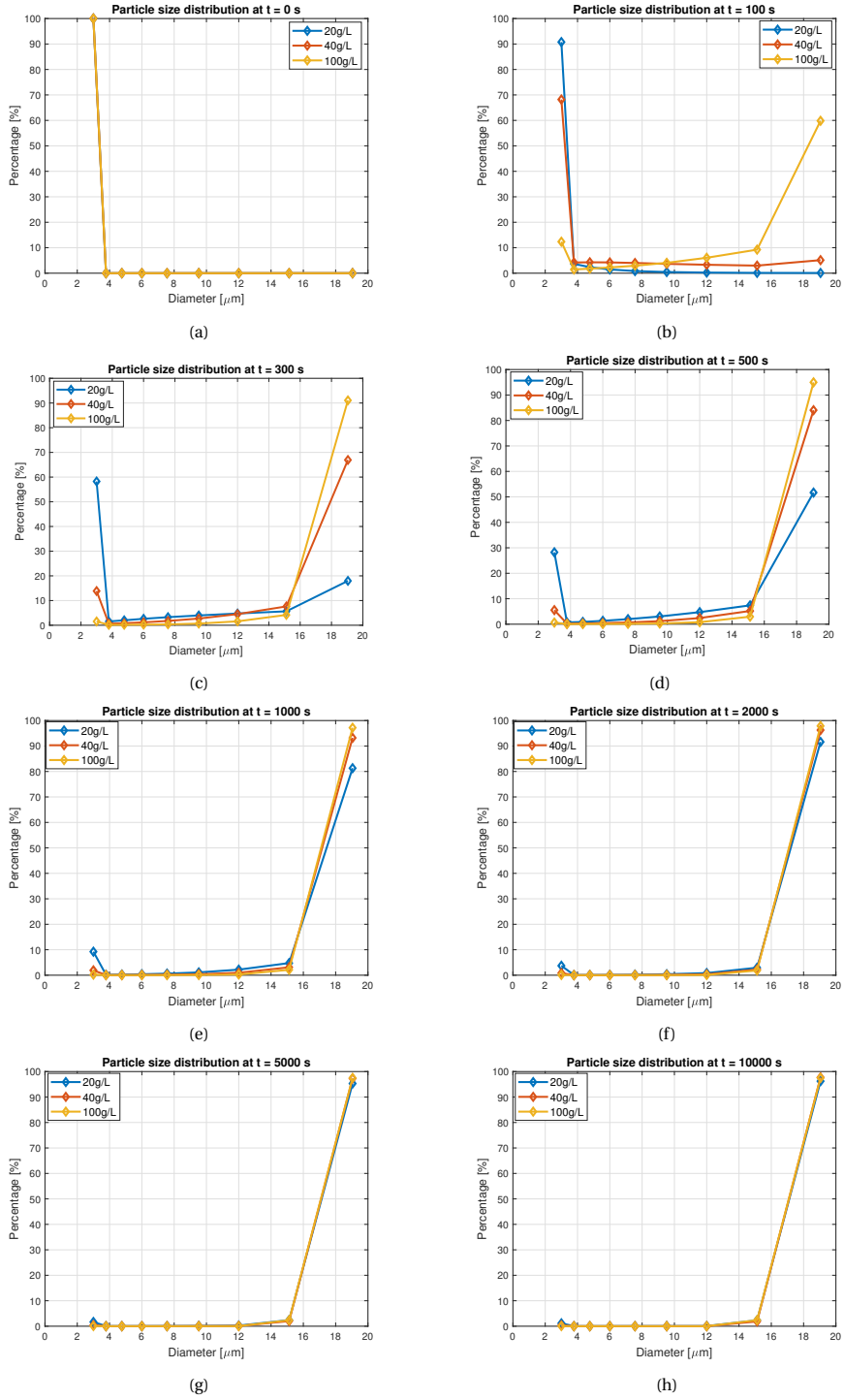


Figure 6.8: The particle size distribution of sediment at t = (a) 0s, (b) 100s, (c) 300s, (d) 500s, (e) 1000s, (f) 2000s, (g) 5000s and (h) 10000s.

6.2.4. COMBINED EFFECT OF SEDIMENT CONCENTRATION AND COLLISION EFFICIENCY

In the simulations of the settling column tests, the parameters related to breakup process has no influence on flocculation and settlement. This is due to $G = 0s^{-1}$ since no source of mixing is present in the column. As a result, breakup of flocs does not occur or is negligible. Therefore, the parameters related to aggregation process control the flocculation process. In this section, the author aims to investigate the effects of collision efficiency on flocculation and settlement of sediment of different concentrations.

NUMERICAL PARAMETERS

Size Classes

The same nine size classes present in Table 6.4 are used in the simulation.

Simulaiton Matrix

The simulation matrix is shown in Table 6.6. The simulations are performed for sediment with an initial mass concentration of 20 g/L, 40 g/L and 100 g/L in order to find the collision efficiencies which gives the best-fit to the experimental settling curves.

Initial mass concentration $[g/L]$	20	40	100
Density of sediment $[kg/m^3]$	2750		
Initial volume concentration	0.00727	0.0145	0.0364
	Homogeneously distributed		
Collision efficiency $A_{i,j}$	0.015, 0.02, 0.035, 0.05, 0.075	0.001, 0.0015, 0.002, 0.003, 0.005	0.0001, 0.00035, 0.0005, 0.00075
Collision frequency function $\Lambda_{i,j}$	$\beta_{i,j} = \frac{2KT(d_i+d_j)^2}{3\mu d_i d_j} + \frac{G}{6}(d_i+d_j)^3 + \frac{\pi}{4}(d_i+d_j)^2\lvert w_{si} - w_{sj}\rvert$		
Breakup frequency function S_i	$S_i = E_b G(\frac{d_i - d_P}{d_P})^{3-n_f} \cdot (\frac{\mu G}{Fy/d_i^2})^{0.5}$ $E_b = 2\cdot 10^{-4}$, $F_y = 10^{-10}$ Pa, $n_f = 2.3$, $d_P = d_1$		
Breakup distribution function $\Gamma_{i,j}$	Binary, $\Gamma_{i,j} = \begin{cases} 2 & \text{j=i+1} \\ 0 & \text{otherwise} \end{cases}$		

Table 6.6: Parameters used for numerical simulations to investigate the combinational effects of initial sediment concentration and collision efficiency A on flocculation and settling behaviour

RESULTS

Non-dimensional height

The mudline height is measured in Enthoven's settling tests while the height of the COM of sediment is provided in the numerical results. The non-dimensional height H/H_{max} is therefore used to compare the experimental data and numerical results. For experimental data, H is the mudline height. H_{max} is the maximum mudline height (0.35 m) which is also the initial height of the mudline. For numerical simulations, H is the height of the COM of sediment. H_{max} is the maximum height of the COM of sediment and it equals to 0.175 m, since the COM is initially located in the middle of the column. Figure 6.9 represents the non-dimensional height calculated from the experimental data and from the numerical simulations with different sediment concentrations and collision efficiencies A.

Absolute and relative 2-norm error are used to quantify the errors in the non-dimensional height. The absolute 2-norm error is defined as:

$$\|E\|_{2,abs} = \sqrt{\sum_{t=t_{start}}^{t_{end}} [(\frac{H}{H_{max}})_{num} - (\frac{H}{H_{max}})_{exp}]^2} \tag{6.4}$$

The relative 2-norm error is defined as:

$$\|E\|_{2,rel} = \sqrt{\frac{\sum_{t=t_{start}}^{t_{end}} [(\frac{H}{H_{max}})_{num} - (\frac{H}{H_{max}})_{exp}]^2}{\sum_{t=t_{start}}^{t_{end}} [(\frac{H}{H_{max}})_{exp}]^2}} \tag{6.5}$$

where $(\frac{H}{H_{max}})_{num}$ and $(\frac{H}{H_{max}})_{exp}$ are the non-dimensional height calculated from the numerical results and from the experimental data respectively. The results are given in Table 6.7.

Initial mass concentration [g/L]	Collision efficiency A [-]	$\|E\|_{2,abs}$ [-]	$\|E\|_{2,rel}$ [-]
20	0.015	3.0330	15.42%
	0.02	2.4798	12.61%
	0.035	7.1603	36.4%
	0.05	9.7617	49.63%
	0.075	11.9211	60.61%
40	0.001	6.4205	36.26%
	0.0015	4.9914	28.19%
	0.002	3.1385	17.73%
	0.003	3.0521	17.24%
	0.005	6.1502	35.74%
100	0.0001	1.3052	5.57%
	0.00035	1.2359	5.27%
	0.0005	0.5677	2.42%
	0.00075	3.2704	13.95%

Table 6.7: Errors in the simulated non-dimensional height

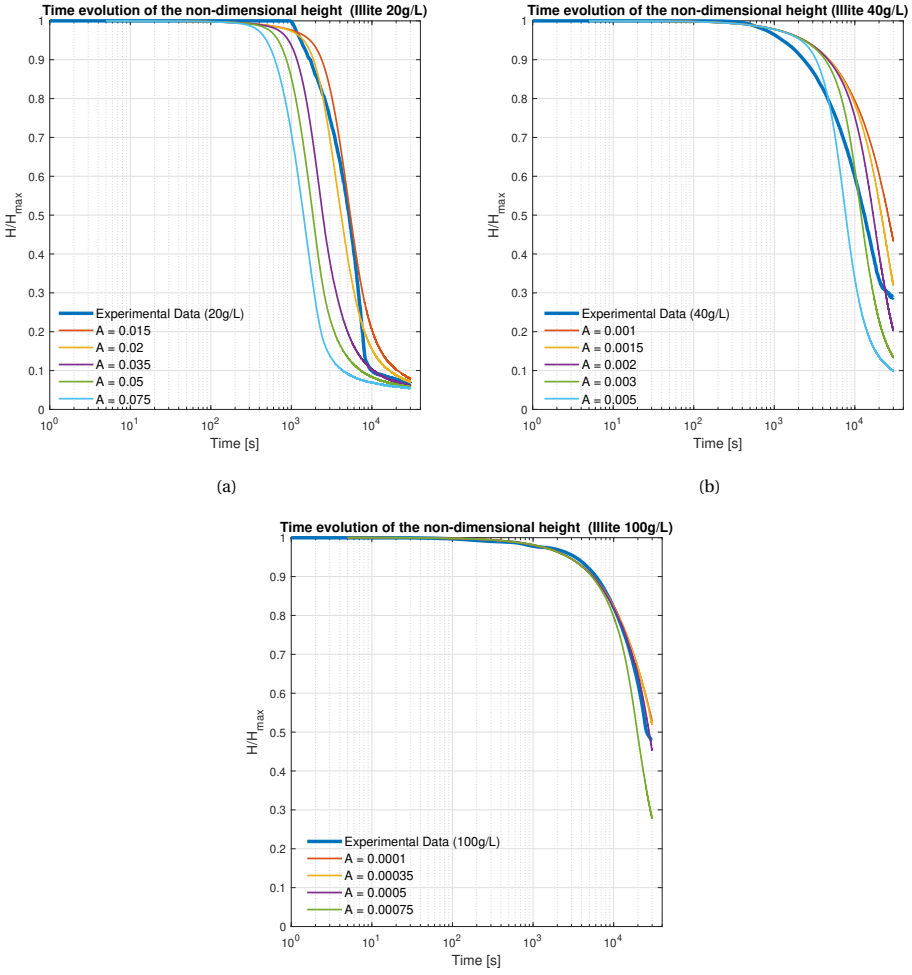


Figure 6.9: Comparison of the non-dimensional height H/H_{max} calculated from the experimental data and the numerical simulations for sediment of (a) 20 g/L, (b) 40 g/L, and (c) 100g/L

.

As summarized in Table 6.7, the error is the lowest (2.42%) for the simulation with a sediment concentration of 100 g/L and a constant collision efficiency $A = 0.0005$. For the simulations with a lower sediment concentration, the error increases to 12.61% for sediment of 20 g/L with $A = 0.02$ and to 17.24% for sediment of 40 g/L with $A = 0.003$. This is due to that the compression settling stage of sediment of 20 g/L and 40 g/L cannot be captured during the numerical simulations since soil compaction is not modelled.
It can be also found that the collision efficiency should decrease with increasing sediment concentration in order to fit the experimental data.

Particle size distribution

An advantage of the numerical simulation is that it can provide PSDs at each instance of time, which is not measured in the experiments. Figure 6.10 shows the PSDs in the numerical simulations which give the best fit to the experimental settling curves for illite of 20 g/L, 40 g/L and 100 g/L. It can be observed that:

- The flocculation process is fastest for illite of 20 g/L and with a collision efficiency A equal to 0.02. 90% of illite particles have aggregated into the largest ones at 15,000 seconds, after which aggregation process becomes much more slowly. By contrast, for illite of 100 g/L, due to a very low collision efficiency (i.e., A =0.0005), only half of primary particles of 3 µm have aggregated into larger ones at the end of the simulation.
- The percentage of primary illite particles and that of the largest size are the two highest in most of the time. In the settling column, particle aggregation is dominant while breakup is negligible due to a very low shear rate in the environment where no source of mixing is present. Consequently, particles have the tendency to aggregate into the largest ones. This has explained why the fraction of the particles belonging to the 9-th size class becomes the highest as flocculation proceeds.

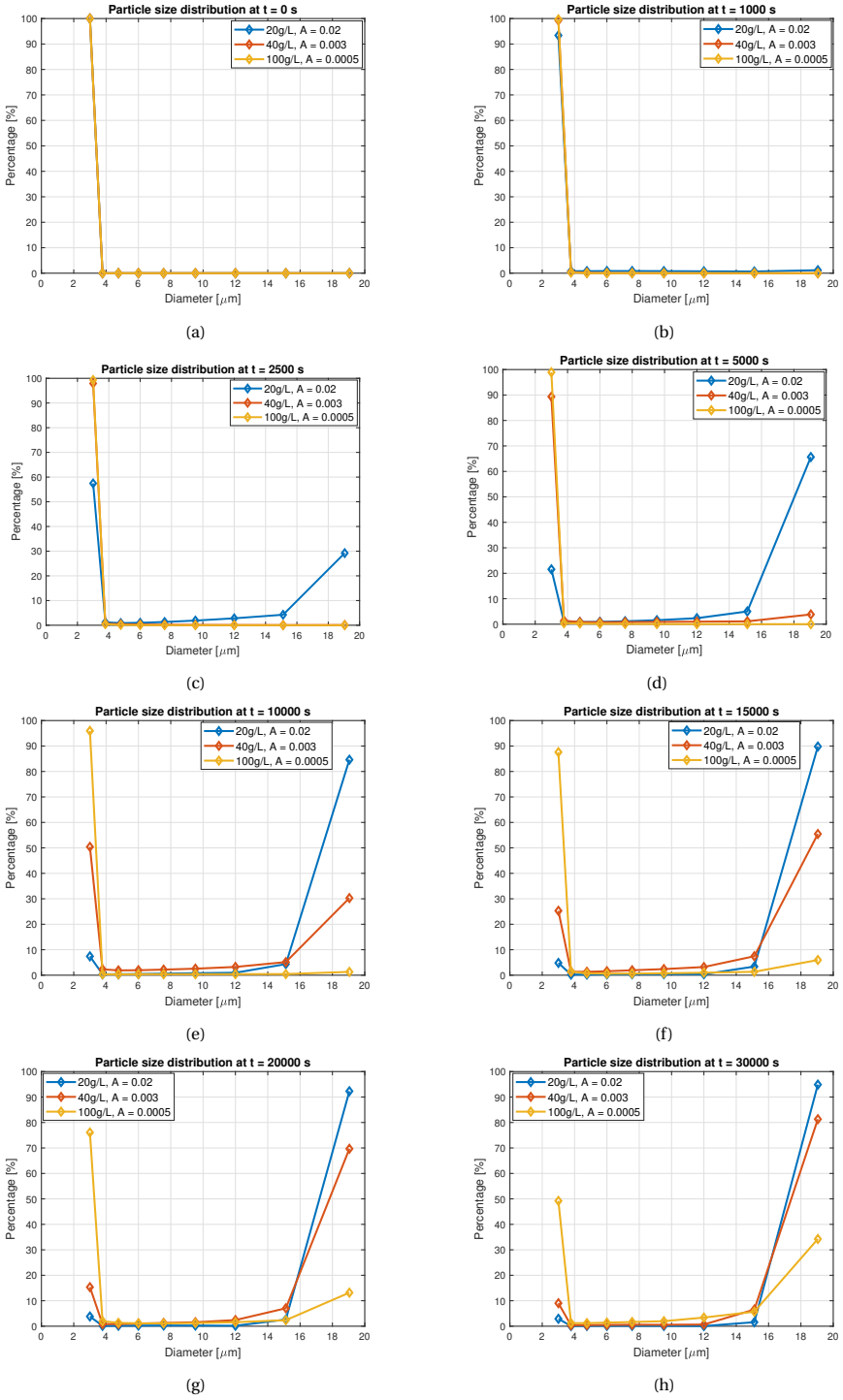


Figure 6.10: The PSD of sediment at t = (a) 0s, (b) 1000s, (c) 2500s, (d) 5000s, (e) 10000s, (f) 15000s, (g) 20000s and (h) 30000s.

6.3. DISCUSSION

6.3.1. SETTLING COLUMN EXPERIMENTS

Figure 6.11 (a) and (b) have shown a gap between the experimental and the numerical results in first 1,000 seconds for illite of 20 g/L. Besides, in the experimental settling curve, a sharp drop around 1,000 seconds can be observed, while in the numerical simulations, the decrease in non-dimensional height is smooth. But for the experiments using sediment of higher concentrations (i.e., 40 g/L and 100 g/L), the decrease in the mudline height is smoother, see Figure A.2. After converting the mudline height to the non-dimensional height, the experimental settling curve for illite of 100 g/L is linear in initial 1500 seconds, which almost coincides the simulation result (see Figure 6.11 (c)). The difference between the experimental and the numerical settling curves for illite of 20 g/L is caused by inaccurate measurement, since the light visibility is blurred for a low-concentration illite, which makes it hard to distinguish water-mixture interface. Consequently, the results are less accurate for low-concentration mixtures (Enthoven, 2021). Besides, the threshold pixel value used to determine the mudline is set manually according to the visual observations. A change in the threshold value will lead to a different experimental result. This may also account for the difference between the experimental data and the numerical results.

6.3.2. NUMERICAL SIMULATIONS

SIZE CLASS

The maximum number of size classes that can be used in the solver without causing divergence of numerical solutions is nine. When the number of size classes is greater than nine, negative values of sediment concentration have been obtained in the solution process, which is clearly not physical. To avoid negative concentration, numerical schemes and time step need to be adjusted. According to Golzarijalal et al. (2017), using 40 particle size classes requires a variable time step ranging from 10^{-7} to 0.5 seconds, which needs a prohibitively large computation power. Since using a small number of particle size classes cannot accurately capture PSD while using a large number of size classes takes an extremely long simulation time and may lead to divergence, it is thus reasonable to use 9 size classes to simulate the settling column tests with a duration of 500 minutes.

INITIAL CONDITION

Initial velocity field

In the settling column experiments, the mixture was still rotating for some time due to inertia after removing the mixer. By comparison, in the numerical simulations, the initial velocity field is assumed to be zero and illite particles begin to flocculate and settle once the simulations start. This could be a potential source of error. However, no obvious difference is observed between the experimental and the numerical settling curves during the initial period, see Figure 6.11 (c). For illite of 20 g/L, the difference is mainly caused by measurement errors due to blurred light visibility for a low concentration sediment and by image processing errors.

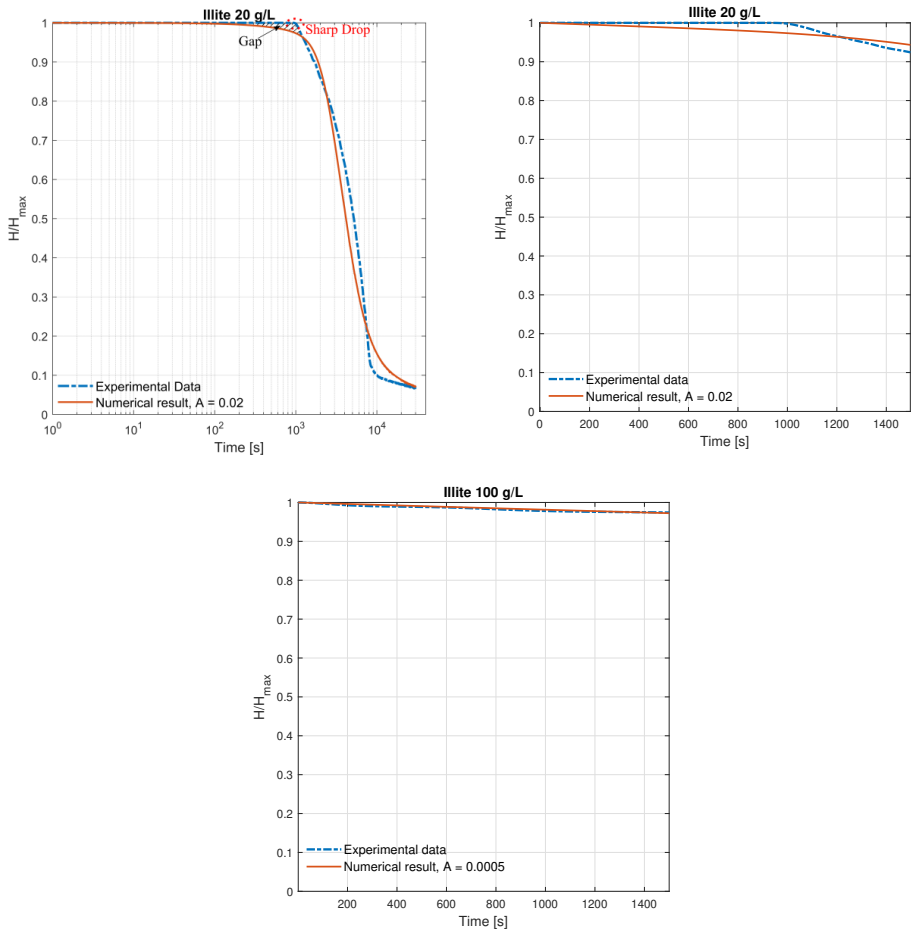


Figure 6.11: A comparison between non-dimensional height calculated from experiments and from numerical results for illite of (a) 20 g/L in semilogarithmic scale, (b) 20 g/L, (c) 100 g/L.

Initial particle size distribution

The primary particle size and the initial PSD also influence the simulated PSD and the settling curves (Maggi, 2005). However, their influence has not been investigated in this research since no experimental PSD is provided for comparison. In addition, the collision efficiency is treated as a fitting parameter in this thesis. If a different primary particle size or a different initial PSD is assumed, simulations need to be re-run to find the collision efficiencies which give good fit to the experimental data. To avoid repetitive work, calibration work is based on the assumption that only primary particles of 3 µm exist at initial time. However, the PSD after the mixing step is not assessed in the settling column experiments. It is therefore not able to verify whether this assumption is physically valid.

FLOCCULATION PARAMETERS

Collision efficiency

It has been found that the collision efficiencies should decrease with increasing illite concentration to fit the experimental data. Researchers have provided different explanations for this:

- According to Quezada, Jeldres, et al. (2020), more flocculant is required to keep acceptable flocculation of a higher proportion of clay. Thus, a higher concentration of clay will reduce collision efficiency if flocculant is absent or not enough.
- Collision efficiency decreases with the increase of particle diameter (Balakin et al., 2012). As flocculation proceeds, larger particles will be formed, which causes a reduction in the collision efficiency. The formation of these larger particles will speed up with increasing initial sediment concentration. Consequently, collision efficiency decreases with the increase of initial sediment concentration.

However, it is argued that collision efficiency is independent from sediment concentration (Maggi, 2005; Shen et al., 2018). But in their cases, sediment concentration is much lower (i.e., several grams per liter) than that (i.e., several tens of grams per liter) in this study.

SOIL COMPACTION

As shown in Figure 6.9 (a), a large difference between the simulated and the experimental settling curves is shown in the compression settling stage. Besides, the tail of the experimental settling curves is not captured by the numerical simulations, see Figure 6.9 (b) and (c). This is due to that soil compaction or consolidation is not modelled since this study focus on the flocculent settling stage and introducing the soil compaction model is out of scope.

6.3.3. ERROR INDUCED BY NONDIMENSIONALIZATION

In the experiments, non-dimensional height is calculated from mud-line height, while in the numerical simulations, it is calculated from the COM of sediment since it is hard to determine the mudline. This may also make differences between the experimental and numerical results.

7

CONCLUSIONS AND RECOMMENDATIONS

7.1. CONCLUSIONS

7.1.1. NUMERICAL MODEL

A similar population balance model has been used in Ansys- CFX-15 software package to investigate the influence of shear rate on flocculation and sedimentation of microalgae (Golzarijalal et al., 2017). However, the system of equations and the solution procedure differ much from those used in this thesis. To model the hydrodynamics, Golzarijalal et al. (2017) have used Euler-Euler method. As described before in Section 3.1, the method requires more computing power in calculating inter-phase forces and in solving momentum equations for each phase. Furthermore, aggregation and breakage of particles will influence interfacial forces, which adds the complexity in calculating the momentum terms. In addition, in their study, the population balance equation is solved separately from the equations of motion, which means that a new type of equations has been introduced. Consequently, as stated in the paper, solving such a large amount of equations becomes prohibitively complex and it requires to use a very small time step in the order of 10^{-6} s (Golzarijalal et al., 2017). Although an adaptive time stepping strategy has been implemented in their simulations, it has taken about 48 hours to simulate a case with a duration of several hundred seconds. Hence, it is infeasible to employ their method to simulate the settling column experiments conducted by Enthoven (2021) which has a duration of 30,000 s.

In this thesis, a multiphase drift-flux solver developed by Dredging & Deep-sea Mining Group of Delft University of Technology is used to describe the motion of the mixture (see Section 3.1.1). A discretized form of population balance equations in terms of volume concentration is used to describe flocculation process of cohesive sediment (see Section 3.2). In this population balance model, particle aggregation process is characterized by a collision frequency function and a collision efficiency function. For clay particles larger than 2 µm, differential settling and turbulent shear govern the collision

frequency. Differential settling is influenced by particle settling velocities and thus by particle size, while turbulent shear is related to turbulence level. Breakup of flocs is induced by fluid shear and the breakup process is characterized by a breakup frequency function related to turbulence level and particle size and by a breakup distribution function. When turbulence is absent (e.g., the settling column experiments conducted by Enthoven (2021)), only particle aggregation occurs under the differential settling mechanism.
For coupling of the two models, the multiphase drift-flux model is extended with a population balance model through using phase transition terms in phase continuity equations to account for particle aggregation and breakup (see Section 3.3). In this way, no additional equation is introduced to the existing equation system, which inherits the characteristics of PBEs and the merits of drift-flux model in reducing computational costs.

7.1.2. VERIFICATION

According to the results obtained in the numerical verification process, the following conclusions can be drawn:

- Mass is conserved during the simulation. This is evidenced by a constant total volume concentration of sediment observed in the simulation.

- Verification has been carried out for two special cases: no particle aggregation occurs and no break-up occurs. In the case that particles do not aggregate into larger ones, only break-up of larger flocs can be observed in the simulations. This is evidenced by the increase in the volume concentration of smaller particles and by the decrease in the volume concentration of larger particles. In the case that particles do not break up, only particle aggregation occurs. This is evidenced by the increase in the volume concentration of larger particles and the decrease in the volume concentration of smaller particles due to particle aggregation.

- All of the residuals are less than the tolerance during the simulation, which means that the iterative convergence condition has been satisfied.

7.1.3. CALIBRATION

A comparison with the experimental data of the settling column tests conducted by Enthoven (2021) has revealed that:

- The model can capture the characteristics of flocculent settling stage in the settling column tests. The numerical result for illite of 100 g/L is the most accurate in all of the simulations.

- The mismatch between experimental and numerical settling curves is attributed to that collision efficiency is dependent on sediment concentration. Calibrating collision efficiency gives satisfactory results.

7.1.4. FLOCCULATION AND SETTLING PROCESS

Cohesive sediment tends to flocculate during the settling process. Figure 6.10 shows that the percentage of larger flocs with higher settling velocities increase with time. Thus, settling of cohesive sediment can be enhanced by flocculation.
This study has also numerically investigated the effect of sediment concentration on flocculation and settling process of illite particles. By calibrating collision efficiency, the numerical experimental settling curves match the experimental ones well. The results have shown that the settling becomes slower as illite concentration increases, which is attributed to a decreasing collision efficiency with increasing concentration obtained in the calibration process.

7.2. RECOMMENDATIONS

- The collision efficiency has been treated as a fitting parameter in this study. Experimental determination of the collision efficiency is expected in the future work to verify the relationship between the collision efficiency and the sediment concentration found in the calibration process.

- The maximum number of size classes is limited to nine in the thesis work. To use a greater number of size classes in order to capture the PSDs more accurately, numerical schemes and time step need to be adjusted in the future work.

- Flocculation process is also influenced by turbulent shear rate (Golzarijalal et al., 2017; Maggi, 2005; van Leussen, 1994). So far, the author's solver has not been used to investigate the effects of shear rate on flocculation. In the future work, the settling column tests with oscillating grids at different frequencies conducted by van Leussen (1994) can be used to calibrate the remaining parameters related to breakup and to validate the model for a wider range of conditions. To simulate the case, it may require the implementation of dynamic mesh in the present model. The sinusoidal motion of the grids in the settling column (see Figure 7.1) can be described using the motion function named *oscillatingLinearMotion* in the dynamic mesh model in OpenFOAM.

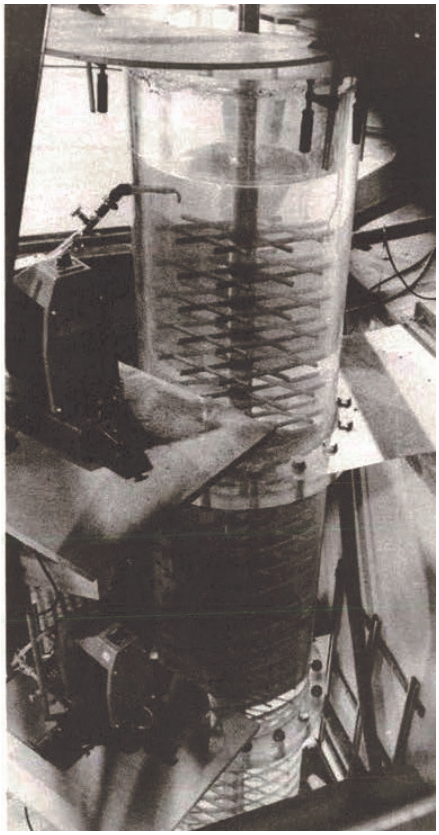


Figure 7.1: Experimental set up for flocculation and settling process influenced by turbulence (van Leussen, 1994)

BIBLIOGRAPHY

Adler, P. (1981). Heterocoagulation in shear flow. *Journal of Colloid and Interface Science*, *83*(1), 106–115. https://doi.org/https://doi.org/10.1016/0021-9797(81)90015-1

Ahern, D. (2017). *Study and improvement of the performance of an industrial settling tank using cfd models in openfoam* (Doctoral dissertation). https://doi.org/10.13140/RG.2.2.22006.70720

AIAA Standards. (1998). Guide: Guide for the verification and validation of computational fluid dynamics simulations (AIAA G-077-1998(2002)). https://doi.org/10.2514/4.472855.001

Balakin, B., Hoffmann, A. C., & Kosinski, P. (2012). The collision efficiency in a shear flow. *Chemical Engineering Science*, *68*(1), 305–312. https://doi.org/https://doi.org/10.1016/j.ces.2011.09.042

Basson, D. K., Berres, S., & Bürger, R. (2009). On models of polydisperse sedimentation with particle-size-specific hindered-settling factors. *Applied Mathematical Modelling*, *33*(4), 1815–1835. https://doi.org/https://doi.org/10.1016/j.apm.2008.03.021

Berlamont, J., Ockenden, M., Toorman, E., & Winterwerp, J. (1993). The characterisation of cohesive sediment properties [Special Issue Coastal Morphodynamics: Processes and Modelling]. *Coastal Engineering*, *21*(1), 105–128. https://doi.org/https://doi.org/10.1016/0378-3839(93)90047-C

Blue Nodules. (2016). Scope of blue nouldes.

Blue Nodules. (2020). Blue nodules results - deep sea mining. https://www.youtube.com/watch?v=JlRGNhwzJvM

Boschen, R. E., Rowden, A. A., Clark, M. R., & Gardner, J. P. (2013). Mining of deep-sea seafloor massive sulfides: A review of the deposits, their benthic communities, impacts from mining, regulatory frameworks and management strategies. *Ocean & coastal management*, *84*, 54–67.

Brennan, D. (2001). *The numerical simulation of two phase flows in settling tanks* (Doctoral dissertation). University of London.

Byishimo, P. (2018). Experiments and 3d cfd simulations of deep-sea mining plume dispersion and seabed interactions.

Camp, P. C. S., T. R. (1943). *Velocity gradients and internal work in fluid motion* (Vol. 30). Journal of Boston Society of Civil Engineering.

Chung, J. S. (2003). *Deep-ocean Mining Technology: Learning Curve I* (Vol. All Days) [ISOPE-M-03-001].

Dover, C. (2011). Tighten regulations on deep-sea mining. *Nature*, *470*, 31–3. https://doi.org/10.1038/470031a

Durden, J. M., Lallier, L. E., Murphy, K., Jaeckel, A., Gjerde, K., & Jones, D. O. (2018). Environmental impact assessment process for deep-sea mining in 'the area'. *Marine*

Policy, *87*, 194–202. https://doi.org/https://doi.org/10.1016/j.marpol.2017.10.013

Dyer, K. R. (1989). Sediment processes in estuaries: Future research requirements. *Journal of Geophysical Research: Oceans*, *94*(C10), 14327–14339. https://doi.org/https://doi.org/10.1029/JC094iC10p14327

Eisma, D. (1986). Flocculation and de-flocculation of suspended matter in estuaries. *Netherlands Journal of Sea Research*, *20*(2), 183–199. https://doi.org/https://doi.org/10.1016/0077-7579(86)90041-4

Elerian, M. (n.d.).

Enthoven, D. (2021). *Plume dispersion of low-density clayey suspension turbidity currents created by deep-sea mining* (Doctoral dissertation). http://resolver.tudelft.nl/uuid:0993ae18-3af3-4ce1-afbd-5cc9fd665912

Ferguson, R., & Church, M. (2004). A simple universal equation for grain settling velocity. *Journal of Sedimentary Research - J SEDIMENT RES*, *74*, 933–937. https://doi.org/10.1306/051204740933

Franco, A., Mosquera-Corral, A., Campos, J., & Roca, E. (2007). Learning to operate anaerobic bioreactors. *Communicating Current Research and Educational Topics in Applied Microbiology*.

Gersten, K. (2009). Hermann schlichting and the boundary-layer theory. In R. Radespiel, C.-C. Rossow, & B. W. Brinkmann (Eds.), *Hermann schlichting – 100 years* (pp. 3–17). Springer Berlin Heidelberg.

Gidaspow, D. (1994). *Multiphase flow and fluidization: Continuum and kinetic theory descriptions*. Academic press.

Gillard, B., Purkiani, K., Iversen, M. H., Vink, A., Chatzievangelou, D., & Thomsen, L. (2018). Physical and hydrodynamic properties of deep sea mining-generated, abyssal sediment plume in the clarion clipperton fracture zone (eastern-central pacific). https://doi.org/10.15146/R3K966

Goeree, J. (2018). *Drift-flux modeling of hyper-concentrated solid-liquid flows in dredging applications* (Doctoral dissertation). Delft University of Technology. https://doi.org/10.4233/uuid:2d432d11-cce4-40de-b951-e89dfebbef27

Golzarijalal, M., Ashtiani, F., & Dabir, B. (2017). Modeling of microalgal shear-induced flocculation and sedimentation using a coupled cfd-population balance approach. *Biotechnology Progress*, *34*. https://doi.org/10.1002/btpr.2580

Grabowski, R. C., Droppo, I. G., & Wharton, G. (2011). Erodibility of cohesive sediment: The importance of sediment properties. *Earth-Science Reviews*, *105*(3), 101–120. https://doi.org/https://doi.org/10.1016/j.earscirev.2011.01.008

Haan, C., Barfield, B., & Hayes, J. (1994). 7 - sediment properties and transport. In C. Haan, B. Barfield, & J. Hayes (Eds.), *Design hydrology and sedimentology for small catchments* (pp. 204–237). Academic Press. https://doi.org/https://doi.org/10.1016/B978-0-08-057164-5.50011-6

Hajdukiewicz, & Magdalena. (2013). Formal calibration methodology relating to cfd models of naturally ventilated internal environments.

Hamaker, H. (1937). The london—van der waals attraction between spherical particles. *Physica*, *4*(10), 1058–1072. https://doi.org/https://doi.org/10.1016/S0031-8914(37)80203-7

Hashemian, N., & Armaou, A. (2016). Simulation, model-reduction, and state estimation of a two-component coagulation process. *AIChE Journal*, *62*(5), 1557–1567. https://doi.org/10.1002/aic.15146

Hein, J. R., Mizell, K., Koschinsky, A., & Conrad, T. A. (2013). Deep-ocean mineral deposits as a source of critical metals for high- and green-technology applications: Comparison with land-based resources. *Ore Geology Reviews*, *51*, 1–14. https://doi.org/https://doi.org/10.1016/j.oregeorev.2012.12.001

Helmons, R. (2019). Development and testing of a hydraulic nodule collector while minimizing its environmental impact, preliminary results of discharge experiments (blue harvesting project). *48th Underwater Mining Conference*.

Hirsch, C. (2007). *Numerical computation of internal and external flows: The fundamentals of computational fluid dynamics*. Elsevier.

Holzmann, T. (2019). *Mathematics, numerics, derivations and openfoam®*.

Hounslow, M. J., Ryall, R. L., & Marshall, V. R. (1988). A discretized population balance for nucleation, growth, and aggregation. *AIChE Journal*, *34*(11), 1821–1832. https://doi.org/https://doi.org/10.1002/aic.690341108

Hundsdorfer, W., & Verwer, J. G. (2013). *Numerical solution of time-dependent advection-diffusion-reaction equations* (Vol. 33). Springer Science & Business Media.

Jankowski, J. A., & Zielke, W. (2001). The mesoscale sediment transport due to technical activities in the deep sea [Environmental Impact Studies for the Mining of Polymetallic Nodul es from the Deep Sea]. *Deep Sea Research Part II: Topical Studies in Oceanography*, *48*(17), 3487–3521. https://doi.org/https://doi.org/10.1016/S0967-0645(01)00054-6

Kármán, T. (1931). Mechanical similitude and turbulence.

Klassen, I. (2017). *Three-dimensional numerical modeling of cohesive sediment flocculation processes in turbulent flows* (Doctoral dissertation). Karlsruher Institut für Technologie (KIT). Karlsruher Institut für Technologie (KIT). https://doi.org/10.5445/IR/1000070360

Kumar, S., & Ramkrishna, D. (1996). On the solution of population balance equations by discretization—i. a fixed pivot technique. *Chemical Engineering Science*, *51*(8), 1311–1332. https://doi.org/https://doi.org/10.1016/0009-2509(96)88489-2

Lee, J. H. W., & Chu, V. (2003). Turbulent jets and plumes: A lagrangian approach.

Lee Byung Joon, M. F. (2014). Numerical simulation of turbulence-induced flocculation and sedimentation in a flocculant-aided sediment retention pond. *Environmental Engineering Research*, *19*(2), 165–174. https://doi.org/10.4491/eer.2014.19.2.165

LeVeque, R. J. et al. (2002). *Finite volume methods for hyperbolic problems* (Vol. 31). Cambridge university press.

Levin, Z., Neiburger, M., & Rodriguez, J. (1973). Experimental evaluation of collection efficiencies and coalescence efficiencies of cloud drops. *Journal of the Atmospheric Sciences*, *30*, 944–946. https://doi.org/10.1175/1520-0469(1973)030<0944:EEOCAC>2.0.CO;2

Lick, W., Huang, H., & Jepsen, R. (1993). Flocculation of fine-grained sediments due to differential settling. *Journal of Geophysical Research: Oceans*, *98*(C6), 10279–10288. https://doi.org/https://doi.org/10.1029/93JC00519

7

Liu, X.-D., & Tadmor, E. (1998). Third order nonoscillatory central scheme for hyperbolic conservation laws. *Numerische mathematik*, *79*(3), 397–425.

Lumborg, U., & Vested, H. (2008). Modelling of cohesive sediment dynamics (chapter 6).

Maggi, F. (2005). Flocculation dynamics of cohesive sediment.

Marchisio, D. L., & Fox, R. O. (2005). Solution of population balance equations using the direct quadrature method of moments. *Journal of Aerosol Science*, *36*(1), 43–73. https://doi.org/https://doi.org/10.1016/j.jaerosci.2004.07.009

Mehta, A. J. (1991). Review notes on cohesive sediment erosion.

Mero, J. L. (1965). Chemical composition of ocean manganese nodules using Quantum emission spectrography (QES) (previously unpublished, published online 2016). https://doi.org/10.1594/PANGAEA.864085

MIDAS. (2016). Managing impacts of deep-sea resource exploitation. https://www.eu-midas.net/

Miller, K. A., Thompson, K. F., Johnston, P., & Santillo, D. (2018). An overview of seabed mining including the current state of development, environmental impacts, and knowledge gaps. *Frontiers in Marine Science*, *4*, 418. https://doi.org/10.3389/fmars.2017.00418

Mirza, S., & Richardson, J. (1979). Sedimentation of suspensions of particles of two or more sizes. *Chemical Engineering Science*, *34*(4), 447–454. https://doi.org/https://doi.org/10.1016/0009-2509(79)85088-5

Protogene, B. (2018). *Experiments and 3d cfd simulations of deep-sea mining plume dispersion and seabed interactions* (Doctoral dissertation). Delft University of Technology.

Pruppacher, H., & Klett, J. (1996). *Microphysics of clouds and precipitation*. Springer Netherlands. https://books.google.nl/books?id=1mXN%5C_qZ5sNUC

Quezada, G. R., Jeldres, M., Robles, P., Toro, N., Torres, D., & Jeldres, R. I. (2020). Improving the flocculation performance of clay-based tailings in seawater: A population balance modelling approach. *Minerals*, *10*(9). https://doi.org/10.3390/min10090782

Quezada, G. R., Ramos, J., Jeldres, R. I., Robles, P., & Toledo, P. G. (2020). Analysis of the flocculation process of fine tailings particles in saltwater through a population balance model. *Separation and Purification Technology*, *237*, 116319. https://doi.org/https://doi.org/10.1016/j.seppur.2019.116319

Richardson, J., & Zaki, W. (1954). The sedimentation of a suspension of uniform spheres under conditions of viscous flow. *Chemical Engineering Science*, *3*(2), 65–73. https://doi.org/https://doi.org/10.1016/0009-2509(54)85015-9

Roberts, C., O'Leary, B., Mccauley, D., Cury, P., Duarte, C., Lubchenco, J., Pauly, D., Sáenz-Arroyo, A., Sumaila, R., Wilson, R., Worm, B., & Castilla, J. (2017). Marine reserves can mitigate and promote adaptation to climate change. *Proceedings of the National Academy of Sciences*, *114*, 201701262. https://doi.org/10.1073/pnas.1701262114

Rodi, W. (2017). *Turbulence models and their application in hydraulics: A state-of-the-art review, third edition*. https://doi.org/10.1201/9780203734896

Rowe, P. (1987). A convenient empirical equation for estimation of the richardson-zaki exponent. *Chemical Engineering Science*, 2795–2796.

Runkana, V., Somasundaran, P., & Kapur, P. (2006). A population balance model for flocculation of colloidal suspensions by polymer bridging [Advances in population balance modelling]. *Chemical Engineering Science*, *61*(1), 182–191. https://doi.org/https://doi.org/10.1016/j.ces.2005.01.046

Saffman, P. G., & Turner, J. S. (1956). On the collision of drops in turbulent clouds. *Journal of Fluid Mechanics*, *1*(1), 16–30. https://doi.org/10.1017/S0022112056000020

Sanz, M. (2018). Flocculation and consolidation of cohesive sediments under the influence of coagulant and flocculant.

Serra, T., & Casamitjana, X. (1998). Effect of the shear and volume fraction on the aggregation and breakup of particles. *Aiche Journal*, *44*, 1724–1730.

Shafrir, U., & Neiburger, M. (1963). Collision efficiencies of two spheres falling in a viscous medium. *Journal of Geophysical Research (1896-1977)*, *68*(13), 4141–4147. https://doi.org/https://doi.org/10.1029/JZ068i013p04141

Shen, X., Lee, B. J., Fettweis, M., & Toorman, E. A. (2018). A tri-modal flocculation model coupled with telemac for estuarine muds both in the laboratory and in the field. *Water Research*, *145*, 473–486. https://doi.org/https://doi.org/10.1016/j.watres.2018.08.062

Shrestha, P. L., & Blumberg, A. F. (2005). Cohesive sediment transport. In M. L. Schwartz (Ed.), *Encyclopedia of coastal science* (pp. 327–330). Springer Netherlands. https://doi.org/10.1007/1-4020-3880-1_95

Smoluchowski, M. V. (1916). Drei Vortrage uber Diffusion, Brownsche Bewegung und Koagulation von Kolloidteilchen. *Zeitschrift fur Physik*, *17*, 557–585.

Spearman, J., Taylor, J., Crossouard, N., Cooper, A., Turnbull, M., Manning, A., Lee, M., & Murton, B. (2020). Measurement and modelling of deep sea sedfviment plumes and implications for deep sea mining. *Scientific Reports*, *10*. https://doi.org/10.1038/s41598-020-61837-y

Sperling, M. V. (2007). Basic principles of wastewater treatment.

Spicer, P. T., & Pratsinis, S. E. (1996). Coagulation and fragmentation: Universal steady-state particle-size distribution. *AIChE Journal*, *42*(6), 1612–1620. https://doi.org/https://doi.org/10.1002/aic.690420612

Spicer, P. T., Pratsinis, S. E., Trennepohl, M. D., & Meesters, G. H. M. (1996). Coagulation and fragmentation:the variation of shear rate and the time lag for attainment of steady state. *Industrial & Engineering Chemistry Research*, *35*(9), 3074–3080. https://doi.org/10.1021/ie950786n

Srivatsan, L., Lake, L., & Bonnecaze, R. (2004). Scaling analysis of deposition from turbidity currents. *Geo-Mar. Lett.*, *24*, 63–74. https://doi.org/10.1007/s00367-003-0162-x

Stone, E., Rebecca. (1999). Entrainment, detrainment, and large-scale horizontal gradients in oceanic deep convection. https://calhoun.nps.edu/handle/10945/39421

The World Bank Group. (2017). *Minerals and metals to play significant role in a low-carbon future* (tech. rep.). http://hdl.handle.net/10986/28312

Thomas, D., Judd, S., & Fawcett, N. (1999). Flocculation modelling: A review. *Water Research*, *33*(7), 1579–1592. https://doi.org/https://doi.org/10.1016/S0043-1354(98)00392-3

Tsuji, Y., Kawaguchi, T., & Tanaka, T. (1993). Discrete particle simulation of two-dimensional fluidized bed. *Powder Technology*, *77*(1), 79–87. https://doi.org/https://doi.org/10.1016/0032-5910(93)85010-7

Vajihinejad, V., & Soares, J. B. (2018). Monitoring polymer flocculation in oil sands tailings: A population balance model approach. *Chemical Engineering Journal*, *346*, 447–457. https://doi.org/https://doi.org/10.1016/j.cej.2018.04.039

Valioulis, I. (1983). Particle collisions and coalescence in fluids. *California Institute of Technology, W. M. Keck Laboratory of Hydraulics and Water Resources, Repor*.

van Ledden, M., van Kesteren, W., & Winterwerp, J. (2004). A conceptual framework for the erosion behaviour of sand–mud mixtures. *Continental Shelf Research*, *24*(1), 1–11. https://doi.org/https://doi.org/10.1016/j.csr.2003.09.002

Van Leer, B. (1974). Towards the ultimate conservative difference scheme. ii. monotonicity and conservation combined in a second-order scheme. *Journal of computational physics*, *14*(4), 361–370.

van Leussen, W. (1994). *Estuarine macroflocs and their role in fine-grained sediment transport*. Universiteit Utrecht, Faculteit Aardwetenschappen. https://books.google.nl/books?id=cYs9NQAACAAJ

Vanni, M. (2000). Approximate population balance equations for aggregation–breakage processes. *Journal of Colloid and Interface Science*, *221*(2), 143–160. https://doi.org/https://doi.org/10.1006/jcis.1999.6571

Verney, R., Lafite, R., Claude Brun-Cottan, J., & Le Hir, P. (2011). Behaviour of a floc population during a tidal cycle: Laboratory experiments and numerical modelling [Proceedings of the 9th International Conference on Nearshore and Estuarine Cohesive Sediment Transport Processes]. *Continental Shelf Research*, *31*(10, Supplement), S64–S83. https://doi.org/https://doi.org/10.1016/j.csr.2010.02.005

Versteeg, H., & Malalasekera, W. (1995). An introduction to computational fluid dynamics - the finite volume method.

Volkmann, S. (2018). *Blue mining—planning the mining of seafloor manganese nodules* (Doctoral dissertation). https://doi.org/10.18154/RWTH-2018-230772

Wadell, H. (1932). Volume, shape, and roundness of rock particles. *The Journal of Geology*, *40*(5), 443–451. https://doi.org/10.1086/623964

Washburn, T. W., Turner, P. J., Durden, J. M., Jones, D. O., Weaver, P., & Van Dover, C. L. (2019). Ecological risk assessment for deep-sea mining. *Ocean Coastal Management*, *176*, 24–39. https://doi.org/https://doi.org/10.1016/j.ocecoaman.2019.04.014

Wilcox, D. (2006). *Turbulence modeling for cfd*. DCW Industries. https://books.google.nl/books?id=tFNNPgAACAAJ

Winterwerp, J. C. (1998). A simple model for turbulence induced flocculation of cohesive sediment. *Journal of Hydraulic Research*, *36*(3), 309–326. https://doi.org/10.1080/00221689809498621

A

SETTLING COLUMN EXPERIMENTS

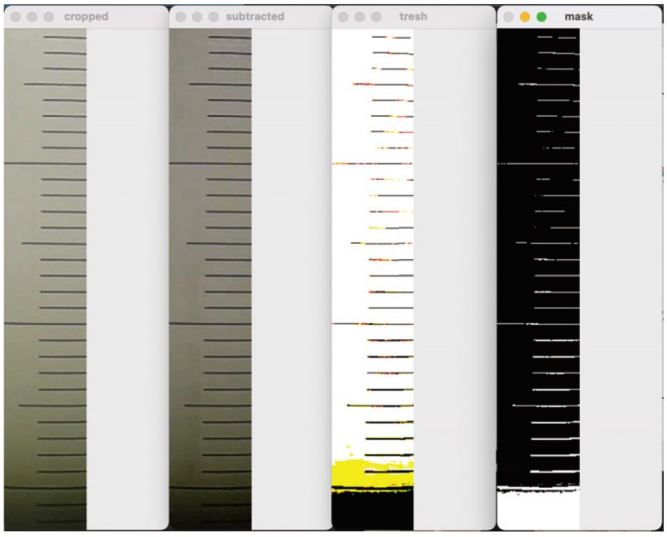


Figure A.1: Cropped, subtracted, tresh and mask pictures in settling column experiments (Enthoven, 2021)

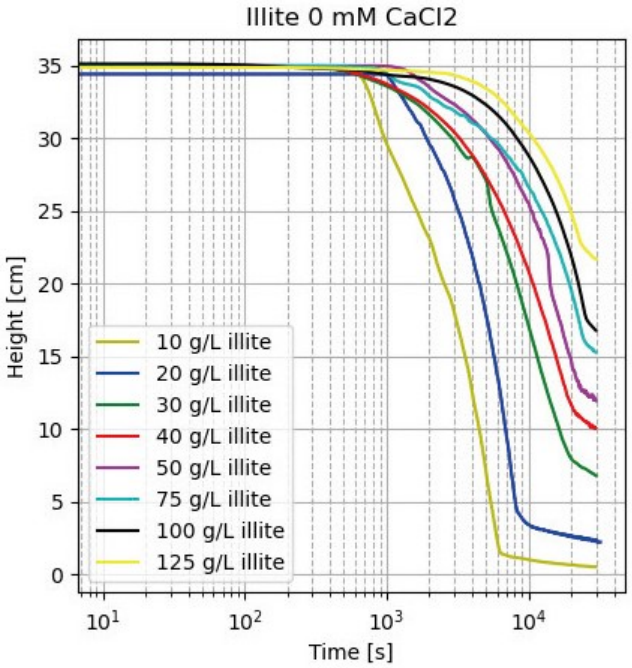


Figure A.2: Time variation of mud-line height for sediment of different initial concentration (Enthoven, 2021)

B

NUMERICAL ALGORITHM

B.1. HEAD FILE

Following code is added to the head file "multiphaseDriftMixture.H"

```
class multiphaseDriftMixture
:
    public IOdictionary,
    public transportModel
{

private:

    // Private data

        //- Name for each phase
        wordList phaseskName_;


      // Breakup Process
        //- An empirical parameter for break-up process
        scalar Eb_;

        //- Floc Strength
        scalar Fy_;

        //- Capacity Dimension
        scalar nf_;

        //- Breakaup Frequency Function
```

```
        PtrList<volScalarField>  S_;

        //- Breakage Distribution Function
        scalar Gamma_;

        //- Shear Rate
        volScalarField G_;

    // Aggregation Process
        //- Collision Efficiency
        scalar Alpha_;

        //- Collision Frequency
        PtrList<volScalarField>  Beta_;

        //- Temperature
        scalar T_;

    // Flocculation = Aggregation + Breakup
        //- A list to save phase transition terms for each sediment
            phase
        PtrList<volScalarField> SrcList;

        //- The summation of all phase transition terms (Should to
            be zero)
        volScalarField SumSource_;

    // Sediment Properties
        //- Diameters of particles
        scalarField diameters_;

        //- Volumes of particles
        scalarField volumes_;

            //- Height of cell centers
        volScalarField CellHeight_;

        //- Volume of cells
            volScalarField CellVolume_;

         // Private member functions

        //- Calculate phase transition terms due to flocculation
        tmp<volScalarField> calSource(label i);
```

```
public:


    // Member Functions

        //- Create a list to save the particle diameter of sediment
        void copyDiameters();

        //- Calculate the particle volume of sediment
        void CalcVolumes();

        //- Calculate turbulent shear rate
        void CalcG();

        //- Calculate collision frequency
        void ColFreq();

        //- Calculate breakup frequency
        void BrkRate();
};
```

B.2. SOURCE CODE FILE

Following code is added to the source code file "multiphaseDriftMixture.C"

```
// * * * * * * * * * * * * * * * * * * Constructors  * * * * * * * * * * *
    * * * * //

Foam::multiphaseDriftMixture::multiphaseDriftMixture
(
    const volVectorField& U,
    const surfaceScalarField& phi
)
:
    ...
    phaseskName_(phasesk_.toc()),
    alphas_
    (
        IOobject
        (
            "alpha.solids",
            mesh_.time().timeName(),
            mesh_,
```

```
            IOobject::NO_READ,
            IOobject::AUTO_WRITE
        ),
        mesh_,
        dimensionedScalar("alphas", dimless, 0.0)
    ),

    // Read collision efficiency
    Eb_(lookupOrDefault<scalar>("Eb", 1e-5)),

    // Read floc strength
    Fy_(lookupOrDefault<scalar>("Fy", 1e-10)),

    // Read capacity dimension
    nf_(lookupOrDefault<scalar>("nf", 2.4)),

    // Initialize the breakup frequency
    S_(phasesk_.size()),

    // Initialize the shear rate
    G_
    (
        IOobject
        (
            "G",
            mesh_.time().timeName(),
            mesh_,
            IOobject::NO_READ,
            IOobject::AUTO_WRITE
        ),
        mesh_,
        dimensionedScalar("G", dimless, scalar(0.0))
    ),

    // Initialize the breakup distribution function
    Gamma_(lookupOrDefault<scalar>("Gamma", 2.0)),

    // Read collision efficiency
    Alpha_(lookupOrDefault<scalar>("Alpha", 1.0)),

    // Initialize the collision frequency
    Beta_(phasesk_.size()*phasesk_.size()),

    // Initialize the temperature
    T_(lookupOrDefault<scalar>("T", 293.0)),
```

```
// Initialize the phase transition source terms
SrcList(phasesk_.size()),

// Initialize the summation of the phase transition source terms
    for all sediment phases
SumSource_
(
    IOobject
    (
        "SumSource",
        mesh_.time().timeName(),
        mesh_,
        IOobject::NO_READ,
        IOobject::AUTO_WRITE
    ),
    mesh_,
    dimensionedScalar("SumSource", dimless/dimTime, scalar(0.0))
),

// Initialize the particle diameter of sediment
diameters_(phasesk_.size()),

// Initialize the particle volume of sediment
volumes_(phasesk_.size()),

CellHeight_
(
    IOobject
    (
        "CellHeight",
        mesh_.time().timeName(),
        mesh_,
        IOobject::NO_READ,
        IOobject::AUTO_WRITE
    ),
    mesh_,
    dimensionedScalar("CellHeight", dimLength, scalar(0.0))
),

CellVolume_
(
    IOobject
    (
        "CellVolume",
```

```
            mesh_.time().timeName(),
            mesh_,
            IOobject::NO_READ,
            IOobject::AUTO_WRITE
        ),
        mesh_,
        dimensionedScalar("CellVolume", dimVolume, scalar(0.0))
    )
{

    // Calculate the height of cell centers
    CellHeight_ = mesh_.C().component(vector::Z);
    forAll(CellHeight_.boundaryField(),cellI )
    {
        CellHeight_[cellI] = 0;
    }
    CellVolume_.ref() = mesh_.V();

    // Create the list for particle diameter of sediment
    copyDiameters();

    // Calculate the particle volume
    CalcVolumes();

    // Initialize the value of collision frequency function for each
        phase
    forAll(Beta_,i)
    {
        word nameBeta("Beta"+ i);
        Beta_.set
        (
            i,
            new volScalarField
            (
                IOobject
                (
                    nameBeta,
                    mesh_.time().timeName(),
                    mesh_
                ),
                mesh_,
                dimensionedScalar(nameBeta, dimless, scalar(0.0))
            )
        );
    }
```

```
// Initialize the value of the phase transition source term for
    each phase
forAll(SrcList,i)
{
    word nameSrc("SrcPhase."+phaseskName_[i]);
    SrcList.set
    (
        i,
        new volScalarField
        (
            IOobject
            (
                nameSrc,
                mesh_.time().timeName(),
                mesh_
            ),
            mesh_,
            dimensionedScalar(nameSrc, dimless/dimTime, scalar
                (0.0))
        )
    );
}

// Initialize the value of breakup frequency function for each
    phase
forAll(S_,i)
{
    word nameS("S"+ i);
    S_.set
    (
        i,
        new volScalarField
        (
            IOobject
            (
                nameS,
                mesh_.time().timeName(),
                mesh_
            ),
            mesh_,
            dimensionedScalar(nameS, dimless, scalar(0.0))
        )
    );
}
```

```
    ...
}

// * * * * * * * * * * * * * * * Member Functions * * * * * * * * * * *
    * * * * //

void Foam::multiphaseDriftMixture::solveAlphas()
{
    ...
    label phasei = 0;

    // Calculate turbulent shear rate
    CalcG();

    // Compute collision frequency
    ColFreq();

    // Compute breakage rate
    BrkRate();

    // Initialize the summation of phase transition terms
    forAll(SumSource_,cellI)
    {
        SumSource_[cellI] = 0;
    }

    // Compute the phase transition terms due to flocculation for
        all sediment phases
    for (label phasei = 0; phasei <= phasesk_.size()-1; phasei++)
    {
        // Compute the phase transition term for phase i
        SrcList[phasei]=calSource(phasei);
        // Check whether the summation of all phase transition terms
            is zero
        SumSource_ += SrcList[phasei];
    }

    forAllIters (phasesk_, iter)
    {
    // Assign the phase transition term for phase i
        volScalarField Sa = SrcList[phasei];

        phaseDrift& alpha = iter();

        surfaceScalarField alphaPhi
```

```
(
    IOobject
    (
        "alphaPhi",
        mesh_.time().timeName(),
        mesh_
    ),
    mesh_,
    dimensionedScalar("0", phi_.dimensions(), 0.0)
);

if (MULESCorr)
{
    // Solve phase continuity equations
    fvScalarMatrix alpha1Eqn
    (
        fv::EulerDdtScheme<scalar>(mesh_).fvmDdt(alpha)
      + fv::gaussConvectionScheme<scalar>
        (
            mesh_,
            phi_,
            upwind<scalar>(mesh_, phi_)
        ).fvmDiv(phi_, alpha)-Sa
    );

    alpha1Eqn.solve();
    alphaPhi = alpha1Eqn.flux();
}

for (int aCorr = 0; aCorr<nAlphaCorr; aCorr++)
{
    if (MULESCorr)
    {
        tmp<surfaceScalarField> talphaPhiCorr
        (
            alphaPhiCorrs[phasei] - alphaPhi
        );

        volScalarField alpha10("alpha10", alpha);

        MULES::correct
        (   //test it
            geometricOneField(),
            alpha,
            alphaPhiCorrs[phasei],
```

```
                talphaPhiCorr.ref(),
                UniformField<scalar>(alphaMax_),
                zeroField()

            );

            // Under-relax the correction for all but the 1st
                corrector
            if (aCorr == 0)
            {
                alphaPhi += talphaPhiCorr();
            }
            else
            {
                refCast<volScalarField>(alpha) = 0.5*alpha +
                    0.5*alpha10;
                alphaPhi += 0.5*talphaPhiCorr();
            }
        }
        else
        {
            surfaceScalarField& alphaPhi = alphaPhiCorrs[phasei
                ];

            MULES::explicitSolve
            (
                geometricOneField(),
                alpha,
                phi_,
                alphaPhi,
                UniformField<scalar>(alphaMax_),
                zeroField()
            );
        }
        calculateAlphaPhi(alphaPhiCorrs[phasei], alpha, phasei);
    }

    // Apply the diffusion term separatelly
    {
    const Time& runTime = mesh_.time();
    volScalarField varible = Foam::fvc::laplacian(sumAlpha);
    volScalarField::Internal varible2 = varible.internalField();
    dimensionedScalar dimcor
    (
    "dimcor",
```

```
dimensionSet(0, -2, 0, 0, 0, 0 ,0),
SMALL   // 1.0e-6
);
dimensionedScalar dimcor1
(
"dimcor1",
dimensionSet(0, 0, -1, 0, 0, 0 ,0),
1
);
//calculation of the kLimit
kLimit_ =
((alphaMax_ - sumAlpha)/ runTime.deltaT().value())
*dimcor1   /(varible +dimcor);
//calcuation of the klim
scalarField deltaX = cbrt(mesh_.V());
forAll (deltaX, i)
{
klim_[i] = deltaX[i] * deltaX[i]/runTime.deltaT().value();
}
//calculation of the alphaDiffusion 1
forAll (alphas_, cellI)
{

     if (alphas_[cellI] > alphaMax_)
        {
        alphaDiffusion_1_[cellI] = alphaDiffusion_;
        }

}
//
fvScalarMatrix alpha1Eqn
(
    fvm::ddt(alpha) - fvc::ddt(alpha)
  - fvm::laplacian(turbulencePtr_->nut() +
    alphaDiffusion_1_, alpha)
);

    alpha1Eqn.solve(mesh_.solver("alpha1Diffusion"));

    alphaPhi += alpha1Eqn.flux();
}

rhoPhi_ += alphaPhi*alpha.rho();
Info<< alpha.name() << "␣volume␣fraction,␣min,␣max␣=␣"
    << alpha.weightedAverage(mesh_.V()).value()
```

```
            << ' ' << min(alpha).value()
            << ' ' << max(alpha).value()
            << endl;
        //
        volScalarField unitalpha
        (
            IOobject
            (
                "unitalpha",
                mesh_.time().timeName(),
                mesh_
            ),
            mesh_,
            dimensionedScalar( "unitalpha", dimless, scalar(1.0))
        );
        sumAlpha += alpha;

        phasei++;
    }
    volScalarField& alphac= phasec_.first();

    //Compute the vlomumetric concentration of the continuous phase
    alphac=1.0-sumAlpha;
    phaseDrift& alpha= phasec_.first();
    surfaceScalarField alphaPhi
        (
            IOobject
            (
                "alphaPhi",
                mesh_.time().timeName(),
                mesh_
            ),
            mesh_,
            dimensionedScalar("0", phi_.dimensions(), 0.0)
        );
    calculateAlphaPhi(alphaPhi, alpha, phasei);
    fixedFluxOnPatches(alphaPhi, alpha);
    rhoPhi_ += alphaPhi*alpha.rho();
    Info<< alpha.name() << " volume fraction , min, max = "
        << alpha.weightedAverage(mesh_.V()).value()
        << ' ' << min(alpha).value()
        << ' ' << max(alpha).value()
        << endl;
    sumAlpha += alpha;
```

```
    calcAlphas();
    Info<< "Phase-sum volume fraction, min, max = "
        << sumAlpha.weightedAverage(mesh_.V()).value()
        << ' ' << min(sumAlpha).value()
        << ' ' << max(sumAlpha).value()
        << endl;
}

// Create a list to save sediment particles' diameters
void Foam::multiphaseDriftMixture::copyDiameters()
{
    label m = 0;
    forAllIter(PtrDictionary<phaseDrift>, phasesk_, iter)
    {
    diameters_[m] = iter().d().value();
    m++;
    }
}

// Create a list to save sediment particles' volumes
void Foam::multiphaseDriftMixture::CalcVolumes()
{
    volumes_[0] =
    1.0/6.0*Foam::constant::mathematical::pi*pow(diameters_[0],3.0);

    for (label j =1; j<=volumes_.size()-1;j++)
    {
        volumes_[j] = pow(2.0,j*1.0)*volumes_[0];
    }

}

// Calculate turbulent shear rate
void Foam::multiphaseDriftMixture::CalcG()
{
    volScalarField epsField = turbulencePtr_->epsilon()();
    volScalarField nuField = phasec_.first().nu()();
    volScalarField Gunit
    (
        IOobject
        (
            "Gunit",
            mesh_.time().timeName(),
            mesh_
        ),
```

```
            mesh_,
            dimensionedScalar("Gunit", dimless/dimTime/dimTime, scalar
                (1.0))
        );
        G_ = pow(epsField/nuField/Gunit,1.0/2.0);
}

// Calculate the collision frequencies \beta_{i,j}
void Foam::multiphaseDriftMixture::ColFreq()
{
    scalarField diameterlist(phasesk_.size());

    label m = 0;
    forAllIter(PtrDictionary<phaseDrift>, phasesk_, iter)
    {
        diameterlist[m] = iter().d().value();
        m++;
    }
    // Unit of velocity [m/s]
    volScalarField Vunit
    (
        IOobject
        (
            "Vunit",
            mesh_.time().timeName(),
            mesh_
        ),
        mesh_,
        dimensionedScalar("Vunit", dimLength/dimTime, scalar(1.0))
    );

    // Unit of viscosity [kg/(m*s)]
    volScalarField muunit
    (
        IOobject
        (
            "muunit",
            mesh_.time().timeName(),
            mesh_
        ),
        mesh_,
        dimensionedScalar("muunit", dimMass/dimLength/dimTime,
            scalar(1.0))
    );
```

```
// Boltzmann constant
scalar Kb = 1.38064852e-23;

forAll(Beta_,i)
{
    word nameBeta("Beta"+ i);
    Beta_.set
    (
        i,
        new volScalarField
        (
            IOobject
            (
                nameBeta,
                mesh_.time().timeName(),
                mesh_
            ),
            mesh_,
            dimensionedScalar(nameBeta, dimless, scalar(0.0))
        )
    );
}

forAll (diameterlist, i)
{
    forAll(diameterlist,j)
    {
        label I = j +i*phasesk_.size();

        // Collision frequency under the mechanism of turbulent
            shear
        volScalarField Beta_shear = (1.0/6.0)*G_*pow((
            diameterlist[i]+diameterlist[j]),3.0);

        // Collision frequency under the mechanism of
            differential settling
        volScalarField Beta_settling =  Foam::constant::
            mathematical::pi/4.0*pow((diameterlist[i]+
            diameterlist[j]),2.0)*(mag(mag(UkmPtr_[i]-UkmPtr_[j
            ])/Vunit));

        // Collision frequency under the mechanism of Brownian
            motion
        volScalarField Beta_Brownian =
        2.0/3.0*Kb*T_/(phasec_.first().mu()()/muunit)
```

```
                *pow((diameterlist[i]+diameterlist[j]),2.0)/(
                    diameterlist[i]*diameterlist[j]);

                // Total collision frequency function
                Beta_[I] = Beta_shear+Beta_settling+Beta_Brownian;
            }
        }

}

// Calculate Breakup frequency
void Foam::multiphaseDriftMixture::BrkRate()
{
    volScalarField muunit
    (
        IOobject
        (
            "muunit",
            mesh_.time().timeName(),
            mesh_
        ),
        mesh_,
        dimensionedScalar("muunit", dimMass/dimLength/dimTime,
            scalar(1.0))
    );
    // Breakup frequency function proposed by Winterwerp
    forAll(volumes_,i)
    {
    S_[i]= Eb_*G_*pow(((diameters_[i]-diameters_[0])/diameters_[0])
        ,3.0-nf_)*pow((phasec_.first().mu()()/muunit*G_)/(Fy_/pow(
        diameters_[i],2.0)),0.5);
    }

}

// Calculate phase transition terms
Foam::tmp<Foam::volScalarField>
Foam::multiphaseDriftMixture::calSource(label i)
{

   //Declare source terms
    volScalarField Source
    (
        IOobject
        (
```

```
            "Source",
            mesh_.time().timeName(),
            mesh_
        ),
        mesh_,
        dimensionedScalar("Source", dimless, scalar(0.0))
    );

    volScalarField Source1
    (
        IOobject
        (
            "Source1",
            mesh_.time().timeName(),
            mesh_
        ),
        mesh_,
        dimensionedScalar("Source1", dimless, scalar(0.0))
    );

    volScalarField Source2
    (
        IOobject
        (
            "Source2",
            mesh_.time().timeName(),
            mesh_
        ),
        mesh_,
        dimensionedScalar("Source2", dimless, scalar(0.0))
    );

    volScalarField Source3
    (
        IOobject
        (
            "Source3",
            mesh_.time().timeName(),
            mesh_
        ),
        mesh_,
        dimensionedScalar("Source3", dimless, scalar(0.0))
    );

    volScalarField Source4
```

```
(
    IOobject
    (
        "Source4",
        mesh_.time().timeName(),
        mesh_
    ),
    mesh_,
    dimensionedScalar("Source4", dimless, scalar(0.0))
);

volScalarField Source5
(
    IOobject
    (
        "Source5",
        mesh_.time().timeName(),
        mesh_,
        IOobject::NO_READ,
        IOobject::AUTO_WRITE
    ),
    mesh_,
    dimensionedScalar("Source5", dimless, scalar(0.0))
);

volScalarField Source6
(
    IOobject
    (
        "Source6",
        mesh_.time().timeName(),
        mesh_,
        IOobject::NO_READ,
        IOobject::AUTO_WRITE
    ),
    mesh_,
    dimensionedScalar("Source6", dimless, scalar(0.0))
);

//- First Term: Aggregation of particles of unequal size
if (i>=2)
{
    for (label j = 0; j <= i-2 ; j++)
    {
        Source1+=pow(2.0,(j-i+1)*1.0)*Alpha_*Beta_[j +(i-1)*
```

```
                phasesk_.size()]*phasesk_[phaseskName_[i-1]]/
                volumes_[i-1]*phasesk_[phaseskName_[j]]/volumes_[j]*
                volumes_[i];
        }
    }

    //- Second Term: Aggregation of particles of equal size
    if (i>=1)
    {
         Source2+= Alpha_*Beta_[i-1 +(i-1)*phasesk_.size()]*phasesk_
             [phaseskName_[i-1]]/volumes_[i-1]*phasesk_[phaseskName_
             [i-1]];
    }

     //- Third Term: Aggregation with smaller particles
    if(i>=1 && i<=phasesk_.size()-2)
    {
        for (label j = 0; j <= i-1 ; j++)
        {
            Source3+=(-1.0)*Alpha_*Beta_[j+i*phasesk_.size()]*
                phasesk_[phaseskName_[i]]/volumes_[i]*phasesk_[
                phaseskName_[j]];

        }
    }

    //- Fourth Term: Aggregation with equal or larger particles
    for (label j = i; j <= phasesk_.size()-2; j++)
    {
         Source4+=(-1.0)*Alpha_*Beta_[j+i*phasesk_.size()]*phasesk_[
             phaseskName_[i]]/volumes_[i]*phasesk_[phaseskName_[j]]/
             volumes_[j]*volumes_[i];
    }

    //- Fifth Term: break-up of particles belonging to size class i
    if (i>=1)
    {
        Source5+=(-1.0)*S_[i]*phasesk_[phaseskName_[i]];
    }

    //- Sixth Term: due to breakage of particles belonging to size
        class i+1
    if(i<=phasesk_.size()-2)
    {
        Source6+=1.0/2.0*Gamma_*S_[i+1]*phasesk_[phaseskName_[i+1]];
```

```
    }

    // Flocculation = Aggregation + Breakup
    Source = Source1+Source2+Source3+Source4+Source5+Source6;

    // Unit of the phase transition terms due to flocculation [s^-1]
    dimensionedScalar dimCorr("dimCorr",dimless/dimTime,1.0);

    // Add the unit
    return Source*dimCorr;
}
```