

Particle Dynamics Simulation toward High-Shear Mixing Process in Many Particle Systems

Siyu Zhu

Submitted in partial fulfillment of the
requirements for the degree
of Doctor of Philosophy
in the Graduate School of Arts and Sciences

COLUMBIA UNIVERSITY

2018

©2018

Siyu Zhu

All Rights Reserved

ABSTRACT

Particle Dynamics Simulation toward High-Shear Mixing Process in Many Particle Systems

Siyu Zhu

Granular materials appear in a broad range of industrial processes, including mineral processing, plastics manufacturing, ceramic component, pharmaceutical tablets and food products. Engineers and scientists are always seeking efficient tools that can characterize, predict, or simulate the effective material properties in a timely manner and with acceptable accuracy, such that the cost for design and develop novel composite granular materials could be reduced.

The major scope of this dissertation covers the development, verification and validation of particle system simulations, including solid-liquid two-phase particle mixing process and foaming asphalt process. High shear mixing process is investigated in detail with different types of mixers. Besides particle mixing study, one liquid-gas two phase foaming asphalt simulation is studied to show the broad capacity of our particulate dynamics simulation scheme. Methodologies and numerical studies for different scenarios are presented, and acceleration plans to speed up the simulations are discussed in detail.

The dissertation starts with the problem statement, which briefly demonstrates the background of the problem and introduces the numerical models built from the physical world. In this work, liquid-solid two-phase particle mixing process is mainly studied. These mixing processes are conducted in a sealed mixer and different types of particles are mixed with the rotation of the mixer blades, to obtain a homogeneous particle mixture. In addition to the solid-liquid particle mixing problem, foaming asphalt problem, which is a liquid-gas two phase flow problem is also investigated. Foaming asphalt is generated by injecting a small amount of liquid additive (usually water) to asphalt at a high temperature. The volume change during this asphalt foaming process is studied.

Given the problem statement, detailed methodologies of particle dynamics simulation are illustrated. For solid-liquid particle mixing, Smoothed Particle Hydrodynamics (SPH) and Discrete

Element Method (DEM) are introduced and implemented to simulate the dynamics of solid and liquid particles, respectively. Solid-liquid particle interactions are computed according to Darcy's Law. Then the proposed SPH coupling DEM model is verified by three classical case studies.

For foaming asphalt problems, a SPH numerical model for foaming asphalt simulation is proposed, and simulations with different water contents, pressures and temperatures are conducted and the results agree with the experiments well. The coupled SPH-DEM method is applied to the particle mixing process, and several particle mixing numerical studies are conducted and these simulations are analyzed in multiple aspects. For the solid-liquid particle mixing problem, liquid plays an important role in the mixing performance. The effects of liquid content and liquid viscosity on mixing performance are studied. The mixing indexes of the mixture are applied to analyze the mixing quality, and the differences between three kinds of mixing indexes are discussed. Then mixers commonly used in industry such as Double Planetary Mixer (DPM) are modeled in mixing simulation and their results are compared with the experiments.

Similar to other numerical simulation problems, the scale of the model and the accuracy of the simulation results are constrained by the computational capacity. Our in-house software package Particle Dynamics Parallel Simulator(PDPS) has been used as a platform to implement the algorithms above and conduct the simulations. Two parallel computing methods of Message Passing Interface (MPI) parallel computing and Graphics Processing Unit (GPU) acceleration have been used to accelerate the simulations. Speedup results for both MPI parallel computing and GPU methods are illustrated in the case studies.

In summary, a comprehensive approach for particle simulation is proposed and applied to particle mixing process and asphalt foaming simulation. The simulation results are analyzed in various aspects to provide valuable insights to the problems studied in this work. Given the improvement of computational capacity, particle dynamics in higher resolution and simulations in more complex configurations can be obtained. This particle simulation platform is general and it can be straightforwardly extended to many-particle systems with more particle phases and solid-liquid-gas dynamics problems.

Table of Contents

LIST OF FIGURES	v
LIST OF TABLES	ix
LIST OF SYMBOLS	xi
CHAPTER 1 Introduction	1
1.1 Particulate Mixing Process	1
1.2 Particulate Simulation Method	3
1.3 Particulate Mixing Simulation Acceleration	5
1.4 Foaming Asphalt	5
1.5 Scope	7
CHAPTER 2 Problem Statement	10
2.1 Multiphase Flow Problems	10
2.2 Asphalt Foaming Process	11
2.2.1 Introduction	11
2.2.2 Experiment	11
2.2.3 Challenges and Simulation Plans	13
2.3 Multiphase High Shear Mixing Problems	15
2.3.1 Introduction	15
2.3.2 Experiment	15
2.3.3 Challenges and Simulation Plans	18
CHAPTER 3 Algorithm and Formulation	21
3.1 Smoothed Particle Hydrodynamics (SPH)	21
3.2 Discrete Element Method (DEM)	24

3.3	Coupling between SPH and DEM	26
3.4	Boundary Conditions	28
3.5	Stability Analysis	29
3.6	Particle Dynamics Parallel Simulator (PDPS)	30
3.6.1	Overview	30
3.6.2	Domain Decomposition	31
3.6.3	Neighbor List Construction	33
3.6.4	Force Calculation	33
3.6.5	Time Integration	34
3.7	Benchmark Studies	35
3.7.1	A single DEM Particle Falling in a Fluid	35
3.7.2	Sedimentation of a Granular Block	37
3.7.3	Liquid Flowing through a Porous Media	40
CHAPTER 4	Foaming Asphalt Simulation	45
4.1	Foaming Characterizations	45
4.2	Model Description	46
4.2.1	Simulation Box	47
4.2.2	Initial Expansion Ratio Computation	48
4.2.3	Bubble Collapsing Model	50
4.2.4	Adaption with Accumulation Process	52
4.3	Simulation Results and Validation	52
4.3.1	Experimental Results	52
4.3.2	Numerical Results	55
4.3.3	Parametric Studies	59
4.4	Conclusions and Discussion	63
CHAPTER 5	Mixing Index Analysis	65
5.1	Simulation Setup	65
5.2	The Mixing Index	67
5.2.1	Homogeneity	67
5.2.2	Randomness	78

5.2.3	Time Series Analysis	80
5.3	Criteria Analysis	84
5.3.1	Centroid Method	84
5.3.2	Local Average Method	85
5.3.3	Time Series Analysis	87
5.4	Summary	87
CHAPTER 6	Effect of Liquid on High Shear Mixing	89
6.1	Mixing Process Analysis	89
6.2	The Effect of Liquid Content on Mixing Property	93
6.3	The Effect of Liquid Viscosity on Mixing Property	95
6.4	Discussion	97
CHAPTER 7	Application to Different Mixers	100
7.1	Double Planetary Mixer	100
7.1.1	Experimental Study	101
7.1.2	Numerical Study and Comparison	102
7.2	Baker Perkins Dual Planetary Mixer	108
7.2.1	Simulation Study	108
7.3	Comparison between Different Mixers	109
7.3.1	Simulation Setup	109
7.3.2	Mixing Index Comparison	115
7.3.3	Liquid Distribution Comparison	118
7.4	Summary	120
CHAPTER 8	Conclusions and Future Work	122
8.1	Summary	122
8.2	Key Results	123
8.2.1	Algorithm and Formulation	123
8.2.2	Foaming Asphalt Simulation	124
8.2.3	Particle Mixing Problem Simulation	124
8.2.4	Computational Speedup	125
8.3	Future Work	126

I	Bibliography	127
II	Appendices	143
A	Parallel Computing on Message Passing Interface(MPI)	144
A.1	Amdahl's Law	144
A.2	Acceleration Study	144
B	General-Purpose Computing on Graphics Processing Units (GPU)	147
B.1	Overview	147
B.2	Memory Allocation in Parallel Computation	148
B.3	Race Condition	148
B.4	GPU Parallel Algorithm of PDPS	149
B.5	Numerical Study and Results	152
B.6	Discussion	156
C	Model Reduction	157
C.1	Model Reduction on Mixing Index	157
C.2	Model Reduction on Mixing Status Profile	159

List of Figures

2.1	Laboratory configuration of the self-developed foamer	12
2.2	LDS for characterization of foaming evolution history	13
2.3	An example of multiphase high shear mixing simulation	16
2.4	Double Planetary Mixer	17
2.5	Orbital blade motion of the blades in Double Planetary Mixer	17
2.6	Final State of a Mixing Experiment Using a Double Planetary Mixer	18
3.1	A schematic illustration of the domain decomposition technique	31
3.2	A flow chart of the Particle Dynamics Parallel Simulator(PDPS)	32
3.3	Sedimentation of a single DEM particle among SPH liquid particles	36
3.4	Sedimentation velocity of a single DEM particle among SPH particles	38
3.5	Snapshots of the process of sedimentation. Green dots are SPH particles(water) and DEM particles(solid) are in orange color	39
3.6	Velocity evolution of the porous block with different porosities	41
3.7	Terminal velocity of the porous block at different porosities	42
3.8	SPH liquid flow through porous media	43
3.9	Height difference of SPH liquid flowing through porous media	44
4.1	Snapshot of the foaming simulation at $t = 10$	48
4.2	Snapshot of Foaming asphalt collapse process at $t=0s, 20s$ and $50s$	51
4.3	Volume adaption for the first 10 seconds accumulation period	53
4.4	Experiment result of foaming asphalt Expansion Ratio (ER)	54

4.5	Comparison of the foaming evolution between experiment and simulation at water content of 2%	56
4.6	Comparison of the foaming evolution between experiment and simulation at water content of 1%	57
4.7	Comparison of the foaming evolution between experiment and simulation at water content of 3%	58
4.8	Comparison of HL of the foamed asphalt at different water contents	59
4.9	Comparison of FI of the foamed asphalt at different water contents	60
4.10	Effect of the pressure in the function chamber on the foaming evolution in the foamed asphalt	61
4.11	Effect of the temperature of the asphalt and the temperature in the function chamber on the foaming evolution in the foamed asphalt	62
5.1	Snapshot of the high shear mixing simulation mixer	66
5.2	Initial state of mixing problem	68
5.3	Schematic illustration of drawbacks of centroid and local average property measurement methods	70
5.4	Centroid of trace particles at x axis	71
5.5	Centroid of trace particles at y axis	72
5.6	Centroid of trace particles at z axis	72
5.7	Coefficient of variation at different AB	76
5.8	Mixing indexes at different AB	77
5.9	Orientational Order	79
5.10	Orientational order index during the mixing process	80
5.11	Choice of Averaging Box (AB) in the simulation domain	82
5.12	Number portion of first type solid particle at the center AB	82
5.13	Number portion of first type solid particle at the middle AB	83
5.14	Number portion of first type solid particle at the rim AB	83
6.1	Topview of particle distribution after mixing (a)solid particles (b)liquid particles . .	90
6.2	Liquid distribution on height	91

6.3	Particle volume fraction on height	92
6.4	Liquid distribution evolution during mixing process	93
6.5	Mixing index at different liquid amount	94
6.6	Liquid distribution comparison of different liquid contents (a) 28.9% liquid (b)11.9% liquid (c) 4.5% liquid	95
6.7	Mixing index at different viscosity	96
6.8	Liquid distribution comparison of high viscosity and low viscosity at different time (a) $\nu = 1m^2/s$ t=0 (b) $\nu = 1m^2/s$ t=2s (c) $\nu = 1m^2/s$ t=5s (d) $\nu = 0.04m^2/s$ t=0 (e) $\nu = 0.04m^2/s$ t=2s (f) $\nu = 0.04m^2/s$ t=5s	98
7.1	Comparison of Double Planetary Mixer (a) experiment (b) simulation	101
7.2	Sample in mixing study to test homogeneity	103
7.3	Initial State of DPM simulation	104
7.4	DPM simulation with 84% solid load	105
7.5	Final State of DPM simulation	106
7.6	Baker Perkins Planetary Mixer	109
7.7	Baker Perkins Planetary Mixer Model	110
7.8	Initial State of Baker Perkins Planetary Mixing	111
7.9	Middle State of Baker Perkins Planetary Mixing	112
7.10	Final State of Baker Perkins Planetary Mixing	113
7.11	Top View of Baker Perkins Planetary Mixing	114
7.12	Final Mixing State when the Blades Are Moved Out	116
7.13	Final Mixing State After the Surface is Flattened	117
7.14	Liquid Distribution at Height at Different Mixers	119
8.1	MPI speedup with multiple CPU cores compared to one CPU core	146
8.2	Program Flow with GPU Computation	150
8.3	Initial state of mixing process	154
8.4	Time Comparison of CPU and both CPU + GPU	155
8.5	Comparison of mixing index for different regression method when $w = 0.75rad/s$. .	160
8.6	Comparison of mixing index for different regression method when $w = 0.85rad/s$. .	161

8.7	Mass density distribution during mixing process	162
8.8	Mass density profile predicted by Linear regression	163
8.9	Mass density profile predicted by kriging regression	164

List of Tables

2.1	DPM experiment with 84% solid load	16
3.1	Comparison of half neighbor list and complete neighbor list	33
3.2	Parameters of the DEM and SPH particles in the sedimentation case study	37
3.3	Parameters of the DEM and SPH particles in the block sedimentation case study	39
3.4	Parameters of particles in the case of a liquid flowing through a porous media case	41
4.1	Parameters of asphat particles in SPH simulation	49
4.2	Physical parameter used in computing gas density	49
5.1	Geometric parameters of the high shear mixer	67
5.2	Parameters of solid and liquid particles in high shear mixing simulation	69
5.3	Distribution of Averaging Box at three different sizes	75
5.4	Mean and standard deviation of number portion in chosen ABs	84
5.5	Time series analysis of number portion in different ABs after 10s	84
7.1	DPM experiment with 84% solid load	102
7.2	Geometric parameters of the Double Planetary Mixer	103
7.3	DPM simulation with 84% solid load	107
7.4	Homogeneity comparison for DPM mixing between experiment and simulation	107
7.5	Geometric parameters of the Double Planetary Mixer	115
7.6	Coefficient of Variation Comparison of Three Mixers	118
8.1	Domain Decomposition Plan for different number of CPU Cores	145
8.2	Computation Time Comparison with Different Number of CPU Cores	147

8.3	Parameters of solid and liquid particles in DPM simulation	153
8.4	Time Comparison of CPU-only and GPU Computation	155

List of Symbols and Abbreviations

x	position	PDPS	Particle Dynamics Parallel Simulator
v	velocity	SPH	Smoothed Particle Hydrodynamics
a	acceleration	DEM	Discrete Element Method
t	time	WMA	Warm Mix Asphalt
F	force	ER	Expansion Ratio
ρ	density	HL	Half Life
P	pressure	FI	Foam Index
Po	porosity	AB	Averaging Box
$W(x)$	kernel function	DPM	Double Planetary Mixer
h	kernel length	BPDPM	Baker Perkins Dual Planetary Mixer
c	speed of sound	MPI	Message Passing Interface
μ	dynamic viscosity	GPU	Graphics Processing Unit
ν	kinematic viscosity	PCA	Principal Component Analysis
η	coefficient of variance		
ξ	mixing index		

Acknowledgments

I am sincerely and heartily grateful to my advisor, Professor Huiming Yin, for the valuable support he gave to me throughout my entire Ph.D. study. It was quite a change in my life when Professor Yin offered me the precious opportunity four years ago to study in his group. I am sure my current achievement would have never been made without his help. Besides, he not only always encouraged me, enlightened me, and provided endless advice and insight into my research and my career, but also set me an example as a man with self-motivation and discipline. It is my great pleasure in my life to be his student at Columbia University.

I am also grateful to the members of my dissertation committee, Professor WaiChing Sun, Professor Haim Waisman of Columbia University and Professor Anthony Rosato at New Jersey Institute of Technology and Dr. Jim Lua at Global Engineering & Materials, INC., for taking their valuable time to participate my defense and to read and comment on this dissertation.

Over the past four years at Columbia University, I have received support from a great number of individuals. Dr. Lingqi Yang gave me the first course on my research topic, and Dr. Fangliang Chen gave me a lot of guidance on how to do research. I am obliged to my colleagues Dr. Liang Wang, Dr. Gan Song, Dr. Chensen Lin, Dr. Qi Tong, Dr. Xin He, Dr. Xiaokong Yu, Mr Qiliang Lin, Mr. Zhenyu Shou, Mr XiangTeng Wang who helped me in both my research and daily life.

I would like to thank my parents, my father Tongli Zhu and my mother Likai Zhao, for their unyielding support. Your endless love to me is the momentum that keeps me moving forward. I also owe sincere and earnest thankfulness to my girlfriend, Jiamin Chen. You always stand with me, back me up, and drive me move forward bravely.

I would like to thank everyone who once helped me in my life.

This work has been sponsored by the National Science Foundation CMMI 1301160 and AFOSR-FA9550-14-C-0058, whose support is gratefully acknowledged. The computational resource from Extreme Science and Engineering Discovery Environment is greatly appreciated

Chapter 1

Introduction

Particle dynamics simulation has attracted significant attention in academia and industry because of its board applications to problems in different industries and scales, from Molecular Dynamics (MD) to large scale astrophysical problems. Despite differences in scales, particle dynamics simulations share certain common features: modeling materials as discrete particles, computing forces between these discrete particles based on physical laws, and updating status of the particles based on the Newtons Law. The focus of this work is particle dynamics simulation in mesoscale in civil engineering, such as particle mixing process and foaming process. The challenge of these problems is how to build the model at the first place, how to simulate both solid and liquid phases in one model, and how to analyze the results of the simulation and compare them with experiments.

1.1 Particulate Mixing Process

The mixing process of powders and granular materials has been the subject of substantial research, driven by the broad industrial applications including pharmaceuticals, food, ceramics, catalysts, metals, and polymer manufacturing. The annual cost of inefficient industrial mixing in the US has been estimated to be as high as 10 billion US\$ with roughly 60% of blended products typically being powder or granular materials [87]. Thus, insightful granular material research and efficient particulate material mixing design would be valuable to both academy and industry.

While considerable progress in understanding the dynamics of dry granular materials has been made in recent years, the physical mechanisms underlying the properties of wet systems or in

viscous fluids remain largely unclear. The addition of liquid will change the original interaction model for dry particles, and lead to aggregate among particle mixtures. Machines and processes that mix, transform and transport the particles within a granular flow are difficult to design because the particle dynamics process is not completely understood. Therefore, understanding mixing mechanisms and identifying critical process and material parameters in slurry flows are crucial to the design and manufacturing of the multiphase materials.

Mixing of granular materials can be performed using a wide variety of different devices such as rectangular blades [12], rotating drum [49], plowshare mixer[5], paddle mixer[54], helical ribbon agitator [66], Bohle Tote blender [9] and so on. These mixers are efficient when mixing pure solid particles or low viscous mixtures. However, when the viscosity of the mixture is high, the forces required for the blades in a mixer to rotate at a high speed increase tremendously, and most of these blades are fixed so that there may be some part of the mixture the blade could not reach. For particle mixture with high viscosity, a fairly common used mixer is the Double Planetary Mixer (DPM) [148; 132], which has been used in many applications such as adhesive, food and cosmetics industry. The advantage of DPM is that it contains two rotating axes that enable the blades to touch almost every corner of the mixer vessel. On the other hand, the blades in DPM are slim and smooth, thus the resistance force between blades and particles are smaller compared to other blades so it is easy for the blades of DPM to rotate at a high speed. Previous studies of DPM focus on experiments [148; 132], and in this work a numerical model for DPM is developed, which is then applied to particle mixing simulation.

Many experimental approaches have been employed in quality control of the mixing process, such as sampling, visual tracking, magnetic resonance imaging, rheometer measurements among others. Although these methods provide valuable insights of the mixing processes, their applications are limited due to the low consistency and high labor costs [24; 30; 12]. Information that would otherwise be very difficult to extract from experiments can be readily obtained from simulations. Moreover, it is time-consuming to develop a new particulate composite through such a trial-error method in experiments. Therefore, numerical simulation becomes an alternative to investigate granular flow dynamics. Virtual experiments can be conducted in computers to save the time and cost in the development of new materials.

1.2 Particulate Simulation Method

Among the mixing problems, liquid-solid mixing becomes more and more common, in which the liquid is used to accelerate the mixing process or function as a binding agent for the final product. The physics involved in liquid-solid mixing is also common in other related topics such suspension [13; 74], fluid bed [84; 116; 80] and complex granular flow [50; 85]. Therefore, it is in great demand to develop a reliable model to simulate the solid-liquid mixing process. Granular material mixing simulation is a highly multi-disciplinary field such as granular mechanics, solid-gas flows, solid-liquid flows, solid-solid flows, gas-liquid flow, mixing and segregation, kinetic theory, rheology, parallel computing and so on. Different approaches have been proposed to tackle the liquid-solid granular flow problems, which could be classified mainly to three categories: continuous media models, discrete particles models, and the combination of both models[9; 24; 12].

Continuous media models treat both solid and liquid phases as interpenetrating continua. This scheme works well for problems with small deformation, but often fails in problems with large deformation, such as mixing problems. In comparison, discrete particle methods can simulate the dynamics of particulate mixtures and are capable of dealing with moving boundary problems in the mixing process. Examples of discrete particle methods are Molecular Dynamic (MD) [59; 18; 52; 11], Discrete Element Method (DEM) [5; 25; 9], Smooth Particle Hydrodynamics (SPH) [119; 78; 76], and Dissipative Particle Dynamics (DPD) [146; 142; 48; 58]. In these methods, some specific treatments are generally required to evaluate the coupling between solid and liquid particles [128; 124; 28].

Other methods combine continuous media and discrete particles, such as CFD coupling DEM [13; 56], where CFD is implemented to simulate the fluid phase as a continuum medium and DEM is applied to simulate solid particles as meshless discrete particles. The coupled CFD-DEM generally requires the meshed elements in CFD to be smaller than the DEM particles, which restricts the number of DEM particles and makes this method only suitable for dilute granular flows. For mixing problems where the solid load is high, but not high enough to apply liquid bridge models [123; 95; 77], the coupled CFD-DEM approach can be time consuming in modeling the moving boundary and discretizing the complex geometry, and SPH provides a practical method to simulate the liquid phase among a large number of solid particles with only a few number of particles.

To tackle mixing problems with intermediate to high solid loads, SPH-DEM coupling is a

promising approach. Cleary [28] proposed a one-way coupling SPH-DEM model based on Darcy's law for prediction of slurry transport. Sun [129] introduced pressure difference and drag forces between liquid-solid interactions, and validated them with rotating drums filled with solid-liquid mixtures. Jonsen [64] employed the DEM stiffness to the coupling of SPH and DEM, and obtained expected torque in a rotation drum. Ricardo [17] applied the SPH momentum equation to the interactions between solid and liquid particles and demonstrated the feasibility by comparing the simulation with experiments in free surface solid-fluid flows. Robinson[112] introduced the local average Navier-Stokes (N-S) equation into coupling methods and validated the scheme with solid sedimentation in liquid. Robin's scheme was later improved and applied to a high-shear mixing problem by Kwon [70] and Tong [133].

In comparison to DEM particles added to SPH liquid, some preliminary work has been conducted towards SPH liquid flow in porous media. Shao [119] and Zhu [150] treated porous structures as boundaries for SPH particles, which are in smaller size compared to the porous structures. Jiang [63] proposed L-J potential between SPH and solid skeleton particles and demonstrated that the scheme can reproduce Kozeny's formula of permeability. Peng [102] presented resistance interactions between solid and liquid SPH particles based on Darcy's law and validated the method with a porous media case study. However, it is still challenging for the coupled SPH-DEM method to simulate particle mixing with a high solid load due to the difficulty of calculating the interaction force between SPH and DEM particles, since previous models are built under the assumption that solid particles are fully saturated by the liquid phase. In this work, DEM-SPH coupling is implemented to examine the solid-liquid mixing problems, in which the solid load can reach 90% in volume and the interaction force between liquid and solid particles significantly depends on the local porosity. To compute the local liquid density, solid particles are considered to be rigid without any overlap with each other. Rather than excluding solid particles by dividing porosity in the density computation [112], solid particles are included in density computation, to avoid potential instability at a high solid load. SPH momentum equation is employed for the interaction between solid and liquid particles, in which viscosity coefficient is varied according to its local porosity following Darcy's Law without any further artificial drag forces. This approach considers the nature of particle interactions in mesoscale, and appears to be more efficient and effective for computation. In this paper, the numerical algorithm and formulation of a two-way DEM-SPH coupling method are introduced

first. Three benchmark studies are subsequently presented to verify the model. Then the coupling method is applied to high shear mixing problems in different mixers, and a series of parametric studies are conducted to investigate the mixing process.

1.3 Particulate Mixing Simulation Acceleration

Generally speaking, granular flow simulation is very time-consuming and high performance computation resource is necessary. In addition to improving the algorithm itself, it is required to adopt an efficient neighbor particle list building strategy for parallel computing on supercomputers. CPUs using Message Passing Interface(MPI) or Open Multi-Processing(OpenMP) are commonly used to implement the parallel computing. With the emerging technology in the General-Purpose computing on Graphics Processing Units (GPGPU) these years, GPU parallelization provides a novel option for high-speed computation. GPUs were initially designed exclusively for three-dimensional image rendering that is not complex but highly computationally intensive. In 2007, NVIDIA released the CUDA programming platform, which is designed for general-purpose computing. Since then, GPUs have been successfully used in many fields of scientific computing.

In recent years, several studies have been conducted with GPU to accelerate the simulation of SPH [144; 100; 141; 40], DEM [147; 125; 103] and DPD (Dissipative Particle Dynamics) [99]. In these studies, the speedup effects range from 20 times [147], 35 times [143] to 40-60 times [40], which depend on the algorithm, as well as the CPU and GPU implemented. Various software packages are applied to GPU simulation such as DualSPHysics [31; 40], Lammmps [99], Trubal [147] and PetaPar [141]. In this work, all the simulations are conducted using the Parallel Dynamics Particle Simulator (PDPS) [146], an in-house software package developed by Pao Sustainable Engineering and Materials Laboratory (Pao Lab) at Columbia University. The software package has functions of both MPI parallel computing and GPU parallel computing. Its parallel algorithms are illustrated and the speedup results are demonstrated.

1.4 Foaming Asphalt

Replacing solid particles with air bubbles, we can simulate the physics in other applications of multiphase flows, such as asphalt foaming. This is a challenge different from the solid-liquid mixing

problems due to the thermomechanical behavior of bubble formation, motion and collapse.

Over 94% of U.S. roadways and 85% airport runways are surfaced with asphalt. During the paving process, a large amount of greenhouse gas and other emissions are generated in the form of blue smoke, which consists of tiny oil droplets resulted from asphalt or reclaimed asphalt pavement (RAP) materials at high temperatures. As a result, asphalt pavement construction projects are often limited to night construction in many urban areas. Warm mix asphalt (WMA) is a group of technologies that has been adopted in Europe and the United States in the past two decades to overcome the environmental and economic challenges of hot mix asphalt (HMA) [15; 21; 38]. These technologies aim at improved workability and reduced viscosity of the asphalt binder using organic or chemical additives or by introducing cool water into the heated molten asphalt under controlled temperature and pressure conditions [8; 61; 117]. This allows lower production and construction temperatures and less asphalt binder to be absorbed by the aggregates than those in traditional HMA alternatives [4]. Because lower temperatures are used at the plant and construction sites, less emissions and odors are produced. Depending on the production temperature, a range of 15-70% reduction of carbon dioxide and other emissions during production have been reported [35; 57]. It is generally accepted that the lower warm mix asphalt production temperatures reduce emissions of volatile organic compounds (VOC) at mix plants [1]. However, it has also been reported that the VOC might be increased resulted from the lower boiling point of compound in WMA due to the presence of steamed water [42; 53], but such an increase could be caused by the poor burner tuning rather than the WMA technology[1].

Recently, WMA produced by water injection has gained increasing popularity among asphalt producers since it involves a relatively inexpensive one-time plant modification and does not require the use of costly additives. According to the National Asphalt Pavement Association (NAPA), in 2012, 88% of WMA produced was through water-based foaming processes, in which cold water is injected into hot asphalt binder and turned into vapor and trapped in numerous tiny bubbles in the asphalt. Therefore, spontaneous foaming is induced, which greatly increases the volume of the asphalt [61]. The foaming processes significantly increase the volume of asphalt binder with large surface area in the unit volume. However, the foam may dissipate in a very short time, often less than one minute. Therefore, the mixing window of the WMA with the foamed asphalt is restricted within a short period while the asphalt binder is still in its foamed state.

When foamed asphalt mixes with aggregate, a strong coating with high shear strength of the mix can be generated. In addition, since foamed asphalt is flexible and has much larger volume, its workability is considerably improved. Therefore, the manufacturing of WMA can be operated at a lower temperature within a shorter time and the optimum asphalt content of a foamed asphalt mix can be lower than that for the non-foamed one [118]. Because of this, the WMA generated by water-based foaming asphalt has various advantages over traditional WMA or HMA, such as reduced binder cost, shorter fabrication time, lower energy consumption, more environmental benignancy, and broader applicability.

The characteristics of foamed asphalt have significant influence on the mixture workability and performance. To effectively use asphalt foaming process to produce WMA, it is very important to understand the characteristics of the foaming asphalt and their effects on the quality of the final WMA product. However, because of the complexity of the foaming process and the difficulty to observe the foaming process in the function chamber, current practices and understandings were mainly based on various empirical methods with many trial-and-error experiments. Real experiments are both expensive and time-consuming to measure and control these environmental parameters; whereas a high fidelity numerical model enables virtual experiments of WMA production and optimization of the asphalt foaming process and the WMA design. However, no such numerical models of the asphalt foaming processes have been found in the literature yet. To advance the fundamental understanding of the asphalt foaming process, we use the Smoothed Particle Hydrodynamics (SPH) to simulate the bubble motion and material microstructure evolution in a heterogeneous material system.

1.5 Scope

The primary objective of this study is to investigate and develop particle dynamics simulation methods, especially multiphase particulate mixing process and foaming process. The materials to be studied include both solid and liquid, or liquid and gas.

The remainder of this dissertation is organized as follows.

Chapter 2 provides the problem statement, in which the specific problems and the challenges

are introduced, including multiphase particulate mixing process and foaming asphalt simulation.

Chapter 3 introduces the basic algorithms and numerical modeling methods to address problems mentioned in Chapter 2. Smoothed Particle Hydrodynamics (SPH) is implemented to simulate the dynamics of liquid and Discrete Element Method (DEM) is implemented to simulate the dynamics of solid particles. After these two fundamental numerical methods are introduced, a novel coupling scheme which applies Darcy's Law to compute the interaction force between liquid and solid particles is proposed. Three kinds of mixing indexes are introduced as homogeneity index, randomness index and time series index. Homogeneity index is used as the major index to test the quality of the mixing process. Then the proposed SPH coupling DEM model is verified by three case studies: 1. one solid particle sedimentation in liquid; 2. one solid block sedimentation in liquid; 3. liquid flow through porous media. Simulation results of these three case studies are compared with theoretical study and several parameters in the SPH-DEM model are determined so that the simulation results can match experiments well.

Chapter 4 investigates the asphalt foaming problem, which is a liquid-gas two phase flow problem. The background and characteristics of asphalt foaming processes are illustrated first, and a SPH model is applied to simulate both asphalt (liquid) and water vapor (gas). To simplify the problem, both liquid and gas phases are treated as SPH particles in the numerical model, and gas evaporation rate is introduced from experimental study to explain the bubble collapse process. Simulations with different water contents, pressure and temperature are conducted and the numerical predictions are consistent with experimental results.

Chapter 5 to 7 address the issues in the coupled SPH-DEM model. The homogeneity of the mixture is the major quality control factor, and the mixing process is mainly analyzed in three aspects: 1. analysis of simulation results on various mixing indexes; 2. the effect of liquid on mixing quality, and; 3. application to different mixers and comparison with experiments. In Chapter 5, various methods to test mixing quality are reviewed and homogeneity of mixing process is explored. Chapter 6 investigates the effect of liquid content and liquid viscosity on the final mixing quality. Chapter 7 compares the mixing processes for three types of mixers, i.e. four-bladed mixer, Ross double planetary mixer (RDPM) and Baker Perkins dual planetary mixer (BPDPM). The simulation results of RDPM are compared with the experimental results and good agreement is obtained.

For particle system simulation problem, computational capacity is always a threshold to improve the accuracy of the problem. This problem is discussed in Appendices. All numerical studies in this work are conducted on our in-house software package Particle Dynamics Parallel Simulator (PDPS). PDPS is accelerated in two approaches, MPI parallel computing and GPU acceleration separately. Speedup results for both MPI parallel computing and GPU computing are illustrated with case studies. After computational acceleration, some model reduction is introduced, to simplify the similar computation in multiple case studies.

Finally, Chapter 8 summarizes the conclusions in this work and provides insights for future work.

Chapter 2

Problem Statement

2.1 Multiphase Flow Problems

Multiphase flows are common in various natural and engineering problems, which is used to refer any fluid flow consisting of more than one phase or component. Multiphase flow occurs in many operations in the chemical, petroleum, and power generation industries [44] as well as natural phenomena such as fog, avalanches, mud slides, sediment transport, debris flows [16]. Depending on the phases involved, multiphase flow could be divided to several types. Solid-liquid two-phase flow problem is usually referred as granular flow, and it is common in a broad range of industrial processes, including mineral processing, plastics manufacturing, ceramic component, pharmaceutical tablets and food products [140; 25]. Liquid-gas two-phase flow problem occurs when evaporation appears within the liquid phase, and also occurs in capillary gas-liquid two-phase flow in increasingly more modern industrial applications, such as compact heat exchangers, microelectronic cooling systems, research nuclear reactors, chemical processing, small-sized refrigeration systems [135]. Gas-solid flows are important to many industrial process such as pulverized coal combustors and fluidized beds [46]. Solid-liquid-gas three-phase problems are also common if air bubble is considered in granular flows, and appropriate models are to be developed to simulate the physics more accurately involved in these multiphase flow problems.

In this work, solid-liquid two-phase flow and gas-liquid two-phase flow are mainly investigated through two specific problems in the following sections.

2.2 Asphalt Foaming Process

2.2.1 Introduction

As mentioned before, foaming asphalt simulation is another topic studied in this work. Water-based foaming asphalt is investigated in this work, in which water is injected into asphalt as an additive to create foamed asphalt. The foaming performance of the foamed asphalt is characterized and the effects of the operating/environmental parameters on the foaming performance are predicted.

In this study, the foaming performance of the water-based foaming asphalt was first evaluated by a self-developed laboratory scale foamer at different water contents. An SPH model was then developed to characterize the bubble motion and the microstructure evolution of the foamed asphalt. The numerical model was validated with the experimental results. Parametric studies were further conducted to investigate the effects of the controlling parameters on the performance of the asphalt foaming process.

2.2.2 Experiment

A self-developed nozzle-based foamer as shown in Figure. 2.1 was developed in the Pao Sustainable Engineering and Materials Laboratory (Pao Lab), to generate the foamed asphalt. This foamer is able to accurately control the binder temperature, injected air pressure, water flowing rate, and expansion chamber temperature. It has an air injection capacity from 0 to 80 psi, which is controlled by the air flow barometer. The water flow rate is controlled by the flow meter combined with the foaming barometer. Under a constant air pressure of 80 psi in the water barometer, the water flow capacity ranges from 0 to 45 ml/min by adjusting the flow switch. The binder container has a capacity of 1 gallon, whose temperature is governed by the asphalt heater band surrounding on the binder container. Upon its full load, the heater is able to sustain a fully filled tank of binder up to 240°C. The binder container is connected with the function chamber via an adjustable valve. The flowing rate of binder from its container to the function chamber is controlled by both the asphalt flow barometer and the adjustable valve. At a fixed position of the valve, the larger the air pressure is applied in the binder container, the higher flow rate of the binder to the function chamber will be obtained. The inject air inlet and water nozzle are located at the two opposite sides of the function chamber. The position of the air inlet is slightly higher than that of the nozzle such that the water

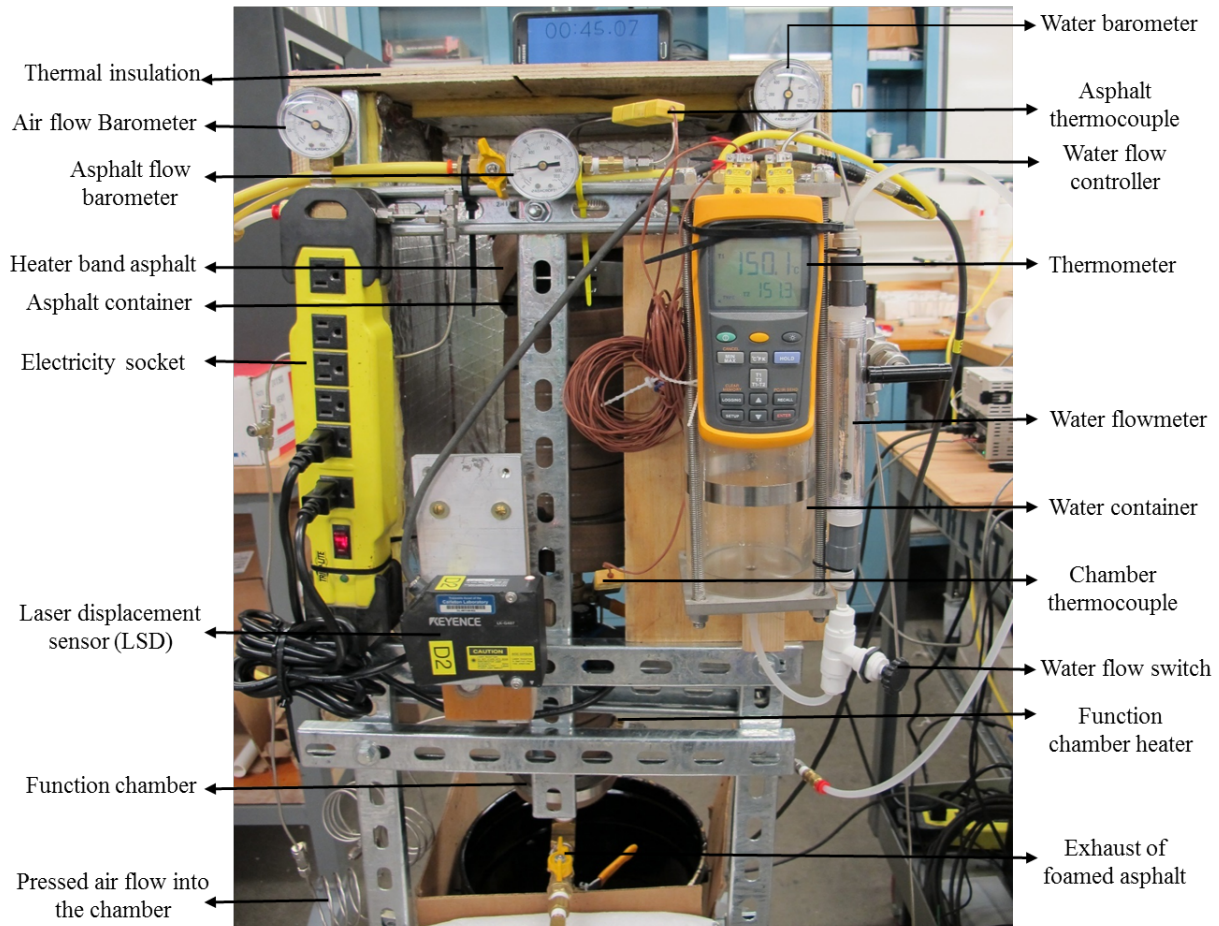


Figure 2.1: Laboratory configuration of the self-developed foamer

spray injected by the nozzle will react with the high temperature binder a little longer in order to generate more bubbles in the binder.

The self-built foamer can accurately control the binder temperature, injected air pressure, water flowing rate, and expansion chamber temperature. The flow rate system configured in this setup can control the flow rate of the injected water down to 2 ml/min with an accuracy of 1%. A high-speed long range Laser Displacement Sensor (LDS) which is shown in Figure. 2.2 (a) is used to characterize asphalt foaming properties, with which the expansion ratio, half-life and foam index can be obtained. The LDS sends a pulse of laser light to a target surface and detects the reflection with a high accuracy of 0.02%. The LDS is linked to the LabVIEW software and

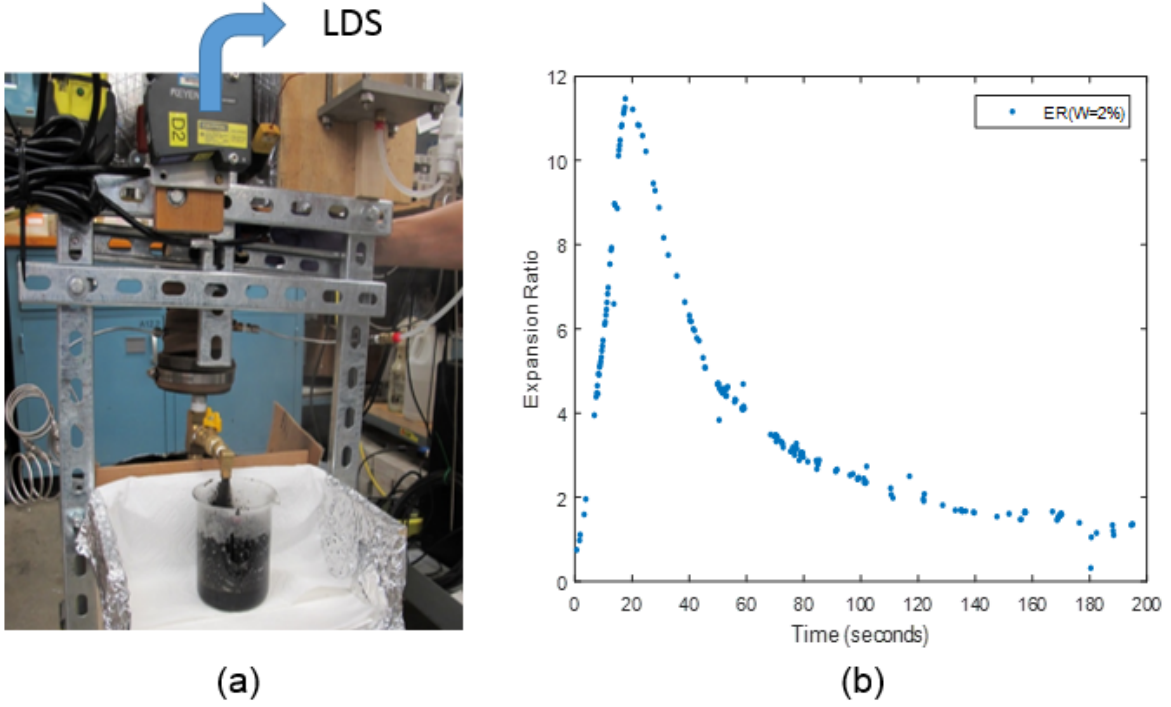


Figure 2.2: LDS for characterization of foaming evolution history

configured with a 5Hz sampling frequency, which makes the LDS is capable to capture detailed information of the foaming evolution. The developed LDS is integrated into the foaming machine and designed to automatically measure the foaming evolution from its generation, expansion, and collapse throughout the entire foaming process. It avoids any operation errors and thus provides objective, automatic, high accurate and repeatable measurements. The evolution of the foamed asphalt captured by the LDS is shown in Figure. 2.2 (b). It shows that this foaming machine is able to automatically capture the detailed information of the entire foaming evolution from its generation, expansion, to collapse.

2.2.3 Challenges and Simulation Plans

As far as the author knows, simulation on asphalt foaming has hardly been studied before. Therefore, there are a lot of challenges to overcome. Some major challenges are listed below:

1. What kind of model should be used. Very few work has been done regarding asphalt foaming simulation so there are not too much literature to refer.

2. How to model phase transition. During the asphalt foaming process, liquid water is injected into asphalt binder. Due to the high temperature of asphalt, liquid water is heated up and transferred to water vapor. Phase transition is a difficult topic for numerical simulation, and it is a challenge to simulate phase transition of water in asphalt foaming process.
3. How to determine the coefficients used in the simulation. Once a numerical model is built, it typically contains several coefficients, which is not related to the property of the materials to be simulated. As no previous work has been done, it is not easy to determine appropriate values for those coefficients used in the simulation.

According to these challenges, corresponding solutions are listed here:

1. Smoothed Particle Hydrodynamics (SPH) is applied to model this problem. Since during the foaming process, there is large volume change and boundary change, and meshless particle methods are more suitable. Depending on the scale of the problem, SPH is the most appropriate one to simulate the liquid in this problem.
2. Since phase transition is indeed a difficult problem, it is not simulated in this work. It is assumed the phase transition is already finished and water is in the form of vapor at the beginning. At the beginning, foaming is achieved and the initial volume of foamed asphalt is the largest. Then the bubble collapse process is simulated, when the volume of foamed asphalt keeps reducing until it reaches a constant value.
3. In this work, due to the lack of relevant information, an empirical model is developed. When the numerical model is built, the first step is to choose the coefficients so that the results of the simulation match the results of the experiment. Then the coefficients are used to produce simulations with different controlling parameters, and some prediction results could be made.

The details of the implementation of the numerical models for asphalt foaming process are explained in Chapter. 4.

2.3 Multiphase High Shear Mixing Problems

2.3.1 Introduction

As mentioned in the introduction, multiphase high shear mixing problem is the main focus of this work. Experiments and simulations will be conducted to investigate the mixing process. During the mixing process, several different types of particulate materials as well as the liquid are placed into a sealed mixer and then mixed. Fig. 2.3 shows an initial state of an simulation example. In this example, two kinds of solid particles that are plotted in red and yellow are mixed with one kind of liquid particles in blue. During the mixing process, the blades of the mixer rotate at a constant speed and all the materials inside the mixer are mixed towards homogeneity.

2.3.2 Experiment

The mixing process could be conducted in various kinds of mixers, and the one used in experiment of this work is Double Planetary Mixer (DPM). DPM is a mixer with two groups of rotating blades around the center, and each group of blades contains two separate blades and rotates around its sub axis, as shown in Fig. 2.4. The advantage of DPM is that its blades can reach larger areas within the container than other mixers, and almost every corner inside the container could be reached during mixing, as illustrated in Fig. 2.5. Following is one example of a mixing experiment conducted in this work.

The mixing experiment is conducted under vacuum of $< 10\text{torr}$ at a temperature of 60°C , and a stabilized rotation speed of 75 rpm. The mixing ingredients include the solid particles, powder sugar and table sugar, and the liquid PDMS. Glass beads (5 wt.% 10 wt.% of the overall solid load) of three different colors were added into the mixture to assess the mixing quality. Average sizes of these particles are: 7-28 μm for Aluminum H12, 25 μm for powder sugar, 250 μm for table sugar, and 1000 μm for glass beads. Density values of the ingredients in the mix design are: $2.7\text{g}/\text{cm}^3$ for Aluminum H12, $1.59\text{g}/\text{cm}^3$ for powder sugar and table sugar, $1.55\text{g}/\text{cm}^3$ for glass beads, $1.0635\text{g}/\text{cm}^3$ for PDMS. The material properties are listed in Table. 2.1

During the mixing process, the base and curing agent of PDMS were premixed for 1.5 minutes before the powder sugar was added and mixed for 3 minutes, followed by adding table sugar and glass beads and mixing for 1 minute. After everything is mixed, the final state of the mixture is

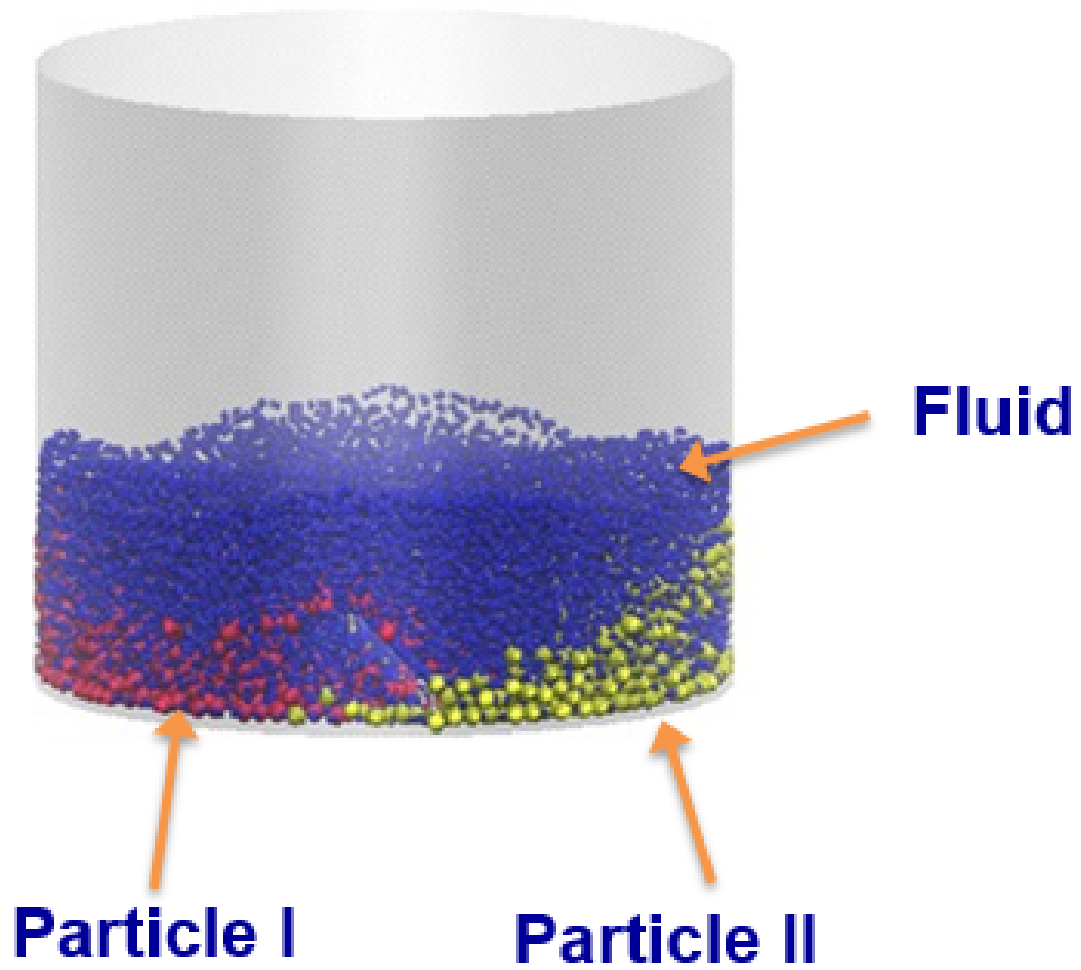


Figure 2.3: An example of multiphase high shear mixing simulation

Table 2.1: DPM experiment with 84% solid load

Materials	Value
Powder sugar	148.7 g
Table sugar	475.7 g
Glass beads	32.9 g
PDMS	125.2g



Figure 2.4: Double Planetary Mixer

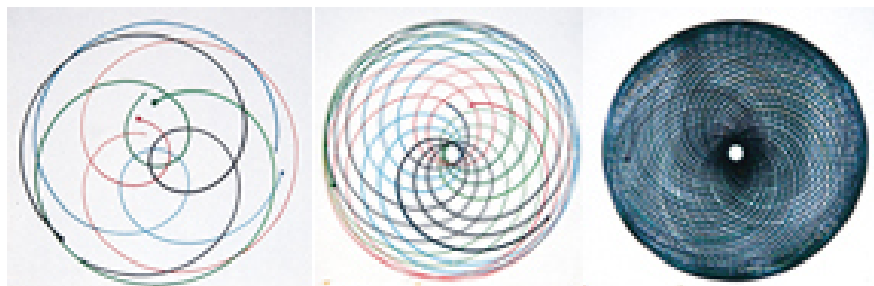


Figure 2.5: Orbital blade motion of the blades in Double Planetary Mixer



Figure 2.6: Final State of a Mixing Experiment Using a Double Planetary Mixer

shown in Fig. 2.6.

2.3.3 Challenges and Simulation Plans

Simulating a mixing process is not easy, and some major challenges are listed below:

1. How to choose the correct numerical model. This is always the first and the most important step before further efforts are made. Choosing the correct model at the beginning would save a lot of time and provide valuable insights to the mixing problem.
2. How to model the interaction forces between solid and liquid. The multiphase mixing process studied in this work contains multiple phases, and different models are used to simulate the dynamics of these various phases. It is a challenge to combine multiple numerical models together and simulate the interaction forces between solid and liquid.

3. How to simulate the rotating blades in mixers. As shown in the figures above, the shape of DPM mixer used in this work is complicated, and it is hard to detect whether a particle has been contacted with the mixer or not.
4. How to compare the simulations with real experiments. Due to computational capacity, there is always some simplifications in the simulation model compared to the real world experiments. It is important to match the time scale and space scale between simulation and experiment, and only when they match, the results obtained in simulation would be useful in the real world.
5. How to accelerate the computational speed in simulation. Typically mixing simulation is very time consuming, since there are so many particles in the dynamics system. There are various ways to accelerate the computation, and fast computational speed would enhance the fundamental study of the mixing process.

According to these challenges, corresponding solutions are listed here:

1. Discrete Element Method (DEM) and Smoothed Particle Hydrodynamics (SPH) are chosen to simulate solid and liquid separately in this work. DEM is widely used for particulate mixing process, and SPH is a meso-scale numerical model for liquid. For the model selection, choosing the correct liquid model is the key. Since the solid phase is simulated as particles, it would be easier to simulate liquid as particles too. Considering the scale of the mixing problem is meso-scale, in which particles could be observed, then particle methods such as Dissipative Particle Dynamics (DPD) that describe Brownian motion in micro-scale is not suitable for this problem, and SPH is an ideal choice.
2. For the interaction of liquid and solid in the simulation, depending on the solid load, various models have been proposed. In previous literatures, DEM-SPH coupling scheme has been applied to liquid-solid particle mixing problems, but only for problems where solid load is low (lower than 50% so that solid particles are fully saturated by liquid). In this work, mixing problems with high solid load (80% to 90%) are studied, and the assumption that solid particles are fully saturated by liquid does not hold anymore. A novel coupling scheme is proposed to solve this issue in this work.

3. When the configuration of rotating blades is simple, whether a particle contacts with the boundary could be detected by computing the distance of the particle to the blades. When the configuration of the blades are complicated, computing the distance between a particle to the blades would be very difficult. Rather than treating the blades as boundary, a better idea is to treat the blades as particles. Each blade is composed of hundreds of fixed particles, and the contact between particles and blades could be transferred to the contact between particles to particles.
4. Time scale in the simulation is matched with the real physics, and the total mixing time in the experiment is the same as the total mixing time in the simulation. For the space scale, because of computational capacity, one particle in the simulation is not equivalent to one particle in the real world. Instead, one particle in the simulation represents a cluster of particles in the real world, such that the results of the simulation could be compared with experiment in a coarse resolution.
5. Mixing simulation is time consuming, and parallel computation is applied to accelerate the computation. Message Passage Interface (MPI) parallel on CPU and GPU parallel computing are implemented separately to speed up the computation. The speedup effect is examined in this work.

The details of the implementation of the numerical models would be in Chapter. 5 to Chapter. 7.

Chapter 3

Algorithm and Formulation

3.1 Smoothed Particle Hydrodynamics (SPH)

The Smoothed Particle Hydrodynamics (SPH) has been widely used for simulating fluid flows, since it was first developed initially for astrophysical problems [45]. It was later used for simulating free surface flow [93] and other applications [88]. SPH is a mesh-free Lagrangian method, which uses particles to discretize continuous fluid field. The property of the fluid field could be smoothed from discrete particles using a kernel function. Even though it is a particle based method, continuous fields of the fluid could be obtained by the weight average of neighbor particles through a kernel function. SPH could be applied to computer visualization and animation [120; 108], as well as calculations on physical systems such as free surface flow [93], shock [90] and violent impact flows [81].

In SPH, each particle has its properties such as position, velocity, mass, pressure and energy. Pressure and temperature are functions of density and kinetic energy, respectively. During the flow process, the mass of each particle stays constant while the density keeps changing as the flow is assumed to be weakly compressible.

For any variable $f(x)$ at point x , SPH discretizes the domain to particles as follows:

$$f(x) = \int f(x')W(x - x', h)dx \quad (3.1)$$

where $W(x - x', h)$ is the kernel function with features of positive, monotonically decreasing with distance, sufficiently smooth and even [76]; h is the Kernel Length which determines the interaction

domain of the kernel function. In this work, Cubic Spline kernel function [94] is implemented:

$$W(x - x', h) = W(r, h) = \alpha \begin{cases} 1 - 1.5 \times \frac{r^2}{h^2} + 0.75 \times \frac{r^3}{h^3} & \text{if } r < h \\ 0.25 \times (2 - \frac{r}{h})^3 & \text{if } h \leq r < 2h \\ 0 & \text{if } r \geq 2h \end{cases} \quad (3.2)$$

where α is $\frac{\pi}{h^3}$ for 3D simulation. The larger the value of h , the more neighboring particles will be counted [83]. Different values of h are chosen for specific problems of interest in order to find a balance between accuracy and efficiency. The density of each particle in SPH is calculated by mass and kernel functions as follows:

$$\rho_i = \sum m_j W_{ij} \quad (3.3)$$

where $W_{ij} = W(x_i - x_j)$. The governing equations of SPH are derived from the Navier-Stokes(N-S) equations.

$$\frac{d\rho}{dt} = -\rho \nabla \cdot \mathbf{v} \quad (3.4)$$

$$\frac{dv}{dt} = -\frac{1}{\rho} \nabla P + g + \nu \nabla^2 v \quad (3.5)$$

In practice, it is reasonable to drop the continuity equation, since each particle in SPH automatically satisfies the principle of mass conservation and is self-consistent [92].

Besides the N-S equations, an important assumption in SPH is that pressure is a function of density through equation of state. Among various equations of state, the Tait equation of state is widely used in fluid mechanics[93; 76], which is expressed as:

$$P(\rho) = \frac{c_0^2 \rho_0}{7} \left[\left(\frac{\rho}{\rho_0} \right)^7 - 1 \right] \quad (3.6)$$

The Tait equation has a parameter of sound speed c_0 . It is usually chosen lower than the physical value, depending on the specific problem size to make the system more stable and yet maintain a low Mach number (lower than 0.1) in an incompressible flow [122; 81].

For any property f of particle i , it could be written as [91]:

$$f_i = \sum_j \frac{f_j m_j}{\rho_j} W_{ij} \quad (3.7)$$

The benefit of using kernel function in the equation is to transfer the spatial derivative of variable to the derivative of the kernel function, and based on the equation above the derivative of f_i is calculated as:

$$\nabla_i f_i = \sum_j \frac{f_j m_j}{\rho_j} \nabla_i W_{ij} \quad (3.8)$$

To discretize the pressure gradient term, a conversion is made first[91]:

$$\frac{\nabla P}{\rho} = \nabla\left(\frac{P}{\rho}\right) + \frac{P}{\rho^2} \nabla \rho \quad (3.9)$$

Then the derivative of pressure could be written as:

$$\nabla_i \frac{P}{\rho} + \frac{P_i}{\rho_i^2} \nabla \rho = \sum_j m_j \frac{P_j}{\rho_j^2} \nabla_i W_{ij} + \frac{P_i}{\rho_i^2} \sum_j m_j \nabla_i W_{ij} = \sum_j m_j \left(\frac{P_j}{\rho_j^2} + \frac{P_i}{\rho_i^2} \right) \nabla_i W_{ij} \quad (3.10)$$

Based on the equation above, the commonly used momentum equation for the interaction of particle i and particle j is applied

$$m_i \frac{dv_i}{dt} = -\sum m_i m_j \left(\frac{P_i}{\rho_i^2} + \frac{P_j}{\rho_j^2} + \Pi_{ij} \right) \nabla_i W_{ij} + F_b \quad (3.11)$$

where

$$\Pi_{ij} = -\frac{\alpha h c_{ij}}{\rho_{ij}} \frac{v_{ij} x_{ij}}{x_{ij}^2 + 0.01 h^2} \quad (3.12)$$

is the artificial viscosity term [90]; c_{ij} is the average sound speed of the two particles i and j ; α is a coefficient parameter which is usually chosen from 0.01 to 0.1 for water [34]; F_b is body force on liquid, such as gravity. x_{ij} is the position differences between particles i and j and v_{ij} is the velocity difference between particles i and j . In this work, rather than artificial viscosity, laminar viscosity[79] is employed as:

$$F_\mu = \sum m_i m_j \left(\frac{4\nu_0 x_{ij} \cdot \nabla_i W_{ij}}{(\rho_i + \rho_j)(x_{ij}^2 + 0.01 h^2)} \right) \quad (3.13)$$

where h is the kernel length mentioned before. The term $0.01 h^2$ is usually used to assure that the dividend is larger than 0 [34].

Artificial viscosity is implemented to make the numerical computation stable when the liquid's viscosity is very low such as water. When the liquid's viscosity is high as in this work, laminar viscosity will be more appropriate to reflect the real viscosity.

Note that by employing different numerical treatments, other forms of SPH equations can be derived for the same partial differential equations [76].

Due to the approximation using discrete particles to represent a continuous flow, the modeling parameters in a particle method will not only depend on the fundamental material properties, but also change with particle size, modeling scale, and neighborhood cutoff-distance, among other factors. To obtain repeatable and reliable results, a procedure for the calibration and verification of the modeling parameters should be conducted before using the method in new problems. Section. 3.7 will demonstrate the procedure to select appropriate parameters based on the existing theoretical results.

3.2 Discrete Element Method (DEM)

The Discrete Element Method (DEM) is a numerical method to simulate the motion and interaction of granular materials as discrete particles. The major physical law is momentum conservation, i.e. Newton's second law, and some DEM methods also consider the rotation of particles so that the torque is taken into account in the governing equations. In conclusion, the governing equations of DEM could be written as:

$$\frac{d\mathbf{r}^I}{dt} = \mathbf{V}^I, \quad m^I \frac{d\mathbf{V}^I}{dt} = \mathbf{F}^I, \quad I^I \frac{d\boldsymbol{\omega}^I}{dt} = \mathbf{T}_r^I \quad (3.14)$$

where I^I , $\boldsymbol{\omega}^I$ and \mathbf{T}_r^I stand for momentum of inertia, rotation speed, and torque of particle with index I . Generally, DEM models could be classified to two types: hard-sphere and soft-sphere models. For hard-sphere model, movements of particles are determined by momentum conservation. Only one collision is permitted at one time and it happens instantaneous; forces between particles are normally not explicit, and the hard-sphere model is mainly used in rapid granular flows [149]. Soft-sphere model is easier to implement, in which DEM particles share a similar algorithm as MD simulation, based on Newton's equation of motion. A small overlap (or deformation) of particles is allowed to calculate the elastic, plastic and frictional forces between them. Therefore, the soft-sphere model has been widely used.

Among the existing soft-sphere models, there are various approaches proposed to solve the relation between particle overlap and interaction forces. The most common method is linear spring-dashpot (LSD) model, in which the interaction between two particles is expressed by a normal spring and dashpot, a tangential spring and dashpot, and torque[33]. It could be further simplified by only considering the spring force based on the relative velocity between particles to simulate the dashpot force[37]. Other contact models for DEM simulations have also been proposed, such as Hertz [73], Mindlin and Deresiewicz theories [139]. However, LSD model is the most widely used one in DEM simulations because of its simplicity and computational efficiency compared to those non-linear models. The selection of force parameters in LSD DEM formulation is crucial to the success of simulation results.

In LSD, there are two types of forces: normal (\mathbf{F}_n^{IJ}) and tangential forces (\mathbf{F}_t^{IJ}), which can be decomposed to spring and dashpot (dissipative force) respectively as follows

$$\mathbf{F}_n^{IJ} = k_n \Delta r_n^{IJ} \mathbf{n}^{IJ} - C_n \mathbf{V}_n^{IJ} \quad (3.15)$$

$$\mathbf{F}_t^{IJ} = \begin{cases} -k_t \Delta r_t^{IJ} \mathbf{t}^{IJ} - C_t \mathbf{V}_t^{IJ} & \text{if } |\mathbf{F}_t^{IJ}| \leq \mu |\mathbf{F}_n^{IJ}| \\ -\mu |\mathbf{F}_n^{IJ}| \mathbf{t}^{IJ} & \text{if } |\mathbf{F}_t^{IJ}| > \mu |\mathbf{F}_n^{IJ}| \end{cases} \quad (3.16)$$

Where $\Delta r_n^{IJ} = |\Delta \mathbf{r}_n^{IJ}|$ and $\Delta r_t^{IJ} = |\Delta \mathbf{r}_t^{IJ}|$ are normal and tangential displacements, respectively. k_n , C_n , and k_t , C_t are spring stiffness and dashpot damping coefficients along normal and tangential directions, respectively, and μ is friction coefficient.

To reflect the physics correctly, it is important to use appropriate DEM parameters, and there are various approaches to predict these parameters analytically [98]. The normal damping coefficient C_n can be determined analytically by normal spring stiffness k_n and restitution coefficient e , as well as the friction coefficient μ , which are normally the inputs to DEM simulation [121; 136].

$$C_n = 2\sqrt{m_{\text{eff}}k_n} \frac{\ln e_n}{\sqrt{\ln^2 e_n + \pi^2}} \quad (3.17)$$

Where $m_{\text{eff}} = m^I m^J / (m^I + m^J)$ is the effective mass of particle I and J , and $e_n = 1$ means pure elastic collision, while $e_n = 0$ is for perfectly inelastic collision.

Several existing works have been proposed to determine normal spring stiffness by matching the maximum strain energy [71], maximum normal overlap [98], and dimensionless contact duration [6]

to the non-linear models. And μ and e_n could be determined by directly applying their physical property.

The tangential spring stiffness k_t and damping coefficient C_t can also be determined as follows [121]:

$$k_t = \frac{2}{7}k_n, \quad C_t = 0.5C_n \quad (3.18)$$

Even though numerous models are proposed, sometimes DEM simulation may still significantly deviate from experimental results since the assumptions made in numerical models may be inaccurate. Therefore these parameters are also determined through comparing numerical results with classic contact mechanics models (Hertzian, JKR, etc.), or calibrated by comparing with experiments. For example, in many cases, the normal spring stiffness k_n , restitution coefficient e , and the friction coefficient μ are directly calibrated by the experimental results.

3.3 Coupling between SPH and DEM

When liquid and solid particles coexist and interact with each other, their coupling force should be considered. In this work a two-way SPH-DEM coupling method is introduced. In contrast to one-way coupling where only one kind of particles exerts force on the other, two-way coupling means that both SPH particles and DEM particles exert forces on each other. Prior to computing the coupling force, the density of SPH particles needs to be revised, otherwise overlap between DEM and SPH particles could occur. To solve this overlap issue, one can introduce the concept of porosity, and then the volume fraction of liquid and solid phase are specified [112]. For an arbitrary particle i , its porosity Po is computed as below:

$$Po_i = \frac{\Sigma_{liquid}V}{\Sigma_{all}V} \quad (3.19)$$

Then density of the SPH liquid particle is calculated by weighted sum of its neighbor particles divided by its porosity:

$$\rho_i = \frac{\Sigma m_j W_j(x_i)}{Po_i} \quad (3.20)$$

In this way, when a DEM particle enters the neighborhood of a SPH particle, the porosity of this SPH particle decreases, and its density increased, generating repulsive forces between its original neighbor SPH or DEM particles. When these neighbor particles reach balance again, the number of SPH or DEM particles in the SPH particles region is decreased, thus making room for the entering DEM particle and no particles are overlapped during this process.

However, at mixing cases where solid load is as high as 80% to 90%, the porosity of SPH particles is inevitably low, which makes the computation of liquid density sensitive to the changes of its position and easy to get unstable. To solve this problem, we propose an alternative way to compute the SPH particles density as below:

$$\rho_i = \Sigma_{liquid} m_j W_j(x_i) + \Sigma_{solid} m_j W_j(x_i) \quad (3.21)$$

Where the mass of solid particles is included in the computation of SPH particles density because the solid particles play a dominant role. Note that the mass of solid particles in the above equation is not exactly the solid mass, but the mass of liquid in the same amount of the solids volume. In other words, solid particles are treated as liquid while computing SPH particles density. In this approach, no overlap between solid and liquid particles would appear, and the scheme is still stable for even high solid loads, because there is no large fluctuations of SPH particle's densities during the simulation process.

As for the coupling forces between DEM and SPH particles, one common approach is to apply buoyancy or pressure gradient forces and drag forces on DEM particles, and then apply the counteractive forces on SPH particles to attain two-way coupling. This approach is based on the assumption that solid particles "float" in liquid, in which liquid take the majority of the space. However, when the solid load is high, the assumption that solid particles are fully saturated by liquid particles does not hold anymore. In this work, a more fundamental formula is derived and the coupling forces are still comprised of two parts, i.e. pressure gradient force and viscous force. For the pressure gradient force, the formula is the same as the SPH pressure gradient force, i.e. treating part of the solid-liquid interaction as SPH pressure gradient forces. Pressure gradient forces are mainly to make SPH and DEM particles stable and do not overlap, while the essential coupling mechanism is the viscous force.

In flows among porous media, when the solid portion of the porous media is high, its macroscopic behavior can be described by Darcy's Law [104]:

$$v = \frac{k}{\mu} \nabla P \quad (3.22)$$

where v is the velocity of the flow, k is the permeability of the porous media, μ is the viscosity of the flow, ∇P is the pressure gradient. Darcy's Law is an equation that describes the flow of a fluid through a porous medium, which was formulated by Henry Darcy based on the results of experiments[36]. Since pressure gradient force is proportional to ∇P , at equilibrium state it could be seen that the viscous force F_d follows:

$$F_d \sim \frac{v\mu}{k} \quad (3.23)$$

In this scheme, since permeability k is a function of porosity, the viscous force between DEM and SPH particles is a function of porosity. When the liquid load is high enough in the simulation, the interactive viscous force between DEM and SPH particles will be close to the viscous force between the SPH particles. Depending on the porosity of the target particles, the coupling viscous force would be multiplied by a coefficient, which is a function of porosity, and satisfies the classical Darcy's Law [63; 150; 104]:

$$f_{ij}^d = \lambda f_{ij}^{d,SPH} \quad (3.24)$$

$$\lambda = 1 + \frac{C\Phi^2}{(1-\Phi)^3 r^2} \quad (3.25)$$

where Φ is the volume fraction of the solid, usually $0 \leq \Phi \leq 0.74$. When Φ is 0 it is pure liquid, $\lambda = 1$. When Φ is high, $\lambda \gg 1$, and the value of λ is approximately proportional to $\frac{\Phi^2}{(1-\Phi)^3 r^2}$, which is the classical result observed in experiment [102]. C is a parameter to be determined by experiments.

3.4 Boundary Conditions

For DEM particles, their interactions with boundary could be simulated using the LSD model, where the forces are computed through the overlap between particles and boundary. The relevant parameters could be determined analytically or through experiments.

For SPH particles, it is not straightforward to set boundary conditions in particle dynamic simulations, because when SPH particles approach a rigid boundary, the support domain of the SPH particles in the kernel function is cut off by the boundary [2]. Periodic boundary condition can avoid this problem. For fixed boundary conditions, one way is to set dummy particles to approximate the interface between the fluid phase and the boundary. The dummy particles will be counted to update normal particles density, but their position and velocity will not be updated. They are predefined to be fixed at the boundary. In this study, to be consistent, SPH particles share the same boundary conditions as DEM particles.

3.5 Stability Analysis

For DEM particles, the critical time step, or the collision duration is expressed as below:

$$t_{c,n} = \pi \left(\frac{k_n}{m_{\text{eff}}} - \frac{C_n^2}{4m_{\text{eff}}^2} \right) \quad (3.26)$$

In convention, the time step is chosen as $\Delta t = \min(t_{c,n}, 50)$ to maintain the stability of the simulation [145].

On the one hand, to approach the physical values, the time step Δt has to be very small, which makes the simulation very time consuming. On the other hand, sometimes the normal spring stiffness does not play a crucial role towards the simulation results so it could be reduced to increase the critical time steps. Considering these two aspects, usually k_n is determined in the scenario when the maximum overlap between two solid particles is less than 1% of their diameter, and in this way the computation efficiency of DEM simulation is substantially improved [29; 110].

For SPH, the main constrain for the time step is that particles do not travel through its neighbor particles in one time step [92], which leads to the criteria as

$$t_1 = \min \sqrt{\left(\frac{h}{f_k} \right)} \quad (3.27)$$

where f_k stands for any subjected force associated with particle k. An additional requirement is that during each unit step a wave does not travel out of the domain [89], which leads to another critical time step:

$$t_2 = \min \frac{h}{c_s + h \max \frac{u_{k,l} x_{k,l}}{x_{k,l}^2}} \quad (3.28)$$

where c_s is the speed of sound in simulation, $x_{k,l}$ and $u_{k,l}$ are position and velocity between particle k and l . The condition for global stability constraint is the combination of both

$$\Delta t = C \min(\Delta t_1, \Delta t_2) \quad (3.29)$$

where C is the Courant number, chosen in ranges of 0.1 to 0.3. For a coupling problem, the time step should satisfy both DEM and SPH requirements.

3.6 Particle Dynamics Parallel Simulator (PDPS)

3.6.1 Overview

All the simulations in this work is done by software package PDPS (Particle Dynamics Parallel Simulator), developed in the Pao Sustainable Engineering and Materials Laboratory (Pao Lab), Columbia University. It was inspired and designed based on the structure from the open source molecular dynamics software LAMMPS [106; 47; 41] (<http://lammps.sandia.gov/index.html>). It brought our interest to develop this new software package when observing the lack of available sources to conduct particle dynamics simulation with various particle based potentials, such as SPH, DEM, DPD, etc. It is our motivation to make an example and provide a general platform for researchers interested in particle dynamics simulation.

PDPS is written in C++, and it used Message Passing Interface (MPI) to conduct the parallel computing in a distributed memory environment. It can be run on comoputers with single processor or multiple processors which can compile C++ and support MPI library. The parallelism is fulfilled by using the domain decomposition technique, which has been demonstrated as the most efficient parallel algorithm towards molecular dynamics simulation[105]. A schematic illustration of the domain decomposition is shown in Fig. 3.1, in which the entire domain is discretized into 3D grids and each grid is assigned to a processor. Only particles inside the local domain belongs to the corresponding processor, and every few time steps the particle list is updated as certain condition is triggered.

Similarly to LAMMPS, the framework of PDPS includes the following unique modules: (1) initialization of the computational model; (2) build neighbor list of particles (3) calculate pairwise forces between particles based on the neighbor list (4) time integration (i.e. velocity-Verlet algorithm) to update particle position and velocity; (5) repeat step (2) if simulation is not finished.

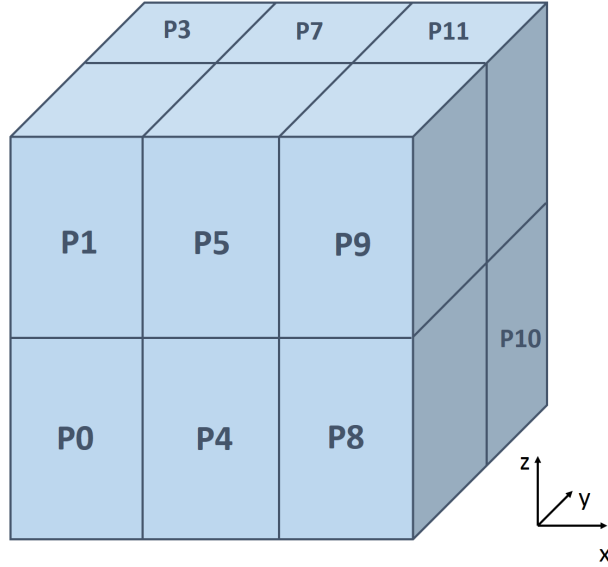


Figure 3.1: A schematic illustration of the domain decomposition technique

A more vivid plot of the flow chart is given in Fig. 3.2. In this work, in addition to executing PDPS with CPU, additional CUDA C++ programming is included in PDPS for GPU parallization purpose. The specific algorithm differences and performances are discussed in Appendix.

3.6.2 Domain Decomposition

As mentioned in Sec.2.6.2, MPI parallel scheme is implemented in PDPS. To make the parallel computing efficient, the simulation domain should be cut uniformly, so as to cut it uniformly is to average the working load for each processor at the simulation pace, as well as to minimize the particle transfer between processors. For example, if 64 cores are used, it could be cut to $4 \times 4 \times 4$ if all the particles are uniformly distributed in all directions, and $8 \times 8 \times 1$ should be implemented if the simulation domain is flat and uniformly distributed horizontally.

During each time step of the simulation, processors communicate with its neighbor to transfer particle information if any particle move out of its original subdomain. Communication time increase as the number of processors used increases, so the computational speed does not increase linearly with the number of processors, and the number of processors used in computation should be chosen reasonably.

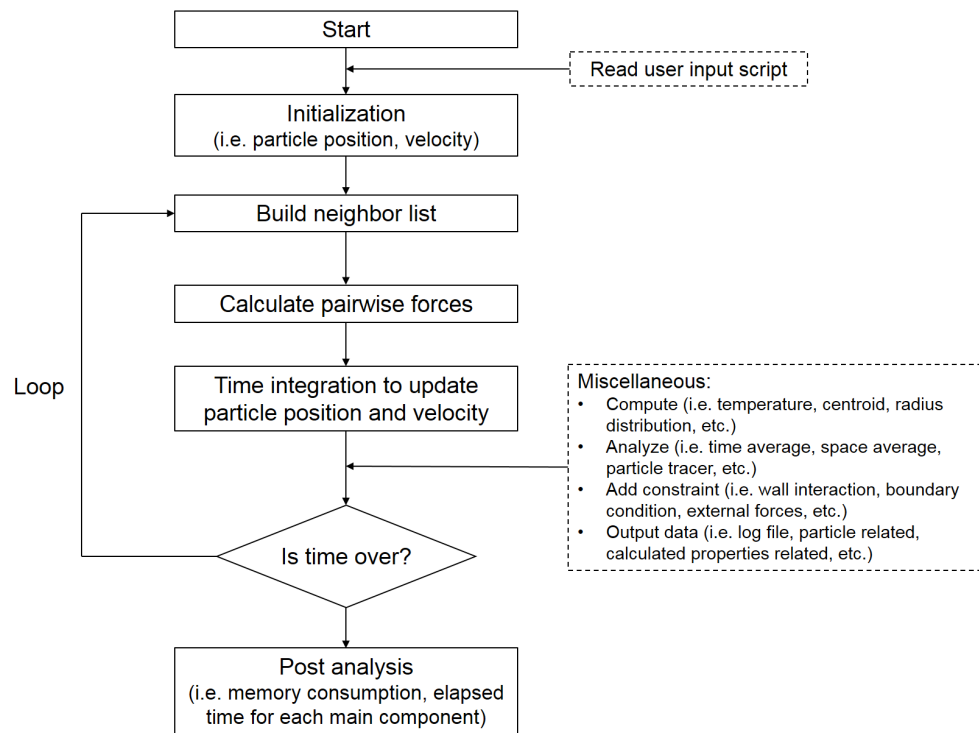


Figure 3.2: A flow chart of the Particle Dynamics Parallel Simulator(PDPS)

3.6.3 Neighbor List Construction

Particle neighbor list is built for every particle in the simulation system, and it stores the information of its neighbor particles that are close enough to have interaction forces. To improve computational efficiency, half neighbor list is built, which means if two particles i and j are neighbors, only j is stored as neighbor particle of i , but i is not neighbor particle of j . Since the forces between two particles have the same magnitude but opposite direction, one force computation is enough for both particles. In this way, the neighborhood information of two particles can be stored only in one of the particles, and the total computation time would be reduced compared to complete neighbor list. For example, if there are 4 particles (1, 2, 3, and 4) and they are neighbors each other, their half neighbor list and complete neighbor list are given in Table 3.1.

Table 3.1: Comparison of half neighbor list and complete neighbor list

Particle ID	Half Neighbor List	Complete Neighbor List
1	2, 3, 4	2,3,4
2	3, 4	1,3,4
3	4	1,2,4
4		1,2,3

3.6.4 Force Calculation

In PDPS, force calculation for particles include pair force, body force and boundary force.

Pair force is the force between individual particles, and the interaction forces between SPH particles, DEM particles and the coupling between these or other force model all belong to pair force. For all these models, the pair force depend on their relative positions and relative velocities. This part is the most time consuming part of the whole simulation process, and neighbor list is built to make this computation more efficient. With the neighbor list, one only needs to loop through all the particles listed in neighbor list to calculate the pair force. Since Newtons Third Law claims that pair force is the same but with opposite direction between two particles, one pair force is only calculated once and two particles force could be updated. Body force is the force that directly applies to each particle such as gravity force, and the calculation of body force is independent with

each other and finished one by one during the simulation.

Boundary force is the interaction force with boundary, and it is independent between particles. The forces of particles with boundaries also depend on their distances and velocities. Mixer blades could be treated as boundary, as well as a fixed amount of particles. When the shape of the blade is easy to simulate such as a rectangular plate, then simulating it as boundary is straightforward. However, when the shape of blade is complicated and the relative distances and relative velocities are difficult to obtained, simulating blades as a bunch of fixed particles is more appropriate.

3.6.5 Time Integration

There are several integration methods available. One of most common one is the Euler algorithm. For the evolution of particle i , its formula is written as following:

$$\begin{cases} r_i(t + \Delta t) = r_i(t) + \Delta t v_i(t) \\ v_i(t + \Delta t) = v_i(t) + \Delta t f_i(t) \\ f(t + \Delta t) = f_i(r(t + \Delta t), v(t + \Delta t)) \end{cases} \quad (3.30)$$

Despite of its simpleness, Euler integration is the first order method with poor accuracy. In particle dynamics simulation, Velocity-Verlet algorithm is the most common choice because of its accuracy and efficiency [130; 115], and it is applied to various particle methods such as Molecular Dynamics (MD) [126], Dissipative Particle Dynamics (DPD) [82], Smoothed Particle Hydrodynamics (SPH) [34] and Discrete Element Method (DEM) [115; 69]. PDPS is a particle based method which can be considered as a coarse-grained particle dynamics simulation. Therefore, Velocity-Verlet algorithm ($\lambda = \frac{1}{2}$) will also be adopted in this code.

$$\begin{cases} r_i(t + \Delta t) = r_i(t) + \Delta t v_i(t) + \frac{1}{2}(\Delta t)^2 a_i(t) \\ \tilde{v}_i(t + \Delta t) = v_i(t) + \lambda \Delta t f_i(t) \\ a_i(t + \Delta t) = a_i(r(t + \Delta t), \tilde{v}(t + \Delta t)) \\ v_i(t + \Delta t) = v_i(t) + \frac{1}{2} \Delta t (a_i(t) + a_i(t + \Delta t)) \end{cases} \quad (3.31)$$

Therefore, in the Eq. 3.31, λ is employed to predict velocity at the next time step ($t + \Delta t$) which denoted by \tilde{v} . \tilde{v} will then be used to approximate force at next time step which will be used to correct the predicted velocity at the next time step. Although PDPS will provide different choice

mentioned above, it is suggested to use the classic Velocity Verlet method with $\lambda = \frac{1}{2}$ for most of particle dynamics simulation.

Velocity Verlet integration method is the most common method used in the particle dynamics simulation, because of its simplicity and accuracy. The specific integration is implemented in two steps.

In the initialization, it will perform:

$$\begin{aligned}\vec{v}(t + \frac{1}{2}\Delta t) &= \vec{v}(t) + \frac{1}{2}\vec{a}(t)\Delta t \\ \vec{x}(t + \Delta t) &= \vec{x}(t) + \vec{v}(t + \frac{1}{2}\Delta t)\Delta t\end{aligned}\tag{3.32}$$

With position updated at time $t + \Delta t$, we are able to derive $\vec{a}(t + \Delta t)$ based on the potential energy. Thus, in the final stage, it will only perform:

$$\vec{v}(t + \Delta t) = \vec{v}(t + \frac{1}{2}\Delta t) + \frac{1}{2}\vec{a}(t + \Delta t)\Delta t\tag{3.33}$$

Some applications of PDPS will be provided in the next Chapter to demonstrate the software and verify the algorithm.

3.7 Benchmark Studies

One of the major concern of particulate dynamics methods is to choose the correct parameters in simulation. For the interaction between liquid and solid particles, there is no available analytical solution that could directly use the fundamental properties of the materials. In this work, the parameters in simulation is chosen from classical benchmark studies. Parameters used in SPH DEM coupling model are chosen to match some classical solid-liquid two phase flow problems, and these parameters are applied to solid-liquid mixing problems.

3.7.1 A single DEM Particle Falling in a Fluid

In this case, a single DEM solid particle falling in a fluid of many SPH particles is investigated. This test is the first step to validate the feasibility of the DEM-SPH coupling method. As shown in Fig. 3.3, the simulation domain is a cylinder tank with radius 20cm, height 30cm.

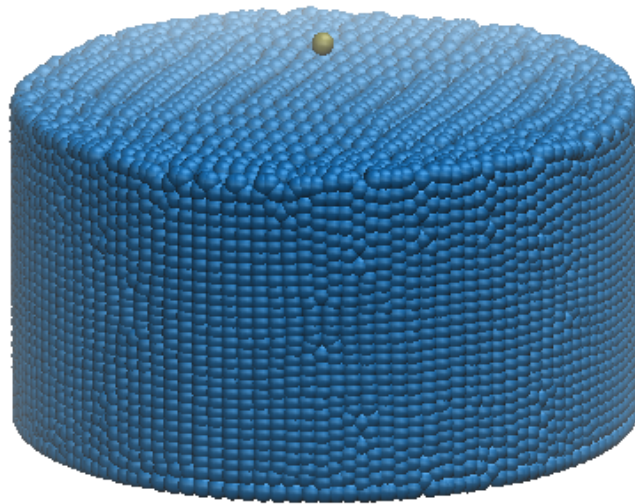


Figure 3.3: Sedimentation of a single DEM particle among SPH liquid particles

Table 3.2: Parameters of the DEM and SPH particles in the sedimentation case study

Parameter	Solid	Liquid
density	$5000kg/m^3$	$1000kg/m^3$
radius	$0.5m$	$h = 1m$
viscosity		$10m^2/s$
numbers	1	30740
velocity	$0.218m/s$	0

Properties of DEM and SPH particles are listed in Table 3.2. Since the Re number in this case is less than 1, it could be considered as a Stokes flow.

Based on the Stokes flow equation, the velocity of this single DEM particle could be expressed as:

$$V = \frac{2\Delta\rho g R^2}{9\mu} = 0.218m/s \quad (3.34)$$

Since the coupling forces between DEM and SPH particles have a range of interaction domain rather than stay on a distinct radius, it is reasonable that we multiply the viscosity forces between DEM and SPH particles by a coefficient to get the correct Stoke flow velocity [10]. It is found when the coefficient is around 3.08, the Stokes flow could be reproduced as shown in Fig. 3.4. Since it is a particle method, it is reasonable that the final setting velocity has some oscillations. This example, for the sedimentation of a single DEM particle among SPH liquid particles shows that the coupling force can effectively captures the physics of the interaction between liquid and solid particles.

3.7.2 Sedimentation of a Granular Block

In this case study, a porous block with equally distributed solid particles was initially rested above a water column. At some certain time, the block was released and fell into water. To focus the study on the drag force and the influence of the porosity, the relative motion among solid particles is restricted, namely, the porous block is treated as a rigid body with constant porosity inside the block. However, when the block drops in the water column, the porosity in space will be

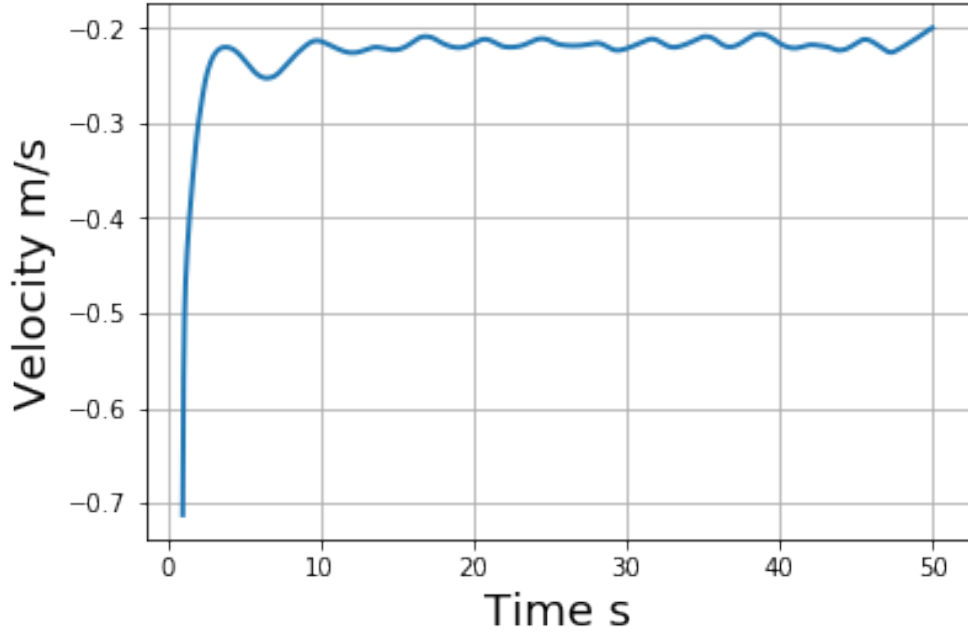


Figure 3.4: Sedimentation velocity of a single DEM particle among SPH particles

changed together with the motion of the block. Parameters of the system are given as follows: the density of water is $\rho_0 = 1000 \text{ kg/m}^3$, viscosity is $\mu = 0.1 \text{ Pa} \cdot \text{s}$, density of the solid particles is $\rho_s = 2000 \text{ kg/m}^3$. These parameters are listed in Table. 3.3. By choosing the desired diameter, the volume of the particle and the porosity in the fluid field could be calculated. A series of diameters ranges $0.0288 \sim 0.0415 \text{ m}$ with the associated porosity $0.9 \sim 0.7$ are tested, and the lattice spacing for both SPH and DEM is 0.2 m . The width of the simulation box is 2.2 m for both sides, and the porous block has dimensions of $1.8 \times 1.8 \times 1.0 \text{ m}$. SPH horizon is chosen as $h = 0.2 \text{ m}$, and time step is 0.0005 s . Since the solid particles are fixed relative to each other, contact forces in DEM are not considered. SPH-DEM coupling is analyzed in this case.

Fig. 3.5 shows the snapshots of the process of sedimentation with a block of constant porosity of 0.8. The motion of the block in the water column are affected by gravity, buoyancy and drag force, where the first two items are relatively unchanged, and the drag force is supposed to decrease with the speed. The three types of forces will finally reach an equilibrium state with a constant velocity. The evolution of each block is shown in Fig. 3.6. Velocities are scaled by the initial value at the moment when each block touches the water surface, and the time is scaled by the period

Table 3.3: Parameters of the DEM and SPH particles in the block sedimentation case study

Parameter	Solid	Liquid
density	$2000kg/m^3$	$1000kg/m^3$
radius	$0.1m$	$h = 0.2m$
viscosity		$10m^2/s$
numbers	500	10584
velocity	$0.0288 \sim 0.0415m/s$	0

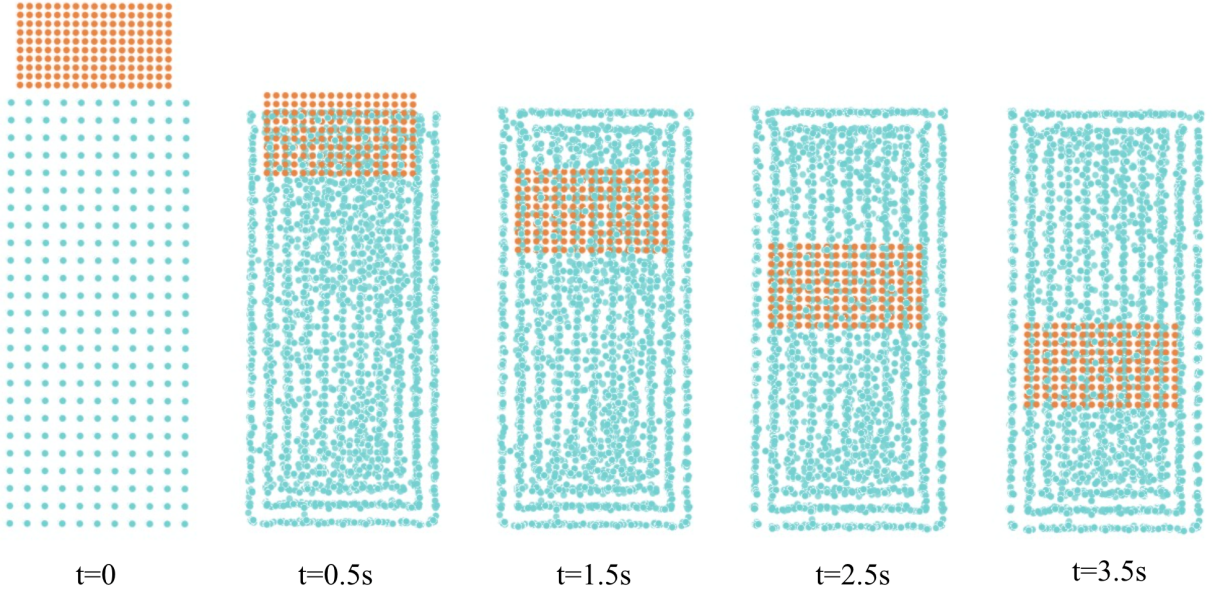


Figure 3.5: Snapshots of the process of sedimentation. Green dots are SPH particles(water) and DEM particles(solid) are in orange color

of relaxation. For blocks with porosity from 0.7 to 0.9, the scaled terminal velocities are increased from 0.23 to 0.42. Because the smaller porosity is associated with the larger volume fraction of the solids, the ratio of resistance (drag and buoyancy) and driving force (gravity) is expected to be larger, which slows down the motion more quickly. Note that the slight oscillations of the curves reflect the fluctuation of the fluid.

The analytical solution of the terminal velocity can be obtained by equating the drag force and the buoyancy to the gravity. The relationship is shown below [112]:

$$0.392Re_p^2 + 6.048Re_p^{1.5} + 23.04Re_p - \frac{4}{3}C\epsilon^\xi = 0 \quad (3.35)$$

where $Re = uD/\nu$ is the Reynolds number with kinematic viscosity ν , diameter D of the solid particle and the relative velocity u between two phases. C is a coefficient depends on densities and viscosity, and ξ depends on the Reynolds number. We compare the simulation results with the analytical solution, which are shown in Fig. 3.7. The terminal velocity of the block is scaled by the one in single particle sedimentation. The error of each sample point is less than 5%. Conceptually, the error is expected to decrease as porosity increase, because the fluctuation of the surrounding fluid is influenced by the geometry of the block. In our simulation, the tendency is barely observable since the measurements may not be precise and the error itself is small enough to distinguish from each other.

3.7.3 Liquid Flowing through a Porous Media

In the third case study, we evaluate the effectiveness of our method on a more complicated scenario, liquid flowing through a porous media. In this case a U tube with a square width of $10m$ and a horizontal length of $60m$ is generated, as shown in Fig. 3.8 below. The vertical height is different between the two sides. In the middle of the tube filled in a cubic porous media with DEM particles accounting for 52.3% of the total volume of the cubic media.

The properties of SPH and DEM particles are listed in Table 3.4.

As the liquid flowing from one side of the U tube to the other side, the flow speed and the height difference between the two sides, are a function of the porosity of the porous media and viscosity of the liquid. Based on the previous studies[102], the height difference between two sides would show an exponential trend:

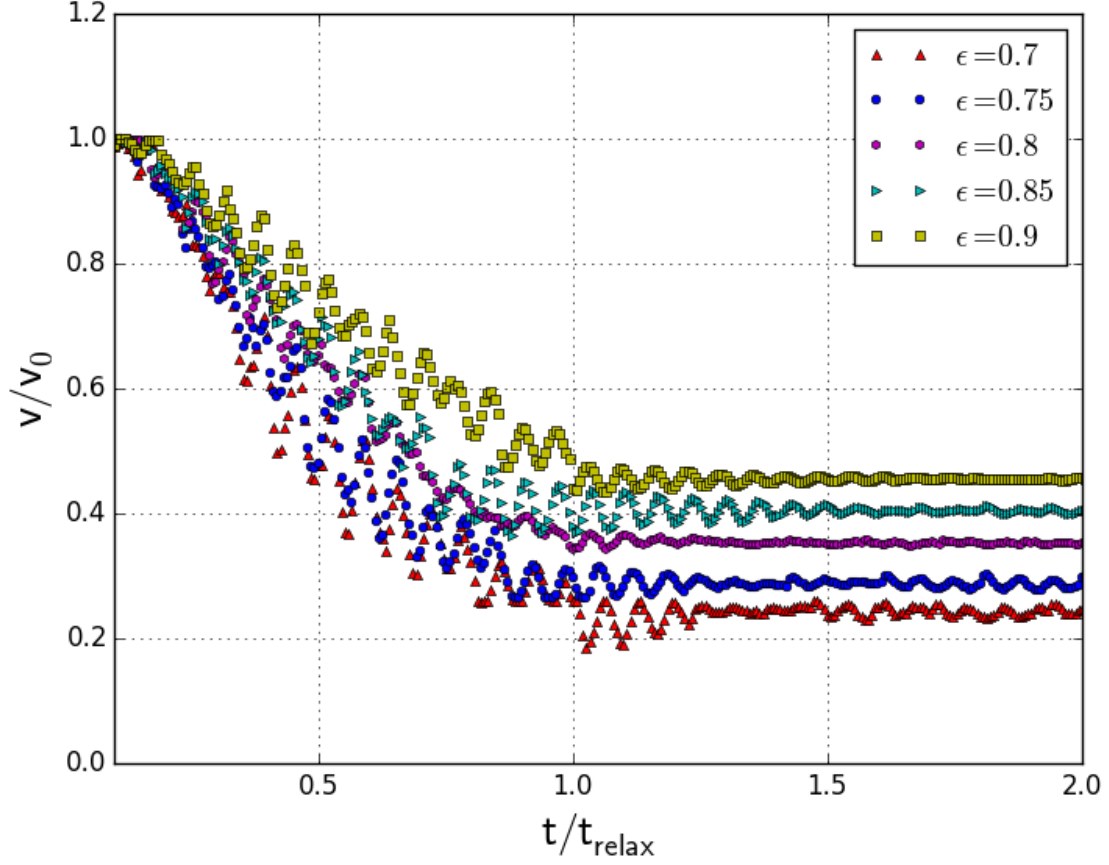


Figure 3.6: Velocity evolution of the porous block with different porosities

Table 3.4: Parameters of particles in the case of a liquid flowing through a porous media case

Parameter	Solid	Liquid
density	5000 kg/m^3	1000 kg/m^3
radius	0.5 m	$h = 1 \text{ m}$
viscosity	depend on porosity	$0.01 \text{ m}^2/\text{s}$
volume fraction	52.36%	47.64%
numbers	1000	7479

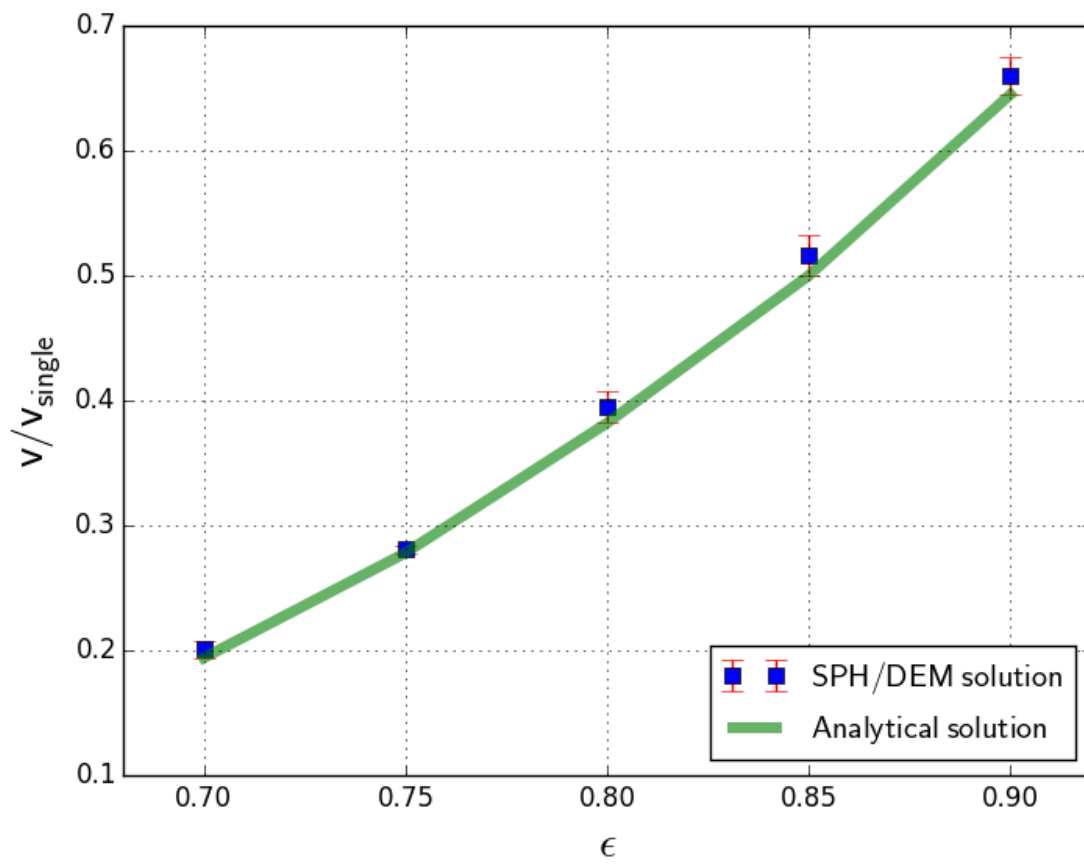


Figure 3.7: Terminal velocity of the porous block at different porosities

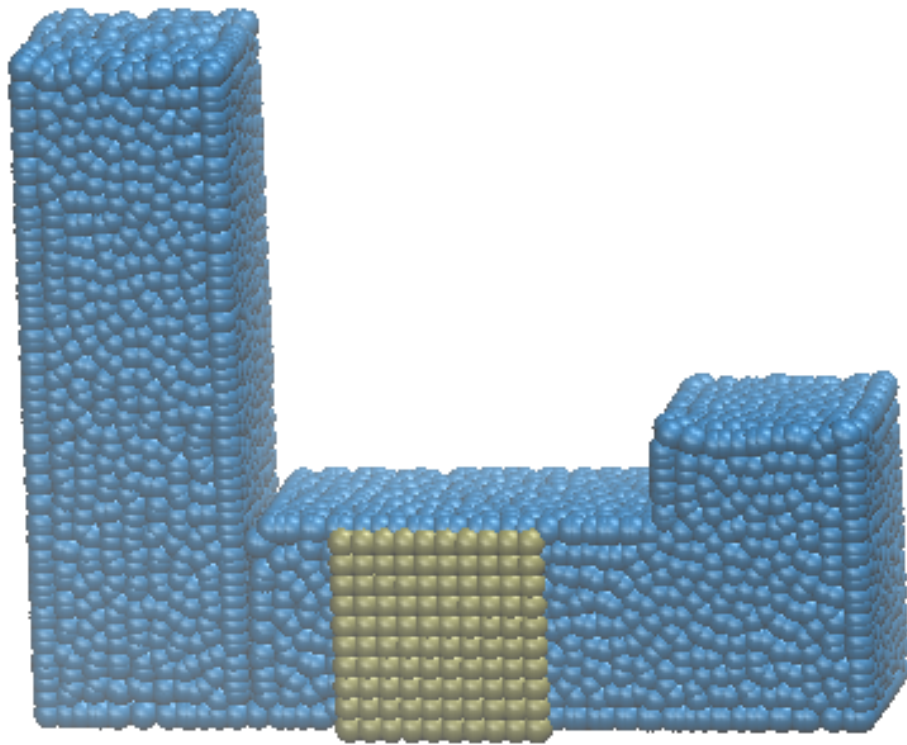


Figure 3.8: SPH liquid flow through porous media

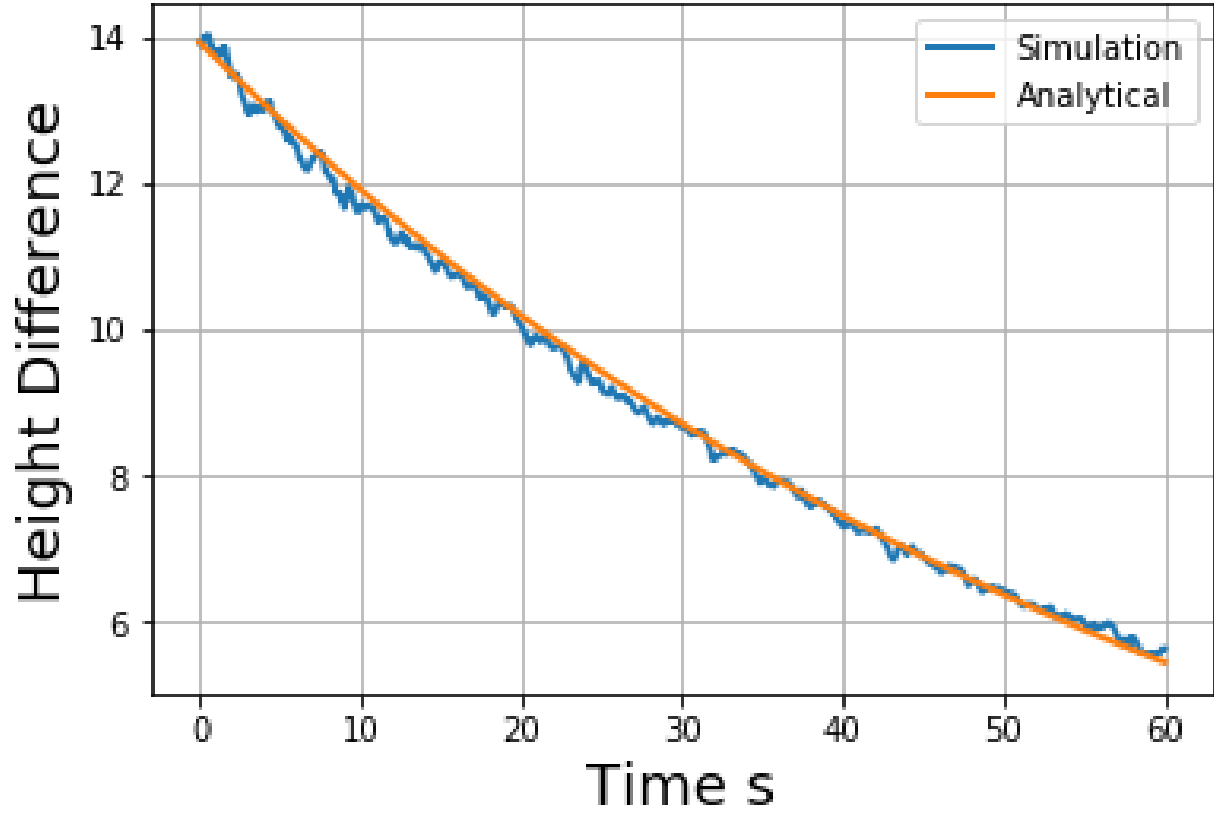


Figure 3.9: Height difference of SPH liquid flowing through porous media

$$\Delta H = \Delta H_0 \exp\left(-\frac{2K_h t}{L}\right) \quad (3.36)$$

$$K_h = \frac{\rho g k}{\mu} \quad (3.37)$$

$$k = \frac{Cr^2(1-\phi)^3}{\phi^2} \quad (3.38)$$

where ϕ is the solid fraction, and C is a problem dependent parameter to be determined. As shown in Fig. 3.9, the height difference between two sides of the U tube follows an exponential curve with $C=8.07 \times 10^{-7}$, which validates our model for viscous forces between SPH and DEM particles.

Chapter 4

Foaming Asphalt Simulation

In this chapter, one application of gas-liquid two-phase flow problem, asphalt foaming process is studied. Smoothed Particle Hydrodynamics (SPH) is applied to solve this problem.

4.1 Foaming Characterizations

The characteristics of foamed asphalt have significant influence on the mixture workability and performance. Therefore in order to effectively use foamed binder to produce WMA, it is important to understand the characteristics of the foamed asphalt.

Since the bubbles are thermodynamically unstable in the foamed asphalt, it is crucial to quantify their evolution with time. Current methods for evaluating foaming characteristics of asphalt are based on measurements of two primary parameters: expansion ratio (ER) and half-life (HL). ER is defined as the increase in volume relative to the original asphalt volume, while half-life is defined as the time in seconds that asphalt foam requires to collapse from the maximum expansion to half of the maximum expansion. Empirically, ER parameter is related to viscosity of the foamed asphalt, and therefore, this parameter affects how foamed asphalt disperses in a mix.

$$ER(t) = \frac{V_{max}}{V_f} \times 100\% \quad (4.1)$$

Where V_{max} = the maximum expanded volume of the foamed asphalt, V_f = the final volume of the asphalt after a certain period.

HL is a measurement of the foamed asphalt stability. It provides an indication of the rate at which the foamed asphalt bubbles collapse with time, which in turn, determines the length of time

the foamed asphalt will remain in the foamed state to allow enough time for coating aggregates in a mix.

$$HL = t_{0.5(V^0-V^f)/V^f} \quad (4.2)$$

Where $t_{0.5(V^0-V^f)/V^f}$ is time at which the overall foam volume is reduced by half from its maximum value.

Another parameter that is gaining widely application to qualify the characteristics is the foam index (FI). The FI is a measurement of the area under the decay curve, i.e., the change in expansion ratio with time, which reflects the stored energy in the foam for a specific asphalt foamed at a known temperature and a determined application rate:

$$FI = \sum_{t=0}^{t_{ER=1}} 0.5 \times (ER_t + ER_{t+1}) \times (t_{t+1} - t_t) \quad (4.3)$$

Where ER_t is ER at time t and ER_{t+1} is the ER at time $t+1$.

4.2 Model Description

From experimental observations, it was found that the asphalt foaming involves a liquid and gas combined two-phase flow. The foaming characteristics, i.e., the maximum ER and HL of the foamed asphalt, depend on how much gas bubble is generated and how quick the foamed asphalt decays thereafter. In this study, an SPH simulation tool is developed to simulate foaming evolution in the foamed asphalt.

Note that some numerical models have been conducted to simulate bubble generation according to gas concentration in SPH liquid phase [27; 120]. In those models, though the animation is satisfying, it is a one-way coupling, where only the force from liquid phase to gas bubble phase was accounted for. A two-way coupled air bubble simulation using SPH was developed to simulate the trapped air [3], where the air bubbles and liquid phase are interactively coupled (two-way coupling) through the drag force. The major difficulty in the water-air two-phase flow simulation is the high-density ratio. One possible way to deal with high-density ratio is to increase the density of air bubbles to one tenth of water and to treat bubbles as rigid body [101]. However, it turns out that this way works numerically but cannot satisfy the physical property. Thereafter, an improved model including thermal effect was proposed to simulate water boiling [108].

In the above-mentioned two-phase models, the interaction between the gas phase and the liquid phase is relatively weak comparing to the large volume increase of foaming phenomenon, where gas phase is no longer surrounded by liquid phase but rather occupied by the majority of air voids. To simulate such a case in foamed asphalt binder with a typical ER of 6 to 8 or even higher, traditional bubble motion in liquid is not suitable anymore. To overcome this difficulty in traditional two-way coupling method, in the numerical simulation model presented in this study, it is assumed that both the gas and liquid phases are generated at the beginning by sharing with the same governing equation. Then the height of the foamed asphalt decreases as the gas bubble collapse at a given decreasing frequency. Therefore, it is the initial gas volume ratio and the collapse rate that determine the expansion ratio and half-life of the foamed asphalt binder. The simulation procedures are described in detail in the following subsections.

4.2.1 Simulation Box

A rectangular simulation box with a size of 10cm in. width, 10cm in. depth, and 40 cm in height is used in this simulation. 6000 fluid particles (including asphalt particles and bubble particles) and 10,000 boundary particles are generated at the simulation box. The simulation time of the foaming duration is 200 seconds and the time step is 0.0002 seconds.

At the beginning, the simulation box was filled up with foaming asphalt, including asphalt particles and bubble particles. The particle numbers of the asphalt and bubble are proportional to their volume, and they share the same governing equation. Then the foamed asphalt began to collapse and the bubble particles disappeared from the simulation box once they collapsed. After the collapsing process finishes, a minority of gas bubbles still got trapped in the asphalt, which cause the volume of the final state slightly larger than its original volume.

Figure 4.1 shows a snapshot during the simulation. The black particles represent asphalt binder, and grey particles represent bubbles, which occupy the majority of volume. Boundary particles are not shown here. During the simulation of foaming process, the number of grey particles decreases while the number of black particles keeps the same. A new parameter surface level is introduced to represent the asphalt height level so that volume expansion ratio can be recorded. The height of the foamed asphalt is calculated through the highest position of asphalt particles. The surface level keeps dropping until the collapse of grey bubble stops. The expansion ratio (ER) curve, half-life

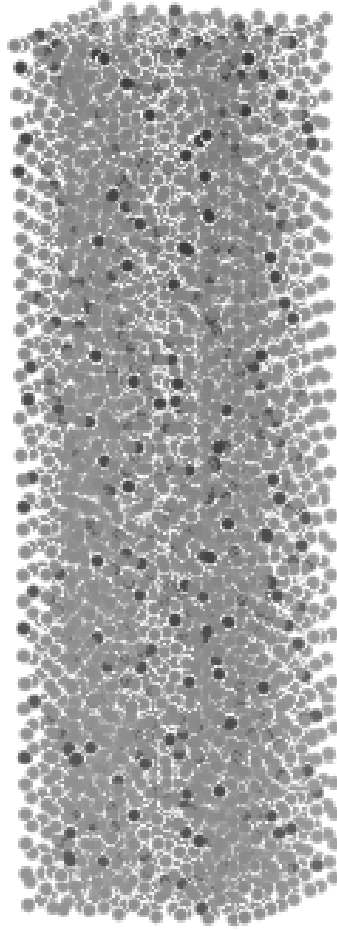


Figure 4.1: Snapshot of the foaming simulation at $t = 10$

(HL) and Foam Index (FI) thus can be computed through the simulation process.

The parameters of particles in the SPH are summarized in Table 4.1

4.2.2 Initial Expansion Ratio Computation

It is assumed that at the beginning the asphalt binder is foamed immediately, which means the liquid agents have evaporated to gas phase and mixed with the asphalt immediately, although the thermodynamic process is considered that the temperature of asphalt decreases since liquid evaporation absorbs a large amount of energy. Once the temperature of asphalt drops below evaporation

Table 4.1: Parameters of asphat particles in SPH simulation

Parameter	Values
Asphalt density	$1030kg/m^3$
Kernel length	1cm
Speed of sound	$30m/s$
Viscosity coefficient	$0.2m^2/s$
Time step	$0.0002s$

Table 4.2: Physical parameter used in computing gas density

Parameter	Values
p	$50psi$
M	$18g/mol$
R	$8.31J \cdot K^{-1} \cdot mol^{-1}$
T	$150^\circ C$

point the evaporation stops, so it is not guaranteed all the liquid additives are evaporated. Once the foaming process finishes, the foamed asphalt is at its maximum volume and then begins to collapse until a stable status is arrived. During this process, the foaming evolution is simulated by tracing the height of the asphalt surface, from which the ER curve, HL and FI are obtained.

At the initial state, both the liquid and bubble particles were generated. An ideal gas model is used here to compute the gas density.

$$\rho = \frac{pM}{RT} \quad (4.4)$$

Where p is the experiment pressure, M is the relative molecular mass, R is the ideal gas constant, and T is the temperature. Their values are listed in Table 4.2.

The water vapor density in the experiments is $1.74g/m^3$. The foaming generation process is an adiabatic process, and the amount of liquid evaporated is determined by energy conservation, in which asphalt heat capacity and water latent heat is introduced. The % of liquid evaporated ϕ is shown below:

$$\phi = \frac{(T_{asphalt} - T_{water-boiling})C_{asphalt}m_{asphalt}}{L_{water}m_{water}} \quad (4.5)$$

Where $T_{asphalt}$ is the temperature of asphalt, $T_{water-boiling}$ is the temperature of water boiling point, $C_{asphalt}$ is heat capacity of asphalt, L_{water} is latent heat of water, $m_{asphalt}$ and m_{water} are mass of asphalt and water accordingly. Given liquid additive content w , based on previous two equations, Expansion Ratio (ER) of foaming asphalt could be expressed as:

$$ER_{initial} = 1 + w\phi \frac{\rho_{waterliquid}}{\rho_{watervapor}} \quad (4.6)$$

Other assumptions made in the present model are as follows:

1. The density of gas in the simulation box is calculated through the ideal gas model in the given pressure and temperature.
2. The volume of the foamed asphalt is the sum of the volume of the original binder and that of the evaporated bubbles.
3. All the energy consumed for water evaporation come from asphalt, and this process happens immediately that heat transfer from environment is neglected

Based on the calculated gas vapor density, the initial ER of the foamed asphalt is calculated. For example, when the water content is 2% and temperature is $150^{\circ}C$, all the water get evaporated and the volume ratio of gas bubble to asphalt is calculated as

$$\frac{1030 \times 2\%}{1.74} = 11.8 \quad (4.7)$$

Then 12.8 is the theoretical maximum volume expansion ratio predicted by the present simulation model when water content is 2% at $150^{\circ}C$.

4.2.3 Bubble Collapsing Model

The essential part in the simulation is to predict the collapsing process of the gas bubbles in the foamed asphalt. In traditional bubble simulations, it was assumed that bubbles only occupy a small amount of the volume and they are mainly surrounded by liquids. According to this assumption, the bubbles rise through the interaction of buoyancy force and drag force. However, in foamed asphalt, bubbles volume is usually up to several times of the initial volume of the asphalt. Instead of rising among liquid, they are surrounded by each other and occupy the majority of the space. Therefore, the quantity of interest in this study is the variation of bubbles volume in the foamed

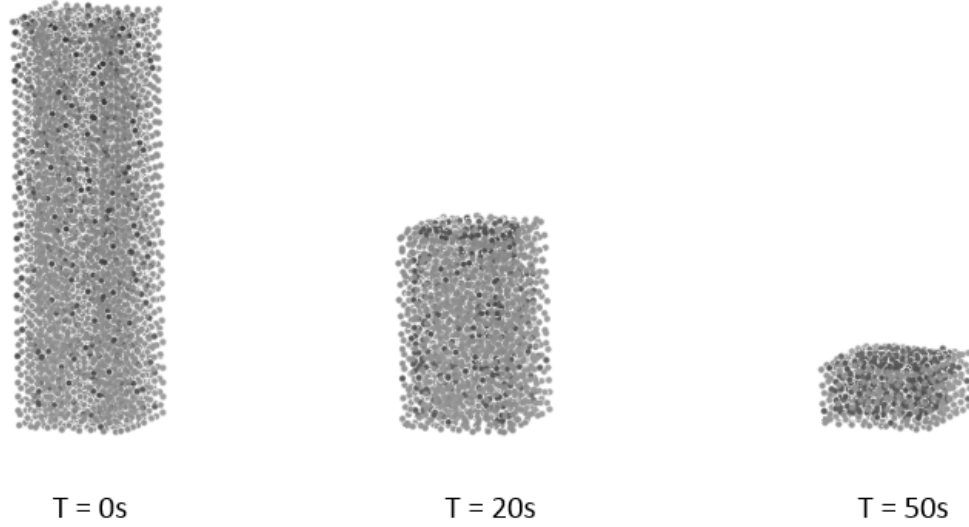


Figure 4.2: Snapshot of Foaming asphalt collapse process at $t=0s$, $20s$ and $50s$

asphalt. Accordingly, the factor influencing the bubbles volume is the bubble collapse rate, rather than its interaction with asphalt. Since asphalt interaction with gas bubble is not that important, they are assumed to share the same governing equation with that of the gas phase to maintain the equilibrium in this study. Similarly, since it is Expansion Ratio(ER) such macro scale properties to be investigated, the specific size of bubble is not that important. Here the bubble size 1 cm (also the kernel length) is a reasonable value, either not too large to cause unstable measurement, nor too small to cost too much computational resource.

Once the bubbles in the foamed asphalt begin to collapse, the gas bubbles whose pressure is lower than the mean asphalt pressure are collapsed at a given decreasing frequency. The pressure in the function chamber is calculated based on the SPH model, i.e. based on density through the equation of state. Once a foamed particle is collapsed, it is removed in the simulation. In this respect, the decreasing frequency is the only parameter that needs to be calibrated from experimental results. Based on the foaming decay curve obtained from experiment at the water content of 2%, the decreasing frequency is determined by correlating the simulation results with the experiment curve. Figure 4.2 shows the bubble collapse process. Since the pressure is the reason

for bubble collision, only the bubbles at surface has chance to collapse. Bubbles can only rise up if the bubbles above them collapse or rise up, and they cannot rise if they are surrounded by asphalt liquid. So at the end of the simulation, some of the bubbles are trapped since the way up to surface of the bubbles is blocked by asphalt liquid. These trapped bubbles explain the phenomenon that the final asphalt volume exceeds its original volume.

Generally, a higher water content lead to a fewer asphalt surrounding each single bubble particle, which makes the generated bubbles collapse quicker. Thus, it is reasonable to assume that the bubble collapsing rate is proportional to the water contents. Therefore, based on the decreasing frequency for the water content of 2%, the bubble-collapsing rate for other water contents is able to be determined.

4.2.4 Adaption with Accumulation Process

In the present SPH simulation model, asphalt is foamed at the beginning and the process to simulate is the collapsing process. However, in real experiments, it takes about twenty seconds to fill up the foamed asphalt container. It is thus assumed that during this bubble accumulation process, the volume of the foamed asphalt increases linearly to its maximum volume.

To capture this foaming accumulation period observed from experiments, an adaption was made in the simulation regarding the foaming accumulation process. The volume of foamed asphalt in simulation is accumulated linearly during the first 20 seconds from the starting point of its original curve. Figure 4.3 illustrates the effect of this adaption for the first 20 seconds.

4.3 Simulation Results and Validation

4.3.1 Experimental Results

Experiments were conducted with water as the foaming agent by the self-developed foamer. Three different water content, 1%, 2% and 3% (by weight of the binder) were applied. All other controlling parameters were kept constant, among which the air pressure in the function chamber is 50 *psi*, the temperature in the function chamber is 150°C. At a fixed valve position, after a series of calibrations, the air pressures applied in the asphalt container are determined, such that a constant volume of binder (1000 g) was flow into the function chamber every minute. With those predeter-

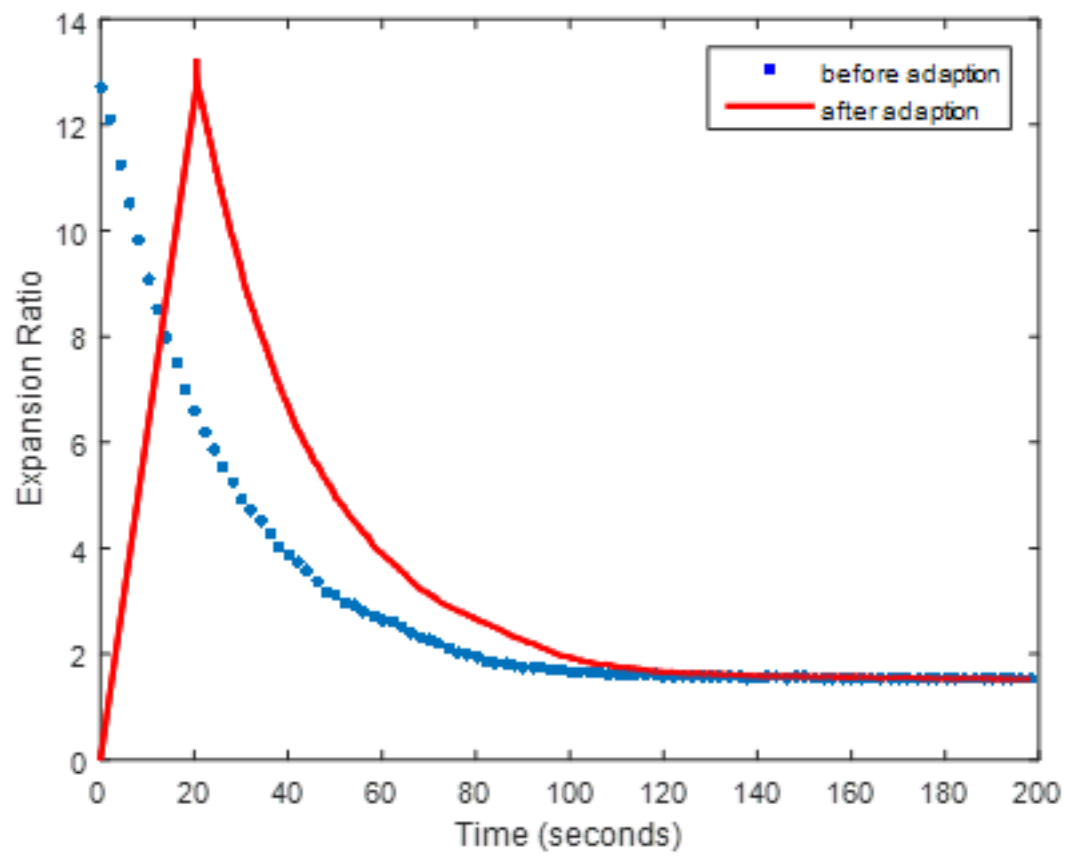


Figure 4.3: Volume adaption for the first 10 seconds accumulation period

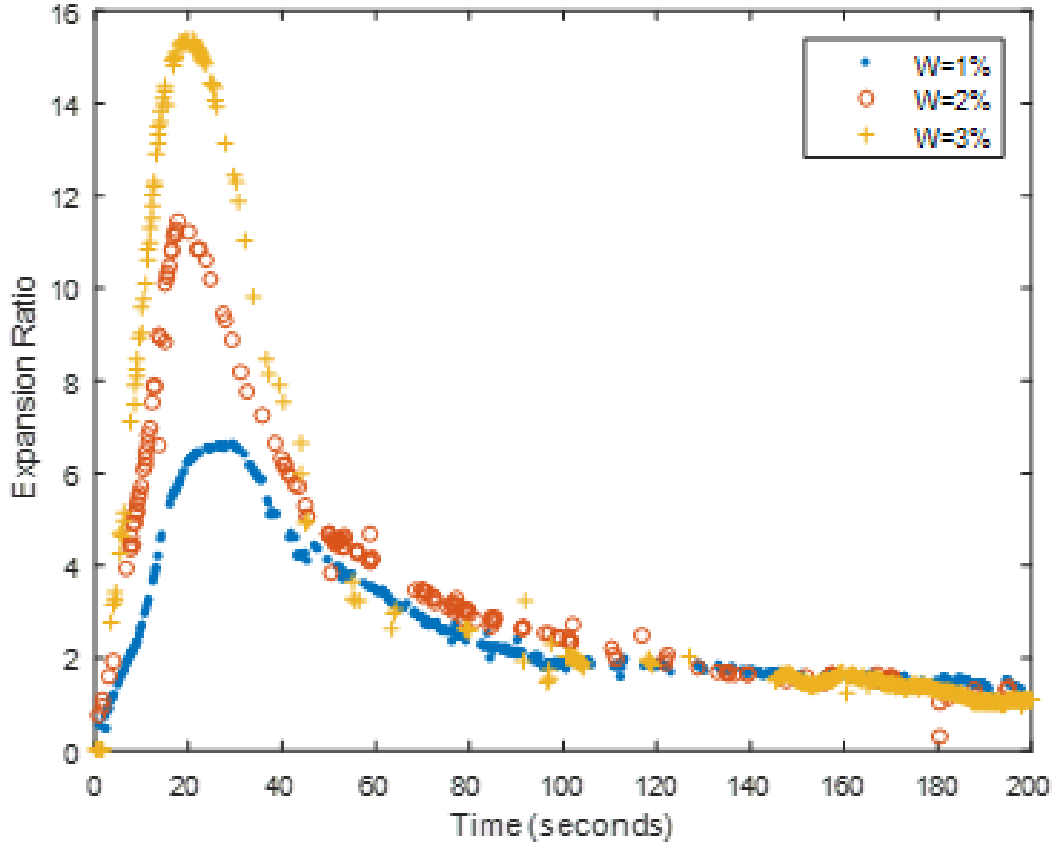


Figure 4.4: Experiment result of foaming asphalt Expansion Ratio (ER)

mined controlling parameters, the water flowing rate was determined, i.e., $10\text{ml}/\text{min}$, $20\text{ml}/\text{min}$, and $30\text{ml}/\text{min}$ for water, such that the pre-designed water contents of 1%, 2% and 3% can be achieved.

The foaming evolution of the foamed asphalt with different water contents are compared in Figure 4.4. In the first stage, the foamed asphalt rapidly accumulates in the container after it is ejected from the function chamber, which is represented by the approximately linearly rising curves. During this stage, some foams collapse as they accumulate in the container, and an equilibrium status arrives when the collapsing rate approaches the injecting rate of the foamed asphalt, which is represented by the peak points shown in Figure 4.4. Once the equilibrium is reached, the ejecting valve is shut down. Thereafter, the foams quickly collapse and the height of the foamed asphalt rapidly decreases in the container, which is represented by the rapidly descending curves in Figure

4.4. Gradually, the foamed asphalt reaches a relatively stable state with a volume expansion about 1.5. This remaining expanded volume is mainly due to the long-lasting small bubbles in the foamed asphalt.

4.3.2 Numerical Results

In this section, the foaming characterization of the foamed asphalt at the water content of 2% predicted by the present simulation model is first examined to obtain the decreasing frequency. From the algorithms presented in Section 4.2, the parameters in the SPH model can be obtained. Firstly, through the accumulation period adaption, the foaming accumulation time is determined. From the decay curve shown in Figure 4.4, the decay time interval in the SPH model is chosen as follows:

$$T_{collapse} = \frac{0.05}{w} \times 2^{tw/50} (t < \frac{50}{w}) \quad (4.8)$$

$$T_{collapse} = \frac{0.05}{w} \times 2^{tw/100} (t \geq \frac{50}{w}) \quad (4.9)$$

Where w is the percent of water content and t refers to time.

With the obtained accumulation period and decreasing frequency, the necessary parameters in the simulation model can be determined. The comparison of the foaming evolution of the foamed asphalt at the water content of 2% predicted by the simulation model and the experimental result is shown in Figure 4.5. It shows that the present numerical model captures the foaming evolution in the foamed asphalt. Note that in the real test, the bubble begins to collapse before the foamed asphalt reaches its maximum ER. Therefore, the maximum ER obtained from the test results is slightly lower than that predicted by the simulation. It should be noted that the simulation parameters obtained only apply for the asphalt used in this experiment. If other asphalt with different properties is employed, the formula could have different parameters. The comparisons of the foaming evolution of the foamed asphalt at other water contents (1% and 3%) predicted by the simulation model and the experimental results are shown in Figure 4.6 and Figure 4.7, respectively.

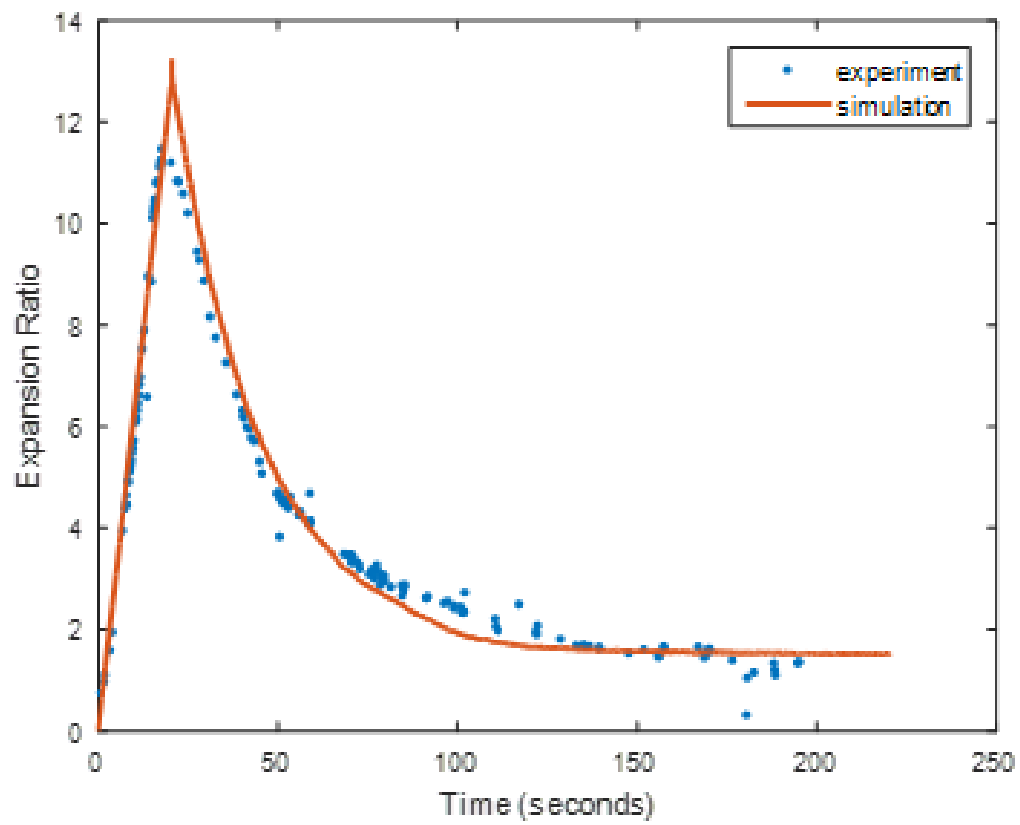


Figure 4.5: Comparison of the foaming evolution between experiment and simulation at water content of 2%

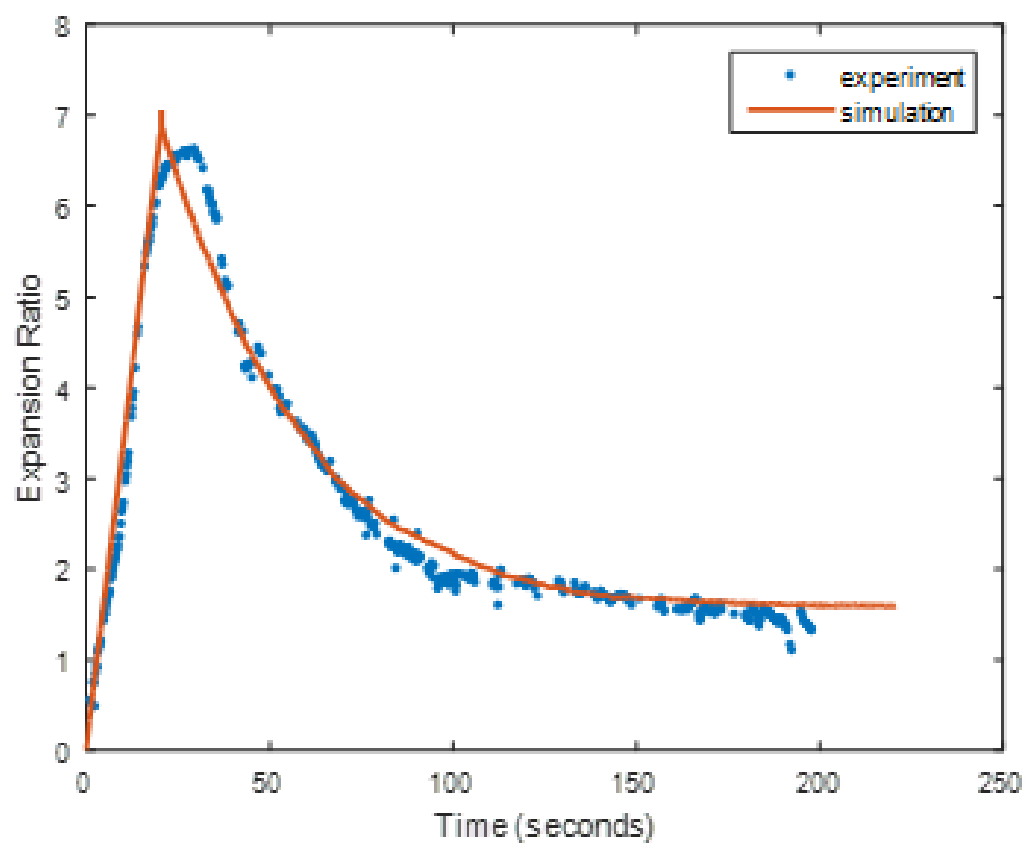


Figure 4.6: Comparison of the foaming evolution between experiment and simulation at water content of 1%

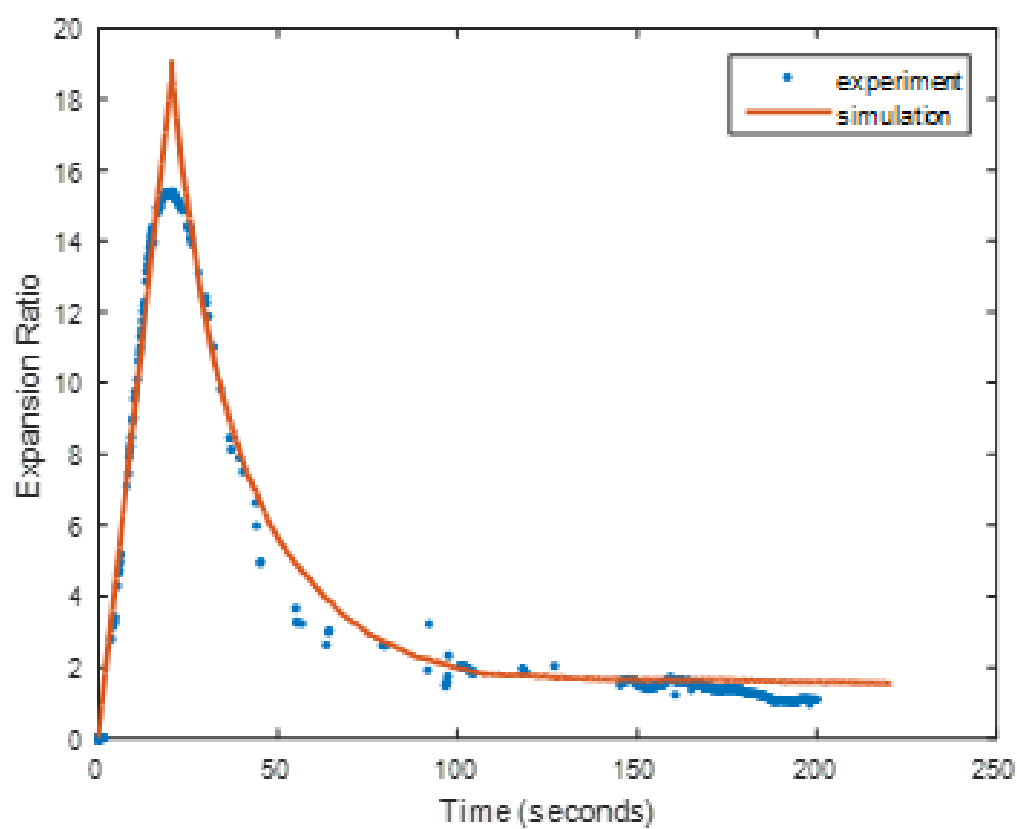


Figure 4.7: Comparison of the foaming evolution between experiment and simulation at water content of 3%

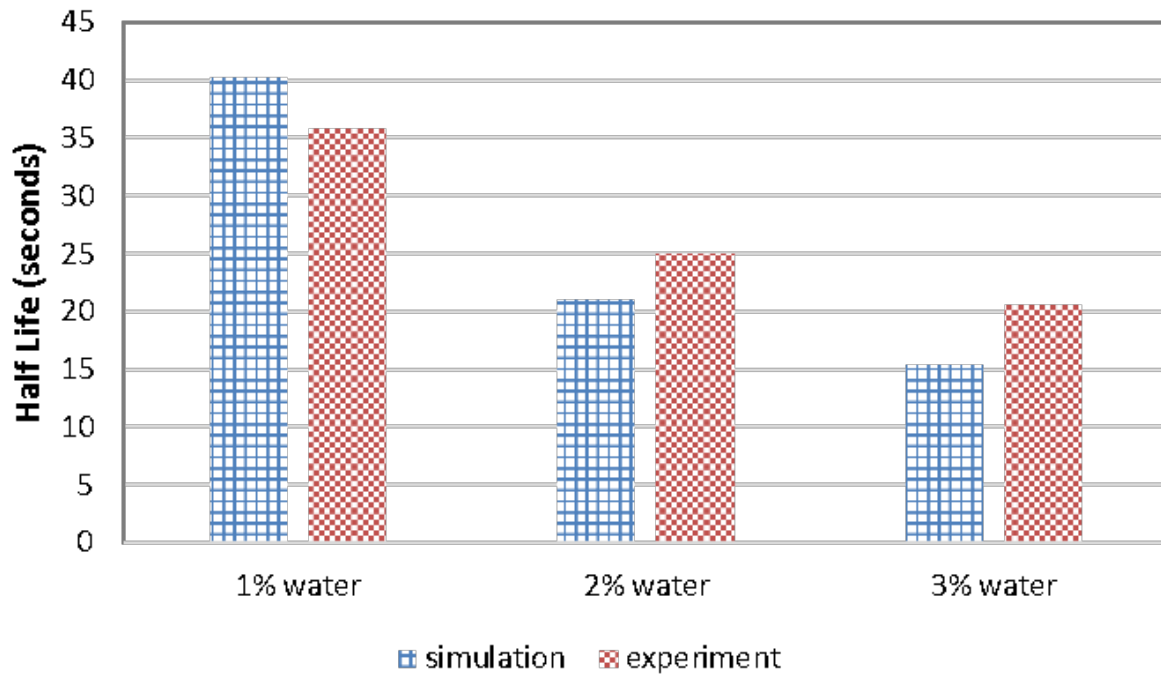


Figure 4.8: Comparison of HL of the foamed asphalt at different water contents

According to Equation 4.2 and 4.3, the other two foaming characteristics, HL and FI, of the foamed asphalt at different water contents can be determined, which are presented in Figure 4.8 and Figure 4.9, respectively. From the comparisons between the present simulation predictions and the experimental results as shown in Figure 4.4 to Figure 4.7, it shows the present simulation model is able to provide predictions of the foaming evolution, HL and FI of the foamed asphalt at different water contents.

4.3.3 Parametric Studies

It was observed from experiments that some experimental conditions such as the pressure in the function chamber, the temperature of the asphalt binder and the temperature in the function chamber would affect the foaming performance of the foamed asphalt. It is difficult to accurately investigate these influences through experimental studies as it is hard to maintain other parameters at a stable state while in evaluating one of them. Therefore, in this section, virtual experiments are conducted by using the validated simulation tool to investigate these influences on the foaming

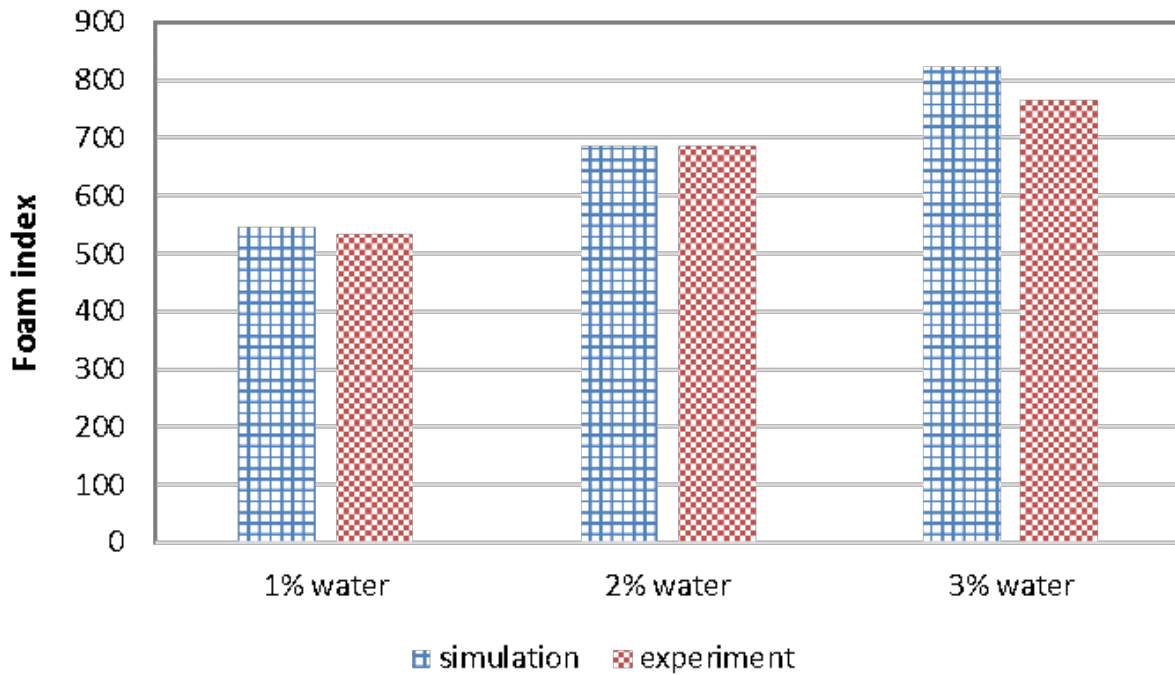


Figure 4.9: Comparison of FI of the foamed asphalt at different water contents

performance.

Effects of Pressure in the Function Chamber

Based on the ideal gas model, a lower pressure leads to a lower gas density and thus a higher ER. However, it requires that all the injected water evaporate in the function chamber and interact with the asphalt binder. In this respect, a higher pressure in the function chamber is helpful to attain a better interaction of the injected water with the asphalt. Figure 4.10 shows the prediction of foaming evolution under different pressures in the function chamber. In this case, all other parameters are kept constant, where the temperature of the asphalt and that in the chamber is kept at 150°C , the water content is 2% water. The predictions shown in Figure 4.10 indicates that the trend of the foaming evolution and HL of the foamed asphalt under different chamber pressures are almost the same, while the maximum ER experience a linear increase as the pressure in the function chamber increases.

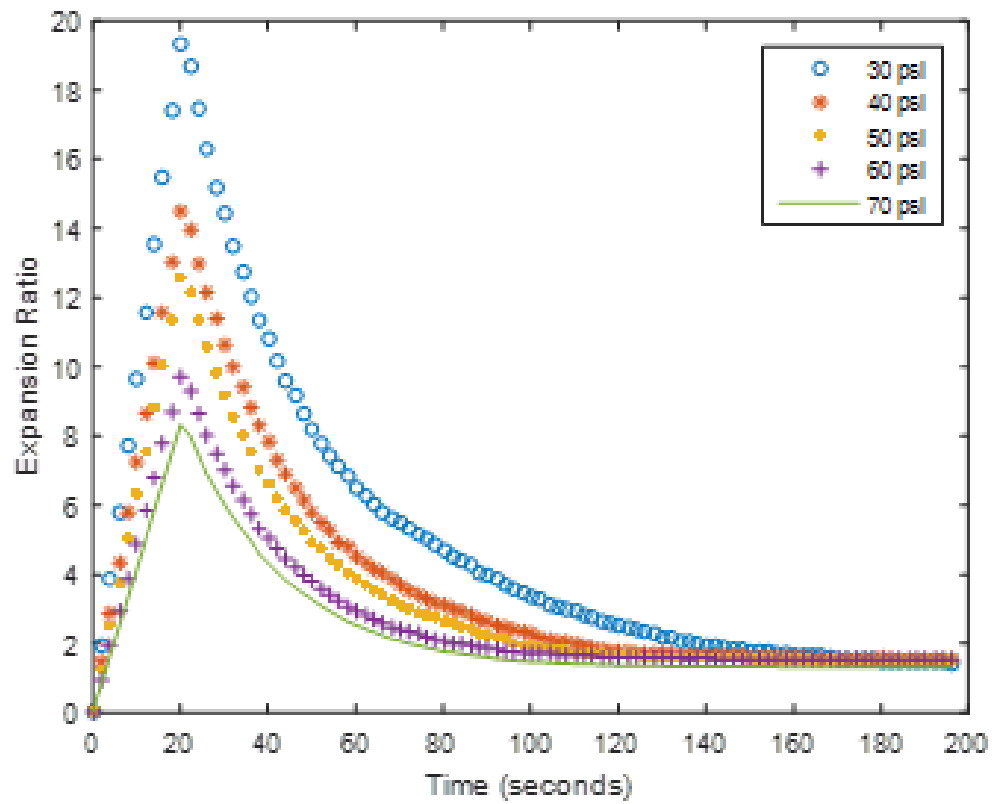


Figure 4.10: Effect of the pressure in the function chamber on the foaming evolution in the foamed asphalt

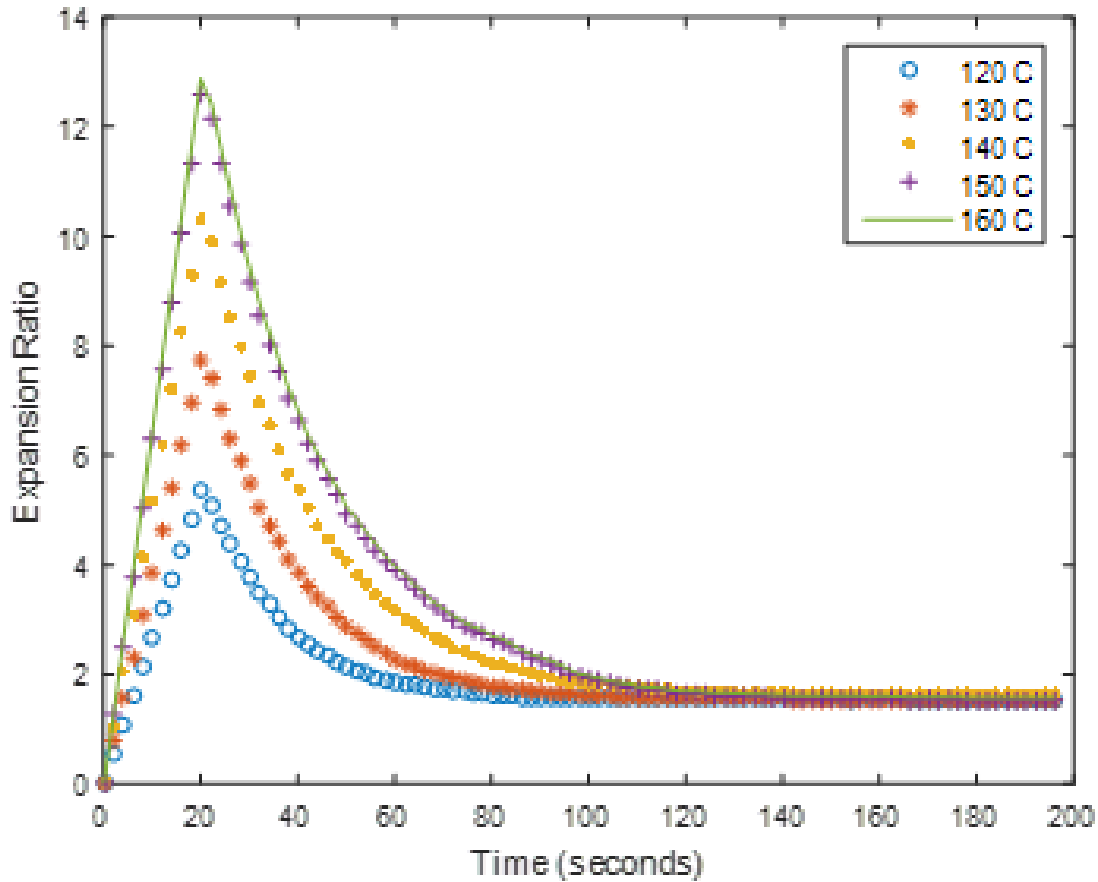


Figure 4.11: Effect of the temperature of the asphalt and the temperature in the function chamber on the foaming evolution in the foamed asphalt

Effects of Different Temperature

Since water evaporation absorbs a large amount of energy, the effect of asphalt temperature is investigated in this case study. Based on the adiabatic process assumption and Equation 4.5 and Equation 4.6, the predicted foaming evolution at different temperature is predicted in Figure 4.11. It shows that generally higher temperature leads to higher ER, while above certain level the influence of temperature becomes trivial. The simulation study showed an expected result based on experience, and more experiments will be conducted to compare with simulation so that a more accurate model could be generated.

4.4 Conclusions and Discussion

A numerical model using the smooth particle hydrodynamics (SPH) was applied in this study to simulate the asphalt foaming process. In this model, it was assumed that both the gas and liquid phases in the foamed asphalt are generated at the beginning by sharing with the same governing equation. Then the height of foamed asphalt decreases as the gas bubble collapse at a given decreasing frequency. An ideal gas theory was used to predict maximum bubble expansion ratio with initial accumulation period adaption, and a linear correlation of bubble collapse frequency with respect to different water contents was applied. A self-developed nozzle-based foamer was applied to generate foamed asphalt binder at different water contents. Three primary parameters, expansion ratio, half-life and foam index that widely applied to evaluate the foaming characteristics of foamed asphalt, were evaluated to study the foaming characteristics of the foamed asphalt.

The foaming evolutions of the foamed asphalt at different water contents were compared well with the experimental results, and thus the present numerical simulation model can capture the foaming evolution in the foamed asphalt. Parametric studies were further conducted by using the numerical model to evaluate the effects of environmental controlling parameters on the foaming characteristics of the foamed asphalt binder. The parametric studies indicated that: (1) the trend of the foaming evolution and HL of the foamed asphalt under different chamber pressures are almost the same, while the maximum ER experience a linear increase as the pressure in the function chamber increases; and (2) the temperature have an obvious influence on the foaming evolution and its corresponding foam characteristics.

In this work water is used as additive to generate foaming asphalt. In addition to water, ethanol could also be used to generate foaming asphalt, and it has lower evaporate point, which requires lower temperature and can save amounts of energy. Only 1% to 2% of ethanol is needed to generate ethanol-based foaming asphalt, and the evaporated ethanol could be reused. This amount of ethanol is safe and can be integrated into existing production process. The machine designed for water could be also used for ethanol with some minor changes.

The simulation tool is helpful to provide guidance in WMA, when choosing the parameters such as temperature, pressure and water content in a workable range. Some parameters in the simulation come from the physical properties such as density and viscosity, but bubble collapse rate is obtained from experiments. The bubble collapse rate is not only dependent on the material

type, but also related to the water content. Another parameter that is hard to obtain directly from physical property is bubble size. Since Expansion Ratio (ER) is the main focus of this work, the bubble size does not make a difference to the final result and it is set as the cutoff size, which is 1cm. These parameters are used to facilitate the simulation, and change with the physics of materials.

This work is the first step to use computer to conduct virtual experiments for the asphalt foaming process, where is very hard to directly measure inside of the foaming chamber. With a few parameters determined by lab tests, it reproduces the complex foaming process on the computer. It can be used by the industry, government agencies, and academia for design, development and optimization of asphalt materials and other composites. We look forward to using it in more applications of this tool in the design and development of novel asphalt materials.

Chapter 5

Mixing Index Analysis

From this chapter, the DEM-SPH coupling model developed in Chapter 3 is applied to multiphase high shear mixing processes, and simulations in various mixers are analyzed in detail.

During the mixing process, the particle distribution keeps changing until a relatively uniform distribution is achieved if possible. It is therefore necessary to quantify the mixing state. The ability to track particle mixing state is one advantage of simulation over experiment, and the measurement of mixing quality is crucial to better analyze and design mixing plan. In this chapter, a multiphase high shear mixing process in a four-bladed mixer is simulated. Three kinds of mixing indexes are proposed to quantify the mixing state, which include homogeneity index, randomness index and time series index.

There are two terms regarding the mixing states, ideally ordered mixture and perfectly random mixture [107]. Ideally ordered mixture refers to the mixture that each sample with the same configuration has the exact same composition. The probability of obtaining such a mixture in mixing is zero. Perfectly random mixture means a mixture has independent distributions in all its samples, and each particle has the same probability to appear in any part of the mixture. This work uses the perfectly random mixture as the reference standard to test the mixing quality.

5.1 Simulation Setup

After the successful demonstration of the method in the three case studies in Chapter 4, this DEM-SPH coupling method is applied to a solid-liquid mixing simulation process in a high shear

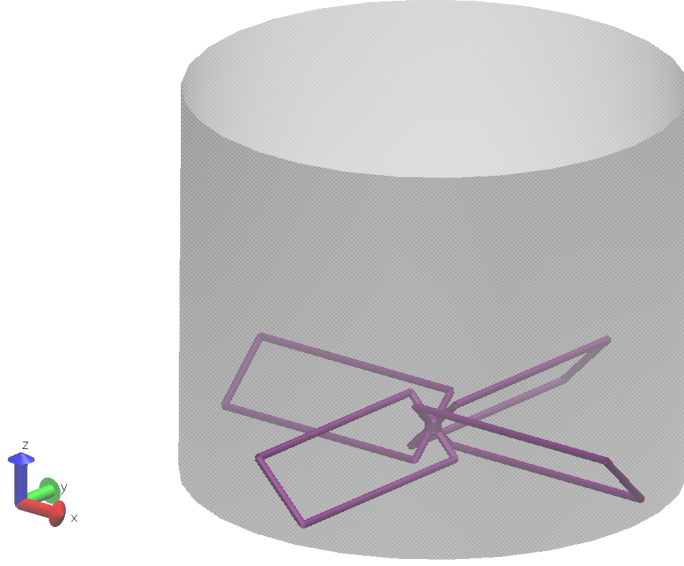


Figure 5.1: Snapshot of the high shear mixing simulation mixer

mixer with four blades. As shown in Fig. 5.1, in a cylinder mix container, four rectangular plane blades are placed at the bottom with uniform angles. The geometry of this mixer is listed in Table. 5.1. To visualize these four blades, particles are created along the boundaries of these blades. The interaction of particles with the four blades is the same as their interaction with the cylinder container, which is Linear Spring-Dashpot (LSD) method with given stiffness and dashpot parameters.

In the simulation conducted below, two kinds of solid particles and one kind of liquid particles are employed to investigate the mixing performance of solid-liquid two-phase flow. Their properties are listed in Table. 5.2. As mentioned before, the sound speed here is chosen as $30m/s$ to increase the time step, while maintaining a low Mach number [122; 81] (The velocity at the tip of the blades is around $2m/s$, and the maximum velocity of particles could not exceed that, so the Mach number is below 0.1).

Prior to mixing, the solid particles are classified into two groups (the orange and tan colors, respectively), with each group filling half side of the container, as shown in Fig. 5.2. The two types of particles are first created and then released to fill up the space in the container, so there are a

Table 5.1: Geometric parameters of the high shear mixer

Parameter	Value
Vessel diameter	400 <i>mm</i>
Vessel height	400 <i>mm</i>
Blade length	192 <i>mm</i>
Blade height	45 <i>mm</i>
Blade rake angle	135°
Rotation speed	10 <i>rad/s</i>
Simulation time	20 <i>s</i>
Time step	5×10^{-5} <i>s</i>

few particles scattered into the other type of particles during the filling process.

Then liquid particles are generated at the top of these solid particles. After these particles fill up the mixing container and reach stable, the blades start to rotate with a constant acceleration rate set at the beginning until a constant given velocity is reached. Then the two groups of solid particles together with liquid particles are mixed towards homogeneity at a constant velocity until the mixing process stopped. In the following parts, the mixing process is analyzed quantitatively in various mixing indexes.

5.2 The Mixing Index

5.2.1 Homogeneity

Centroid difference

The centroid measurement based on particles' spatial information history. All particles will be divided and grouped into different cells based on their initial positions, and one group of particles are marked as trace particles to track their centroid position. From an experimental perspective, different groups of particles will be assigned with different color, hence, particles segregation and mixing can be easily observed [86]. From a simulation perspective, each cell has its own centroid and the distance of the centroids from one group to the global center quantitatively characterizes

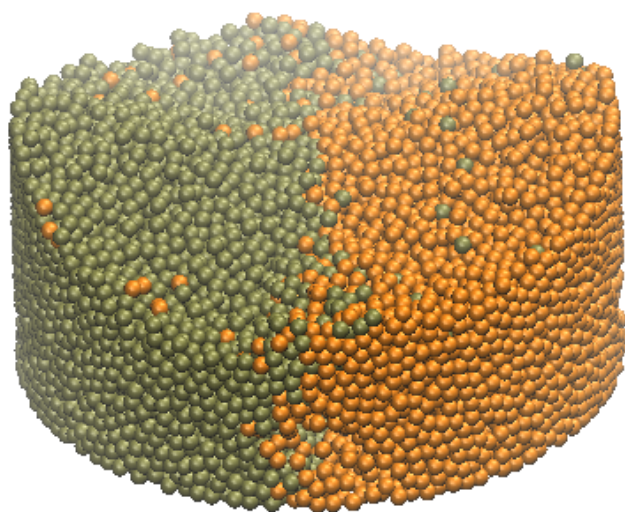


Figure 5.2: Initial state of mixing problem

Table 5.2: Parameters of solid and liquid particles in high shear mixing simulation

Parameter	Solid
Diameter	10mm
Density	1000kg/m ³
Mass	0.5236g
Number	19440
Normal Stiffness	$7 \times 10^7 N/m$
Tangential Stiffness	$2 \times 10^7 N/m$
Normal Damping Coefficient	8000N · s/m
Tangential Damping Coefficient	4000N · s/m
Coefficient of restitution	0.8
Friction coefficient	0.5
Parameter	Liquid
Diameter	10mm
Density	1000kg/m ³
Mass	1g
Number	1060
speed of sound	30m/s
viscosity coefficient	0.2m ² /s

the degree of mixing of that group of particles. At the beginning, they are well segregated. Ideally, if the particle mixing is perfect, the distance between the centroid of one group to the overall center should decay to zero. Although centroid measurement method is very convenient, the homogeneity is limited to particle position only. Certain scenario with bad mixing mixture but zero centroid difference could happen. Such scenario can be illustrated in Fig. 5.3. Apparently, with the pattern shown at right-upper picture, centroid difference fails.

Application of centroid difference in mixing simulation

The centroid difference measurement is applied to test the homogeneity in case study mentioned in Section. 5.1. At the initial state of the mixing process, all the solid particles of type 1 in region

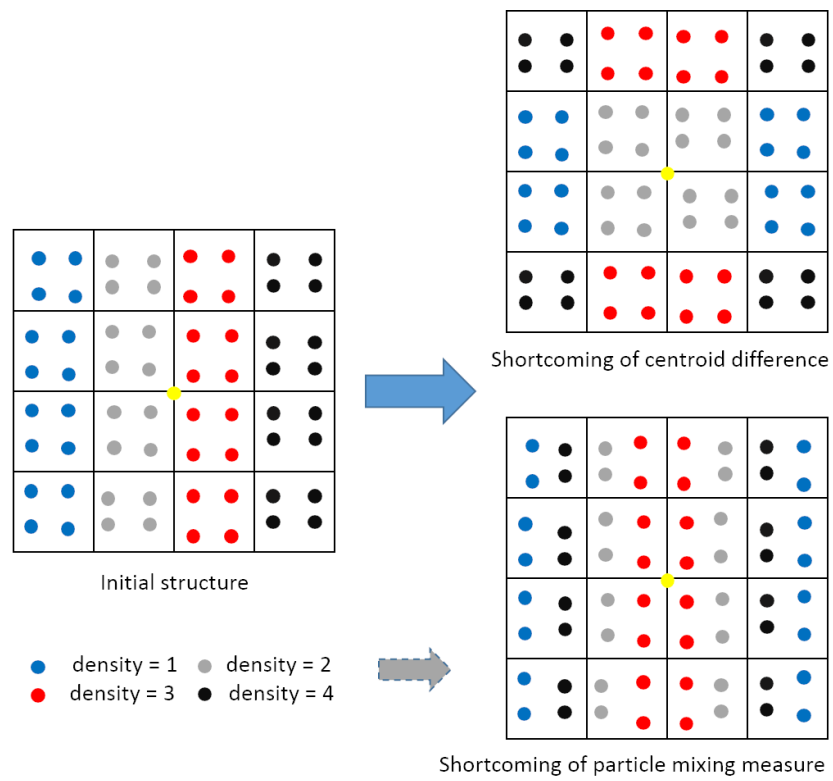


Figure 5.3: Schematic illustration of drawbacks of centroid and local average property measurement methods

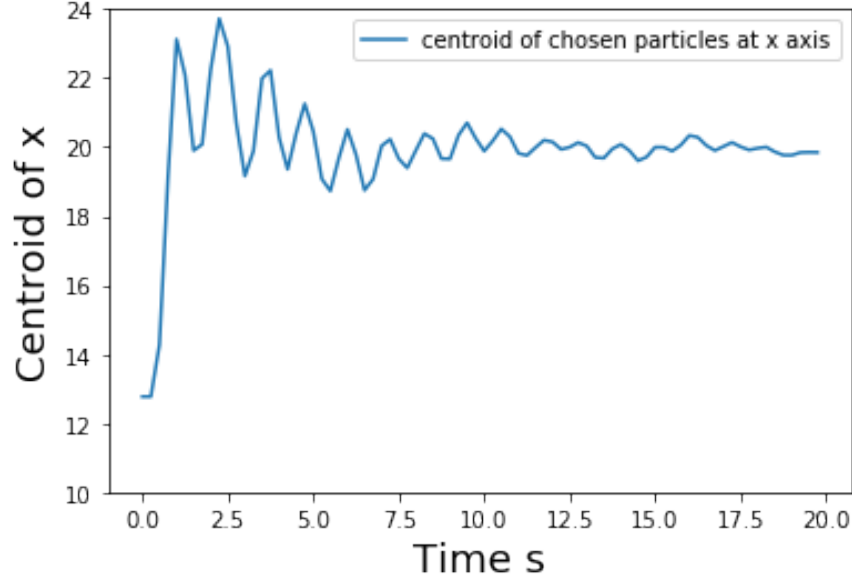


Figure 5.4: Centroid of trace particles at x axis

$x < 20mm, y < 20mm$ are chosen as trace particles to compute their centroid during the mixing process. Fig. 5.4, Fig. 5.5 and Fig. 5.6 show the centroid of these particles at x, y and z directions. If all the particles are mixed perfectly, the centroid of these chosen particles will be the same as the centroid of all particles in the system, which is the centroid of the simulation domain in this case by symmetry. It is demonstrated in Fig. 5.4 and Fig. 5.5 that the centroid of these trace particles approaches to the center of the mixing container ($x = 20mm, y = 20mm$). For the centroid at z direction, it is hard to tell where the expected position is in this case. It is found in Fig. 5.6 that the centroid at z direction first increases from 6.5 to around 9, maintains a value between 8.5 and 9 during the mixing process, and drops back to 6.5 at the end. The reason for this movement is that the rotating blades lift up these trace particles during the mixing process, so that their centroid at z direction increases during this period.

To quantify these results in centroid difference computation, the theoretical distribution of centroid at uniform distribution is calculated and compared with the simulation results. Assume particles could appear anywhere in the cylinder mixer with radius R , then the standard deviation for each particle in both x and y direction is calculated as $R/2$. Then if there are n particles chosen to trace the centroid, based on Central Limit Theory, the centroid of these trace particles in both

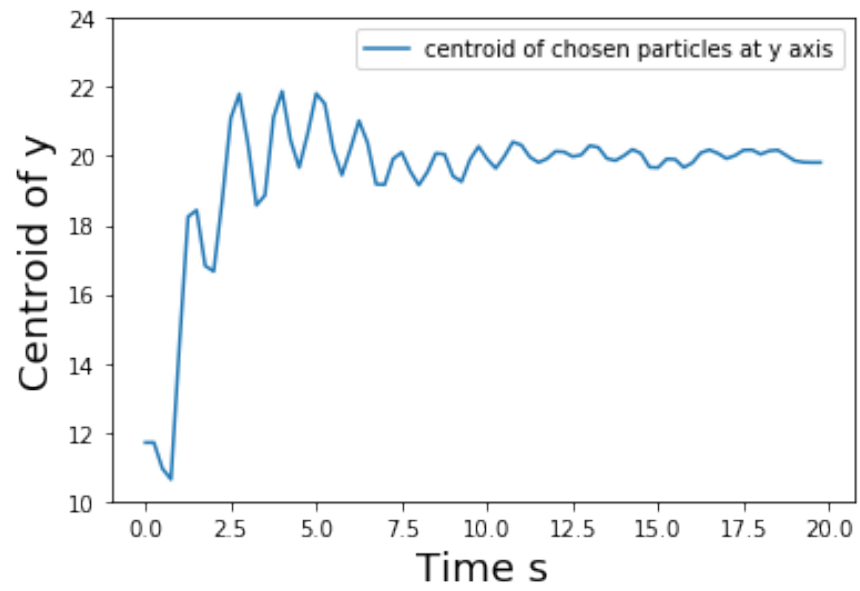


Figure 5.5: Centroid of trace particles at y axis

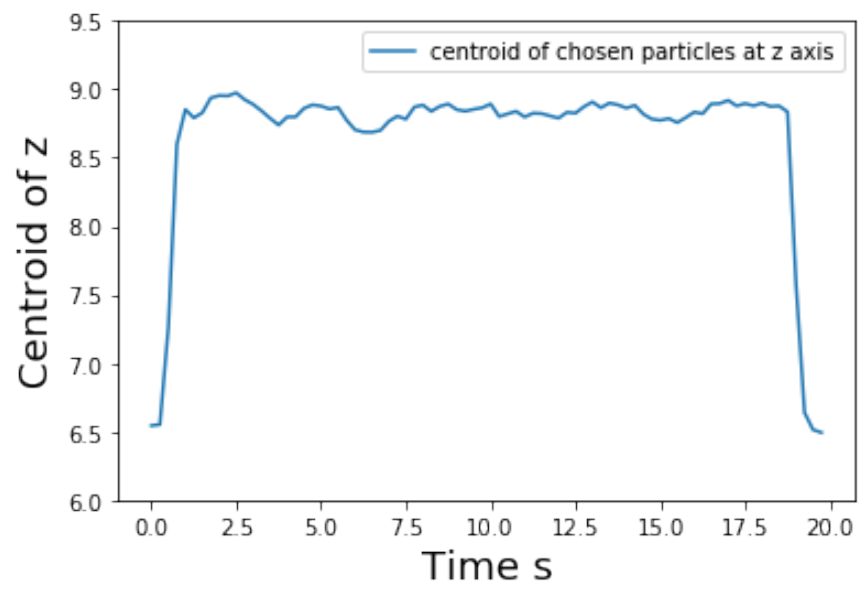


Figure 5.6: Centroid of trace particles at z axis

x and y directions follows normal distribution $N(R, \frac{R^2}{4n})$. The detail proof is provided in Sec. 5.3. In this case, $R = 20$, $n = 2430$, the mean value of centroid is 20, and the standard deviation is calculated as 0.29. In the simulation, the differences between the center of the mixer and the centroid of trace particles are 0.15, 0.18 at x axis and y axis separately, smaller than one standard deviation, which means the mixture is mixed uniformly at x and y direction based on this centroid difference index calculation.

Centroid difference is a convenient way to quantify the mixing quality, but it is also a rough method, in which there are obvious oscillations in the curve as shown in Fig. 5.4 and Fig. 5.5. Centroid difference method describes the mixture in a global view, but lack information of details in the mixture, so the convergence of centroid difference is necessary for homogeneous mixture, but not sufficient.

Local average property measurement

In this work, in addition to centroid difference, the local average index is implemented to test the homogeneity of the mixture. In this method, a 3D grid will be constructed over the chosen domain. The size of each cell in the 3D grid, i.e. the Averaging Box (AB) should be at least five times of the average particle diameter[29] so that the local averages of a selected property (e.g., mass, density, diameter, etc.) are statistically meaningful. A value such as number density of certain particle (the number of particles in a constant space) is chosen to compute as local average value. Then the local average values will be calculated using all the particles within the cell, so that the probability distribution for the local averages can be constructed. Afterwards, the mean, standard deviation, and coefficient of variation η (defined as the standard deviation divided by the mean) will be computed over the entire domain of local cells at each time step. In this work, the number density of certain type of particles in each AB is chosen as the target value to test, and its value in each AB and its variance among all ABs are calculated. If the mixture is homogeneous, the number density of any type of particles would be similar in each AB and the variance among all ABs is low. Two ideal limits will also be calculated for η for the fully segregated η_s and fully random mixed states η_r , and the calculation for these two values are illustrated in Sec. 5.3. Then η can be normalized to the range of 0 – 1 to characterize the quality of particles mixing which is expressed

as

$$\xi = (\eta - \eta_r)/(\eta_s - \eta_r) \quad (5.1)$$

where $\xi = 0$ indicates a fully mixed states; while $\xi = 1$ suggests a fully segregated state. In this work, ξ is referred as Mixing Index. Both η and ξ are useful to quantify the mixing process, in which η embodies the absolute value of homogeneity, while ξ demonstrates the relative value of homogeneity. The local average index is not only a good indicator of the homogeneity of the final mixture, but also a natural measure of the mixing rate. Moreover, this method is very effective in the experiment perspective. Depending on the sizes of the particles in the mix design, different ABs could be chosen to test the mixing quality.

The choice of Averaging Box

In previous chapters, local average homogeneity matrix has already been applied to the analysis of mixing performance. To apply the local average homogeneity matrix, Averaging Box must be chosen first [26]. Averaging Box (AB) is a sample volume within the simulation domain, which is used to analyze the feature of the particles inside the box [26; 29]. Choosing the correct size of AB is critical to the analysis of experiment and simulation [51; 67]. The size of AB could not be so small, because small AB may not contain enough information of the system; it can not be larger either, otherwise the total number of ABs is not enough to analyze the variance between these ABs. Usually the size of AB is chosen in a way that each AB contains 50-100 particles [29]. The size of AB should correspond to the use of the mixture, and it should be comparable to the size of the mixture used in one single unit, such as one tablet in pharmaceutical product.

In this work, the size of AB is chosen with similar standard, so that around 50-100 particles are contained in each AB. Larger AB is also applied to compare with smaller AB to demonstrate the effect of the size of AB.

Simulations results comparison with different sizes of Averaging Box

To investigate the effect of the size of AB, three different sizes of AB, $30mm \times 30mm \times 40mm$, $50mm \times 50mm \times 53.3mm$, $100mm \times 100mm \times 80mm$ are also applied. The details of the distributions

Table 5.3: Distribution of Averaging Box at three different sizes

Averaging Box	Small	Medium	Large
Length on X Axis	30mm	50mm	100mm
Length on Y Axis	30mm	50mm	100mm
Length on Z Axis	40mm	53.3mm	80mm
Start Point on X Axis	50mm	50mm	50mm
Start Point on Y Axis	50mm	50mm	50mm
Start Point on Z Axis	10mm	10mm	10mm
Numbers on X Axis	10	6	3
Numbers on Y Axis	10	6	3
Numbers on Z Axis	4	3	2
Total Numbers	400	108	18

of these ABs are listed in Table. 5.3. Their coefficients of variation are shown in Fig. 5.7, and the mixing indexes are shown in Fig. 5.8.

In Fig. 5.7 and Fig. 5.8, three curves in different colors represent three different sizes of AB. In general, they show a similar trend with the following features:

1. The value of coefficient of variation η decreases with the size of the AB. It is clearly seen that in Fig. 5.7, the red curve that represents the smallest AB lies on top of the three curves, and the yellow curve that represents the largest AB lies on the bottom of the figure. This phenomenon means that the mixing quality of the mixture really depends on how one observes the mixture. When the AB size is larger, it means that the whole mixture is observed in a coarse scale, so it is easier to be mixed well on a coarse scale. When it goes to fine scale, the same mixture that looks uniform on coarse scale may not be mixed well in this scale. This observation implies that before determining whether a mixture is homogeneous, the standard of resolution must be set first. The results of these mixing indexes in this simulation demonstrate the homogeneity of the mixture in a specific scale, and the scale is the size of the AB.
2. All the three curves move simultaneously and reach a constant value at the same time in

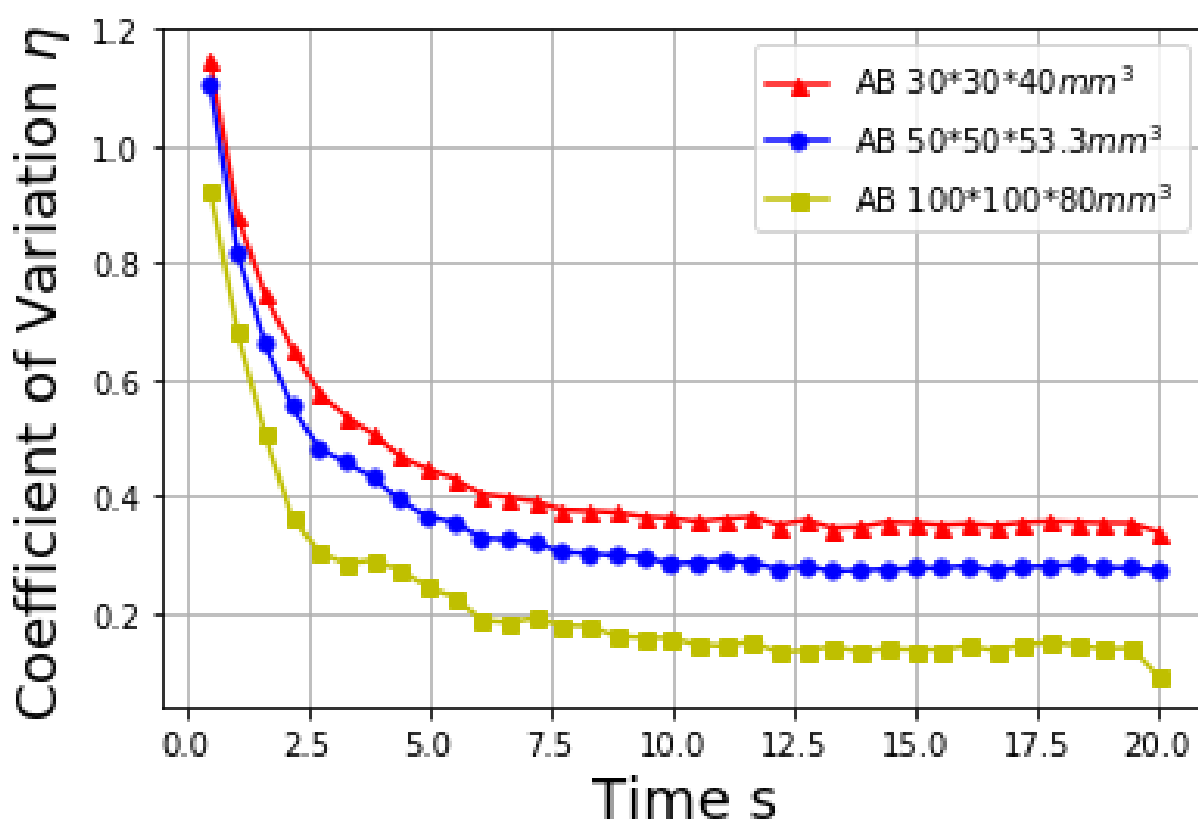


Figure 5.7: Coefficient of variation at different AB

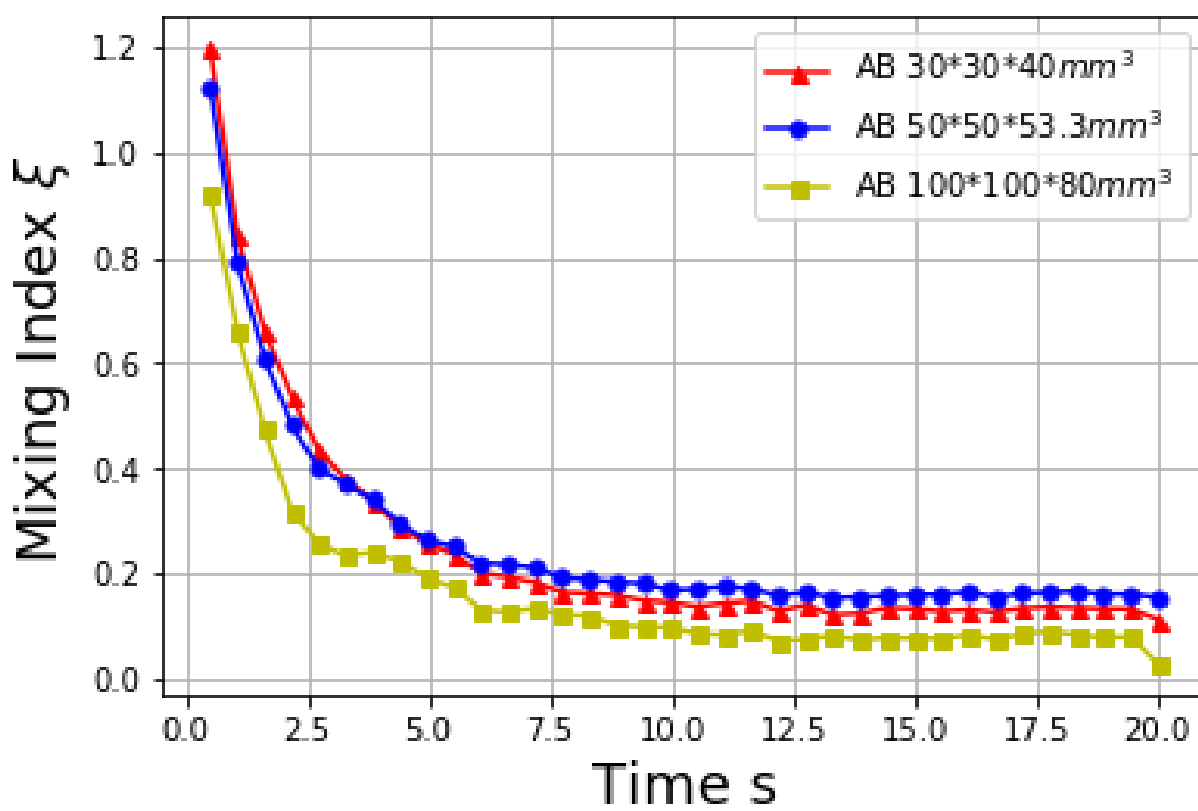


Figure 5.8: Mixing indexes at different AB

both Fig. 5.7 and Fig. 5.8. This means that the mixing process continues in all scales, and as long as the mixture is mixed towards homogeneity, the mixing indexes will decrease in all scales. In Fig. 5.8, it is found that the mixing index ξ of these three curves have similar shape and reach similar values at the end, which demonstrates that the mixing process in different scales continues with consistent speed.

3. It is noted that the curves of mixing index in Fig. 5.8 are much more closer to each other than the curves in Fig. 5.7. It is because the coefficients of variances are normalized by standard values to obtain the mixing index, and mixing index shows the general mixing quality that does not have too much relation with the size of AB. Applying mixing index to analyze the mixture could reduce the influence of AB size.
4. Generally speaking, there is a goal of standard to achieve for the mixing process, and the size of AB should be chosen according to the desired resolution. Thus, the selection of AB should also reflect the use of the mix. For example, if the use is for medicine, then the size of a capsule is a good one for AB.

5.2.2 Randomness

Randomness is an important concept in particle dynamics. It means that the position distribution of particles is random, in which each particle has the same possibility to appear anywhere, so the ordered packing structures such as Face Centered Cubic(FCC) or Simple Cubic(SC) are not likely to appear in. One random order metric is implemented in this work, which is orientational order.

Orientalional Order

Orientalional order was originally proposed by Steinhardt [127] to study the bond-orientational order in molecular-dynamics simulations of supercooled liquids and in models of metallic glasses. It was wildy used later in the research of sphere packing. Rintoul and Torquato gave a series values of Orientalional order in various crystal structures [111] and Torquato [134] employed the Orientalional order to quantify the order of sphere packing in a jammed structure. Orientalional order has been proved reasonable in detecting crystal structruers, and can be extended to test the order or randomness of general structures.

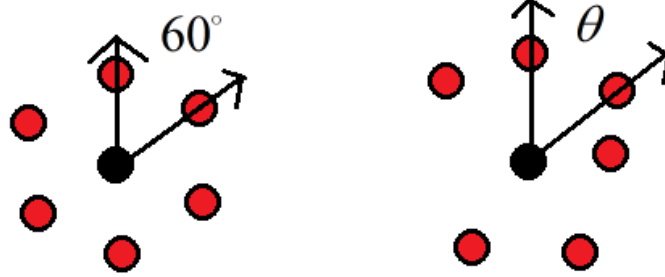


Figure 5.9: Orientational Order

For every particle, the Orientational order (usually denoted as Q value) measures the angles between the bonds of its nearest neighbor particles, which are calculated using the spherical harmonics to get the final bond-orientational order metric. Normalized by the Q value of the Face Centered Cubic (FCC) structure, the magnitude of Q reaches the largest value of 1 when the structure is FCC, and reaches the smallest value of 0 when the microstructure is perfectly random. The equation to compute orientational order is shown below:

$$Q_6 = \left(\frac{4\pi}{13} \sum_{m=-6}^6 \left| \frac{1}{N_b} \sum_{i=1}^{N_b} Y_{6m}(\theta_i, \phi_i) \right|^2 \right)^{\frac{1}{2}} \quad (5.2)$$

$$Q = \frac{Q_6}{Q_{6,FCC}} \quad (5.3)$$

where N_b is the number of neighbors for a particle, and $Q_{6,FCC}$ is the Q_6 value for Face Centered Cubic (FCC). For a particle P, all the particles within 1.5 times the distance of the particle P's nearest neighbor particle are neighbors of the particle P. $Y_{6m}(\theta, \phi)$ are the spherical harmonic functions, and their corresponding values could be easily get in relevant textbook.

Simulation Study

Orientational Order is applied to mixing process in this part, and the results are shown in Fig. 5.10 for the first 3 seconds. It is clear that this value drops immediately at the beginning, and maintains a stable low value during the process.

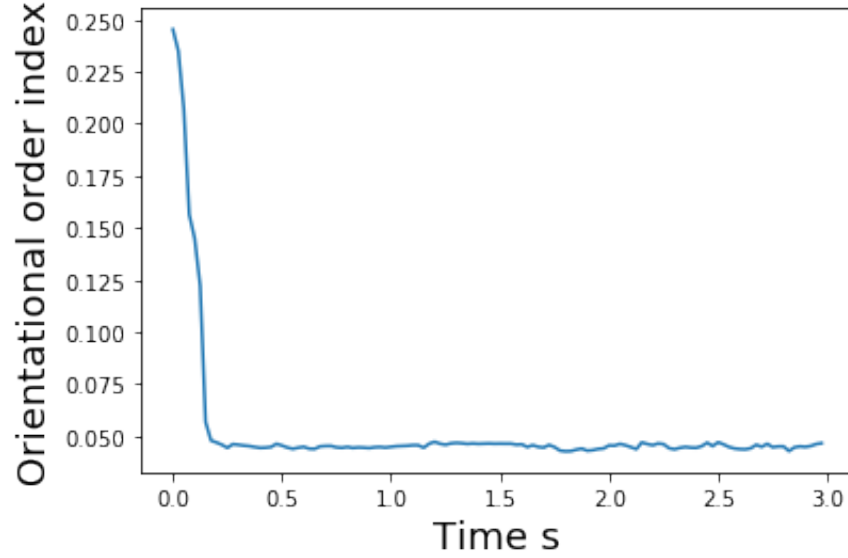


Figure 5.10: Orientational order index during the mixing process

At the beginning, all the particles are generated in an ordered way, so this value is high at the beginning. Then all the particles fall in their position and the initialization is finished. This curve means that the mixture achieves random distribution at the beginning stage, after all the particles fall in their position, and then it is always random during the mixing process. It is obvious that the mixture is far from homogeneous at the beginning, and this index does not show any meaningful movement during the mixing process.

As described before, randomness is about the way particles are packed, and it is not related to the homogeneity of the mixture. During the mixing process, all the particles are moving constantly and it is impossible to let particles packed with order, so it is randomly distributed during the whole mixing process, and this randomness index does not provide any information about the mixing quality. This method is analyzed in this part, and the solution is that it is not an ideal index to test the mixing quality.

5.2.3 Time Series Analysis

Previous methods measure the current status of particle system and determined the mixing quality. [55] developed a new order metric to calculate the time series of the measurement. The measurement

can be density or something else that can be easily measured.

$$P_k = \frac{\sum_{j=1}^{m-k} (s_j - \bar{s})(s_{j+k} - \bar{s})}{\sum_{j=1}^m (s_j - \bar{s})^2} \quad (5.4)$$

where \bar{s} is the mean of this measurement, s_j and s_{j+k} are the values of this measurement at time j and $j + k$.

An Averaging Box (AB) is chosen in the system, and time series is used here to measure the volatility of the system. A short time interval is needed to get the time series.

This equation calculates the covariance between two time series with time lag k , where m is the present time, and k should be smaller than $m/4$. Based on time series theory [19], if the system is completely random, the value P_k calculated above should obey normal distribution $N(0, \frac{1}{m})$, and lie between $(-\frac{2}{\sqrt{m}}, \frac{2}{\sqrt{m}})$ with 95% confidence. The details of this proof is given in Sec. 5.3.

Time series analysis is applied to analyze the mixing process. The AB size $30mm \times 30mm \times 40mm$ is set and three ABs are selected to compute the particle portion (percentage of certain number of particles among all the particles) of one kind of solid particles in this AB. The particle portion of the first kind of solid particles is calculated in three different ABs which located at $x = 20mm, y = 20mm, z = 9mm$, then $x = 26mm, y = 26mm, z = 9mm$ and $x = 32mm, y = 32mm, z = 9mm$ separately. These three ABs are at the same height, and their horizontal distributions are shown in Fig. 5.11. These time series values are shown in Fig. 5.12, Fig. 5.13, and Fig. 5.14.

The mean and standard deviation of these number portion series are analyzed in Table. 5.4. It is noted that at the center AB, the solid portion is lower than other positions, and this is consistent with previous observations in Chapter 5, because the rotation of the blades makes the liquid sink at the center. The volatility of the number portion is lower at the rim AB of the simulation domain, and this implies that the mixing status of particles away from the center is less likely to change, compared to the particles at the center AB or middle AB part of the container.

Then these time series are analyzed based on the Equation. 5.4. To analyze the mixing process, 400 snapshots are used in this case, which means $m = 400$, and time lag $k = 1$ and $k = 10$ are tested separately for these ABs after $t = 10s$. The results of these time series analysis are shown in Table. 5.5. Based on the Equation. 5.4, if the mixing status is stable, the value should be in

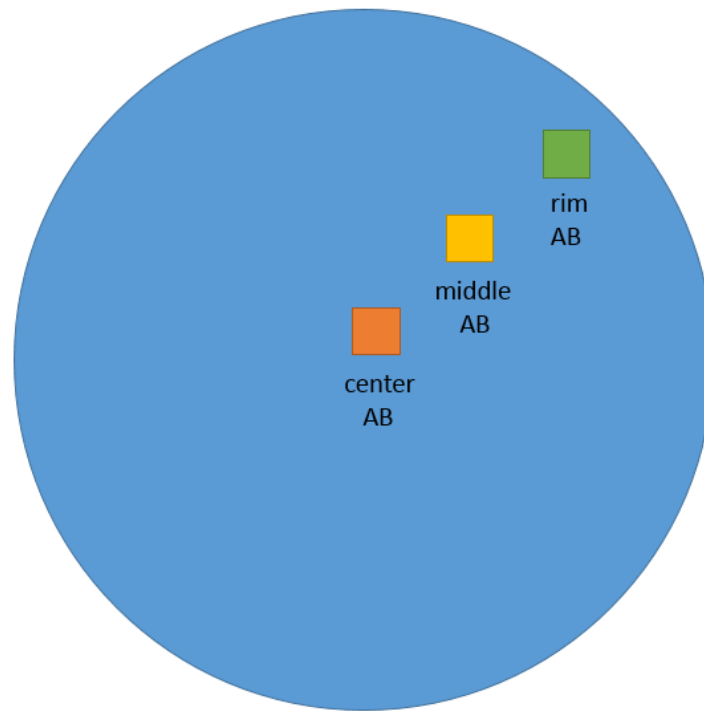


Figure 5.11: Choice of Averaging Box (AB) in the simulation domain

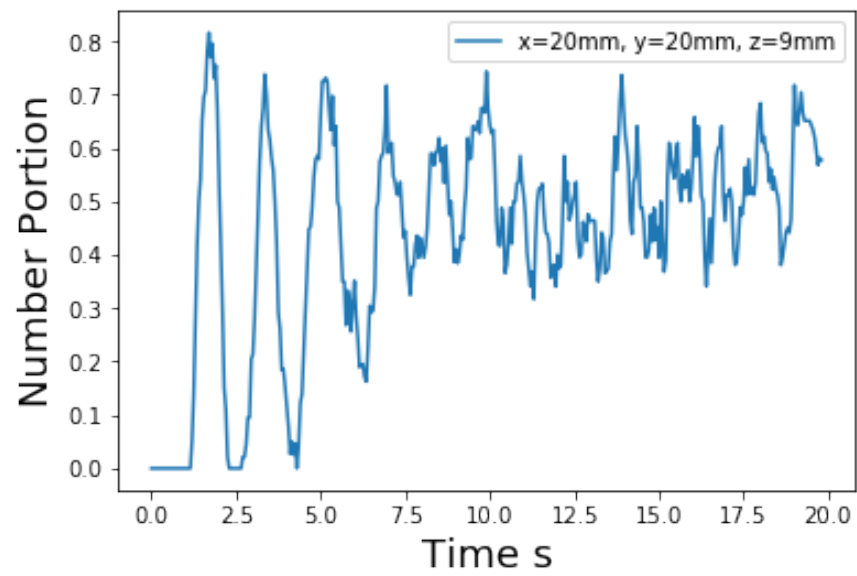


Figure 5.12: Number portion of first type solid particle at the center AB

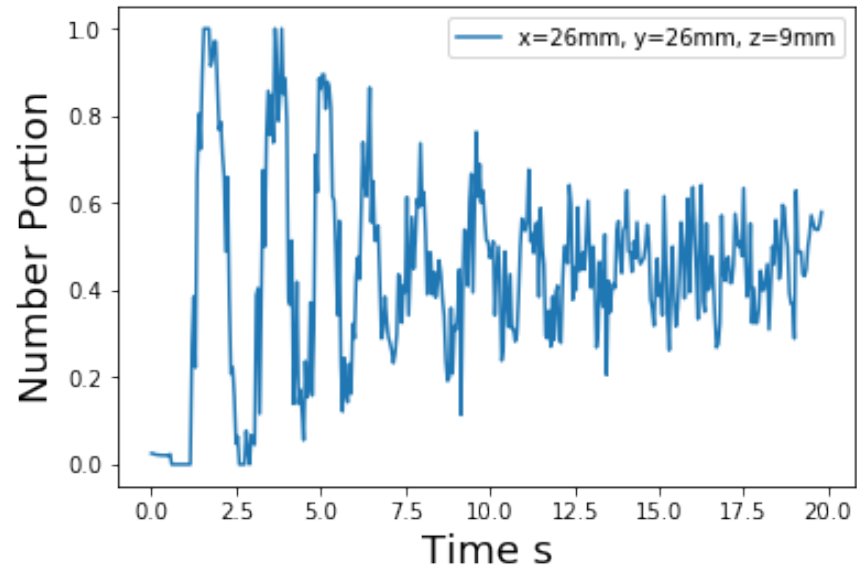


Figure 5.13: Number portion of first type solid particle at the middle AB

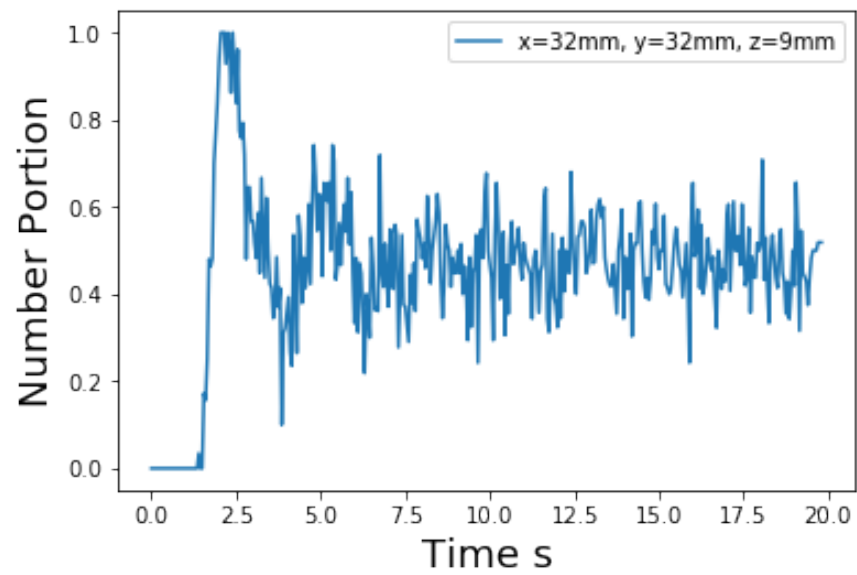


Figure 5.14: Number portion of first type solid particle at the rim AB

Table 5.4: Mean and standard deviation of number portion in chosen ABs

AB	mean of number portion	variance of number portion
center	0.439	0.200
middle	0.461	0.200
rim	0.457	0.185

the range $(-0.1, 0.1)$. And it is noted that all values in Table. 5.5 are in this range except the one for center AB when $k = 1$. If the cyclical rotation of the blades is considered and a larger k is chosen, then this value is also in the range. These values demonstrate that after $t = 10s$, the mixing process does not make any significant progress towards homogeneity.

Table 5.5: Time series analysis of number portion in different ABs after 10s

AB	time lag $k = 1$	time lag $k = 10$
center	0.142	0.046
middle	0.040	-0.019
rim	0.021	0.012

On the other hand, when this time series is stationary, it does not mean the mixture is already uniform. It could be that the distribution of particles stays still and nothing happens to mix it better. This index is useful to check the amplitude of the change in the mixture, but is not able to illustrate the homogeneity of the mixture at current status.

5.3 Criteria Analysis

In this section, the detailed criteria to determine random mixing is examined for centroid method, local average method and time series analysis method.

5.3.1 Centroid Method

In this part the proof of a statement at Section. 5.2 is given. The statement is that at uniform distribution inside a cylinder container with radius R , the centroid of n trace particles at x or y

axis follows the normal distribution $N(R, \frac{R^2}{2n})$.

By symmetry, the mean of the centroid position should be at R. The first step is to prove that for each single particle, the variance of the distribution of its position is $\frac{R^2}{2}$. It is assumed that at uniform distribution, each particle has the same probability to appear in anywhere within the cylinder container. Assume x represents the relative position of the centroid of any trace particle to the center of the cylinder, then its variance is expressed below:

$$Var(x) = E(x^2) - E(x)^2 = \frac{\int_0^{2\pi} \int_0^R \rho^2 \rho d\rho d\theta}{\pi R^2} - 0 = \frac{\int_0^{2\pi} d\theta \int_0^R \rho^3 d\rho}{\pi R^2} = \frac{2\pi \frac{R^4}{4}}{\pi R^2} = \frac{R^2}{2} \quad (5.5)$$

The next step is to calculate the variance for n trace particles. It is assumed that these n particles are independent, and for each particle its centroid to the center of the mixing container x follows the distribution $x \sim N(0, \frac{R^2}{2})$.

Central Limit Theorem claims that if x_i is a series of independent and identically distributed random variables which has expected value μ and finite variance σ^2 , and their sample average is defined below:

$$S_n = \frac{\sum_{i=1}^n x_i}{n} \quad (5.6)$$

Then

$$S_n \sim N(\mu, \frac{\sigma^2}{n}) \quad (5.7)$$

Recall $\mu = 0$, $\sigma^2 = \frac{R^2}{2}$ in this case, then

$$S_n \sim N(0, \frac{R^2}{2n}) \quad (5.8)$$

With the value $R = 20$, $n = 2430$, the variance and standard deviation of S_n are calculated as 0.0823 and 0.2869.

5.3.2 Local Average Method

In this part, the theoretical value for coefficient of variation η (standard deviation divided by mean) in fully segregated states η_s and fully mixed states η_r are calculated.

Fully segregated state is defined as the state that given the Averaging Boxes (AB), particles distribute on half of the ABs uniformly, while the other half ABs are empty. Assume there are m ABs and half the ABs have n particles each while the others half have 0, then

$$\eta_s = \frac{n/2}{n/2} = 1 \quad (5.9)$$

On the other hand, for fully random mixed states, it is assumed that each particle has the same possibility to appear in any of the ABs.

Assume there are n particles, m ABs and the volume ratio between all the particles and one AB is M , and the number of particles in each AB is denoted as x .

When there is only one particle, if the distribution is uniform, the expected number of particles in each AB is $\frac{1}{M}$ and the variance is $\frac{M-1}{M^2}$. Then for n particles, based on Central Limit Theorem, the value x in each RVE follows the distribution

$$x \sim N\left(\frac{n}{M}, \frac{(M-1)n}{M^2}\right) \quad (5.10)$$

Then consider $Var(x)$,

$$Var(x) = \sum_{i=1}^m \left(x - \frac{n}{M}\right)^2 = \frac{(M-1)n}{M^2} \chi_m^2 \quad (5.11)$$

where χ_m^2 is Chi Square distribution with m degree of freedom. In the simulation $n = 9720$, $m = 400, 108, 18$ separately. Assume the height of the mixture is 140mm in the simulation, then $M = 651, 176, 29$ separately. Referring to the Chi Square table, the 95% percent confidence value for χ_m^2 are 447.8, 133.6 and 29.2. Then the 95% percent confidence value for

$$\eta = \frac{std(x)}{E(x)} \quad (5.12)$$

would be 0.25, 0.14, 0.06 separately.

If the structure is homogeneous, then there is 95% possibility that the coefficient of variance is below this value. This value could be used as a reference to determine whether a structure is homogeneous and is applied to the calculation of mixing index in Equation. 5.1.

5.3.3 Time Series Analysis

Recall Equation. 5.4 (which is also listed below) in Chapter. 5.

$$P_k = \frac{\sum_{j=1}^{m-k} (s_j - \bar{s})(s_{j+k} - \bar{s})}{\sum_{j=1}^m (s_j - \bar{s})^2} \quad (5.13)$$

In this part, the statement that P_k calculated above obeys normal distribution $N(0, \frac{1}{m})$, and lie between $(-\frac{2}{\sqrt{m}}, \frac{2}{\sqrt{m}})$ with 95% confidence is proved below.

Assume the time series s_i is a random process with constant mean \bar{s} and constant variance σ^2 , and $s_j - \bar{s}$ obey normal distribution $N(0, \sigma^2)$. Also assume s_{j+k} and s_j are independent, then

$$E(s_j - \bar{s})(s_{j+k} - \bar{s}) = 0 \quad (5.14)$$

And the variance of this term is

$$Var(s_j - \bar{s})(s_{j+k} - \bar{s}) = E(s_j - \bar{s})^2(s_{j+k} - \bar{s})^2 - [E(s_j - \bar{s})(s_{j+k} - \bar{s})]^2 = \sigma^4 \quad (5.15)$$

On the other hand, the expected value of $\sum_{j=1}^m (s_j - \bar{s})^2$ is $m\sigma^2$, then

$$Var(P_k) = \frac{(m-k)\sigma^4}{m^2\sigma^4} = \frac{m-k}{m^2} \quad (5.16)$$

Usually $k \ll m$, so the variance is close to $\frac{1}{m}$. Since all the terms in the dividend of P_k are independent, when m is large, based on Central Limit Theory $P_k \sim N(0, \frac{1}{m})$

It is calculated that the standard deviation of P_k is $\frac{1}{\sqrt{m}}$, and $(-\frac{2}{\sqrt{m}}, \frac{2}{\sqrt{m}})$ is in the range of two standard deviations, so it has 95% confidence to lie within this range.

5.4 Summary

In this chapter, three kinds of mixing indexes, which include homogeneity index, randomness index and time series index are examined. Homogeneity index is the major criteria to check the mixing quality, and two kinds of homogeneity metrics, centroid difference and local average are introduced and applied to high shear mixing simulation.

The centroid difference method is easy to implement and understand, but this criterion is rough and easy to satisfy. The local average method is more accurate and popular. Before local average

method is applied, the size of Averaging Box (AB) should be chosen, and different sizes of AB would lead to different local average results. Randomness is another measurement for particle dynamics system, and it is useful to check whether the structure between particles is ordered, but it is not useful to judge the mixing quality. Time series analysis examines the volatility of the mixing status, and even though it does not provide the current mixing status of the mixture, one can tell whether the mixing quality is improving or keeping still. The ideal values at random mixing state for these mixing indexes are examined in detail and set as the standard to quantify the mixing status. In general, a satisfying mixing quality is achieved in the high shear mixing simulation.

In conclusion, among the mixing indexes, homogeneity index, especially the local average measurement, is the most important index to check the mixing quality. Meanwhile, time series analysis could provide some information on whether the mixing quality is still changing or not.

Chapter 6

Effect of Liquid on High Shear Mixing

6.1 Mixing Process Analysis

Since at the beginning two kinds of solid particles are placed in symmetry, the radial distribution and height distribution of the two types of solid particles are expected to be comparable after certain time of mixing as shown in Fig. 6.1(a). However, there are still some small aggregates in the mixture, which is caused by the strong viscous force of the liquid particles inside, shown in Fig. 6.1(b).

In addition to the top view, the distribution of liquid at different height of the cylindrical container is expressed in Fig. 6.2 and Fig. 6.3. Fig. 6.2 shows the distribution of different types of particles at different height levels, and the sum of all the distributions of certain type equals to 1. Fig. 6.3, on the other hand, shows the volume fraction of different types of particles at certain height level, and the sum of all the volume fractions at every height level equals to 1.

Notice that while the solid particles are distributed homogeneously along the vertical direction of the cylinder, the liquid mainly stays in the middle range. Therefore, the liquid is unable to be mixed well vertically within the solid particles in current a configuration. In Fig. 6.4 the distribution of liquid particles in the whole mixture during the mixing process is illustrated. Solid particles that are in orange and tan are plotted smaller purposely to let those blue liquid particles eases to identify. It is shown that liquid particles mainly lie in the middle of the mixture vertically at the end of the mixing process. Due to resolution issue, it looks like some part of the mixture is not covered by liquid particles. If enough particles are simulated, those parts which looks like not

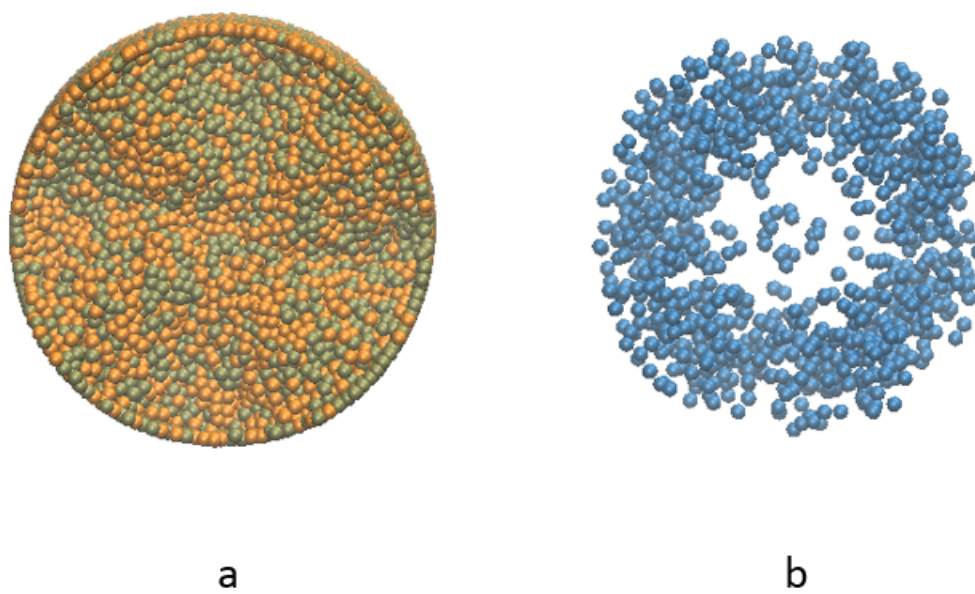


Figure 6.1: Topview of particle distribution after mixing (a)solid particles (b)liquid particles

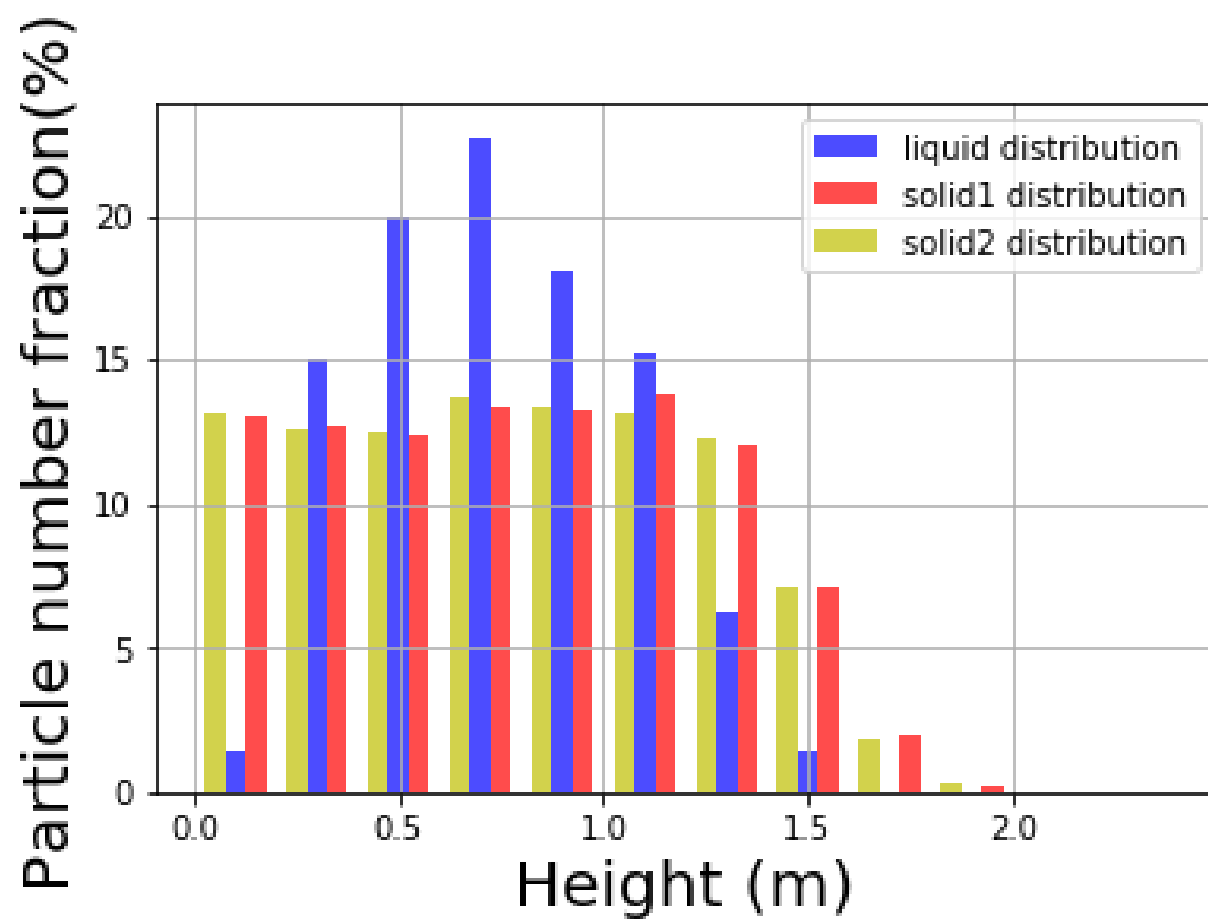


Figure 6.2: Liquid distribution on height

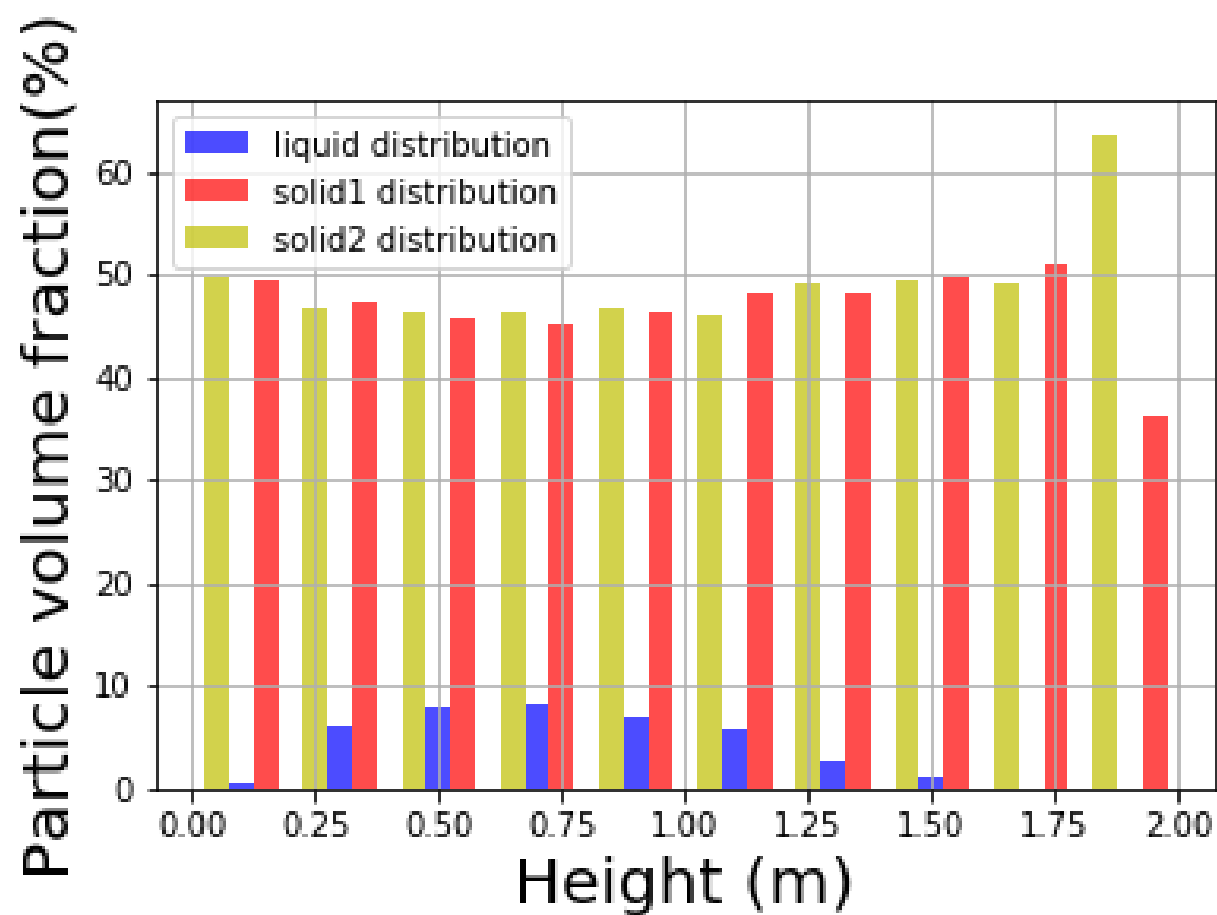


Figure 6.3: Particle volume fraction on height

covered by liquid particles now will be covered by some liquid particles, but the distribution may still be unbalanced.

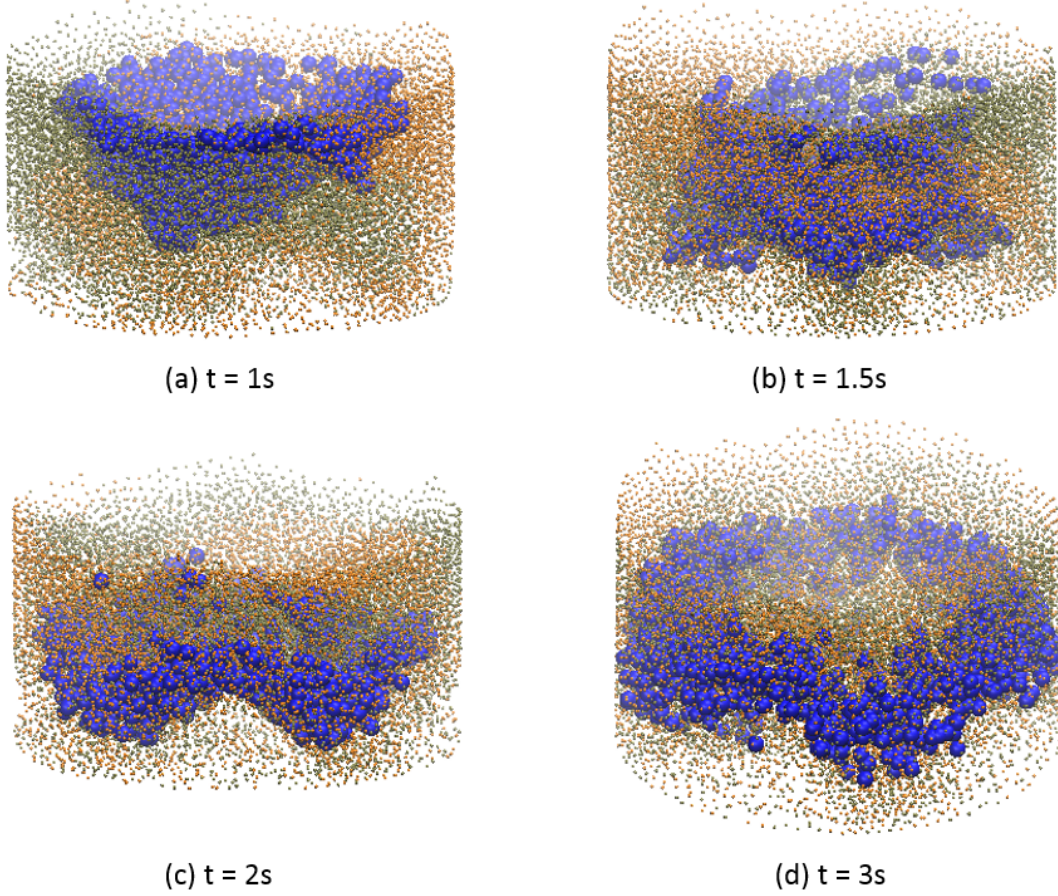


Figure 6.4: Liquid distribution evolution during mixing process

6.2 The Effect of Liquid Content on Mixing Property

In this part, the effect of liquid amount on mixing performance is studied. To test the mixing quality, local average mixing index is adopted here to examine the homogeneity of the particles with a cubic Averaging Box (AB) of size $30mm \times 30mm \times 40mm$. In the following case studies, the mixing index is defined the same way. Liquid contents vary from 0 to 30% and their mixing indices are shown below.

Topview of the distribution of liquid particles is shown in Fig. 6.6. Generally speaking, the

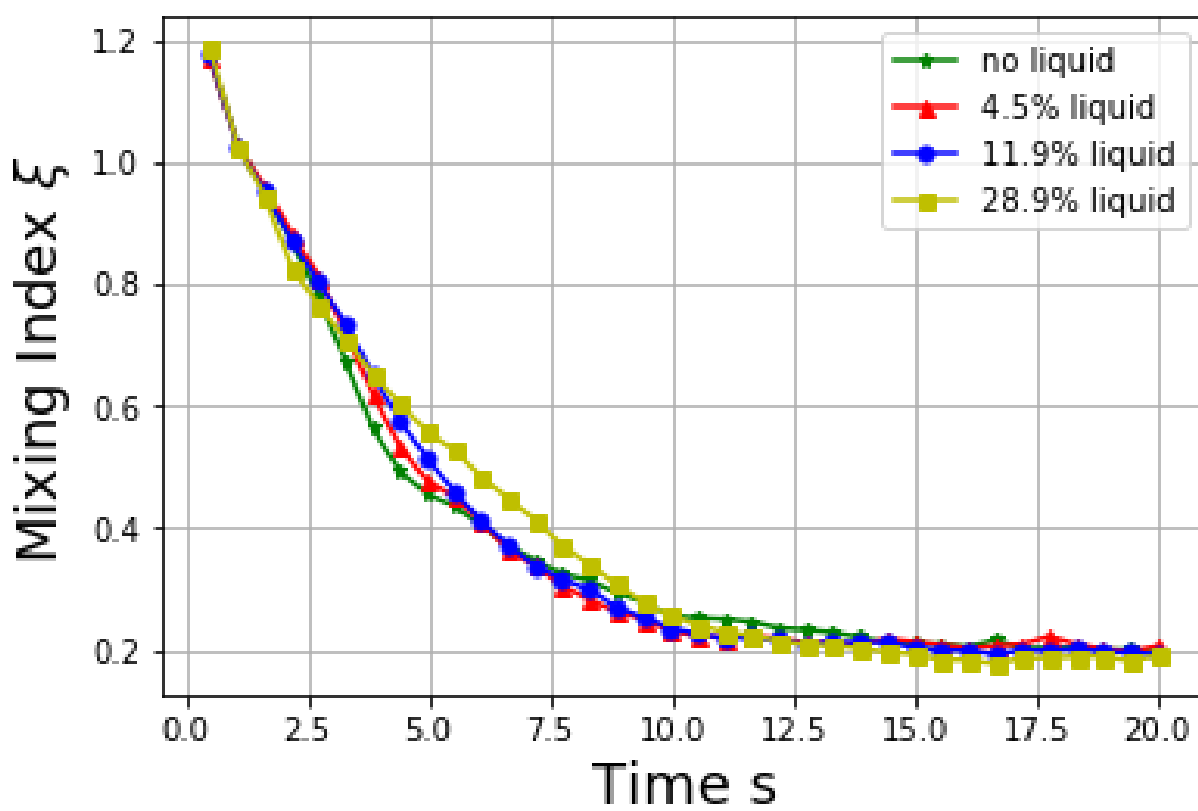


Figure 6.5: Mixing index at different liquid amount

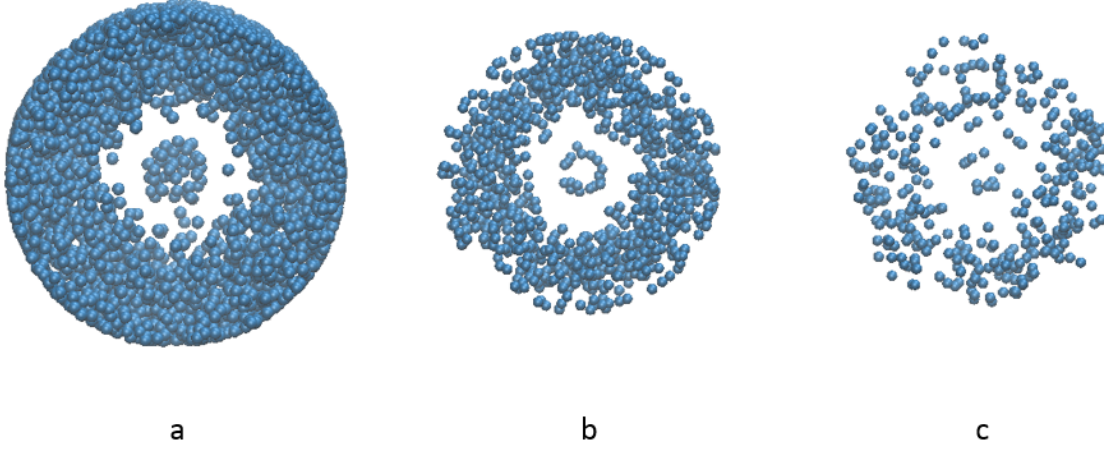


Figure 6.6: Liquid distribution comparison of different liquid contents (a) 28.9% liquid (b) 11.9% liquid (c) 4.5% liquid

distributions of liquid particles are similar, and the distributions of solid particles are not influenced much. However, when the liquid amount is as high as 28.9% the mixing process is slightly slower than other cases (i.e., it takes a longer time for the mixing index to reach the asymptotic value).

6.3 The Effect of Liquid Viscosity on Mixing Property

The viscosity of liquid has a significant effect on the mixing performance. Three different kinematic viscosity $0.04m^2/s$, $0.2m^2/s$ and $1m^2/s$ are applied while other parameters keep the same. The evolutions of mixing indices with time in these simulations are listed in Fig. 6.7.

A high viscosity inhabits the mixing process. When the viscosity of liquid is low, its effect on the mixing performance is not obvious, but when the viscosity of liquid is high, the viscous force between liquid and solid is also high. The liquid particles act like binders and hold solid particles together as local aggregates, which makes it hard to mix the solid particles completely. However, even though a high viscosity decelerates the mixing progress, solid particles could mix well eventually given a long mixing time in Fig. 6.7.

In this part, a group of figures of liquid particles distribution are shown in Fig. 6.8 from top

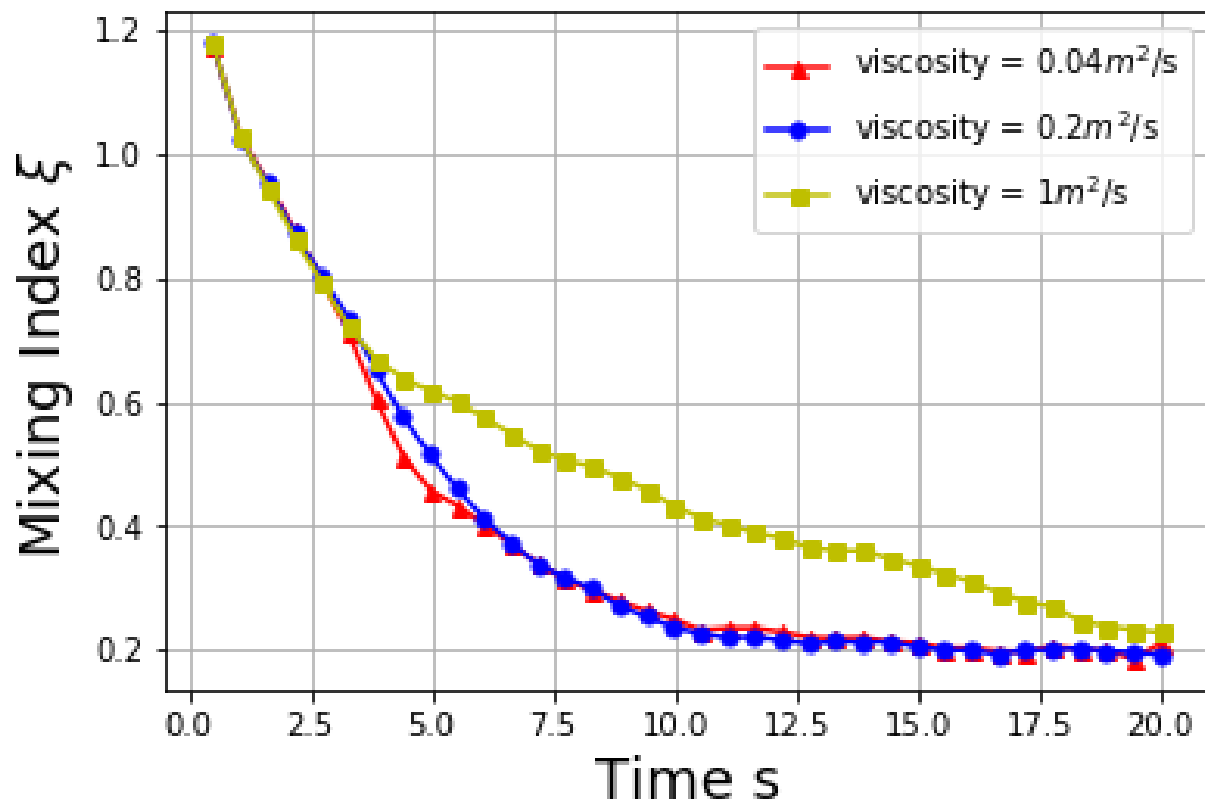


Figure 6.7: Mixing index at different viscosity

view. When the viscosity of liquid particles is high, the viscous liquid would shrink towards the center first because of the rotation of the mixtures as in Fig. 6.8(b), and then extend to the rim gradually, eventually reach an almost homogeneous distribution horizontally, as in Fig. 6.8(c). On the other hand, when the viscosity of liquid is low, all the liquid particles are distributed to the rim immediately and remain along the boundary for the rest of the simulation process. One can tell that there is no difference from Fig. 6.8(e) to Fig. 6.8(f). Appeared in the middle of the domain is a little liquid trapped by the blades, and the majority is distributed on the rim of the container, thus the liquid particles are hard to mixed homogeneously with the solid particles if the viscosity is too low.

6.4 Discussion

In this part, the two-way SPH-DEM coupling method is applied to high shear mixing problems in a four-bladed mixer. Two types of solid particles with same properties and one type of liquid particles are mixed in a four-bladed mixer. Mixing index measured by local average method is the main criterion to quantify the quality of the mixing status. The mixing processes with various amounts of liquid and different liquid viscosity values are presented, and some interesting results are listed below:

1. In general solid particles could be mixed well, but liquid is not able to distribute uniformly among the mixture. As shown in Fig. 6.1(b) from top view and Fig. 6.2 for vertical distribution, the distribution of liquid particles is not balanced. It may be caused by the nature of the rotating blades, and liquid particles tend to distribute more on the rim rather than the center.
2. For a complete view of the track of liquid particles shown in Fig. 6.4, it is observed that liquid particles first shrink to the center, then sink at center, and finally distribute to the rim at a lower height compared to the initial position. The liquid content of the mixture depends on the distance to the axis center. At the beginning, liquid particles concentrate at the center, and they distribute at the rim in the end. Depending on the liquid distribution radially, one can deduce the current stage of the mixing process.

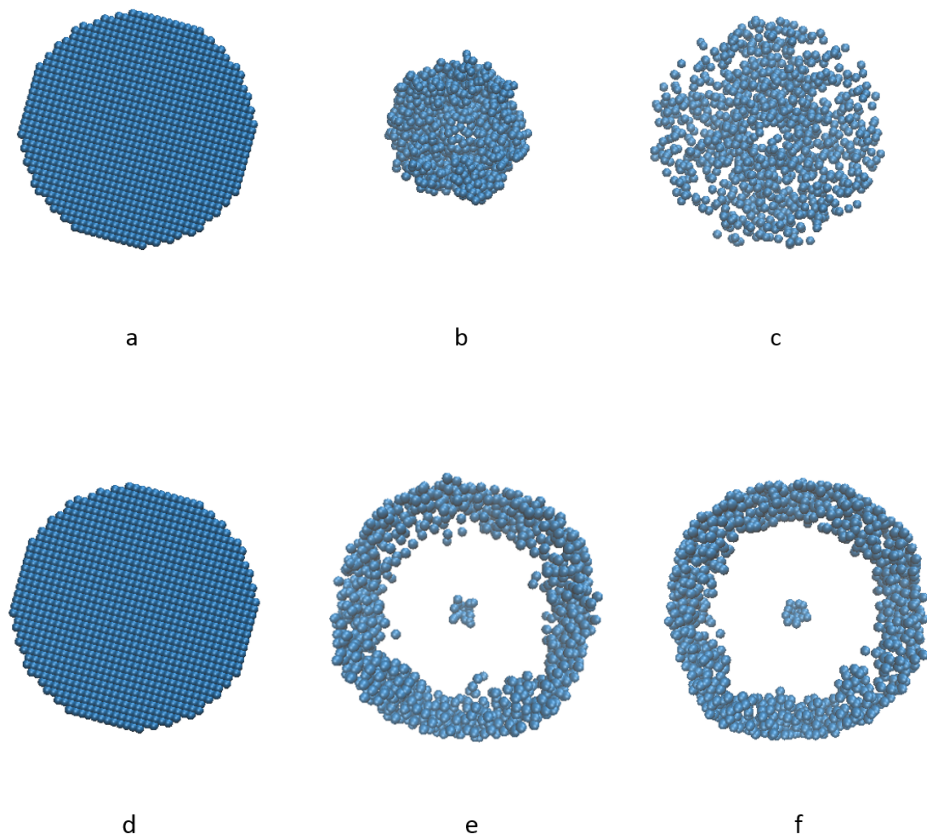


Figure 6.8: Liquid distribution comparison of high viscosity and low viscosity at different time
 (a) $\nu = 1 \text{ m}^2/\text{s}$ $t=0$ (b) $\nu = 1 \text{ m}^2/\text{s}$ $t=2\text{s}$ (c) $\nu = 1 \text{ m}^2/\text{s}$ $t=5\text{s}$ (d) $\nu = 0.04 \text{ m}^2/\text{s}$ $t=0$ (e) $\nu = 0.04 \text{ m}^2/\text{s}$ $t=2\text{s}$ (f) $\nu = 0.04 \text{ m}^2/\text{s}$ $t=5\text{s}$

3. As for the effect of liquid amount on the mixing quality, it is observed that it does not have too much influence on the final result, as shown in Fig. 6.6. The mixing quality here refers to the mixing quality of solid particles, and for liquid distribution, the pattern looks very similar in Fig. 6.6(a), (b) and (c), for different liquid contents.
4. As for the effect of liquid viscosity on the mixing quality, it does have obvious impact on the mixing quality, and high viscosity of liquid would inhibit the mixing process in Fig. 6.7. The liquid distribution is also inhibited in Fig. 6.8. However, for long term, the final state will converge, high viscosity would slow down the mixing process, but the results obtained at high viscosity would eventually be the same as that at low viscosity.

Chapter 7

Application to Different Mixers

In addition to the four-bladed mixer modeled in previous chapters, two kinds of mixers commonly used in industry are introduced and modeled. The simulation results are analyzed and the effects of different mixers on mixing performance will be discussed. Experiments are conducted using one of the mixer and the results are compared with simulation.

7.1 Double Planetary Mixer

Double Planetary Mixers (DPM) have been widely used for the particle mixing process with high viscosity. DPM used in experiments is shown in Fig. 7.1 (a) and our simulation model for DPM is shown in Fig. 7.1 (b). One DPM contains two groups of blades, connected through a common fixed axis in the center of the mixer, as shown in Fig 7.1. Each group of blades has two single blade that are connected by their sub-axis, and each blade rotates 90° towards its center. The subaxis in each group is not fixed and rotate around the fixed axis in the center. With two kinds of rotating axes, the motion of DPM is more flexible and almost every corner in the mixer container could be reached. A detailed description of the configuration of DPM is shown in Fig. 7.1 (b), and their values used in this simulation are listed in Table. 7.2.

Compared to the Four-Bladed Blades modeled in previous chapters, the shape of the blades of DPM is curved, not straight. The complicated shape of DPM blades makes it hard to determine whether a particle contacts with DPM or not. To avoid this problem, DPM is created as a group of fixed solid particles, which are included in the particle systems. The interaction forces between

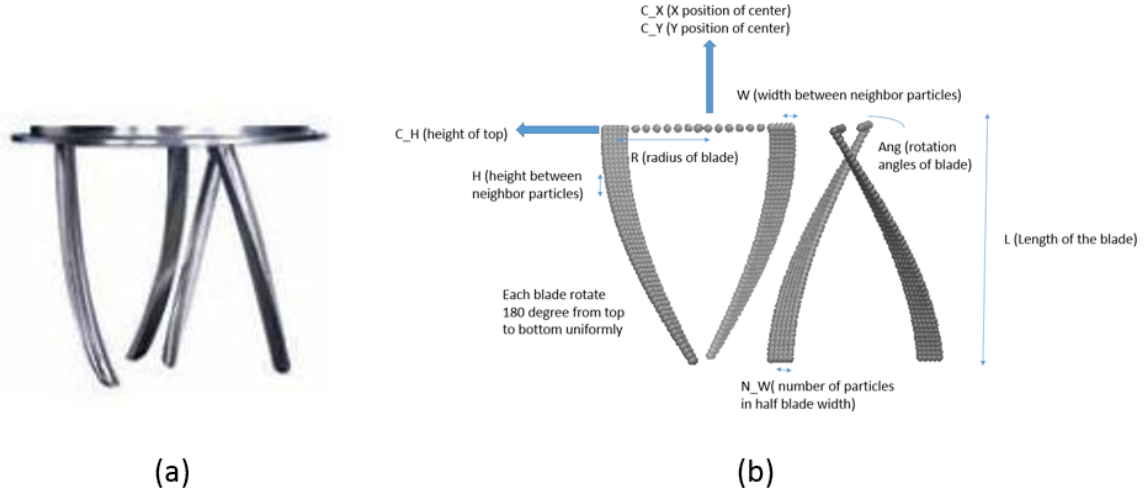


Figure 7.1: Comparison of Double Planetary Mixer (a) experiment (b) simulation

DPM particles and other particles follow the same rule, but DPM particles are bonded with rigid connection and their motion is not affected by these interaction forces, and they follow the DPM mixer rotation pattern as designed at the initialization by the user.

7.1.1 Experimental Study

The mixing experiment is conducted under a vacuum of $< 10 \text{ torr}$ at a temperature of 60° , and a stabilized rotation speed of 75 rpm . Powder sugar, table sugar and PDMS are mixed during the mixing test. Glass beads (5 wt.% 10 wt.% of the overall solid load) of three different colors were added into the mixture to assess the mixing quality.

Average sizes of these particles are: $7\text{-}28 \mu\text{m}$ for Aluminum H12, $25 \mu\text{m}$ for powder sugar, $250 \mu\text{m}$ for table sugar, and $1000 \mu\text{m}$ for glass beads. Density values of the ingredients in the mix design are: 2.7 g/cm^3 for Aluminum H12, 1.59 g/cm^3 for powder sugar and table sugar, 1.55 g/cm^3 for glass beads, 1.0635 g/cm^3 for PDMS. The material properties are listed in Table. 7.1

The base and curing agent of PDMS were premixed for 1.5 minutes before powder sugar was added and mixed for 3 minutes, followed by table sugar and glass beads (mixing for 1 minute).

To observe how well the different particles are mixed as mixing time increases, we took out

Table 7.1: DPM experiment with 84% solid load

Materials	Value
Powder sugar	148.7g
Table sugar	475.7g
Glass beads	32.9g
PDMS	125.2g

certain amount of the granulate composite after the mixing process finished and cast them into 8 petri dishes ($5mm$ (height) \times $38mm$ (diameter)). One example of the sample is shown in Figure. 7.2. Once these samples are cured, they were taken out of the petri dish with a spatula, and the black glass beads at the bottom face of the petri dishes within an intact circle of $24mm$ (diameter) were counted for evaluation of the mixing quality. Notice that because the black color provided the best contrast against the ingredients in the mix, only the black beads were counted. Samples were taken out at each minute. To ensure representative sampling, we took out samples from different locations of the mixing vessel. We also pre-calculated to the total mass of the mix to guarantee that enough materials were left at the end of the mixing so that the mix design did not change qualitatively as the overall mass of the mix was continuously reduced by sampling.

7.1.2 Numerical Study and Comparison

Following the experiment study, simulation is designed to compare with this Double Planetary Mixer experiments. Since the radius of powder sugar is too small, which is hard to simulate together with other particles whose radius is ten times larger than it, the powder sugar in the PDMS is treated together as one homogenized liquid material. The other two solid materials table sugar and glass beads are simulated for the mixing process. The parameters used in simulation is listed in Table. 7.4

Fig. 7.3 shows the initial state of the Double Planetary Mixing process. The grey particles represent Double Planetary Blades, the yellow particles represent table sugars, the purple particles represent the glass beads and the blue particles represent liquid (mixture of PDMS and powder sugar). Fig. 7.4 shows a snapshot during the mixing process. The final state is shown in Fig. 7.5

At the beginning of the simulation, one kind of liquid particles and two kinds of solid particles

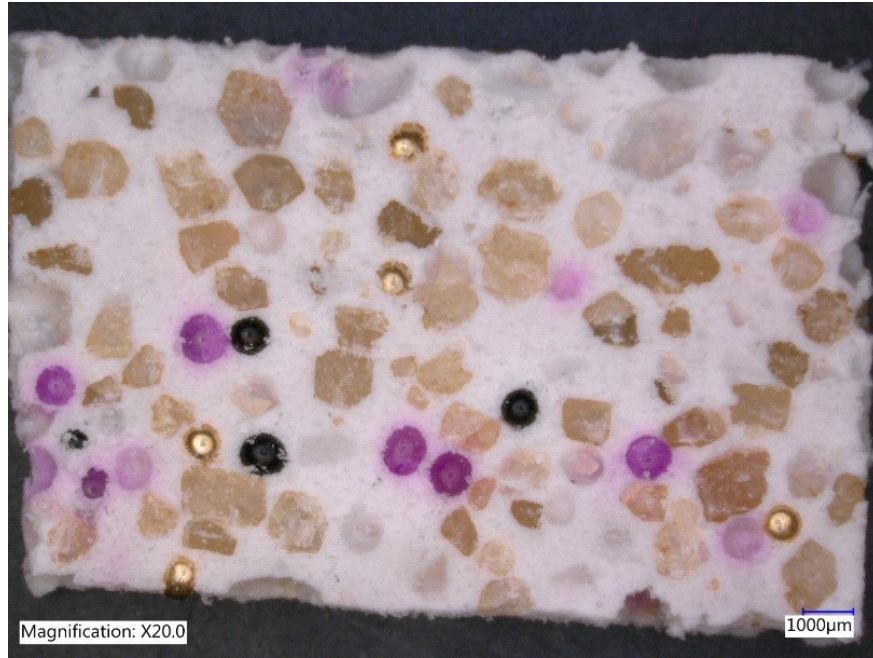


Figure 7.2: Sample in mixing study to test homogeneity

Table 7.2: Geometric parameters of the Double Planetary Mixer

Parameter	Value
Vessel diameter	200mm
Vessel height	200mm
Blade length (L)	96mm
Blade top height (H)	150mm
Center fixed axis radius (R)	50mm
Sub rotating axis radius (r)	45mm
Blade rotation angle (Ang)	90°
Width between neighbor particles (W)	2.5mm
Height between neighbor particles (D)	2.5mm
Number of particles in half blade width (N)	4
Center axis rotation speed	1.25rad/s
Sub axis rotation speed	1.25rad/s

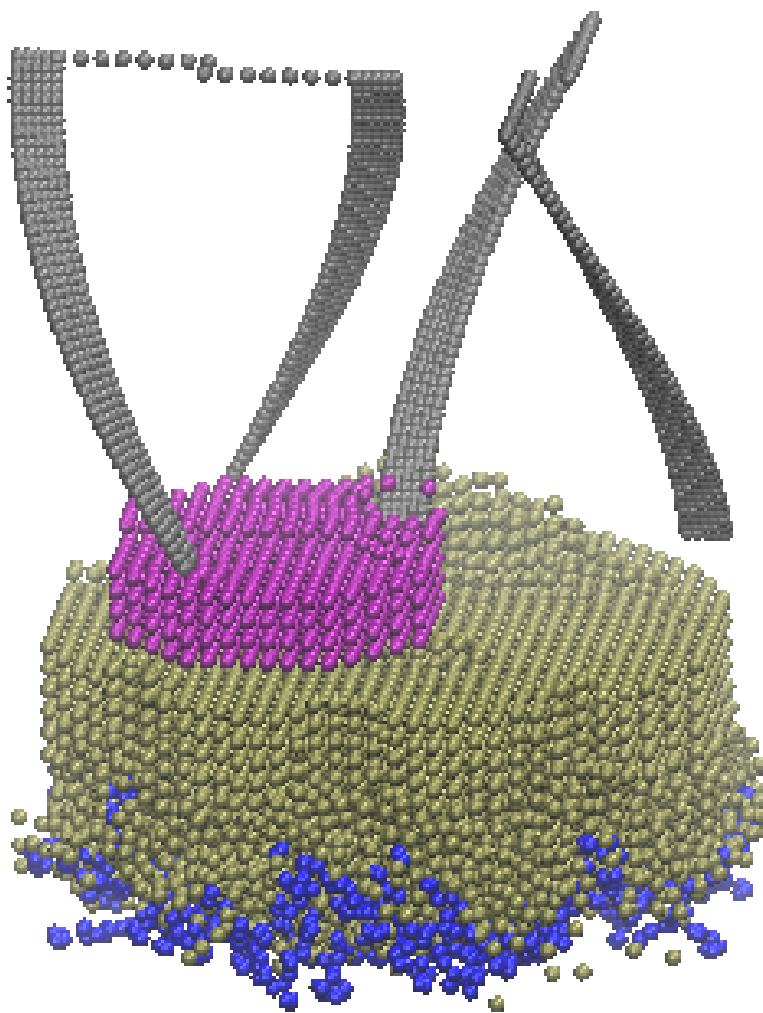


Figure 7.3: Initial State of DPM simulation

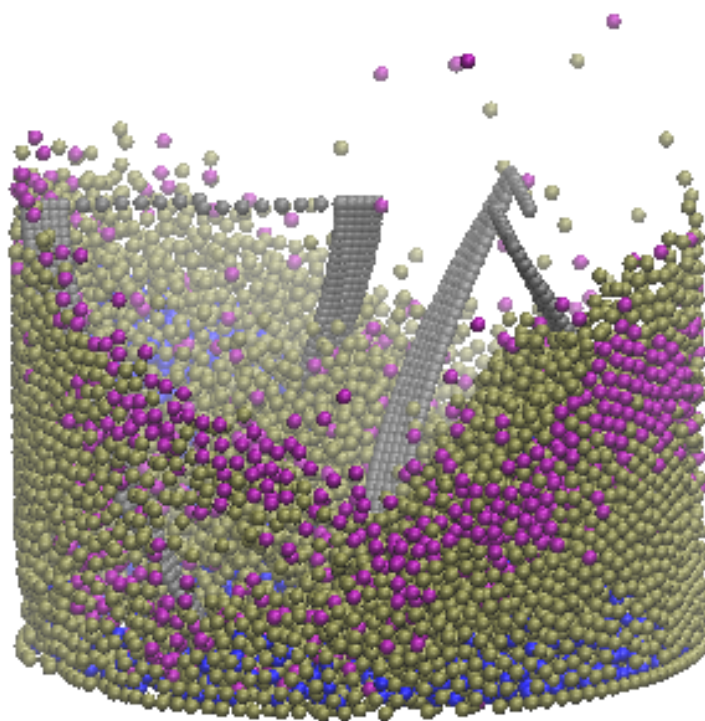


Figure 7.4: DPM simulation with 84% solid load

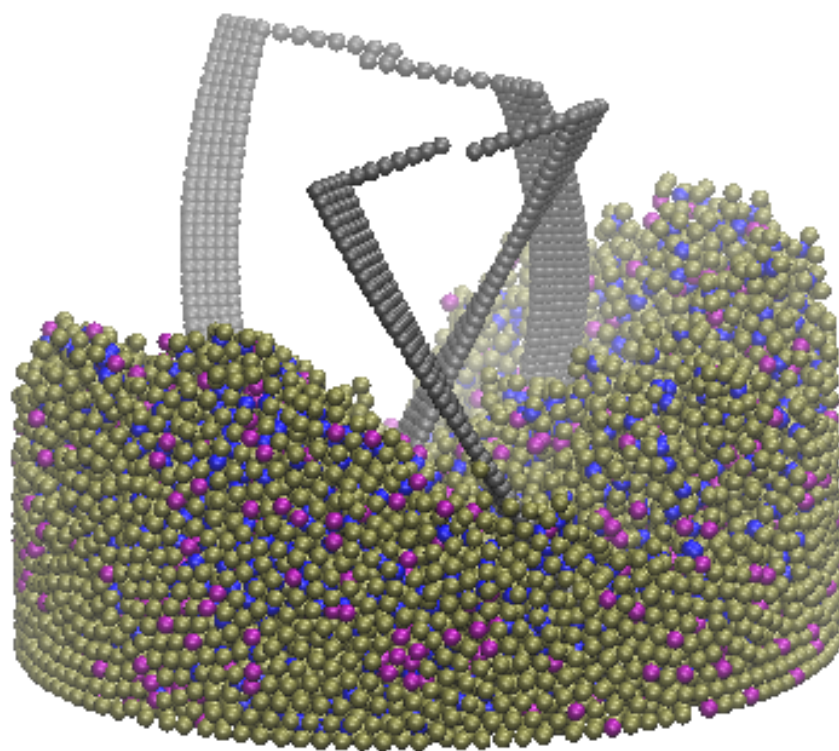


Figure 7.5: Final State of DPM simulation

Table 7.3: DPM simulation with 84% solid load

Property	Value
Liquid (PDMS and powder sugar) density	$1.064g/cm^3$
Table sugar density	$1.590g/cm^3$
Glass beads	$1.550g/cm^3$
Table sugar radius	$2.5cm$
Glass beads	$2.5cm$
Liquid (PDMS and powder sugar) particle numbers	2478
Table sugar particle numbers	11424
Glass bead particle numbers	972
Time step	$5e - 6$
Simulation Time	$60s$

Table 7.4: Homogeneity comparison for DPM mixing between experiment and simulation

Properties	Experiment (numbers in sample)	Simulation (number density in sample)
Mean	7.4	0.081
Standard Deviation	1.9	0.0387
Ration	0.26	0.39

are generated. They experience a free falling stage until an equilibrium condition is reached. Then the DPM made of particles move into the particles and start to rotate at a constant rate. During the mixing process, all three kinds of particles get mixed together and merged with each other. The purple particles which represent glass beads are used as tracers to test the homogeneity of the mixture. Similar as 8 samples in experiment, 8 cells with size $40mm \times 40mm \times 5mm$ are chosen to calculate homogeneity of the whole mixture. Their number densities of glass beads in space are calculated and compared with the experiments. Since the sizes of particles in simulation and experiment are different, the ratio between standard deviation and mean value is used for comparisons.

From the comparison of mean and standard deviation ratio, it is shown that they are in similar range. This value demonstrates the uniformity of the distribution of glass beads, and it is also a sign of the homogeneity of the whole mixture. In general our simulation catches the physics, and more accurate results could be obtained with much more particles.

7.2 Baker Perkins Dual Planetary Mixer

In addition to Double Planetary Mixer, another kind of mixer, Baker Perkins Dual Planetary Mixer (BPDPM) is also widely used in mixing process. Fig. 7.6 shows an illustration of Baker Perkins Dual Planetary Mixer. The rotation pattern of BPDPM is the same as Double Planetary Mixer, in which two blades rotate around the center axis, and each blade rotates around itself in sub-axis. The differences between BPDPM and DPM lie in the blades. In BPDPM, two blades are different. One blade is similar to the blade in DPM, but the lower ends of the blades are connected. The other blade is a curved plate without holes, and the shape of its boundary is the same as that in DPM. The configuration of BPDPM makes it have larger contact surface with mixtures than DPM.

Since the rotation pattern of BPDPM is similar to DPM, the model of BPDPM could be easily obtained by making some changes as mentioned above. The simulation model of BPDPM is shown in Fig. 7.7.

7.2.1 Simulation Study

In this part, a simulation study is conducted in BPDPM and compared with DPM. All the parameters of particles and the rotation pattern of the blades are the same, only the configuration of blades are different. 30 seconds of BPDPM high shear mixing simulation is conducted. Fig. 7.8, Fig. 7.9 and Fig. 7.10 show the initial, middle state and final state of the mixing process.

It is noted that during the mixing process, all the particles move along with the blades and form larger aggregates. As the mixing process start, the distribution of the mixture in the container becomes unbalanced. From the top view of the mixture in Fig. 7.11, some parts are empty while other parts aggregate more particles. Because of the space distribution of all the particles are unbalanced and it changes with the rotation of blades, tracing Averaging Box (AB) and computing



Figure 7.6: Baker Perkins Planetary Mixer

local average mixing index curve is not an appropriate way to measure the mixing process.

Even when the mixing process stops, the distribution of all the particles is still not balanced in the container, as shown in Fig. 7.10. Some part of the container is empty while other parts are higher. Recall DPM in Sec. 7.1, at final state all the particles fill the container in a more balanced way and several ABs could be chosen to test homogeneity. However, in BPDPM mixer the filling pattern of all the particles has been completely changed because the contact surface between BPDPM blades and DPM is much larger than DPM. Therefore, the simulation results for BPDPM is hard to quantify.

7.3 Comparison between Different Mixers

7.3.1 Simulation Setup

In this section, simulations are conducted and compared with three different mixers, which are four-bladed mixer, Double Planetary Mixer (DPM) and Baker Perkins Dual Planetary Mixer (BPDPM). The configuration of the mixing vessel and all the particles to be mixed are the same in all simu-

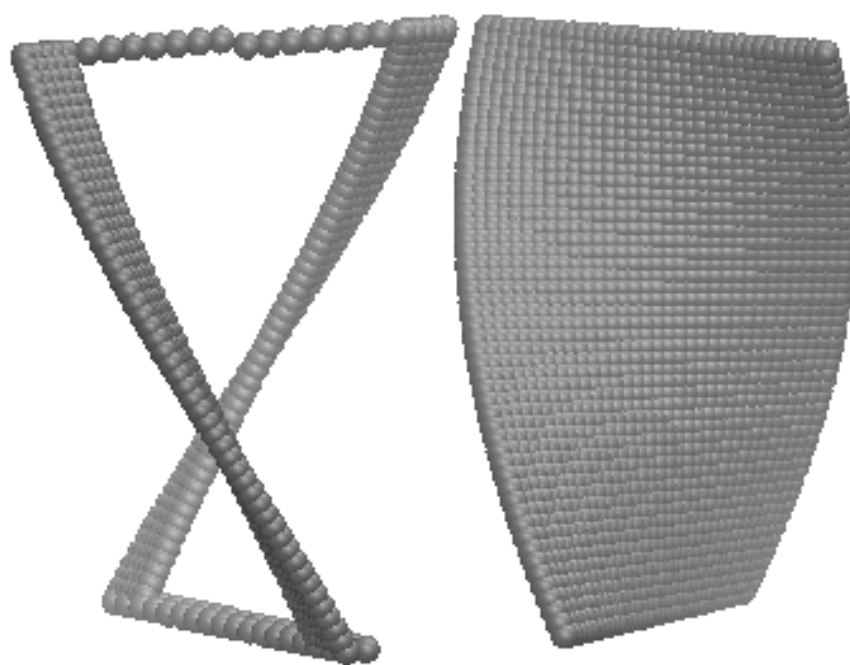


Figure 7.7: Baker Perkins Planetary Mixer Model

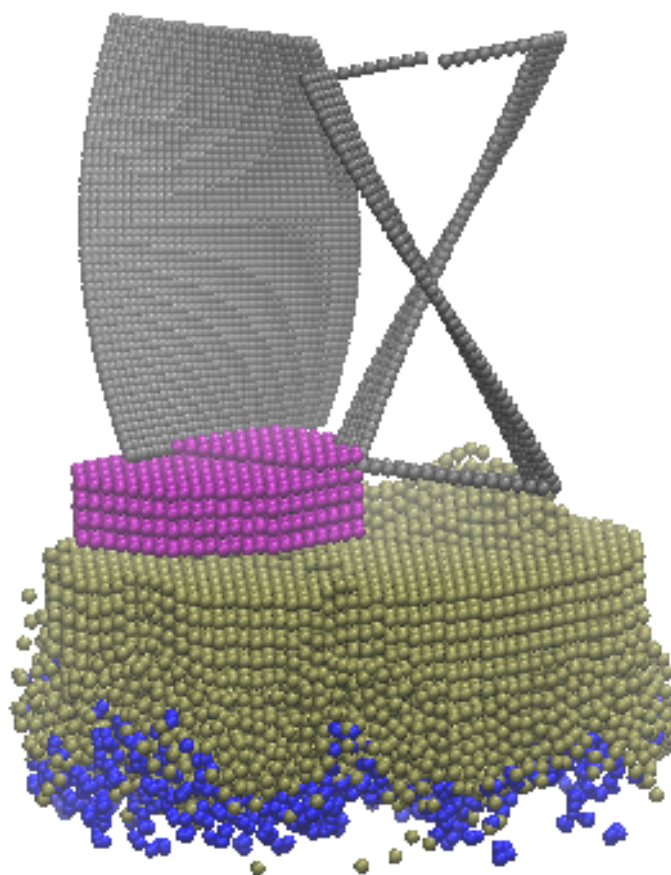


Figure 7.8: Initial State of Baker Perkins Planetary Mixing

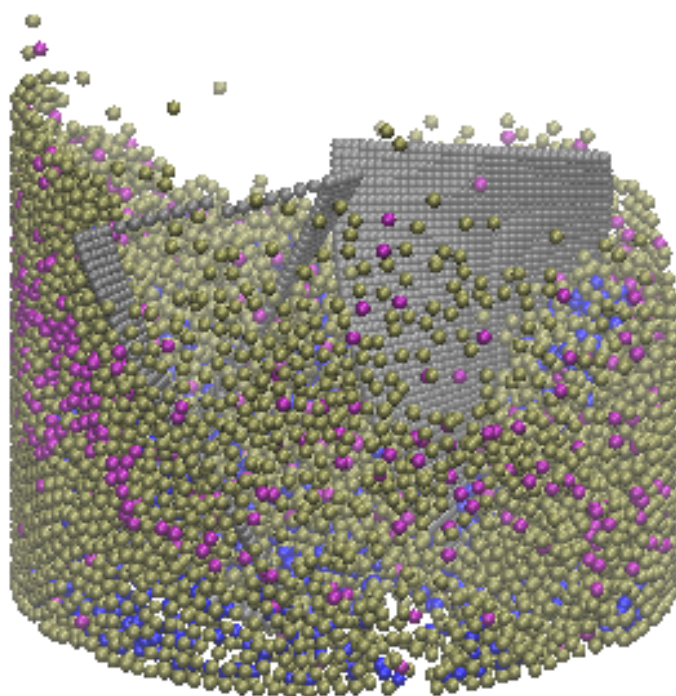


Figure 7.9: Middle State of Baker Perkins Planetary Mixing

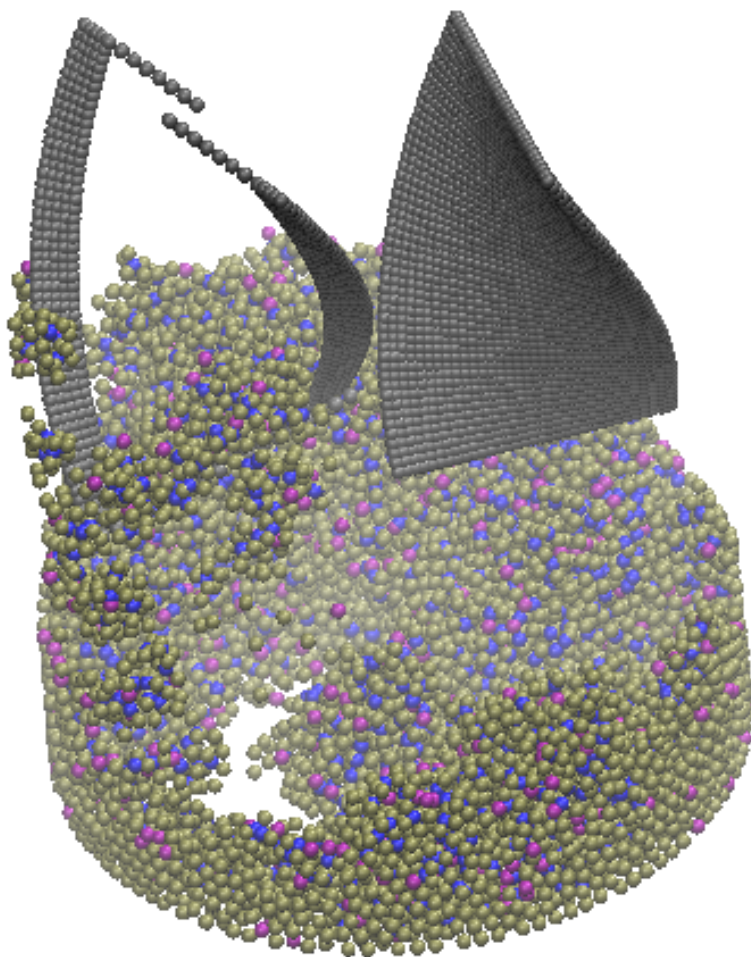


Figure 7.10: Final State of Baker Perkins Planetary Mixing

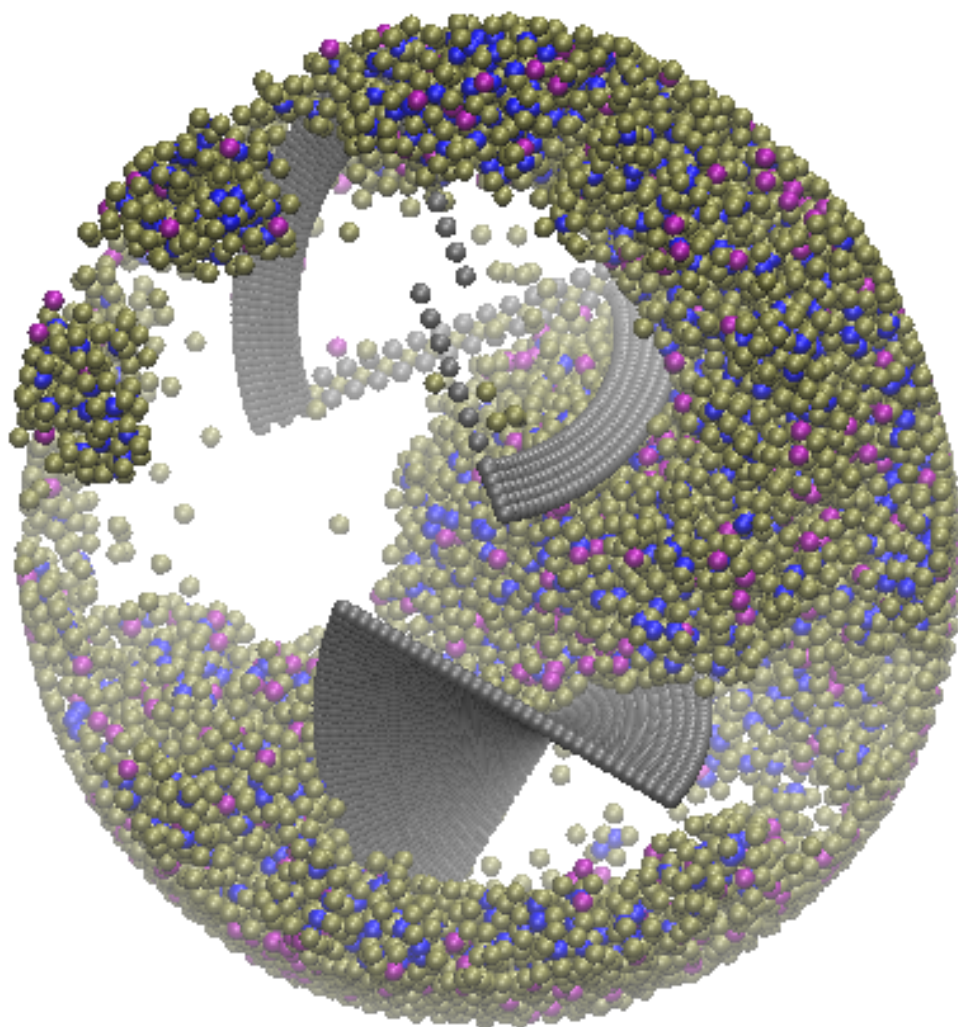


Figure 7.11: Top View of Baker Perkins Planetary Mixing

Table 7.5: Geometric parameters of the Double Planetary Mixer

Parameter	Value
Vessel diameter	400mm
Vessel height	400mm
Blade length (L)	192mm
Blade top height (H)	300mm
Center fixed axis radius (R)	100mm
Sub rotating axis radius (r)	90mm
Blade rotation angle (Ang)	90°
Width between neighbor particles (W)	5mm
Height between neighbor particles (D)	5mm
Number of particles in half blade width (N)	4
Center axis rotation speed	5rad/s
Sub axis rotation speed	5rad/s

lations, only the mixing blades are different. To make the simulation consistent, the configuration of DPM is set in Table. 7.5, and the configuration of BPDPM is changed correspondingly. In all three cases with different mixers, 20s of the high shear mixing process is simulated.

7.3.2 Mixing Index Comparison

As mentioned in Sec. 7.1 and Sec. 7.2, the distribution of all the particles within the vessel is unbalanced because of the rotation of the blades, the local average method is not suitable to measure the mixing process. Even though the blades move out of the mixture in Fig. 7.12, the surface of the mixture is still not flat. Fortunately, a plate of particles could be generated to flatten the surface, which are the grey particles in Fig. 7.13

Then Local average method is applied to test the homogeneity of the mixtures in all three simulations, and the Averaging Box (AB) is chosen the same way as the one with dimension $30 \times 30 \times 30mm^3$ in Chapter. 5. There are 400 ABs chosen to compute the coefficients of variation η and the results for three different mixers are compared in Table. 7.6.

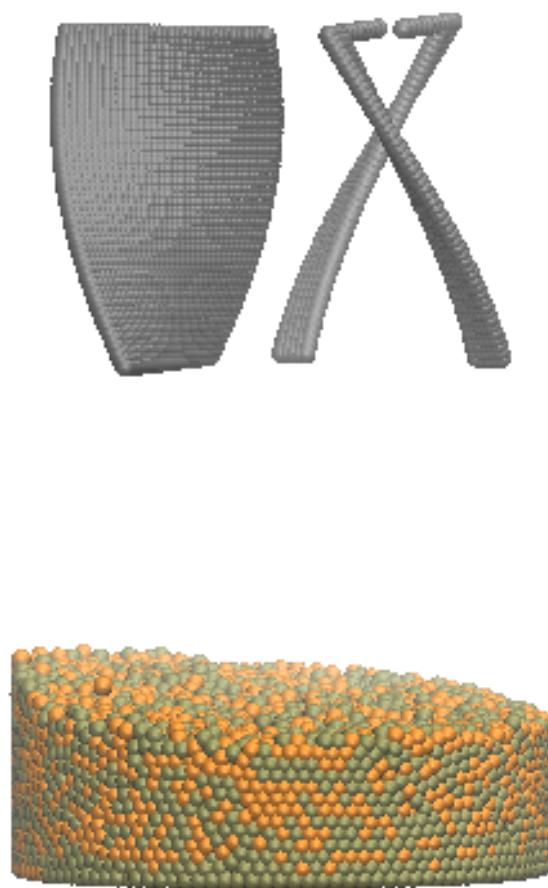


Figure 7.12: Final Mixing State when the Blades Are Moved Out

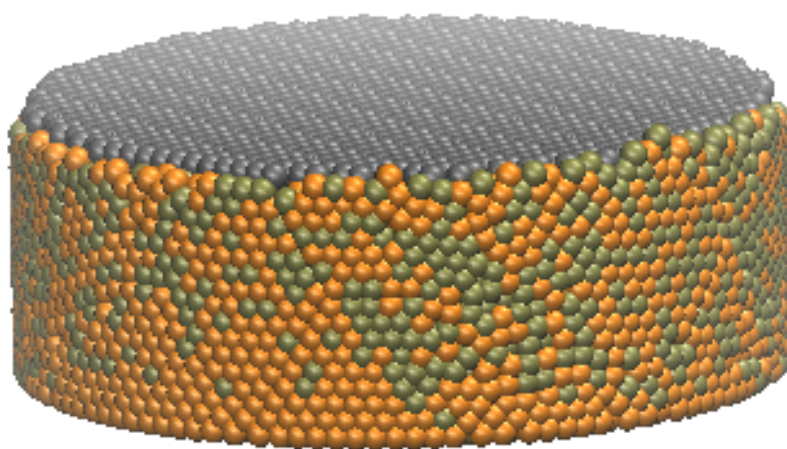


Figure 7.13: Final Mixing State After the Surface is Flattened

Table 7.6: Coefficient of Variation Comparison of Three Mixers

Mixer	Coefficient of Variation η
Four-Bladed Mixer	0.211
Double Planetary Mixer	0.235
Baker Perkins Dual Planetary Mixer	0.231
Homogeneous Reference Value	0.269

The homogeneous reference value is an indicator that if the mixture is homogeneous, there is 95% possibility that the coefficient of variance is below this value, and it could be used as a reference to determine whether the mixture is homogeneous or not. It could be seen that in all three mixers, the coefficient of variation η is lower than the reference value for homogeneous structure, which means that the solid particles are mixed homogeneously in all the mixers. The values for DPM and BPDPM are similar, while Four-Bladed Mixer achieves the best result.

It should be noticed that this index only demonstrates that the solid particles are distributed homogeneously among the mixture, while liquid particles are not.

7.3.3 Liquid Distribution Comparison

To illustrate the distribution of liquid, rather than the mixing index, the distributions of liquid at height are compared in Fig. 7.14.

It is interesting that the distribution of liquid at height shows obvious differences among three mixers. For Four-Bladed Mixer, the liquid mainly distribute at the middle height level, and the center has more liquid than the end. DPM and BPDPM have similar liquid distribution, in which liquid mainly distribute on the bottom of the vessel, and on higher level liquid is rare. The difference of liquid distribution between Four-Bladed Mixer and the other two is caused by their rotating pattern. In the Four-Bladed Mixer, the blades are mounted at the bottom of the vessel, and during mixing process the rotation of the blades lift up all the particles to a distance. As for DPM and BPDPM, the blades are mounted at the top, and their contact areas with the mixture are less compared to Four-Bladed Mixer, so liquid is not lifted and sediment to the bottom. Between DPM and BPDPM, because BPDPM has larger surface contact with the mixture, the liquid distribution in BPDPM is lifted up a little compared to DPM.

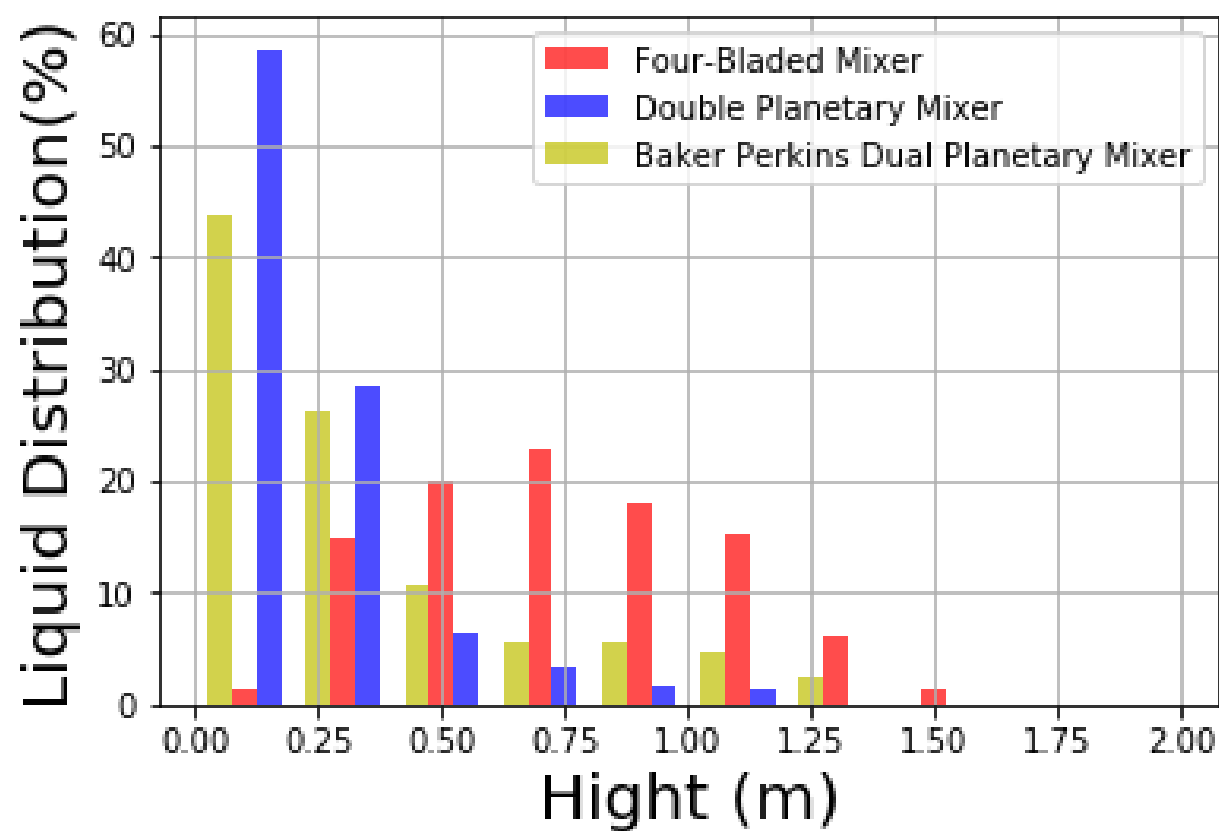


Figure 7.14: Liquid Distribution at Height at Different Mixers

7.4 Summary

In this Chapter, two kinds of mixers in industry, Double Planetary Mixer (DPM) and Baker Perkins Dual Planetary Mixer (BPDPM) are modeled, and simulations are conducted for these two mixers. For DPM, experiment is also conducted. After the mixing process in DPM, some samples are chosen in both experiment and simulation to test the homogeneity in a coarse scale, and the results obtained from these samples agree well. For BPDPM, particles are mixed in an unbalanced distribution within the container, which makes the mixing quality hard to analyze.

The rotation pattern of BPDPM is the same as DPM, and their configurations are similar. BPDPM blades are much wider and have larger contact surface with the mixture than DPM. This difference leads the final state of BPDPM mixing hard to analyze, because all the particles are stirred in a wiggled shape, which makes it unable to choose appropriate AB for analysis.

The analysis of homogeneity is based on the assumption that a series of Averaging Box (AB) could be chosen and then the data of particles within these AB are analyzed. However, in larger blades such as Double Planetary Mixer (DPM) and Baker Perkins Dual Planetary Mixer (BPDPM), during the mixing process, particles are pushed to aggregate by the mixer and the distribution of all the particles within the container is unbalanced, which makes it hard to choose reasonable ABs. For DPM, AB could be chosen at final state, but for BPDPM it is still hard to choose and quantify the mixing quality at final state. For this kind of blades that change the distribution of particles significantly, a novel mixing index should be proposed to test its mixing quality.

The mixing performances between DPM, BPDPM and Four-Bladed Mixer are compared. As discussed before that local average method is not an ideal indicator for mixing process of BPDPM and DPM because of the unbalanced distribution of particles, so only the final states of the mixture in DPM and BPDPM are applied to local average method, and the particles are flattened before analysis. It is found that all three blades achieve very good results and the solid particles are mixed homogeneously. As for liquid, the Four-Bladed Mixer has a more balanced liquid distribution, while the liquid mainly stay at the bottom for the other two mixers.

In general, the performance of DPM and BPDPM is similar. Four-Bladed Mixer achieves a better liquid distribution than DPM and BPDPM. However, when the mixture is highly viscous, the drag forces on the blades of Four-Bladed Mixer is very high because they have large contact surfaces with the mixture, which may make the blades hard to rotate at a desired speed. In this

condition, DPM and BPDPM are more applicable to produce homogeneous mixture.

Chapter 8

Conclusions and Future Work

8.1 Summary

This dissertation focuses on the particle system simulation, especially solid-liquid two-phase particle mixing problem. Methodologies and numerical studies are presented for different scenarios. Two kinds of problems, foaming asphalt problem and multiphase high shear mixing problem are investigated.

An Smoothed Particle Hydrodynamics(SPH) model is developed for asphalt foaming process, which is a liquid-gas two-phase flow problem. The background and characteristics of foaming asphalt is illustrated first, and an SPH model is applied to simulate both asphalt (liquid) and water vapor (gas). To simplify the problem, both liquid and gas phases are treated the same in the numerical model, and gas evaporation rate is introduced from experimental study to explain the bubble collapse process. Simulations with different water contents, pressure and temperature are conducted and the results agree with the experiments well.

For the multiphase high shear mixing problem, Smoothed Particle Hydrodynamics(SPH) is implemented to simulate the dynamics of liquid and Discrete Element Method(DEM) is implemented to simulate the interaction of solid particles. After these two fundamental numerical methods are introduced, a novel coupling scheme which applies Darcy's Law to compute the interaction force between liquid and solid is proposed. The proposed SPH coupling DEM model is verified by three case studies. Simulation results of these three case studies are compared to the theoretical results with very good agreement. Then the coupled SPH-DEM model is applied to particle mixing prob-

lem. The mixing problem is studied in detail in several aspects: the effect of liquid on mixing performance; analysis of mixing quality on various mixing index; model and simulation of mixers in industry. In this problem, the experiments are conducted in Double Planetary Mixer (DPM), and a simulation model for DPM is built. Experimental study and simulation are compared and their results show good agreement.

For particle system simulation problem, computational capacity is always a threshold to improve the accuracy of the problem. All the numerical studies in this work are conducted with our in-house software package Particle Dynamics Parallel Simulator(PDPS). PDPS is accelerated in two ways, MPI parallel computing and GPU acceleration respectively. Speedup results for both MPI parallel computing and GPU computing are illustrated with case study.

In summary, a comprehensive approach for particle simulation is proposed in this work and applied to solve multiphase high shear mixing problem and foaming asphalt problem. This method is able to be extended to more complex problems.

8.2 Key Results

In this work, the key results are presented in four parts: algorithm and formulation for coupled SPH-DEM model, mixing problem simulation and comparison with experiments, foaming asphalt simulation and comparison with experiments, computational speedup with MPI and GPU.

8.2.1 Algorithm and Formulation

In this part, traditional SPH and DEM algorithm and formulations are given. A two-way SPH-DEM coupling method that covers the full spectrum of solid loads, especially for a high solid load, is presented, to study high shear mixing problems. In this coupling scheme, porosity is introduced for each SPH particle to calculate its density. The coupling force between SPH and DEM comes from two parts, pressure gradient force and viscous force. SPH pressure gradient equation is implemented to compute the pressure gradient force between DEM and SPH particles. For the viscous force, a relationship between viscous force and particles porosity is built based upon the Darcy's Law. Three verification case studies are conducted to verify the SPH-DEM coupling model as follows: a single solid particle sedimentation among liquid particles; a porous solid block sedimentation

among liquid particles; liquid flowing through a porous media. This SPH-DEM coupling method is applied to the multiphase high shear mixing process.

8.2.2 Foaming Asphalt Simulation

In this part, a SPH model was applied in this study to simulate the asphalt foaming process.

In this model, it was assumed that both the gas and liquid phases in the foamed asphalt are generated at the beginning by sharing with the same governing equation. Then the height of foamed asphalt decreases as the gas bubble collapse at a given decreasing frequency. An ideal gas theory was used to predict maximum bubble expansion ratio with initial accumulation period adaption, and a linear correlation of bubble collapse frequency with respect to different water contents was applied. Three primary parameters, expansion ratio, half-life and foam index that widely applied to evaluate the foaming characteristics of foamed asphalt, were evaluated to study the foaming characteristics of the foamed asphalt.

The foaming evolutions of the foamed asphalt at different water contents were compared well with the experimental results, and thus the present numerical simulation model can capture the foaming evolution in the foamed asphalt. Parametric studies were further conducted by using the numerical model to evaluate the effects of environmental controlling parameters on the foaming characteristics of the foamed asphalt binder. The parametric studies indicated that: (1) the trend of the foaming evolution and HL of the foamed asphalt under different chamber pressures are almost the same, while the maximum ER experience a linear increase as the pressure in the function chamber increases; and (2) the temperature have an obvious influence on the foaming evolution and its corresponding foam characteristics.

8.2.3 Particle Mixing Problem Simulation

After the verification, the coupling method is applied to high shear mixing problems. The simulation is investigated in three aspects: effect of liquid particles on mixing performance, mixing quality analysis on various mixing index, application to industry mixer.

For the liquid effect, it is found that adding more liquid does not affect the mixing process for a high solid load mix, and increasing the viscosity of the liquid will inhabit the mixing process tremendously. Among these tests, although the solid particles could be mixed well eventually, the

liquid distribution is far from homogeneous, especially when the viscosity of liquid is too low.

Three kinds of criteria to analyze the mixing performance are introduced in this work, which are homogeneity measurement, randomness measurement and time series measurement. Homogeneity measurement is the most important index to measure the mixing quality, and the effect of the choice of Averaging Box (AB) on these mixing criteria is studied. It is found that the mixing process continues in all the different scales, and it is easier to be mixed well on a coarse scale rather than a fine scale. Time series analysis could provide useful information on whether the mixing is still making process, but is unable to quantify the current mixing status.

Three kinds of mixers, Four-Bladed Mixer, Double Planetary Mixer and Baker Perkins Dual Planetary Mixer are simulated in this work and their results are analyzed. In all three simulations, solid particles are mixed homogeneously in the end, while liquid distribution is more uniform in Four-Bladed Mixer. The experiment for the mixing problem is conducted in double planetary mixer(DPM), and the numerical model for DPM is given in detail. The results of experiments and simulations of mixing problem in DPM mixer are compared and they agree well.

8.2.4 Computational Speedup

Particle mixing simulation is typically time consuming, and the speedup of simulation is very important. Both MPI parallel computing and GPU acceleration is implemented in our software package PDPS to speedup the simulation. And model reduction method is proposed at the end.

The principle of MPI acceleration is explained and the speedup with the number of CPU cores used is illustrated. It is found that when the number of cores is not too high, the speed increases almost linear with the number of cores. However, when the number of processors keep increasing, the time spent on communication will eventually terminates the speedup effect. It is found 512 processors would achieve an very ideal speedup result.

Then the GPU acceleration algorithm is applied to a solid-liquid two-phase mixing case study. In this case study, two types of solid particles and one type of liquid particles are mixed by a double planetary mixer. For the same case study, an E5-2620 CPU and a Tesla K40C are implemented for CPU only computation and GPU computation and their speeds are compared for different parts and totally. Among all the computation components, pair force computation take almost 80% of the time, so the speedup of particle pair force almost determines the total speedup performance.

A 56.6 times speedup is achieved for particle pair force computation, leading to a 57.7 times speedup for total simulation, which is a reasonable result. The speedup of GPU would also be influenced by the CPU and GPU used for computation, and GPU has more advantages for larger computation problems since it has more memories than CPU. In future, more advanced GPU would be implemented to increase the speedup performance of GPU parallelism.

For model reduction, both mixing index and density surface of particles during the simulation are predicted based on previous simulation results. Statistical methods such as Linear regression, Kriging Regression and Principle Component Analysis (PCA) are applied to the model reduction. It is found that the predicted results could achieve reasonable accuracy compared with the simulation results.

8.3 Future Work

In future, the mixing process would be investigated in more detail and simulation results would be compared with experiments in more aspects. With the improvement of computational capacity, more accurate and more complex simulation results would be obtained, and specific patterns and phenomenon will be studied. As for foaming asphalt simulation, a fundamental approach would be explored, which can lead to a general solution to these gas-liquid two phase flow. Besides these two problems, other problems with similar physics would be explored.

In addition to the simulation of solid and liquid particles in this work, gas particles would be considered in future and three-phase(solid, liquid, gas) model will be proposed. These approaches will all bring simulation closer to the reality.

Part I

Bibliography

Bibliography

- [1] Warm mix asphalt (WMA) emission reductions and energy savings. In *2nd International Warm-Mix Conference, St. Louis, MO*, 2011.
- [2] S. Adami, X. Y. Hu, and N. A. Adams. A generalized wall boundary condition for smoothed particle hydrodynamics. *Journal of Computational Physics*, 231(21):7057–7075, 2012.
- [3] Markus Ihmsen Julian Bader Gizem Akinci and Matthias Teschner. Animation of air bubbles with SPH. 2011.
- [4] Ahmad Alhasan, Ala Abbas, Munir Nazzal, Samer Dessouky, Ayman Ali, Sang-Soo Kim, and David Powers. Low-temperature characterization of foamed warm-mix asphalt produced by water injection. *Transportation research record: journal of the transportation research board*, (2445):1–11, 2014.
- [5] Meysam Alian, Farhad Ein-Mozaffari, and Simant R. Upreti. Analysis of the mixing of solid particles in a plowshare mixer via discrete element method (DEM). *Powder Technology*, 274:77–87, 2015.
- [6] D. Antypov and J. A. Elliott. On an analytical solution for the damped hertzian spring. *EPL (Europhysics Letters)*, 94(5):50004, 2011.
- [7] Wilkins Aquino. An object-oriented framework for reduced-order models using proper orthogonal decomposition (POD). *Computer Methods in Applied Mechanics and Engineering*, 196(41):4375–4390, 2007.

- [8] Zelalem A. Arega, Amit Bhasin, Wei Li, David E. Newcomb, and Edith Arambula. Characteristics of asphalt binders foamed in the laboratory to produce warm mix asphalt. *Journal of Materials in Civil Engineering*, 26(11):04014078, 2013.
- [9] P. E. Arratia, Nhat-hang Duong, F. J. Muzzio, P. Godbole, and S. Reynolds. A study of the mixing and segregation mechanisms in the bohle tote blender via DEM simulations. *Powder Technology*, 164(1):50–57, 2006.
- [10] G. K. Batchelor. *An Introduction to Fluid Mechanics (Cambridge Mathematical Library)*. Cambridge: Cambridge University Press, 2000.
- [11] Herman JC Berendsen, JPM van Postma, Wilfred F. van Gunsteren, ARHJ DiNola, and J. R. Haak. Molecular dynamics with coupling to an external bath. *The Journal of chemical physics*, 81(8):3684–3690, 1984.
- [12] F. Bertrand, L.-A. Leclaire, and G. Levecque. DEM-based models for the mixing of granular materials. *Chemical Engineering Science*, 60(8):2517–2531, 2005.
- [13] Bruno Blais, Manon Lassaigne, Christoph Goniva, Louis Fradette, and Franois Bertrand. Development of an unresolved CFDDEM model for the flow of viscous suspensions and its application to solidliquid mixing. *Journal of Computational Physics*, 318:201–221, 2016.
- [14] Fani Boukouvala, Yijie Gao, Fernando Muzzio, and Marianthi G. Ierapetritou. Reduced-order discrete element method modeling. *Chemical Engineering Science*, 95:12–26, 2013.
- [15] Nathan Bower, Haifang Wen, Shenghua Wu, Kim Willoughby, Jim Weston, and Joe DeVol. Evaluation of the performance of warm mix asphalt in washington state. *International Journal of Pavement Engineering*, 17(5):423–434, 2016.
- [16] Christopher Earls Brennen and Christopher E. Brennen. *Fundamentals of multiphase flow*. Cambridge university press, 2005.
- [17] Ricardo B. Canelas, Jose M. Domnguez, Alejandro JC Crespo, Moncho Gmez-Gesteira, and Rui ML Ferreira. A smooth particle hydrodynamics discretization for the modelling of free surface flows and rigid body dynamics. *International Journal for Numerical Methods in Fluids*, 78(9):581–593, 2015.

- [18] Richard Car and Mark Parrinello. Unified approach for molecular dynamics and density-functional theory. *Physical review letters*, 55(22):2471, 1985.
- [19] Chris Chatfield. *The analysis of time series: an introduction*. CRC press, 2016.
- [20] Libei Chen, Yves Hontoir, Dexian Huang, Jie Zhang, and A. Julian Morris. Combining first principles with black-box techniques for reaction systems. *Control Engineering Practice*, 12(7):819–826, 2004.
- [21] Arif Chowdhury and Joe W. Button. A review of warm mix asphalt, 2008.
- [22] P. Christofides. Nolinear and robust control of PDE systems. *Methods and Applications to Transport-Reaction Processes*, 2001.
- [23] P. G. Cizmas, A. Palacios, T. O’Brien, and M. Syamlal. Proper-orthogonal decomposition of spatio-temporal patterns in fluidized beds. *Chemical engineering science*, 58(19):4417–4427, 2003.
- [24] Paul W. Cleary. DEM simulation of industrial particle flows: case studies of dragline excavators, mixing in tumblers and centrifugal mills. *Powder Technology*, 109(1):83–104, 2000.
- [25] Paul W. Cleary. Particulate mixing in a plough share mixer using DEM with realistic shaped particles. *Powder technology*, 248:103–120, 2013.
- [26] Paul W. Cleary, Guy Metcalfe, and Kurt Liffman. How well do discrete element granular flow models capture the essentials of mixing processes? *Applied Mathematical Modelling*, 22(12):995–1008, 1998.
- [27] Paul W. Cleary, Soon Hyoun Pyo, Mahesh Prakash, and Bon Ki Koo. Bubbling and frothing liquids. In *ACM Transactions on Graphics (TOG)*, volume 26, page 97. ACM, 2007.
- [28] Paul W. Cleary, Matt Sinnott, and Rob Morrison. Prediction of slurry transport in SAG mills using SPH fluid flow in a dynamic DEM based porous media. *Minerals Engineering*, 19(15):1517–1527, 2006.
- [29] Paul W. Cleary and Matthew D. Sinnott. Assessing mixing characteristics of particle-mixing and granulation devices. *Particuology*, 6(6):419–444, 2008.

- [30] Stephen L. Conway, Azzeddine Lekhal, Johannes G. Khinast, and Benjamin J. Glasser. Granular flow and segregation in a four-bladed mixer. *Chemical Engineering Science*, 60(24):7091–7107, 2005.
- [31] Alejandro JC Crespo, Jos M. Domnguez, Benedict D. Rogers, Moncho Gmez-Gesteira, S. Longshaw, R. Canelas, Renato Vacondio, A. Barreiro, and O. Garca-Feal. DualSPHysics: Open-source parallel CFD solver based on smoothed particle hydrodynamics (SPH). *Computer Physics Communications*, 187:204–216, 2015.
- [32] Noel AC Cressie. Spatial prediction and kriging. *Statistics for Spatial Data, Revised Edition*, pages 105–209, 1993.
- [33] Peter A. Cundall and Otto DL Strack. A discrete numerical model for granular assemblies. *geotechnique*, 29(1):47–65, 1979.
- [34] R. A. Dalrymple and B. D. Rogers. Numerical modeling of water waves with the SPH method. *Coastal engineering*, 53(2):141–147, 2006.
- [35] John A. D’Angelo, Eric E. Harm, John C. Bartoszek, Gaylon L. Baumgardner, Matthew R. Corrigan, Jack E. Cowser, Thomas P. Harman, Mostafa Jamshidi, H. Wayne Jones, and David E. Newcomb. Warm-mix asphalt: European practice, 2008.
- [36] Henry Darcy. *Les fontaines publiques de la ville de Dijon: exposition et application...* Victor Dalmont, 1856.
- [37] Alberto Di Renzo and Francesco Paolo Di Maio. Comparison of contact-force models for the simulation of collisions in DEM-based granular flow codes. *Chemical engineering science*, 59(3):525–541, 2004.
- [38] Marisa Dinis-Almeida and Mrcia Lopes Afonso. Warm mix recycled asphalta sustainable solution. *Journal of Cleaner Production*, 107:310–316, 2015.
- [39] Alexander IJ Forrester and Andy J. Keane. Recent advances in surrogate-based optimization. *Progress in Aerospace Sciences*, 45(1):50–79, 2009.

- [40] G. Fourtakas and B. D. Rogers. Modelling multi-phase liquid-sediment scour and resuspension induced by rapid flows using smoothed particle hydrodynamics (SPH) accelerated with a graphics processing unit (GPU). *Advances in Water Resources*, 92:186–199, 2016.
- [41] Benjamin FrantzDale, Steven J. Plimpton, and Mark S. Shephard. Software components for parallel multiscale simulation: an example with LAMMPS. *Engineering with Computers*, 26(2):205–211, 2010.
- [42] Vincent Gaudefroy, Francois Olard, Myriam Ridane, Etienne Beduneau, and Chantal De La Roche. Laboratory environmental assessment of half-warm mix asphalts by means of the factorial experiment design approach. In *TRB 2009 Annual Meeting*, pages 24–p, 2009.
- [43] David H. Gay and W. Harmon Ray. Identification and control of distributed parameter systems by means of the singular value decomposition. *Chemical Engineering Science*, 50(10):1519–1539, 1995.
- [44] Dimitri Gidaspow. *Multiphase flow and fluidization: continuum and kinetic theory descriptions*. Academic press, 1994.
- [45] Robert A. Gingold and Joseph J. Monaghan. Smoothed particle hydrodynamics: theory and application to non-spherical stars. *Monthly notices of the royal astronomical society*, 181(3):375–389, 1977.
- [46] R.A. Gore and C.T. Crowe. Effect of particle size on modulating turbulent intensity. *International Journal of Multiphase Flow*, 15(2):279–285, 1989-04.
- [47] Christina Grindon, Sarah Harris, Tom Evans, Keir Novik, Peter Coveney, and Charles Laughton. Large-scale molecular dynamics simulation of DNA: implementation and validation of the AMBER98 force field in LAMMPS. *Philosophical Transactions of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, 362(1820):1373–1386, 2004.

- [48] Robert D. Groot and Patrick B. Warren. Dissipative particle dynamics: Bridging the gap between atomistic and mesoscopic simulation. *The Journal of chemical physics*, 107(11):4423–4435, 1997.
- [49] Nan Gui, Jie Yan, Wenkai Xu, Liang Ge, Daling Wu, Zhongli Ji, Jinsen Gao, Shengyao Jiang, and Xingtuan Yang. DEM simulation and analysis of particle mixing and heat conduction in a rotating drum. *Chemical Engineering Science*, 97:225–234, 2013.
- [50] Yu Guo and Jennifer Sinclair Curtis. Discrete element method simulations for complex granular flows. *Annual Review of Fluid Mechanics*, 47:21–46, 2015.
- [51] Andrei A. Gusev. Representative volume element size for elastic composites: a numerical study. *Journal of the Mechanics and Physics of Solids*, 45(9):1449–1459, 1997.
- [52] James M. Haile. *Molecular dynamics simulation: elementary methods*, volume 1. Wiley New York, 1992.
- [53] G. A. Harder. LEA half-warm mix paving report, 2007 projects for NYSDOT. *McConnaughay Technologies, Cortland, NY*, 2008.
- [54] Ali Hassanpour, Hongsing Tan, Andrew Bayly, Prasad Gopalkrishnan, Boonho Ng, and Mojtaba Ghadiri. Analysis of particle motion in a paddle mixer using discrete element method (DEM). *Powder Technology*, 206(1):189–194, 2011.
- [55] D. He, N. N. Ekere, and L. Cai. Computer simulation of random packing of unequal particles. *Physical review E*, 60(6):7098, 1999.
- [56] Yurong He, Wengen Peng, Tianqi Tang, Shengnan Yan, and Yunhua Zhao. DEM numerical simulation of wet cohesive particles in a spout fluid bed. *Advanced Powder Technology*, 27(1):93–104, 2016.
- [57] Brian Hill, Salman Hakimzadeh, and W. G. Buttlar. An investigation of the effects of three warm mix asphalt additives on asphalt binder and mixture properties 2. *Urbana*, 51(61801):10, 2011.
- [58] P. J. Hoogerbrugge and JMVA Koelman. Simulating microscopic hydrodynamic phenomena with dissipative particle dynamics. *EPL (Europhysics Letters)*, 19(3):155, 1992.

- [59] Wm G. Hoover. Molecular dynamics. In *Molecular Dynamics*, volume 258, 1986.
- [60] Mohamed Azlan Hussain. Review of the applications of neural networks in chemical process controlsimulation and online implementation. *Artificial intelligence in engineering*, 13(1):55–68, 1999.
- [61] K. J. Jenkins, M. F. C. Van de Ven, and J. L. A. De Groot. Characterisation of foamed bitumen. In *7th Conference on asphalt pavements for Southern Africa*, volume 18, 1999.
- [62] Zhenya Jia, Eddie Davis, Fernando J. Muzzio, and Marianthi G. Ierapetritou. Predictive modeling for pharmaceutical processes using kriging and response surface. *Journal of Pharmaceutical Innovation*, 4(4):174–186, 2009.
- [63] Fangming Jiang, Mnica SA Oliveira, and Antonio CM Sousa. Mesoscale SPH modeling of fluid flow in isotropic porous media. *Computer Physics Communications*, 176(7):471–480, 2007.
- [64] Par Jonsen, Jan F. Stener, Bertil I. Palsson, and Hans-Ake Hggblad. Validation of a model for physical interactions between pulp, charge and mill structure in tumbling mills. *Minerals Engineering*, 73:77–84, 2015.
- [65] O. Kahrs and W. Marquardt. The validity domain of hybrid models and its application in process optimization. *Chemical Engineering and Processing: Process Intensification*, 46(11):1054–1066, 2007.
- [66] Yasunobu Kaneko, Takeo Shiojima, and Masayuki Horio. Numerical analysis of particle mixing characteristics in a single helical ribbon agitator using DEM simulation. *Powder Technology*, 108(1):55–64, 2000.
- [67] T. Kanit, S. Forest, Ia Galliet, Va Mounoury, and D. Jeulin. Determination of the size of the representative volume element for random composites: statistical and numerical approach. *International Journal of solids and structures*, 40(13):3647–3679, 2003.
- [68] Jack PC Kleijnen. Kriging metamodeling in simulation: A review. *European journal of operational research*, 192(3):707–716, 2009.

- [69] Harald Kruggel-Emden, M. Sturm, Siegmund Wirtz, and Viktor Scherer. Selection of an appropriate time integration scheme for the discrete element method (DEM). *Computers & Chemical Engineering*, 32(10):2263–2279, 2008.
- [70] Jihoe Kwon and Heechan Cho. Simulation of solid-liquid flows using a two-way coupled smoothed particle hydrodynamics-discrete element method model. *Korean Journal of Chemical Engineering*, 33(10):2830–2841, 2016.
- [71] Yidan Lan and Anthony D. Rosato. Macroscopic behavior of vibrating beds of smooth inelastic spheres. *Physics of Fluids*, 7(8):1818–1831, 1995.
- [72] Yidong Lang, Stephen E. Zitney, and Lorenz T. Biegler. Optimization of IGCC processes with reduced order CFD models. *Computers & Chemical Engineering*, 35(9):1705–1717, 2011.
- [73] P. A. Langston, U. Tzn, and D. M. Heyes. Discrete element simulation of granular flow in 2d and 3d hoppers: dependence of discharge rate and wall stress on particle interactions. *Chemical Engineering Science*, 50(6):967–987, 1995.
- [74] Alessandro Leonardi, Falk K. Wittel, Miller Mendoza, and Hans J. Herrmann. Coupled DEM-LBM method for the free-surface simulation of heterogeneous suspensions. *Computational Particle Mechanics*, 1(1):3–13, 2014.
- [75] Y. C. Liang, H. P. Lee, S. P. Lim, W. Z. Lin, K. H. Lee, and C. G. Wu. Proper orthogonal decomposition and its applicationspart i: Theory. *Journal of Sound and vibration*, 252(3):527–544, 2002.
- [76] M. B. Liu and G. R. Liu. Smoothed particle hydrodynamics (SPH): an overview and recent developments. *Archives of computational methods in engineering*, 17(1):25–76, 2010.
- [77] P. Y. Liu, R. Y. Yang, and A. B. Yu. DEM study of the transverse mixing of wet particles in rotating drums. *Chemical Engineering Science*, 86:99–107, February 2013.
- [78] Xin Liu, Haihua Xu, Songdong Shao, and Pengzhi Lin. An improved incompressible SPH model for simulation of wavestructure interaction. *Computers & Fluids*, 71:113–123, 2013.
- [79] Edmond YM Lo and Songdong Shao. Simulation of near-shore solitary wave mechanics by an incompressible SPH method. *Applied Ocean Research*, 24(5):275–286, 2002.

- [80] Kun Luo, Shuai Wang, Shiliang Yang, Chenshu Hu, and Jianren Fan. Computational fluid dynamicsdiscrete element method investigation of pressure signals and solid back-mixing in a full-loop circulating fluidized bed. *Industrial & Engineering Chemistry Research*, 56(3):799–813, 2017.
- [81] Salvatore Marrone, M. Antuono, A. Colagrossi, G. Colicchio, D. Le Touz, and G. Graziani. -SPH model for simulating violent impact flows. *Computer Methods in Applied Mechanics and Engineering*, 200(13):1526–1542, 2011.
- [82] Nicos S. Martys and Raymond D. Mountain. Velocity verlet algorithm for dissipative-particle-dynamics-based models of suspensions. *Physical Review E*, 59(3):3733, 1999.
- [83] Pierre Maruzewski, David Le Touz, Guillaume Oger, and Franois Avellan. SPH high-performance computing simulations of rigid solids impacting the free-surface of water. *Journal of Hydraulic Research*, 48:126–134, 2010.
- [84] Jean-Philippe Matas, Sylvain Marty, Mohamed Seydou Dem, and Alain Cartellier. Influence of gas turbulence on the instability of an air-water mixing layer. *Physical review letters*, 115(7):074501, 2015.
- [85] Viktor Mechtcherine, Annika Gram, Knut Krenzer, Jrg-Henry Schwabe, Sergiy Shyshko, and Nicolas Roussel. Simulation of fresh concrete flow using discrete element method (DEM): theory and applications. *Materials and structures*, 47(4):615–630, 2014.
- [86] Guy Metcalfe, Troy Shinbrot, J. J. McCarthy, and Julio M. Ottino. Avalanche mixing of granular solids. *Nature*, 374(6517):39–41, March 1995.
- [87] Maher Moakher, Troy Shinbrot, and Fernando J. Muzzio. Experimentally validated computations of flow, mixing and segregation of non-cohesive grains in 3d tumbling blenders. *Powder Technology*, 109(1):58–71, 2000.
- [88] J. J. Monaghan. Smoothed particle hydrodynamics and its diverse applications. *Annual Review of Fluid Mechanics*, 44:323–346, 2012.
- [89] J. J. Monaghan, R. A. F. Cas, A. M. Kos, and M. Hallworth. Gravity currents descending a ramp in a stratified tank. *Journal of Fluid Mechanics*, 379:39–69, 1999.

- [90] J. J. Monaghan and R. A. Gingold. Shock simulation by the particle method SPH. *Journal of computational physics*, 52(2):374–389, 1983.
- [91] J. J. Monaghan and H. Pongracic. Artificial viscosity for particle methods. *Applied Numerical Mathematics*, 1(3):187–194, 1985.
- [92] Joe J. Monaghan. Smoothed particle hydrodynamics. *Annual review of astronomy and astrophysics*, 30(1):543–574, 1992.
- [93] Joe J. Monaghan. Simulating free surface flows with SPH. *Journal of computational physics*, 110(2):399–406, 1994.
- [94] Joseph J. Monaghan and John C. Lattanzio. A refined particle method for astrophysical problems. *Astronomy and astrophysics*, 149:135–143, 1985.
- [95] Yoshitsugu Muguruma, Toshitsugu Tanaka, and Yutaka Tsuji. Numerical simulation of particulate flow with liquid bridge between particles (simulation of centrifugal tumbling granulator). *Powder Technology*, 109(1):49–57, 2000.
- [96] D. My-Ha, K. M. Lim, B. C. Khoo, and K. Willcox. Real-time optimization using proper orthogonal decomposition: Free surface shape prediction due to underwater bubble dynamics. *Computers & fluids*, 36(3):499–512, 2007.
- [97] Raymond H. Myers, Douglas C. Montgomery, G. Geoffrey Vining, Connie M. Borrer, and Scott M. Kowalski. Response surface methodology: a retrospective and literature survey. *Journal of quality technology*, 36(1):53, 2004.
- [98] Helio A. Navarro and Meire P. de Souza Braun. Determination of the normal spring stiffness coefficient in the linear springdashpot contact model of discrete element method. *Powder technology*, 246:707–722, 2013.
- [99] Trung Dac Nguyen and Steven J. Plimpton. Accelerating dissipative particle dynamics simulations for soft matter systems. *Computational Materials Science*, 100:173–180, 2015.
- [100] Daisuke Nishiura, Mikito Furuichi, and Hide Sakaguchi. Computational performance of a smoothed particle hydrodynamics simulation for shared-memory parallel computing. *Computer Physics Communications*, 194:18–32, 2015.

- [101] Seungtaik Oh and Il Kwon Jeong. Single-phase trapped air simulation in water flow. 2014.
- [102] Chong Peng, Guofang Xu, Wei Wu, Hai-sui Yu, and Chun Wang. Multiphase SPH modeling of free surface flow in porous media with variable porosity. *Computers and Geotechnics*, 81:239–248, 2017.
- [103] Li Peng, Ji Xu, Qingshan Zhu, Hongzhong Li, Wei Ge, Feiguo Chen, and Xinxin Ren. GPU-based discrete element simulation on flow regions of flat bottomed cylindrical hopper. *Powder Technology*, 304:218–228, 2016.
- [104] Gerald G. Pereira, Mahesh Prakash, and Paul W. Cleary. SPH modelling of fluid at the grain level in a porous medium. *Applied Mathematical Modelling*, 35(4):1666–1675, 2011.
- [105] Steve Plimpton. Fast parallel algorithms for short-range molecular dynamics. *Journal of computational physics*, 117(1):1–19, 1995.
- [106] Steve Plimpton, Roy Pollock, and Mark Stevens. Particle-mesh ewald and rRESPA for parallel molecular dynamics simulations. In *PPSC*. Citeseer, 1997.
- [107] M. Poux, P. Fayolle, J. Bertrand, D. Bridoux, and J. Bousquet. Powder mixing: some practical rules applied to agitated systems. *Powder Technology*, 68(3):213–234, 1991.
- [108] Mahesh Prakash, Paul W. Cleary, Soon Hyoun Pyo, and Fletcher Woolard. A new approach to boiling simulation using a discrete particle based method. *Computers & Graphics*, 53:118–126, 2015.
- [109] Dimitris C. Psychogios and Lyle H. Ungar. A hybrid neural network-first principles approach to process modeling. *AIChE Journal*, 38(10):1499–1511, 1992.
- [110] Stefan Radl, Eva Kalvoda, Benjamin J. Glasser, and Johannes G. Khinast. Mixing characteristics of wet granular matter in a bladed mixer. *Powder Technology*, 200(3):171–189, 2010.
- [111] M. D. Rintoul and Salvatore Torquato. Computer simulations of dense hard-sphere systems. *The Journal of chemical physics*, 105(20):9258–9265, 1996.

- [112] Martin Robinson, Marco Ramaioli, and Stefan Luding. Fluidparticle flow simulations using two-way-coupled mesoscale SPHDEM and validation. *International journal of multiphase flow*, 59:121–134, 2014.
- [113] Amanda Rogers and Marianthi G. Ierapetritou. Discrete element reduced-order modeling of dynamic particulate systems. *AIChE Journal*, 60(9):3184–3194, 2014.
- [114] Reinout Romijn, Leyla zkan, Siep Weiland, Jobert Ludlage, and Wolfgang Marquardt. A grey-box modeling approach for the reduction of nonlinear systems. *Journal of Process Control*, 18(9):906–914, 2008.
- [115] E. Rougier, A. Munjiza, and N. W. M. John. Numerical comparison of some explicit time integration schemes used in DEM, FEM/DEM and molecular dynamics. *International journal for numerical methods in engineering*, 61(6):856–879, 2004.
- [116] Maysam Saidi, Hassan Basirat Tabrizi, John R. Grace, and C. Jim Lim. Hydrodynamic investigation of gas-solid flow in rectangular spout-fluid bed using CFD-DEM modeling. *Powder Technology*, 284:355–364, 2015.
- [117] M. F. Saleh. Characterisation of foam bitumen quality and the mechanical properties of foam stabilised mixes. 2006.
- [118] Charles W. Schwartz and Sadaf Khosravifar. Design and evaluation of foamed asphalt base materials, 2013.
- [119] Songdong Shao. Incompressible SPH flow model for wave interactions with porous media. *Coastal Engineering*, 57(3):304–316, 2010.
- [120] Xuqiang Shao, Zhong Zhou, and Wei Wu. Particle-based simulation of bubbles in watersolid interaction. *Computer Animation and Virtual Worlds*, 23(5):477–487, 2012.
- [121] J. Shfer, S. Dippel, and D. E. Wolf. Force schemes in simulations of granular materials. *Journal de physique I*, 6(1):5–20, 1996.
- [122] Leonardo Di G. Sigalotti, Jaime Klapp, Eloy Sira, Yasmin Melen, and Anwar Hasmy. SPH simulations of time-dependent poiseuille flow at low reynolds numbers. *Journal of computational physics*, 191(2):622–638, 2003.

- [123] S. J. R. Simons and R. J. Fairbrother. Direct observations of liquid binderparticle interactions: the role of wetting behaviour in agglomerate growth. *Powder Technology*, 110(1):44–58, 2000.
- [124] Barbara Solenthaler, Jrg Schlfi, and Renato Pajarola. A unified particle model for fluidsolid interactions. *Computer Animation and Virtual Worlds*, 18(1):69–82, 2007.
- [125] Matthew Spellings, Ryan L. Marson, Joshua A. Anderson, and Sharon C. Glotzer. GPU accelerated discrete element method (DEM) molecular dynamics for conservative, faceted particle simulations. *Journal of Computational Physics*, 334:460–467, 2017.
- [126] Q. Spreiter and M. Walter. Classical molecular dynamics simulation with the velocity verlet algorithm at strong external magnetic fields. *Journal of Computational Physics*, 152(1):102–119, 1999.
- [127] Paul J. Steinhardt, David R. Nelson, and Marco Ronchetti. Bond-orientational order in liquids and glasses. *Physical Review B*, 28(2):784, 1983.
- [128] Xiaosong Sun and Mikio Sakai. Three-dimensional simulation of gassolidliquid flows using the DEMVOF method. *Chemical Engineering Science*, 134:531–548, 2015.
- [129] Xiaosong Sun, Mikio Sakai, and Yoshinori Yamada. Three-dimensional simulation of a solidliquid flow by the DEMSPH method. *Journal of Computational Physics*, 248:147–176, 2013.
- [130] William C. Swope, Hans C. Andersen, Peter H. Berens, and Kent R. Wilson. A computer simulation method for the calculation of equilibrium constants for the formation of physical clusters of molecules: Application to small water clusters. *The Journal of Chemical Physics*, 76(1):637–649, 1982.
- [131] Mandar V. Tabib and Jyeshtharaj B. Joshi. Analysis of dominant flow structures and their flow dynamics in chemical process equipment using snapshot proper orthogonal decomposition technique. *Chemical Engineering Science*, 63(14):3695–3715, 2008.
- [132] P. A. Tanguy, F. Thibault, C. Dubois, and A. Ait-Kadi. Mixing hydrodynamics in a double planetary mixer. *Chemical Engineering Research and Design*, 77(4):318–324, 1999.

- [133] Qi Tong, Siyu Zhu, and Huiming Yin. Scale-up study of high-shear fluid-particle mixing based on coupled SPH/DEM simulation. *Granular Matter*, 20(3):34, 2018.
- [134] Salvatore Torquato, Thomas M. Truskett, and Pablo G. Debenedetti. Is random close packing of spheres well defined? *Physical review letters*, 84(10):2064, 2000.
- [135] K. A. Triplett, S. M. Ghiaasiaan, S. I. Abdel-Khalik, and D. L. Sadowski. Gasliquid two-phase flow in microchannels part i: two-phase flow patterns. *International Journal of Multiphase Flow*, 25(3):377–394, 1999.
- [136] M. A. Van der Hoef, M. Ye, M. van Sint Annaland, A. T. Andrews, S. Sundaresan, and J. A. M. Kuipers. Multiscale modeling of gas-fluidized beds. *Advances in chemical engineering*, 31:65–149, 2006.
- [137] Amit Varshney and Antonios Armaou. Reduced order modeling and dynamic optimization of multiscale PDE/kMC process systems. *Computers & Chemical Engineering*, 32(9):2136–2143, 2008.
- [138] Amit Varshney, Sivakumar Pitchaiah, and Antonios Armaou. Feedback control of dissipative PDE systems using adaptive model reduction. *AIChE journal*, 55(4):906–918, 2009.
- [139] Otis R. Walton and Robert L. Braun. Viscosity, granular-temperature, and stress calculations for shearing assemblies of inelastic, frictional disks. *Journal of Rheology*, 30(5):949–980, 1986.
- [140] G.Q. Wang and J.R. Ni. The kinetic theory for dilute solid/liquid two-phase flow. *International Journal of Multiphase Flow*, 17(2):273–281, 1991-03.
- [141] Yingrui Wang, Leisheng Li, Jingtao Wang, and Rong Tian. GPU acceleration of smoothed particle hydrodynamics for the navier-stokes equations. In *Parallel, Distributed, and Network-Based Processing (PDP), 2016 24th Euromicro International Conference on*, pages 478–485. IEEE, 2016.
- [142] Patrick B. Warren. Dissipative particle dynamics. *Current opinion in colloid & interface science*, 3(6):620–624, 1998.
- [143] Xilin Xia and Qiuhua Liang. A GPU-accelerated smoothed particle hydrodynamics (SPH) model for the shallow water equations. *Environmental Modelling & Software*, 75:28–43, 2016.

- [144] Qingang Xiong, Bo Li, and Ji Xu. GPU-accelerated adaptive particle splitting and merging in SPH. *Computer Physics Communications*, 184(7):1701–1707, 2013.
- [145] Lingqi Yang. *Particle dynamics simulation of microstructure evolution towards functionally graded material manufacturing*. Columbia University, 2015.
- [146] Lingqi Yang and Huiming Yin. Parametric study of particle sedimentation by dissipative particle dynamics simulation. *Physical Review E*, 90(3):033311, 2014.
- [147] Xiaoqiang Yue, Hao Zhang, Chunhai Ke, Congshu Luo, Shi Shu, Yuanqiang Tan, and Chunsheng Feng. A GPU-based discrete element modeling code and its application in die filling. *Computers & Fluids*, 110:235–244, 2015.
- [148] G. Zhou, P. A. Tanguy, and C. Dubois. Power consumption in a double planetary mixer with non-newtonian and viscoelastic materials. *Chemical Engineering Research and Design*, 78(3):445–453, 2000.
- [149] H. P. Zhu, Z. Y. Zhou, R. Y. Yang, and A. B. Yu. Discrete particle simulation of particulate systems: theoretical developments. *Chemical Engineering Science*, 62(13):3378–3396, 2007.
- [150] Yi Zhu, Patrick J. Fox, and Joseph P. Morris. A pore-scale numerical model for flow through porous media. *International journal for numerical and analytical methods in geomechanics*, 23(9):881–904, 1999.

Part II

Appendices

A Parallel Computing on Message Passing Interface(MPI)

A.1 Amdahl's Law

In computer science, Amdahl's Law is a formula which gives the theoretical maximum speedup for a computer program in parallel computing. It should be noted that in parallel computing, there is always some parts of the program that must do in series, and the proportion of parallel program p is smaller than 1. For example, if 10% of the program cannot be paralleled (10% means that this part of program consumes 10% of the time in serial computing of the whole program), then $p = 0.9$.

When the program is parallel with n processors, if we neglect communication times and assume n processors speedup perfectly n times in the parallel part, then the total speedup for the program including serial computing and parallel computing is:

$$S = \frac{1}{1 - p + p/n} \quad (8.1)$$

When n goes to infinite, the upper bound of theoretical speed up is obtained

$$S_{max} = \frac{1}{1 - p} \quad (8.2)$$

Amdahl's Law provides us important information that using more cores to compute the problem does not always bring significant speedup, if only limited part of the program could be paralleled. The value of p is the key factor to choose a reasonable number of cores to parallel the computation. Fortunately, mixing problem is highly parallelable and the value of p is really large (larger than 0.99), and it will be proved by case study below.

A.2 Acceleration Study

In this part, the same particle mixing problem is conducted with various number of cores within finite time steps to test the time consumed in computation. The speedup curve with CPU cores used are plotted and compared. In this part, 240000 steps of simulation is extracted from the beginning stage of previous Double Planetary Mixer simulation, which has 20476 particles of both liquid and solid. Since the nature of the simulation problem is uniform distributed horizontally but not vertically, the simulation domain is cut in a way that both x and y directions are divided

uniformly but in z direction all the processors share the same height. During the simulation, the particles all distribute at the bottom of the vessel and the height of all the particles are much less than the length of both x and y directions, so in z direction the domain is not divided. The specific dividing plan for different CPU cores are listed in Table. 8.1.

Table 8.1: Domain Decomposition Plan for different number of CPU Cores

Number of Cores	Domain Decomposition
1	1
2	2 1 1
4	2 2 1
8	4 2 1
16	4 4 1
32	8 4 1
64	8 8 1
128	16 8 1
256	16 16 1
512	32 16 1
1024	32 32 1

The results of computational time spent at different numbers of cores are shwon in Table. 8.2 and Fig. 8.1.

It is surprising that in most cases, the speedup value of the simulation is larger than the number of cores used in computation. PDPS is written in a way that no matter how many cores are used, particle transfer is always implemented, and if there is only one core, then this processor will transfer particle close to boundary with itself. This communication process may slowdown the speed which cause the speedup larger than the number of cores. As the number of cores increases, the computation time for each core decrease, but the communication time between processors increase. When the number of cores reach 1024, its speedup starts to decrease, since communication is in series and using more cores does not always produces faster result. In this problem, using 512 cores is the best among the cases.

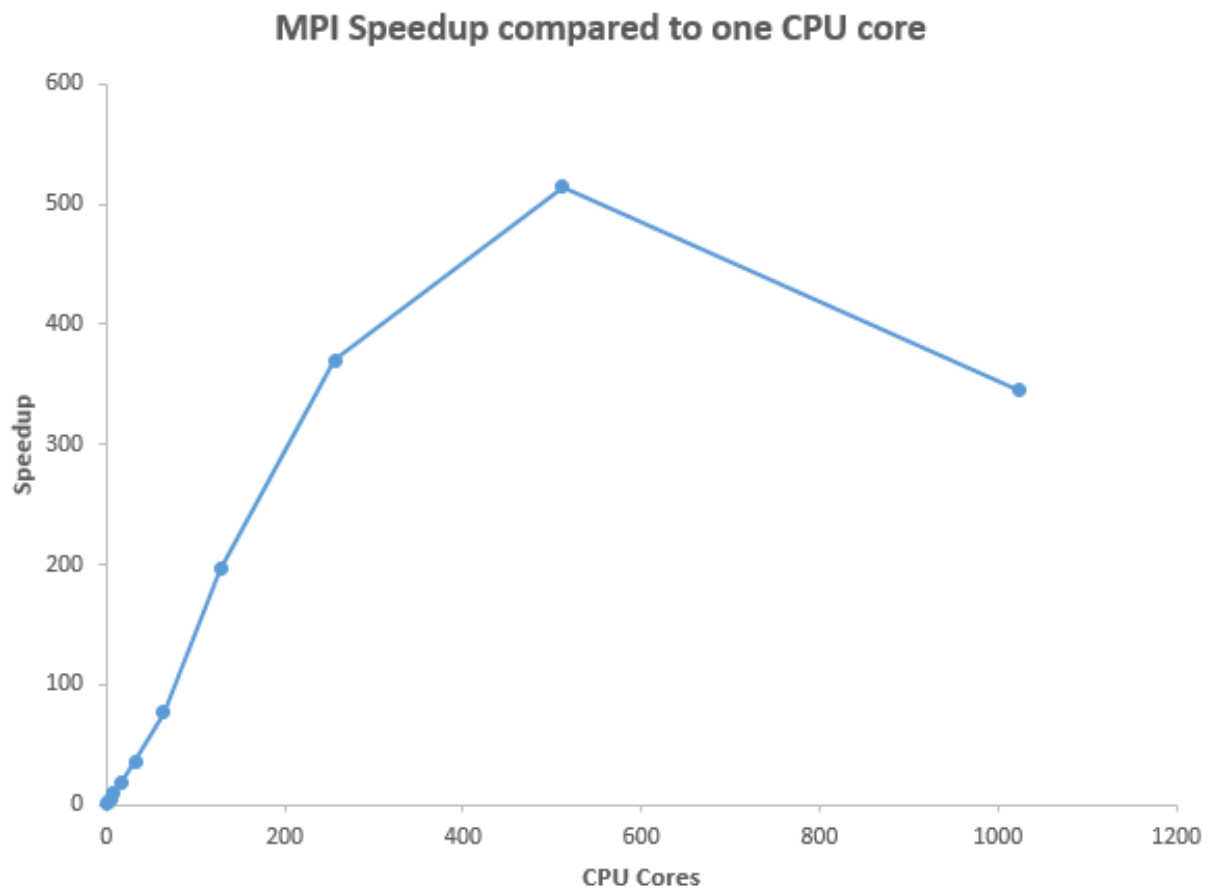


Figure 8.1: MPI speedup with multiple CPU cores compared to one CPU core

Table 8.2: Computation Time Comparison with Different Number of CPU Cores

Number of Cores	Time Spent (s)	Speedup
1	127242	1
2	54456.3	2.3
4	24595.5	5.2
8	13685.5	9.3
16	7162.57	17.8
32	3525.89	36.1
64	1667.76	76.3
128	649.414	195.9
256	343.515	370.4
512	247.138	514.9
1024	368.741	345.1

B General-Purpose Computing on Graphics Processing Units (GPU)

B.1 Overview

Efficient large-scale modeling remains a key issue for all kinds of particle simulations, and the GPGPU has been attracting extensive interests recently to enable real-world simulations. A GPU board have thousands of cores for computation, which can tremendously speed up the computation if managed correctly. GPGPU is a shared memory computation system, in which a GPU board has thousands of cores sharing the same memory. Within all the cores in GPU, every 32 cores form a block which has its own local memory only accessible to these 32 cores and each core (sometimes called thread in computation) also has its own cache.

In GPU computation, CPU is still necessary to give command. Besides, one cannot directly watch the value of variables in GPU memory, thus the data transfer between CPU and GPU is crucial to the accuracy and efficiency of GPU computation.

Programming in GPU is not trivial, and there are many details to be considered carefully to obtain the desirable efficiency. In this work, CUDA C++ is implemented for GPU computation in addition to the original C++ programming language. CUDA C++ is a package based on C++

and designed specifically for GPU programming by NVIDIA.

B.2 Memory Allocation in Parallel Computation

In parallel computing, depending on the memory usage in computing process, there are mainly two types of parallel scheme: distributed memory parallelization and shared memory parallelization.

Distributed memory parallelization refers to a multiprocessor computer system, in which each processor or core has its own memory and only access its own memory during the computation. The computation task is usually cut to small pieces of work for each processor, and all the processors finish their own work and communicate with each other if there is a need to transfer data. One commonly used distributed memory parallelization is Message Passing Interface (MPI), which is applied to our software package PDPS and used widely in particle mixing problem.

In shared memory parallelization, all threads reach data in one place and no communication is needed. GPU parallelization is a shared memory parallel system, in which thousands of threads all share the same reading and writing access to global data. However, in a GPU board it has more hierarchy for data memory. Except the global data every thread can access, every thread has its own cache and every block of threads (usually there are 32 threads in a block) has a local shared memory only they can access. The local memory in each block has faster access speed than global memory, and it is recommended to put frequently used variable to local memory to reduce data access time.

B.3 Race Condition

Race condition is a situation happened in parallel computing when multiple threads tend to access the same data. If they are not arranged in right order, race condition would lead to error in computation which will not occur in serial computing. One common way to avoid race condition is to make constrain to all the threads that only one can access the data. In parallel computing, it is inevitable that we are faced with situations where multiple threads accessing the same data, and the speed will be reduced since the data could only be accessed by one thread at one time in series if the race condition reaches. In higher level, we try to avoid this situation, parallel as much as possible, but small problem could be paralleled completely as the race condition may not happen.

B.4 GPU Parallel Algorithm of PDPS

General Structure of GPU-Parallelized Particle Simulation

This section presents the structure of GPU-parallelized particle simulation. As shown in Fig 8.2, the system is initialized and all variables are stored in CPU at the beginning of the simulation. Then, all the data are transferred from CPU memory to GPU memory before the main time step loop starts. All the computations, associated with neighbor list construction, particle interactions, and system update, are performed on GPU memory with GPU-parallelized algorithms at different time steps without requiring any communication between GPU and CPU. The code structure is designed to minimize the communication between CPU and GPU since this part is pretty time consuming, and the communication is only executed from GPU to CPU if any result is required. Data transfer from GPU memory to CPU memory is necessary, since the data in the GPU memory are not directly accessible for outputting. This structure keeps the original flow in CPU-only computation and minimizes the data transfer between CPU and GPU.

Neighbor List Construction Speedup

While the majority of this particle simulation algorithm remains the same when transferred to GPU, the particle neighbor lists implemented in CPU and GPU systems are different.

As mentioned before, in CPU-only computation, to improve computational efficiency, a half neighbor list is built. However, in GPU computation, multiple threads simultaneously compute the particle interaction forces and it is possible that two neighbor particles are computed simultaneously, and during this condition if a half neighbor list is implemented, one variable may be altered by two threads, which would lead to a computational error. Therefore, in GPU computation, a complete neighbor list is built to guarantee that the computational result is correct.

With half neighbor list changed to a complete neighbor list, the force computation algorithm should also be adjusted accordingly. The force computation for coupled SPH-DEM has been modified from a half neighbor list based to complete neighbor list based, which means in GPU parallelization the interaction force between each pair of particles i and j are computed twice separately.

In addition to the above-mentioned differences of the half neighbor list, the algorithm of neighbor

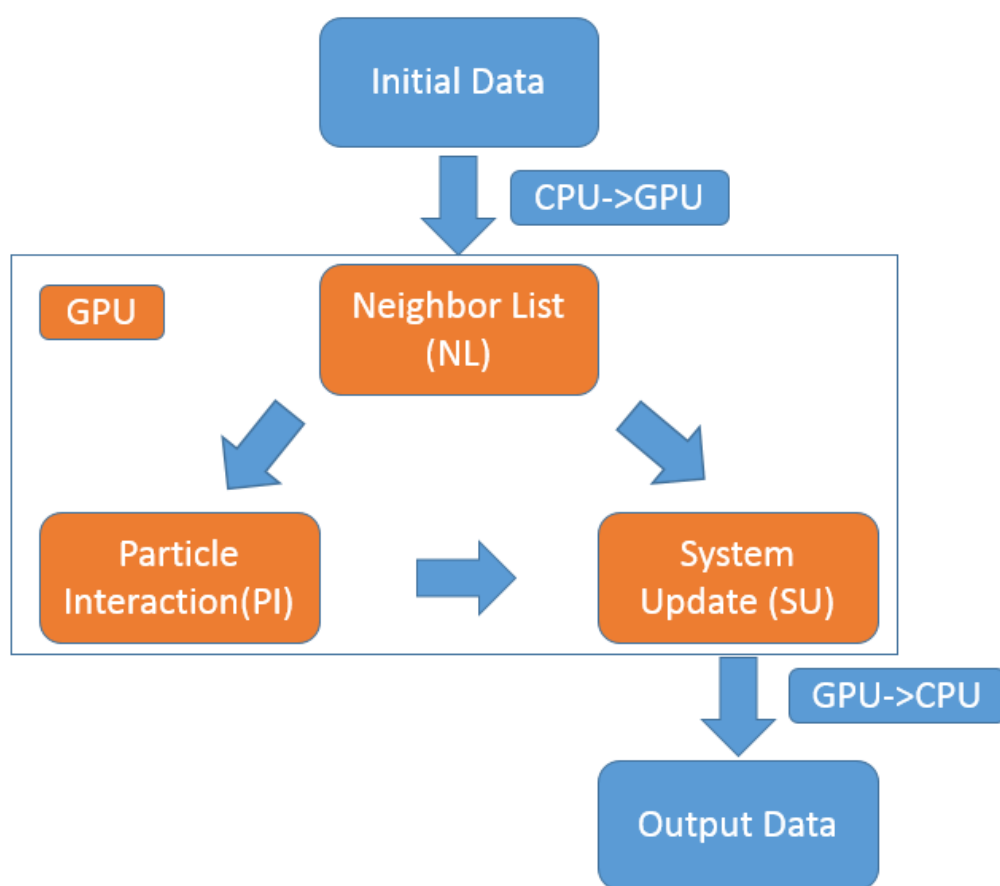


Figure 8.2: Program Flow with GPU Computation

list construction should be mentioned. The neighbor list construction consists of two steps, building a local linked list and constructing neighbor list from the local linked list. Since the number of particles is usually large, if one directly loops through all the particles to build the neighbor list, many unnecessary computations of particles, which lie far away from each other, is conducted, which is time-consuming. A general solution is to build a local linked list first, to sort particles that share similar location together, and then the neighbor list could be constructed from the local linked list. However, the local linked list could not be parallelized so far because if so it will introduce the problem of race condition. Fortunately, neighbor list construction only takes a small portion of time among the total computations, and it will not influence the final speedup significantly.

Particle Pair Force Computation Speedup

Based on the change on neighbor list, the algorithm of particle pair force which involves neighbor list changes accordingly, while others remain similarly to the CPU-only computation. In GPU computation, each thread is in charge of the computation of one particle, and it loops through all its neighbor particles to calculate the forces between it and its neighbor particles. Even though the interaction force between two different particles is calculated twice separately, it is guaranteed the neighbor list information are stored in different parts of the complete neighbor list and the forces are added to different particles, thus no variable is accessed simultaneously by multiple threads.

During the particle pair force computation, particle force coefficients used constantly are stored in local memory to reduce its memory access time. Usually there are only a few force coefficients, and the limited local memory is enough to store these variables.

External Force Computation and System Update Speedup

To compute external forces such as body force and boundary force, each thread is in charge of the associated computations for one particle since the external force computation for different particles are independent from each other. Similarly to particle pair force computation, the frequently used parameters such as gravity would be stored in local shared memory. Updating the position and velocity of each particle is performed by assigning one thread to one particle since each computation is also independent from others.

B.5 Numerical Study and Results

Simulation Setup

In this work, a mixing case study of two types of solid particles and one type of liquid particle in a DPM is conducted in both CPU-only computation and GPU computation.

During the simulation process, two kinds of solid particles with same properties and one kind of liquid particle are generated at the beginning at the bottom of the container. To pack these particles tightly at the bottom, they experience a free falling process at the beginning from their initial position. Then double planetary blades are generated as clusters of relatively fixed solid particles, which share the same properties with solid particles to be mixed. After the particles reach a stable status, the DPM moves down from above and merges into the particles, as shown in Fig. 8.3

The parameters used for particle interactions are shown in Table. 8.3.

Then it starts to rotate with a constant acceleration rate. Two groups of solid particles together with one group of liquid particles are mixed towards homogeneous until the rotation of the blade stopped. One complete simulation is finished with GPU computation, and another short simulation that consists of 120000 steps with time step 0.00025s is conducted on both CPU-only computation and GPU computation for comparison. Their speed comparison is shown in the following section.

Acceleration Analysis

The computation time for simulations using one CPU-only and both CPU and GPU is listed in Table. 8.4 and Figure. 8.4. The CPU core used is Intel Xeon CPU E5-2620 v3, and the GPU board is Tesla K40C. Pair force computation, which is the force computation between particles, spends most of the time. If this part could be reduced significantly, the total speedup would be improved accordingly. In our algorithm this pair force computation speeds up almost 56.6 times. It should be noted that since different neighbor list structures are used for CPU and GPU, the actual work implemented in this case study for pure CPU computation and GPU computation is different. The one in pure CPU computation only needs to compute half the amount of work as the work in GPU computation. In this condition the speedup of 56.6 times is a significant. Another part which has been speed up tremendously is the boundary force computation, which is more than

Table 8.3: Parameters of solid and liquid particles in DPM simulation

Parameter	Solid
Diameter	$10mm$
Density	$1000kg/m^3$
Mass	$0.5236g$
Number	19440
Normal Stiffness	$7 \times 10^7 N/m$
Tangential Stiffness	$2 \times 10^7 N/m$
Normal Damping Coefficient	$8000N \cdot s/m$
Tangential Damping Coefficient	$4000N \cdot s/m$
Coefficient of restitution	0.8
Friction coefficient	0.5
Parameter	Liquid
Diameter	$10mm$
Density	$1000kg/m^3$
Mass	$1g$
Number	1060
speed of sound	$30m/s$
viscosity coefficient	$0.2m^2/s$

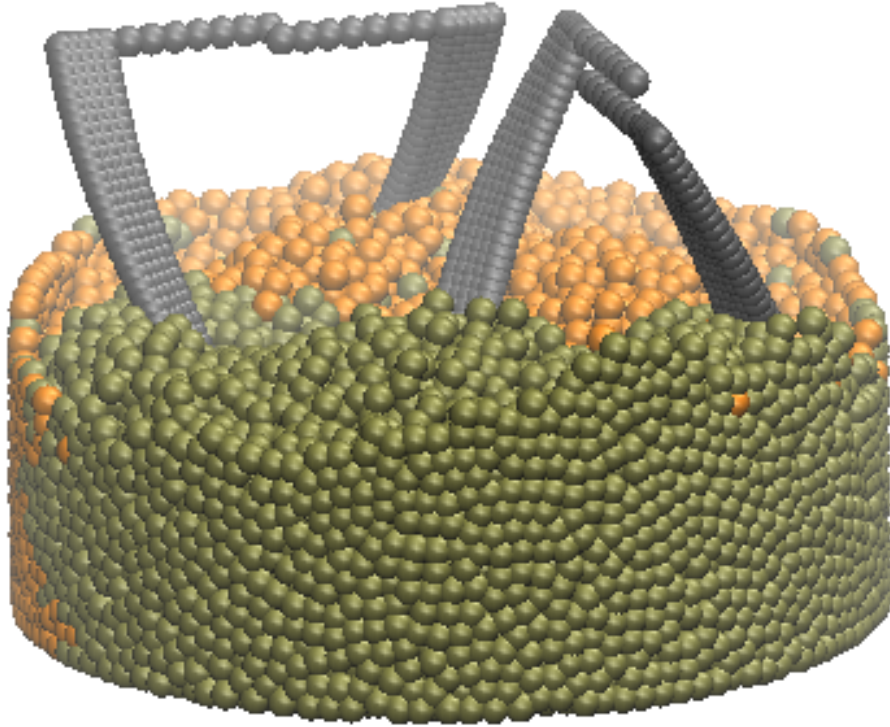


Figure 8.3: Initial state of mixing process

one thousand times. Since boundary force is much more easy to parallel, it should achieve higher speedup. Neighbor list construction speeds up 36.9 times speedup, because part of neighbor list construction is difficult to parallel. Among the remaining parts, initialization and output are only executed in CPU, so the general speedup of other parts is 4.3 in total. Generally, the total speedup of the mixing case study is around 60 times, which is very promising.

Table 8.4: Time Comparison of CPU-only and GPU Computation

Computation	Time on CPU(s)	Time on GPU(s)	SpeedUp
Pair Force	96539	1705	56.6
Boundary Force	7738	6	1290
Neighbor List	1918	52	36.9
Others	349	82	4.3
Total	106544	1845	57.7

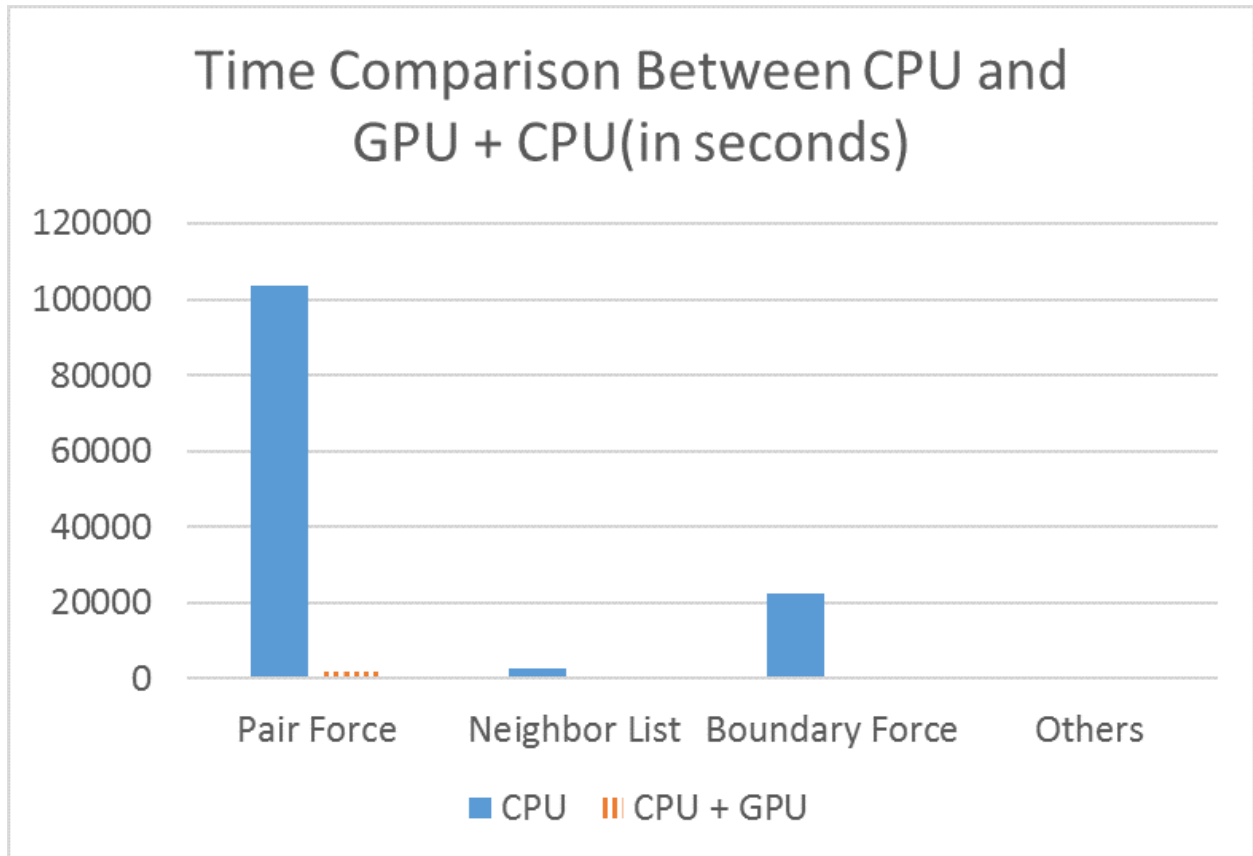


Figure 8.4: Time Comparison of CPU and both CPU + GPU

B.6 Discussion

In order to achieve high efficiency using the GPGPU, one needs to reduce data transfer between CPU and GPU, allocate variable memory in GPU properly, and use appropriate algorithms to parallelize the program. To reduce data transfer, all the data are transferred from CPU memory to GPU memory after initialization, and they transfer back to CPU memory only when they are required. To allocate memory properly, local memory in GPU is used and the frequently used variables such as particle pair force coefficients are stored in GPU local memory. As for GPU computation algorithms, different parts take different strategies. The most simple strategy is in body force computation and system update, where each thread is in charge of one particle, and they loop over all the particles. For neighbor list construction, it is difficult to make the process parallelized, so a linked list is built first serially, and then it is paralleled based on the linked list. Some of the remaining part must be implemented in CPU and cannot be parallelized, but fortunately, these parts do not take too much computing time during the whole process.

Then the GPU acceleration algorithm is applied to a solid-liquid two-phase mixing case study. Among all the computation components, pair force computation take almost 80% of the time, so the speedup of particle pair force almost determines the total speedup performance. A 56.6 times speedup is achieved for particle pair force computation, leading to a 57.7 times speedup for total simulation, which is a reasonable result.

The speedup of GPU would also be influenced by the CPU and GPU applied for computation, since the differences of computational speed between different CPUs or GPUs are significant. GPU has more advantages for larger computation problems since it has more memories than CPU. When the number of particles simulated reaches a certain level, CPU is unable to store such a large volume of data, while GPU still has enough memory. On a general desktop, there are only dozens of CPUs, so the speedup effect is not very significant using CPUs on desktop. In this condition, using an efficient GPU board on a desktop would achieve satisfying speedup effect compared to CPU parallelism. CPU parallelism is usually implemented on supercomputers, where thousands of cores could be used. If no resources of supercomputers are available, using GPU on desktop is more convenient to provide fast results.

Parallel computing on CPU is usually implemented on supercomputers, since the speedup effect with thousands of cores on supercomputers is more significant compared to dozens

In future, more advanced GPU would be implemented to increase the speedup performance of GPU parallelism.

C Model Reduction

Though previous research, it is noticed that DEM-SPH coupling simulation can be very time consuming. There is a great need to speed up the simulation or develop a surrogate-simplified model. The work to speedup is contained in the development of MPI parallel computing or GPU acceleration, and a surrogate-simplified model based on statistical method is proposed to save time.

Generally speaking research models can be classified to three types: white-box (Analytical models), grey-box (Semi experimental models) [109; 20; 65; 114] and black-box (Empirical models) [60; 97; 39; 68]. White-box model is implemented when the simulation is conducted from basic physical laws such as momentum conservation, and the whole process is investigated clear. Black-box model is a prediction of the results to explore based on existing experimental or simulation data, and the process is a mystery to the researcher just as something in a black box could not be seen though, and the only thing can get is the results. Grey-box is a mixture of both, in which some parts are known and others are not. The understanding in these types of model is best in white box model, least in black model, and their computational speed is vice versa. The simulation process developed previously in this work is a white box model, and is time consuming. Here the surrogate prediction model is a time efficient black box model.

The prerequisite of the surrogate black box model is to have multiple simulations results available, and choose the target property in terms of mixing index. Once the simulation results of some parameters are obtained, results of other parameter combinations can be predicted with current result data [14; 113]. In this part a surrogate black box model is developed to predict mixing index and mixture density profile.

C.1 Model Reduction on Mixing Index

Kriging Regression

Kriging is one of the regression method [32; 68; 72], which can be regarded as a more advanced linear regression method. Linear regression method predicts new data with the linear combina-

tion of parameters and minimize its variance. Kriging method also predicts new data with linear combination, but it uses linear combination of existing data point, not parameters.

The idea is to predict result as the weight combination of existing data result:

$$f(x_k) = \sum_{i=1}^n w_i f(x_i) \quad (8.3)$$

$$\sum_{i=1}^n w_i = 1 \quad (8.4)$$

The essential part in Kriging is the way to choose weight. And this is the flexible advantage of Kriging. It can distinguish the dependences between existing data and give less weight to similar results.

The variance of kriging formulation is shown below:

$$-Var(f(x_k) - f(x_i)) + \sum_{j=1}^M w_j Cov(f(x_j) - f(x_i)) + \lambda = 0 \quad (8.5)$$

where λ is a constant, and the covariance Cov is defined below:

$$Cov(h) = \sigma_{max} - \gamma(h) \quad (8.6)$$

where the variance γ could be predefined or formulated with existing data, and σ_{max} is the value of $\gamma(h)$ when $h = h_{max}$. Several choices of predefined γ is shown below [62]:

$$\gamma_{sph}(h) = \alpha_{sph} \begin{cases} (1.5 \frac{h}{h_{max}} - 0.5 (\frac{h}{h_{max}})^3) & h \leq h_{max} \\ 1 & h > h_{max} \end{cases} \quad (8.7)$$

$$\gamma_{gaus}(h) = \alpha_{gaus} (1 - \exp(-\frac{3h^2}{h_{max}^2})) \quad (8.8)$$

$$\gamma_{exp}(h) = \alpha_{exp} (1 - \exp(-\frac{3h^2}{h_{max}^2})) \quad (8.9)$$

The values of coefficients α_{sph} , α_{gaus} , and α_{exp} are chosen as 4, and h_{max} is chosen as 1 in these cases [62].

And the data driven variance is shown below:

$$\gamma(h) = \frac{1}{2N_{Clus}(h)} \sum_{N(h)} [f(x_i) - f(x_j)]^2 \quad (8.10)$$

where $N_{Clus}(h)$ is the number of pairs whose distances are within $[h - \epsilon, h + \epsilon]$, and ϵ is defined by user depending on the problem. Through this user defined variance definition, various data points could be assigned proper weights.

Model Reduction Results

For this model reduction method, a mixing case containing three types of particles and one liquid is conducted with different blade rotation speed at $0.4rad/s$, $0.5rad/s$, $0.6rad/s$, $0.7rad/s$, $0.8rad/s$, $0.9rad/s$, $1.0rad/s$. And the prediction of the mixing index in the whole process for blade speed at $0.75rad/s$ and $0.85rad/s$ are shown in Fig 8.5 and Fig 8.6:

Both linear regression and kriging regression showed good agreements with complete simulation results. For one kriging regression has not shown advantage over linear regression, and it remains to be seen at cases with multiple parameters.

C.2 Model Reduction on Mixing Status Profile

Principle Component Analysis(PCA)

Mixing index is just one variable thus it is relatively easy to predict. Fig 8.7 shows a particle mass density profile, which is another feature that can be used to test whether the structure has been mixed well or not. If the structure is mixed homogeneously, the density profile should be a plane, otherwise it would have rough surfaces.

Prediction of density profile is much more complicated since it has more variables than the mixing index. Theoretically, it is possible to predict all the variables in density profiles one by one. However, to be practical, a better approach is to find the minimum essential variables that contain the majority information of the density profile (for example, 95%). Usually the variables in a profile are not independent, for example, the density profile could be generated once the essential variables have been predicted, so that a lot of effort could be saved through this approach. During this process, the Principle Component Analysis(PCA) is implemented to save computational time.

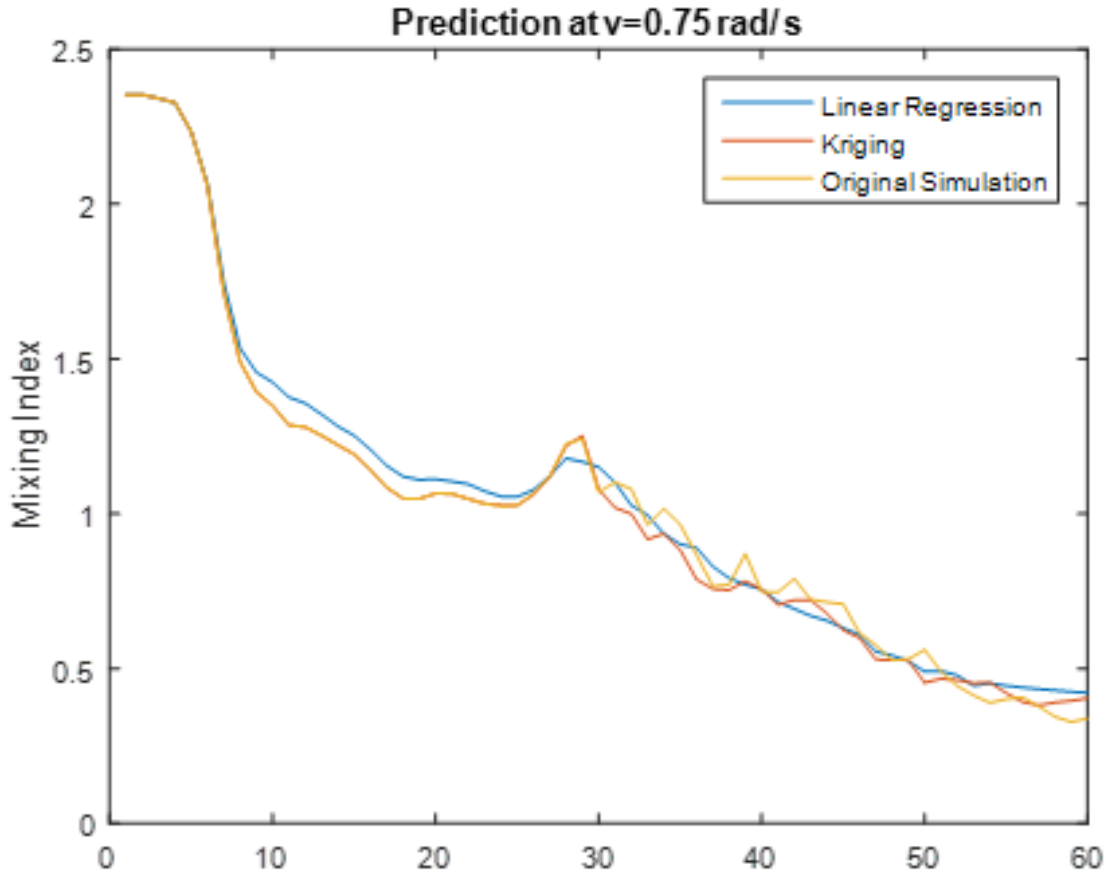


Figure 8.5: Comparison of mixing index for different regression method when $w = 0.75 \text{ rad/s}$

PCA is a statistical procedure that uses matrix transposition to convert a set of possibly correlated variables to a set of linearly uncorrelated variables, which is called principal components. PCA have been used in various disciplines for model reduction including image processing, signal analysis, structural analysis, data compression, process identification and control [43; 22; 75; 23; 7; 96; 131; 137; 138].

For a matrix Z , the purpose of PCA is to find a matrix W so that matrix T in the Equation. 8.11 has less dimensions than Z and all the components in T are linear independent.

$$T = ZW \quad (8.11)$$

The specific application will be illustrated in next section.

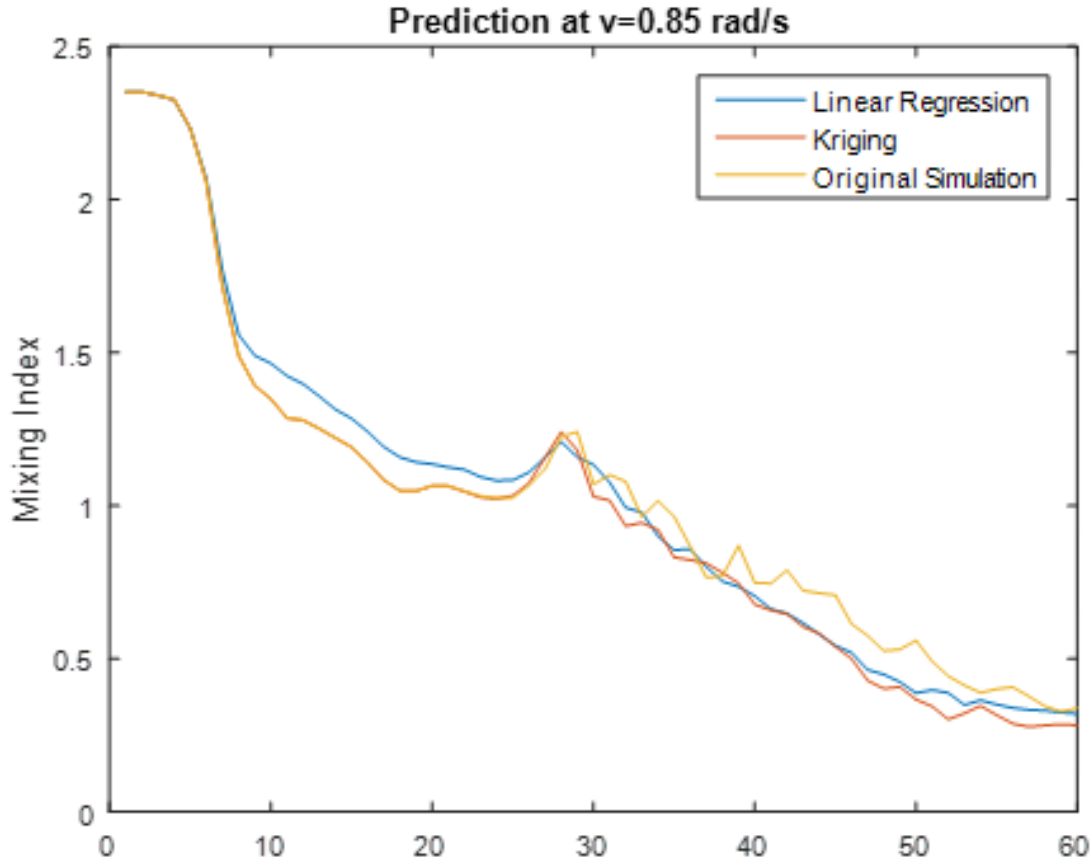


Figure 8.6: Comparison of mixing index for different regression method when $w = 0.85 \text{ rad/s}$

Model Reduction Results

The same case is studied as an example to show the PCA approach.

1. There are 36 case studies, 100 variables for density profile, and these data form a state matrix $Z_{36 \times 100}$. All the 36 case studies with 3 different parameters form a input matrix $X_{36 \times 3}$. List the two matrix and the problem is to find the relationship between $X_{36 \times 3}$ and $Z_{36 \times 100}$.
2. Implement PCA on matrix $Z_{36 \times 100}$ to obtain

$$Z_{36 \times 100} = T_{36 \times 35} P_{35 \times 100} + Z_{mean 36 \times 100} \quad (8.12)$$

Where $P_{35 \times 100}$ is 35 principal components, and $T_{36 \times 35}$ is the combination coefficients of the 35 principal components in 36 cases. $Z_{mean 36 \times 100}$ is the mean of 100 variables in 36 cases.

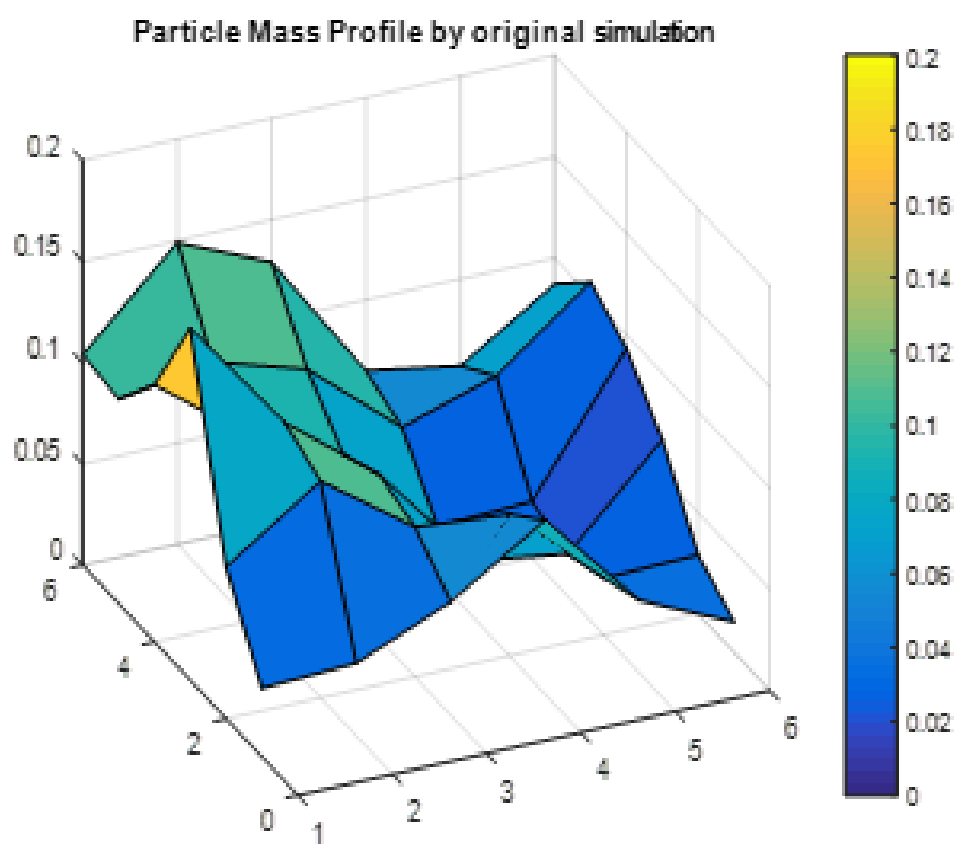


Figure 8.7: Mass density distribution during mixing process

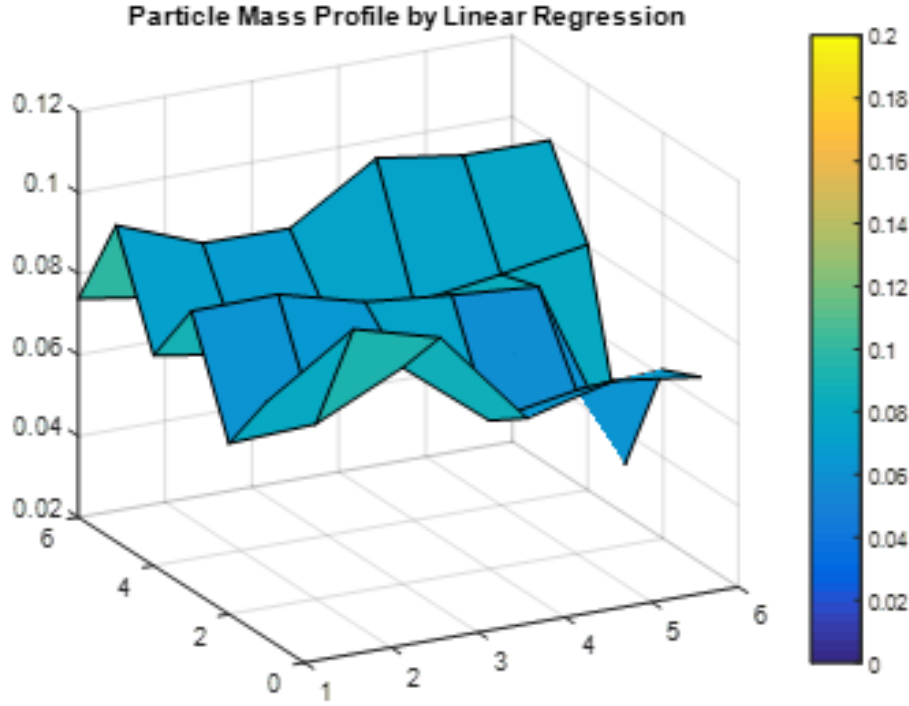


Figure 8.8: Mass density profile predicted by Linear regression

3. For new prediction case $Z_{new1 \times 100}$, instead to predict all the 100 variables of Z , only 35 coefficients $T_{1 \times 35}$ or less are to predict. Then multiplying the principle components $P_{35 \times 100}$ the final answer, the density profile could be obtained.
4. The prediction of $T_{new1 \times 35}$ could be linear regression or kriging regression of $X_{36 \times 3}$ and $T_{36 \times 35}$. For each coefficient in $T_{1 \times 35}$ one regression is implemented, so there are total 35 regressions to implement to obtain the new coefficients $T_{new1 \times 35}$. The variables we want can be expressed as:

$$Z_{new1 \times 100} = T_{new1 \times 35} P_{35 \times 100} + Z_{mean36 \times 100} \quad (8.13)$$

Based on the PCA methods, the density profile can be predicted by both the linear and kriging regressions, which are provided in Fig 8.8 and Fig 8.9, respectively. It shows that, in this density profile prediction, kriging gives better results than linear regression.

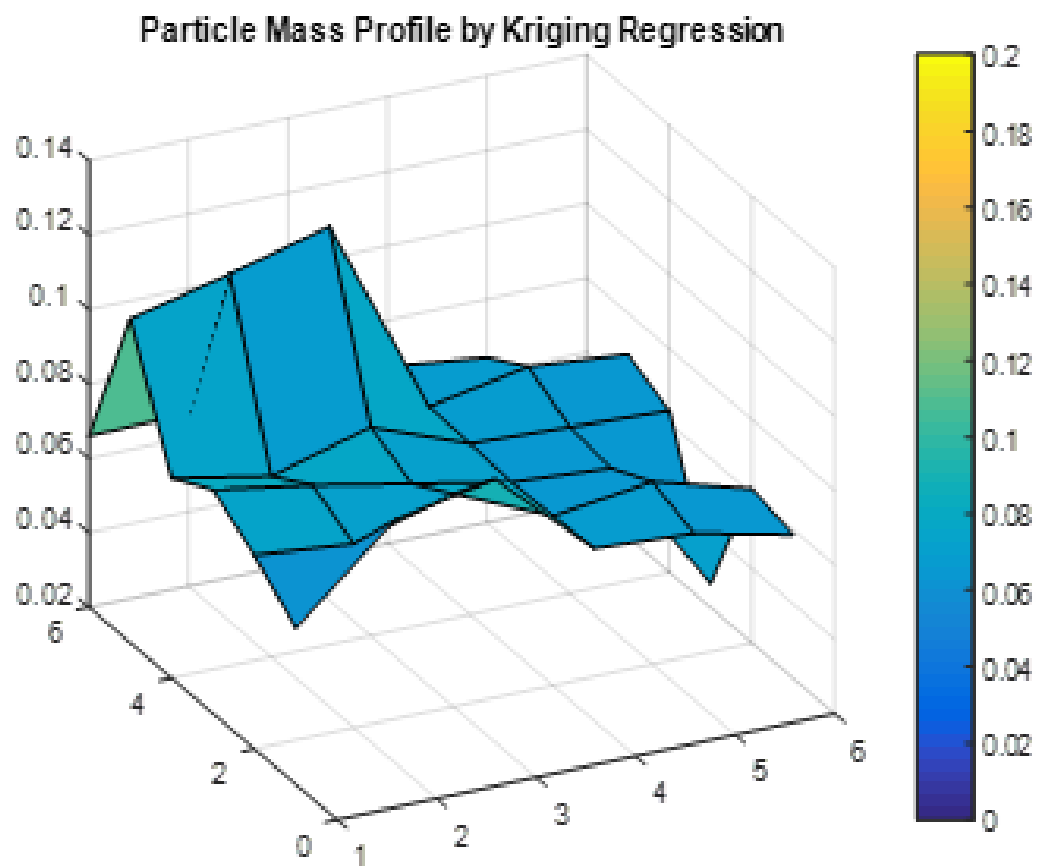


Figure 8.9: Mass density profile predicted by kriging regression