An improved CFD-DEM framework for gas-liquid-solid multiphase free surface flow

Zur Erlangung des akademischen Grades
Doktor der Ingenieurwissenschaften (Dr.-Ing.)
von der KIT-Fakultät für Maschinenbau des
Karlsruher Instituts für Technologie (KIT)

angenommene
Dissertation
von

M.Eng. Huihuang Xia
aus Anhui, China

Tag der mündlichen Prüfung:	25.07.2023
Referent:	Prof. Dr. Marc Kamlah
Korreferent:	Prof. Dr. Britta Nestler

Abstract

Gas-liquid-solid multiphase systems are ubiquitous in engineering applications, e.g. direct inkjet printing, spray drying and spray coating. Direct inkjet printing is a promising additive manufacturing technology for fabricating temperature-sensitive components. However, inkjet-printed structures suffer from the so-called “coffee-ring effect” or “coffee-stain effect”, which leads to a ring-like deposition pattern with more material deposited around the edge but much less material left inside the printed ink-ring. Understanding the physics behind evaporating inkjet-printed droplets with many suspended solid particles inside helps to suppress the coffee-ring effect and to guarantee more uniform material deposition patterns. Accordingly, developing a numerical framework for modelling the gas-liquid-solid multiphase system with evaporation is of great significance. In the present thesis, an improved Computational Fluid Dynamics-Discrete Element Method (CFD-DEM) coupling framework is proposed and developed to model the gas-liquid-solid multiphase system with and without evaporation.

For modelling such a surface-tension-dominant multiphase system, some fundamental scientific problems must be addressed: particle transport and accumulation, surface tension and free surface capturing, droplet wetting and evaporation, contact line pinning, particle-fluid interactions, etc. The DEM is adopted to track the trajectory of solid particles, and CFD is used to model surface tension, evaporation, contact line pinning, etc. Furthermore, coupling CFD to DEM is employed to calculate the complex particle-liquid interactions.

On the DEM side, the conventional DEM is extended to model micro-sized particles. The non-contacting surface forces, e.g. Van der Waals, electrostatic and

Derjaguin-Landau-Verwey-Overbeek (DLVO) forces, and a Brownian-motion-induced random force model are implemented into the open-source DEM code LIGGGHTS. Extensive numerical validations prove that these newly implemented force models can predict either non-contacting or random forces with fairly good numerical accuracy. On the CFD side, the improved Coupled Level Set and Volume of Fluid (i-CLSVoF) framework is developed and implemented into the open-source C++ library OpenFOAM to model surface-tension-dominant flow. The i-CLSVoF framework can capture sharp free surface with less interface diffusion. The improved surface tension force model implemented in i-CLSVoF can predict the surface tension force more accurately with promising suppression of un-physical spurious velocities. Three evaporation models have been implemented into i-CSLVoF for modelling the evaporation of the liquid phase. Numerical validations demonstrate that these evaporation models can accurately model the phase change from liquid to vapour. Furthermore, a contact line pinning model is incorporated to describe droplet evaporation with a constant contact radius mode.

The resolved CFD-DEM approach represents the interactions between continuous fluid and discrete solid phases in a locally resolved way. An improved resolved CFD-DEM model developed in this thesis is capable of modelling the gas-liquid-solid multiphase free surface flow with and without evaporation of the liquid phase. The i-CLSVoF framework is adopted to model the surface tension effect and capture the sharp free surface. An improved capillary force model is developed to compute the capillary interactions for partially floating solid particles at a free surface. Two well-known benchmark cases, namely drag coefficient calculation and the single sphere settling are conducted to validate the resolved CFD-DEM model. It turns out that the resolved CFD-DEM model developed in this thesis can accurately calculate the fluid-solid interactions and predict the trajectory of solid particles interacting with the liquid phase. Numerical demonstrations, e.g. two particles moving along a free surface when the liquid phase evaporates, and particle transport and accumulations inside an evaporating sessile droplet show the performance of the resolved model in modelling complex particle-liquid interactions.

The resolved CFD-DEM approach accurately resolves the flow fields around solid particles but is computationally expensive. Thus, it is only applicable to computationally model the multiphase system with a limited number of particles (less than 1000). Accordingly, a so-called un-resolved CFD-DEM model is further developed in this thesis. In the un-resolved CFD-DEM model, each CFD cell contains several solid particles and thus can be used to investigate the global behaviour of many solid particles (up to 10^6). A new drag force model with a corrected drag coefficient, which is in good agreement with extensive experimental data, has been implemented. This drag force model is applicable over a wide range of Reynolds number ($10^{-4} - 10^6$). Extensive numerical validations have been conducted to validate the drag force model and the un-resolved CFD-DEM approach. A 3D dam break benchmark case demonstrates that the un-resolved CFD-DEM model can realize the four-way coupling between the solid and liquid phases involving around 4000 solid particles. Comparison between numerical simulations and the corresponding experimental studies proves that the accuracy of the un-resolved CFD-DEM model is reasonable.

Kurzfassung

Gas-Flüssig-Fest-Mehrphasensysteme sind in technischen Anwendungen allgegenwärtig, z. B. beim direkten Tintenstrahldruck, der Sprühtrocknung und der Sprühbeschichtung. Der direkte Tintenstrahldruck ist eine vielversprechende additive Fertigungstechnologie für die Herstellung temperaturempfindlicher Bauteile. Allerdings leiden tintenstrahlgedruckte Strukturen unter dem so genannten „Kaffeering-Effekt“ oder „Kaffeefleck-Effekt“, der zu einem ringförmigen Abscheidungsmuster führt, bei dem mehr Material um den Rand herum abgeschieden wird, aber viel weniger Material im Inneren des gedruckten Tintenrings übrig bleibt. Das Verständnis der physikalischen Zusammenhänge beim Verdampfen von mit Tintenstrahldruckern gedruckten Tröpfchen mit vielen suspendierten Feststoffpartikeln im Inneren hilft, den Coffee-Ring-Effekt zu unterdrücken und gleichmäßigere Materialabscheidungsmuster zu gewährleisten. Dementsprechend ist die Entwicklung eines numerischen Rahmens für die Modellierung des Mehrphasensystems Gas-Flüssigkeit-Feststoff mit Verdampfung von großer Bedeutung. In der vorliegenden Arbeit wird ein verbessertes CFD-DEM (Computational Fluid Dynamics-Discrete Element Method)-Kopplungssystem vorgeschlagen und entwickelt, um das Mehrphasensystem Gas-Flüssigkeit-Feststoff mit und ohne Verdampfung zu modellieren.

Für die Modellierung eines solchen Mehrphasensystems, in dem die Oberflächenspannung dominiert, müssen einige grundlegende wissenschaftliche Probleme angegangen werden: Partikeltransport und-akkumulation, Oberflächenspannung und Erfassen der freien Oberfläche, Tröpfchenbenetzung und Verdampfung, Kontaktlinien-Pinning, Partikel-Fluid-Wechse- lwirkungen usw. Die DEM wird eingesetzt, um die Trajektorie von Feststoffpartikeln zu verfolgen, und CFD wird

verwendet, um Oberflächenspannung, Verdunstung, Kontaktlinien-Pinning usw. zu modellieren. Darüber hinaus wird die Kopplung von CFD und DEM eingesetzt, um die komplexen Partikel-Flüssigkeits-Wechselwirkungen zu berechnen.

Auf der DEM-Seite wird die konventionelle DEM erweitert, um mikroskopisch kleine Partikel zu modellieren. Die berührungslosen Oberflächenkräfte, z.B. Vander-Waals-, elektrostatische und Derjaguin-Landau-Verwey-Overbeek-(DLVO)-Kräfte, sowie ein durch Brownsche Bewegung induziertes Zufallskraftmodell werden in den Open-Source DEM-Code LIGGGHTS implementiert. Ausführliche numerische Validierungen zeigen, dass diese neu implementierten Kraftmodelle sowohl berührungslose als auch zufällige Kräfte mit recht guter numerischer Genauigkeit vorhersagen können. Auf der CFD-Seite wird das verbesserte Coupled Level Set and Volume of Fluid (i-CLSVoF)-Framework entwickelt und in die Open-Source-C++-Bibliothek OpenFOAM implementiert, um die oberflächenspannungsdominierte Strömung zu modellieren. Das i-CLSVoF-Framework kann scharfe freie Oberflächen mit wenig Grenzflächendiffusion erfassen. Das in i-CLSVoF implementierte verbesserte Modell der Oberflächenspannungskraft kann diese genauer vorhersagen und bietet eine Unterdrückung unphysikalischer Störgeschwindigkeiten. Drei Verdunstungsmodelle wurden in i-CSLVoF implementiert, um die Verdunstung der flüssigen Phase zu modellieren. Numerische Validierungen zeigen, dass diese Verdunstungsmodelle den Phasenübergang von Flüssigkeit zu Gas genau modellieren können. Darüber hinaus ist ein Kontaktlinien-Pinning-Modell integriert, um die Tröpfchenverdampfung mit einem konstanten Kontaktradius zu beschreiben.

Der sogenannte aufgelöste CFD-DEM-Ansatz stellt die Wechselwirkungen zwischen kontinuierlichen flüssigen und diskreten festen Phasen in ortsaufgelöster Form dar. Ein in dieser Arbeit entwickeltes verbessertes aufgelöstes CFD-DEM-Modell ist in der Lage, die mehrphasige freie Oberflächenströmung zwischen Gas, Flüssigkeit und Feststoff mit und ohne Verdampfung der flüssigen Phase zu modellieren. Der i-CLSVoF-Rahmen wird verwendet, um den Oberflächenspannungseffekt zu modellieren und die scharfe freie Oberfläche zu erfassen.

Ein verbessertes Kapillarkraftmodell wird entwickelt, um die Kapillarinteraktionen für teilweise schwimmende Feststoffteilchen an einer freien Oberfläche zu berechnen. Zur Validierung des aufgelösten CFD-DEM-Modells werden zwei bekannte Vergleichsfälle durchgeführt, nämlich die Berechnung des Widerstandsbeiwert und das Absinken einer einzelnen Kugel. Es zeigt sich, dass das in dieser Arbeit entwickelte aufgelöste CFD-DEM-Modell die Fluid-Feststoff-Wechselwirkungen genau berechnen und die Trajektorie von Feststoffpartikeln, die mit der flüssigen Phase wechselwirken, vorhersagen kann. Numerische Demonstrationen, z.B. zwei Partikel, die sich entlang einer freien Oberfläche bewegen, wenn die flüssige Phase verdampft, sowie Partikeltransport und -ansammlungen innerhalb eines verdampfenden Tropfens auf einem Substrat zeigen die Leistungsfähigkeit des aufgelösten Berechnungswerkzeugs bei der Modellierung komplexer Partikel-Flüssigkeits-Wechselwirkungen.

Der aufgelöste CFD-DEM-Ansatz löst die Strömungsfelder um Feststoffpartikel genau auf, ist aber rechenintensiv. Daher eignet er sich nur für die rechnerische Modellierung eines Mehrphasensystems mit einer begrenzten Anzahl von Partikeln (weniger als 1000). Daher wird in dieser Arbeit ein sogenanntes unaufgelöstes CFD-DEM-Modell weiterentwickelt. In dem unaufgelösten CFD-DEM-Modell enthält jede CFD-Zelle mehrere Feststoffteilchen und kann somit zur Untersuchung des globalen Verhaltens vieler Feststoffteilchen (bis zu 10^6) verwendet werden. Ein neues Widerstandskraftmodell mit einem korrigierten Widerstandsbeiwert, der mit umfangreichen experimentellen Daten gut übereinstimmt, wurde implementiert. Dieses Widerstandskraftmodell ist über einen weiten Bereich der Reynoldszahl ($10^{-4} - 10^6$) anwendbar. Zur Validierung des Widerstandskraftmodells und des unaufgelösten CFD-DEM-Ansatzes wurden umfangreiche numerische Validierungen durchgeführt. Ein 3D-Dammbruch-Benchmark-Fall zeigt, dass das unaufgelöste CFD-DEM-Modell die Vier-Wege-Kopplung zwischen der festen und der flüssigen Phase mit etwa 4000 Feststoffpartikeln realisieren kann. Der Vergleich zwischen numerischen Simulationen und den entsprechenden experimentellen Studien beweist, die Genauigkeit des unaufgelösten CFD-DEM-Modells.

Acknowledgement

First of all, I would like to express my sincere gratitude to my advisor Prof. Marc Kamlah. I still remember I received an email from Marc that reads, "I am happy to host you for your PhD in my group" from Marc in 2017 when I was studying for my master's degree in China. I benefited a lot from helpful discussions with Marc during the past years. His patience, motivation and enthusiasm helped me a lot to remove all the barriers on my journey towards my PhD.

I would like to thank Prof. Britta Nestler for reviewing my doctoral thesis. Also, many thanks to Prof. Robert Stieglitz for chairing my doctoral thesis defence on July 25th, 2023.

I would like to thank Prof. Yixiang Gan from the University of Sydney. Helpful discussions, and his recommendations mean a lot to me. I would like to thank Prof. Dominic Vella from the Mathematical Institute at the University of Oxford. I got helpful suggestions from Prof. Vella during my research stay at Oxford Mathematical Institute. The three-month stay at Oxford was funded by the Research Travel Grant from the Karlsruhe House of Young Scientists (KHYS).

Many thanks to my colleagues from Kamlah's group. Helpful discussions, suggestions and interesting questions from them always inspire me a lot. I would like to thank Kaiju (Tony) Lu, Tao Zhang, Cong Wang, Mathias Jetter and Qian Yuan for enriching my life in Karlsruhe.

I would like to thank Dr. Chenguang Zhang (Polytechnique Montréal), Prof. Weimin Song (Central South University), Prof. Kai Wu (Xi'an Jiaotong University), Dr. Xiang Xu (Anhui Agricultural University), Wenhao Sun (Northwestern Polytechnical University), Arnav Ajmani (University of Stuttgart), Dr.

Yong Tang (Arizona State University), Yefeng Yang (Dalian University of Technology), Ting Zhou (Dalian University of Technology), Shengnan Li (Liaoning Technical University), Miao Jin (Harbin Engineering University) and Zhiquan Chen (Southeast University). Some email conversations with some well-known experts in our community also helped me out of difficulties, e.g. Dr. Julien Maes, Dr. Abd Essamade Saufi, Prof. Steffen Hardt, Dr. Jing Lu, Henning Scheufler, Johannes Kind and Henrik Sontheimer.

The BwUniCluster 2.0 is acknowledged for providing powerful high-performance parallel computing. The financial support from the China Scholar Council (CSC) and the Helmholtz Association in Germany is sincerely acknowledged.

Many thanks to the developers of some excellent open-source code and software I used during my PhD, e.g. LIGGGHTS, OpenFOAM, YADE, ParaView, Voro++, Gnuplot, Matplotlib, Octave, POV-Ray, WebPlotDigitizer, Atom, Inkscape and LaTeX.

Last but not least, many thanks to my parents for their endless love and support. Thanks to my younger sister Qingqing and her husband.

Karlsruhe, August 8, 2023 *Huihuang Xia*

Contents

Abstract i

Kurzfassung v

Acknowledgement ix

1 Introduction 1
1.1 State of the art 4
1.2 Motivation and objectives 13
1.3 Outline of the thesis 15

2 The extended DEM approach for fine particles 17
2.1 The conventional DEM approach 17
2.2 The extended DEM approach 20
2.2.1 The DLVO force model 21
2.2.2 The Van der Waals force 21
2.2.3 The electrostatic force 24
2.2.4 Combination of the Van der Waals and electrostatic forces . 26
2.3 Brownian motion and its computational modelling 28
2.4 Summary 30

3 The i-CLSVoF framework for surface-tension-dominant flow 33
3.1 Mathematical formulation 35
3.2 Numerical method 36
3.2.1 The VoF method 37
3.2.2 The LS method 38

3.2.3 the i-CLSVoF framework 39
3.3 Numerical implementations and discretization 45
3.3.1 Finite volume discretization of the governing equations . . 46
3.3.2 The overall solution procedure 52
3.4 Numerical demonstrations 52
3.4.1 Dam break with obstacle 52
3.4.2 Droplet relaxation with surface-tension force only 54
3.4.3 Spurious currents around sessile droplets 59
3.5 Summary 64

4 Modelling droplet evaporation with the i-CLSVoF framework 67
4.1 Governing equations and numerical method 68
4.2 The i-CLSVoF framework with evaporation 71
4.2.1 The constant mass flux evaporation model 71
4.2.2 The thermally driven evaporation model 71
4.2.3 The vapour mass fraction gradient evaporation model . . . 73
4.3 Contact angle hysteresis and contact line pinning 77
4.3.1 Contact angles and contact line velocity 79
4.3.2 The Robin boundary condition 83
4.4 Numerical validation and case study 85
4.4.1 Droplet evaporation with the constant mass flux evaporation model 86
4.4.2 Droplet evaporation with the thermally driven evaporation model 91
4.4.3 Droplet evaporation with vapour mass fraction gradient evaporation model 94
4.4.4 Droplet evaporation with contact line pinning 97
4.5 Summary 100

5 The improved resolved CFD-DEM framework for multiphase system with and without evaporation 103
5.1 Mathematical formulation 105

5.2 Numerical method 109
5.2.1 Calculations of the interaction forces and torque 109
5.2.2 The numerical procedure 117
5.2.3 Coupling algorithm 119
5.3 Numerical validation 122
5.3.1 Validation of calculations of the drag coefficient 122
5.3.2 Single particle settling 125
5.4 Application 128
5.4.1 Two particles moving along a free surface with evaporation 128
5.4.2 Particle transport and accumulation in an evaporating droplet with contact line pinning 132
5.5 Summary 136

6 The improved un-resolved CFD-DEM framework for gas-liquid-solid multiphase flow 139
6.1 Mathematical formulation 142
6.1.1 Governing equations of the fluid phase 143
6.1.2 Governing equations of the solid phase 144
6.2 Numerical method 144
6.2.1 Calculation of the interaction forces 144
6.2.2 Calculation of the local fluid volume fraction 148
6.2.3 Finite volume discretization of the governing equations . . 151
6.2.4 Coupling algorithm 155
6.3 Numerical validation and application 157
6.3.1 Single particle settling in viscous liquids 157
6.3.2 Single particle settling from air to liquid 159
6.3.3 3D dam break simulations with particles 161
6.4 Summary 165

7 Summary and outlook 167

Bibliography 173

List of Figures 197

List of Tables **203**

List of symbols and abbreviations **205**

A Appendix **209**
A.1 Derivation of the VoF/LS equation 209
A.2 Derivation of equations for the un-resolved CFD-DEM 210
A.2.1 Derivation of the continuity equation 210
A.2.2 Derivation of the momentum equation 211
A.2.3 Derivation of the VoF equation 213
A.3 Voronoi tessellation for granular media: calculations of the local packing fraction with Voro++ 213
A.4 Code list 215
A.5 Quantities in different unit systems 234

B Publication list **237**

1 Introduction

Three-dimensional (3D) direct inkjet printing is one of the additive manufacturing technologies to fabricate 3D components in a contactless way with high flexibility, low cost and without contact-induced deformation of printed components [1, 2]. The most promising aspect of the direct inkjet printing technology is that the fabrication temperature is not too high compared to the laser beam melting or sintering and that the direct inkjet printing can even fabricate 3D structures at room temperature [3–5]. The general process of direct inkjet printing is shown in Figure 1.1. Micro-sized suspension droplets (1–100 picoliters) are generated from the nozzle in two different ways (either continuous inkjet or drop-on-demand) [6]. Sessile droplets sitting on the substrate undergo their drying

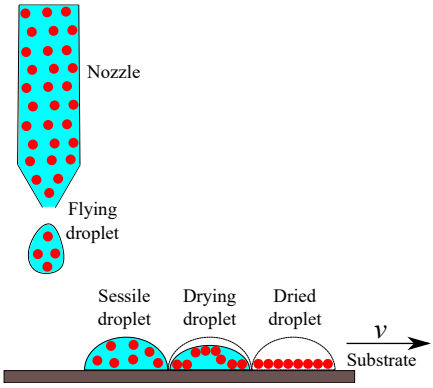


Figure 1.1: Schematic diagram of the 3D direct inkjet printing process.

process. The suspended nanoparticles are dragged by the internal capillary flow

inside evaporating droplets by which the materials deposition patterns are thus achieved. Moving the substrate with given velocities in vertical and horizontal directions continuously guarantees the fabrication of 3D structures layer by layer.

However, ring-like deposition patterns are found after drying droplets on substrates, i.e. much more material is deposited around the contact line region and with relatively less material left inside the droplet (as shown in Figure 1.2). The uneven material deposition is reported to endanger the surface quality and functionality of the printed structures or devices [4]. The ring-like material deposition pattern is the well-known coffee-ring effect reported in the literature [7, 8]. The

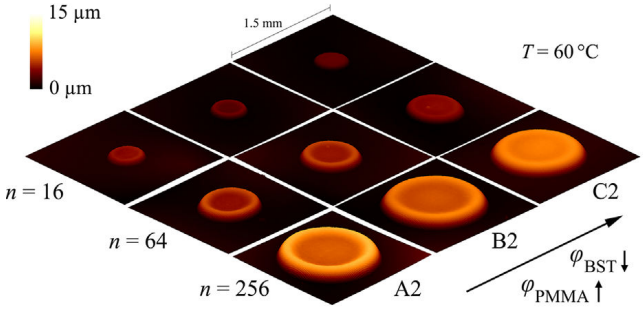


Figure 1.2: Ring-like material deposition patterns found in the structures fabricated by the direct inkjet printing [5].

complex physics behind the coffee-ring effect has been investigated during the past decades [9]. Contact line pinning, larger evaporation mass flux around the triple contact line region and radial capillary flow from the droplet center to the edge are regarded as three most essential factors resulting in the ring-like material deposition pattern [10, 11]. The competition between droplet evaporation and particle movement is found to influence the resulting deposition patterns, and the ring-like deposition may cease if the droplet evaporation is faster than particle movement [12]. The initial contact angle and volume fraction of particles are found to influence the deposition patterns as well, such that a lower initial

contact angle and smaller volume fraction lead to ring-like deposits [13]. The dimensionless quantity capillary number Ca is used to quantify the dominance between the viscous drag force and the surface-tension force with Ca being given by

$$\mathrm{Ca} = \frac{\mu U}{\sigma}, \tag{1.1}$$

where μ and U are the dynamic viscosity and characteristic liquid velocity, respectively, and σ the surface-tension coefficient. The capillary number Ca is reported to influence the internal flow. A large capillary number leads to a more dominant viscous effect than surface tension, and thus suppresses the radial movement of particles from droplet center to edge [14]. Increasing liquid viscosity leads to a large capillary number and makes the viscous force more dominant since a large viscous force hinders the radial capillary flow and thus suppresses the coffee-ring effect [5, 15].

Further theoretical, numerical and experimental studies are conducted to suppress the coffee-ring effect. Particle shape is found to be important in changing material deposition patterns as ellipsoidal particles lead to a more uniform material deposition compared to ideally spherical particles. Further studies indicate that even just adding a small fraction of non-spherical particles (ellipsoids) into the suspension can generate a much more uniform deposition pattern [16, 17]. The underlying reason is that the non-spherical particle shape deforms the liquid interface and then enhances the inter-particle capillary interactions which are relatively weak for spherical particles. Drying droplets on heated substrates changes the internal flow fields and accordingly suppresses the non-uniform material deposition [18]. Manipulating the internal flow inside evaporating droplets with the help of thermally-driven surface-tension gradient (also known as the Marangoni effect) has been found to be an effective way to control the deposition patterns. Namely, the Marangoni effect reverses the radial capillary flow and thus alleviates the ring-like material deposition [19, 20]. Electric fields are used to alter the internal flow directions and to suppress the coffee-ring effect in a simple way [21]. A simple yet efficient method to deposit material uniformly is to add some ethanol into the droplet suspension. Namely, coffee-ring-free devices are found

to have superior performance in capacitance, capability, and contact resistance compared to the devices suffering from the coffee-ring effect [22]. An inward particle drift near the triple contact line is obtained by changing the particle diameter in the range between 3 µm and 10 µm, and a related experimental study provides simple methods to manipulate the particle transport and accumulations [23]. Droplet shape does matter in forming the material depositions and it is confirmed that the contact line curvature influences the development of the coffee ring (large material deposition in sharp concave region and less in convex region) [24, 25].

1.1 State of the art

In addition to the coffee-ring effect, some other non-uniform material deposition patterns have been found in inkjet-printed structures [26]. One goal of direct inkjet printing is to fabricate components with a smooth surface. Since non-uniform deposition patterns affect the surface quality of printed structures, it is worth investigating how to suppress the surface non-uniformity. The evaporation-induced material deposition patterns have been investigated experimentally, theoretically and numerically during the past decades. In this chapter, the literature review of some existing remarkable methods reported in the literature is classified into the following sub-sections.

Investigations without particles

A simple and computationally effective method to model the coffee-ring effect is not to track trajectories of lots of particles inside evaporating droplets but to model particles as an insolvable solute. Accordingly, the particle transport and

accumulation can be modelled by updating the solute concentration during the evaporation process. The solute concentration equation is given by

$$\frac{\partial \alpha_s \rho_s}{\partial t} + \nabla \cdot (\alpha_s \rho_s \mathbf{U}) = \dot{m}_{ls}, \tag{1.2}$$

where α_s is the solute volume fraction, ρ_s the solute density and the source term $\dot{m}_{ls}$ accounts for a gradually increasing solute concentration [27]. Solving the aforementioned solute concentration equation during the drying process with the Constant Contact Angle (CCA) evaporation mode was reported in the literature. There, the ambient temperature is found to influence the deposition patterns significantly (higher temperature leads to a hollow pattern but more uniform deposition is achieved with lower temperature) [27]. A multiphase numerical model has been developed to model the transition between rings and uniform deposits by updating the local particle concentrations with a coupled inhomogeneous evaporation model by which rings are found when the scaled inverse capillary number is much large than one [28]. A two-dimensional (2D) model has been developed to investigate the deposition patterns when drying droplets on an inclined substrate. It turns out that the lower contact line remains pinned during the whole droplet lifetime while the upper contact line starts to move eventually and terminates the accumulation of solute around the upper contact line region [29]. Particle deposition thickness relates to the evaporation flux around the triple contact line, and a large evaporation flux results in more ring-like material deposition [7]. Presence of a neighbouring droplet is found to weaken the evaporation flux at the nearest region of two interacting droplets. This was demonstrated by numerical simulations (as shown in Figure 1.3), and the conclusion was confirmed further by corresponding experimental studies [30].

A 2D lubrication-theory-based model was developed to model droplet evaporation on inclined heated substrates where a convection-diffusion equation was solved to track the transport of colloidal particles inside evaporating droplets

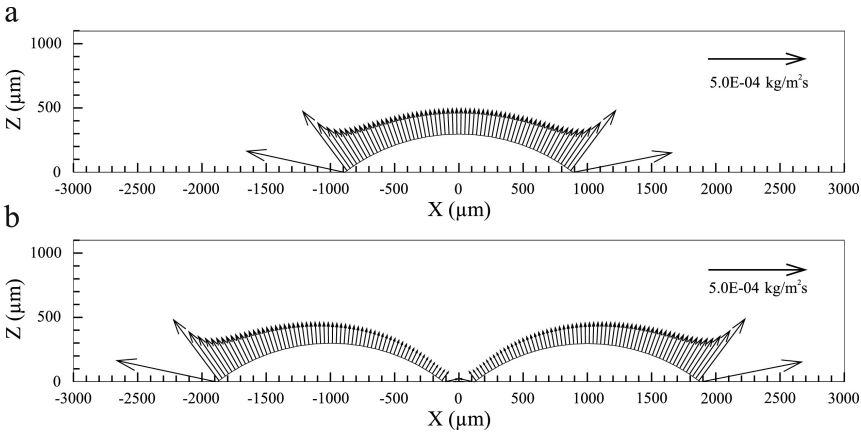


Figure 1.3: Evaporation flux around evaporating droplets: (a) single droplet, (b) two interacting droplets sitting near each other [30].

[31]. In this study, it was found that steeper smooth substrates speed up the droplet evaporation irrespective of the Bond number given by

$$\mathrm{Bo} = \frac{\Delta\rho g L^2}{\gamma}, \tag{1.3}$$

where $\Delta\rho$, g, L and γ are the density difference of two phases, the gravitational acceleration, the characteristic length and the surface tension, respectively. Droplet evaporation on rough and inclined substrates was found to have some dependence on the Bond number Bo.

Investigations with particles

Solution concentration solved from Eqn. 1.2 gives only limited information related to the material deposition, for instance, particle volume fraction. However, the particle deposition pattern of a drying suspension droplet is found to have an interesting transition from regular to irregular packing around the triple contact

line [32]. For example, consider the resistance of a given body which can be calculated with Pouillet's law given by

$$R = \int_l \frac{\rho}{A} ds, \tag{1.4}$$

where ρ is the resistivity of the element, A is the local cross section and the line integral has to be calculated along the length of the body in transport direction. What can be learned from Eqn. 1.4 is that the ratio between length and local cross section determines the resistance, and thus the material deposition patterns are important to understand the relationship between structure and property.

A one-way coupled approach based on Brownian dynamics is proposed to study transport and accumulation of nanoparticles around the contact line for modelling the coffee-ring effect with prescribed internal flow fields given by analytical solutions inside an evaporating droplet [33]. However, this simple approach neglects the influence of the presence of suspended particles on the internal flow fields inside an evaporating droplet, especially when the particle volume fraction becomes large in the final stage of droplet evaporation. Monte Carlo simulations were conducted to investigate the drying of bi-dispersed colloidal droplets on hydrophilic substrates. It was found that smaller particles travel closer to the contact line than larger particles. Furthermore, none of particles reach the triple contact line but deposit at a small distance away from the contact line due to the presence of the surface tension around the contact line [34]. A simple 2D coupled Lattice Boltzmann Method (LBM) and Discrete Element Method (DEM) model was developed to model the coffee-ring effect with both contact line pinning and mixed free and fixed contact line modes [35]. An extensive parameter study on the effect of particle volume fraction, droplet contact angle and substrate temperature was conducted to understand their influence on the deposition patterns. One interesting finding of this study is that increasing the substrate temperature leads to a change from the single ring to the coffee-eye deposition pattern as the Marangoni convection flow changes the internal flow.

In addition to the LBM-DEM coupling, the coupling of Computational Fluid Dynamics (CFD) with DEM, i.e. CFD-DEM coupling bridges the complicated interaction between the continuous fluid phase and the discrete particle phase. CFD-DEM coupling has potential applications in civil engineering [36], chemical engineering [37] and more [38]. CFD-DEM coupling can be classified into three different models: the resolved CFD-DEM coupling, the un-resolved CFD-DEM coupling and the so-called semi-resolved CFD-DEM coupling (a hybrid resolved and un-resolved approach). The resolved CFD-DEM approach resolves the fluid flow more accurately as the particle size is larger than the CFD cell[1] size but it is also computationally expensive [39]. In the un-resolved CFD-DEM approach, the motion of the fluid phase is governed by the volume-averaged Navier–Stokes equations, while the motion of solid phase is described by solving Newton's second law of motion for each particle, separately. Each CFD cell contains several particles, and this approach can be used to study the global behaviour of the fluid and particle system with large number of particles (up to 10^6) with relative low computational cost [36, 40]. To find the balance between numerical accuracy in resolving the fluid flow and computational cost, the semi-resolved CFD-DEM coupling model has been developed and reported in the literature [41]. It can also be used to model a granular system with particles of a large size ratio. The locally averaged un-resolved method is used to resolved the motion of fine particles while the movements of large particles can be computed with the resolved method with more accuracy and less computational costs [42]. CFD-DEM coupling approaches for general particle and fluid system are developed to understand complex phenomena in suspension rheology [43], geomechanics [36, 44], powder-based selective laser melting [45] and blood flow with irregular red blood cells [46]. However, little literature can be found related to CFD-DEM coupling with phase change or evaporation, which is crucial for modelling evaporation-induced particle movement and accumulation.

[1] Here, cell refers to a mesh of small volume bounded by arbitrary polygonal faces after discretizing the computational domain.

Concerning computational modelling of droplet evaporation with CFD, several phase-change models have been developed to model the phase change from liquid to vapour. These include the constant mass flux model [47, 48], the thermally driven model [49–51], and the vapour mass fraction gradient model [48, 52, 53]. The challenging aspect in modelling phase change or evaporation is to address the velocity jump at the interface, which results in some numerical difficulties. Kunkelmann developed an approach that removes the source terms at the interface cells and defines positive and negative mass sources in the most adjacent liquid and gas cells, respectively. This approach was demonstrated to show good performance in modelling boiling [54]. A similar method was also implemented into the open-source code Gerris for modelling droplet evaporation subject to a large mass transfer rate [55]. Both methods are highly dependent on the mesh resolution at the interface. Normally, Adaptive Mesh Refinement (AMR) is needed to cut the interface region into two different regions with negative and positive mass sources accurately. The contact line is fixed during the most of the evaporation time of droplets due to the surface roughness of the substrate [7] and the self-pinning of the contact line [56]. Computational modelling of the contact line pinning with CFD is crucial for studying the coffee-ring effect numerically.

The diameter of particles inside suspension droplets of 3D direct inkjet printing is in the range of 30 nm $< D_p <$ 130 nm [5]. Larger particle sizes can also be found in some experimental investigations of the coffee-ring effect [22, 57]. All these particles are either micro-sized or sub-micro-sized, however, modelling the mechanical behaviour of micro-scale particles and nanoparticles is beyond the capability of the conventional soft-sphere-based DEM approach [58]. The reason lies in some non-contacting surface forces (the Van der Waals and electrostatic force) becoming crucial. They can not be neglected anymore when the particle size gets smaller. In contrast, the conventional DEM approach only accounts for contact forces between two interacting particles when they contact with each other [59–61].

Some extended contact force models adopted from molecular dynamics have been incorporated into DEM to extend the contact laws. The agglomerations of

nanoparticles with a diameter of 20 nm have been numerically studied with the extended DEM by incorporating the Van der Waals force model [62]. The results indicate that the local packing structure of nanoparticles is more dense than that of the micro-sized particles although the overall packing fraction of nanoparticles decreases. Numerical simulations with DEM show that the cohesiveness of granular materials is controlled not only by the inter-particle cohesion (what is commonly known to us) but also by the stiffness and inelasticity of particles [63]. Aggregation in colloidal suspensions was investigated numerically using a mathematical model based on the Langevin equations, which confirmed that the hydrodynamic interactions influence the growth of aggregates significantly. Namely, the growth rate of aggregates is overestimated if the hydrodynamic interactions are neglected [64]. The electrostatic force plays a significant role in governing the packing of fine particles, since the electrostatic force acts as an repulsive force which relates the packing to some key material parameters. Namely, the DEM simulations with electrostatic force indicate that a high zeta potential guarantees a well-stabilized colloidal system while a lower zeta potential leads to agglomerations of fine particles [65]. The Derjaguin, Landau, Verwey, and Overbeek (DLVO) theory combines the Van der Waals attractive force and the electrostatic repulsive force, and thus explains the stability of colloidal systems [61]. Random motion of nanoparticles in colloidal suspensions is governed by the Brownian motion. Brownian motion correlates to both the particle and fluid properties. A Brownian force model with random vector components was developed to model the random force with randomness in both magnitude and direction [60, 66].

Different evaporation-induced particle deposition patterns

The mobility of the contact line is found to significantly result in different particle deposition patterns, e.g. ring-, flat- and dome-like after drying small droplets (their radius $R < 1$ mm) with small contact angles ($\theta < 45^\circ$) [67]. Furthermore, a mobile contact line is found to suppress the ring-like deposition patterns. Also,

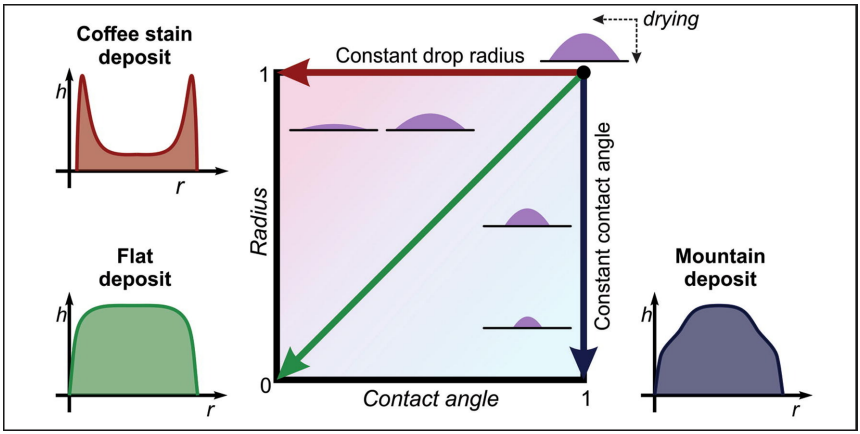


Figure 1.4: Different deposition patterns and some crucial governing parameters [67].

the deposition pattern highly depends on the solvent evaporation profile, evolution of the droplet radius relative to its contact angle, and the ratio between initial and maximum solute concentration. A good combination of these parameters can be used to manipulate the deposition patterns of drying droplets to get a uniform and flat deposition morphology (shown in Figure 1.4). A receding contact line does not lead to a ring-like deposition pattern but was found to result in a mountain-like deposition pattern after dying a circular droplet with uniform evaporation rate [68]. The competition between retreating contact line and solute precipitation leads to different depositions, e.g. concentrated stain, ring-like stain and also combined structure [69]. Experimental studies indicate that more particle deposit at the advancing side of an evaporating droplet resting on an inclined substrate with different inclined angles except for the vertically oriented case [70, 71]. Furthermore, some other deposition patterns are found both experimentally and theoretically. The ring-like to mountain-like deposition pattern transition can be achieved by manipulating the mobility of contact line and the evaporation rate [72]. Increasing the substrate temperature changes the

ring-like deposition to eye-like deposition [73]. Detailed experimental investigations found that particle deposition patterns show a transition from regular particle packing to irregular packing around the contact line [32]. Various patterns including ring-like, bump-like, and finger-like are found experimentally and the interplay between inward Marangoni flow and the coffee-ring-induced outward flow results in different patterns [74]. Understanding the governing parameters behind the different deposition patterns helps to manipulate the deposition morphologies in a better way.

Manipulating the deposition patterns

The Relative Humidity (RH) is found to influence the internal capillary flow inside drying droplets. Large RH suppresses the capillary flow in both sessile and pendant droplets, and a transition from ring-like to homogeneous deposition morphology is obtained by increasing RH from 33% to 63% for sessile droplets [75]. Furthermore, disc-like deposition with RH=33% changes to dotted deposition by increasing RH to 63% for pendant droplets. The RH influences deposit structures after drying droplets on soft substrates, and a circular spot is found with high RH value. Additionally, a low RH value tends to result in ring-like deposition patterns [76]. Droplet geometry provides a promising way to tailor the particle deposition pattern by manipulating the evaporative dynamics of droplets. A drying droplet in a confined geometry is reported to alter the internal flow to inward circulation flows inside evaporating droplets and to guarantee a more uniform material deposition, such that this method provides a simple way to deposit materials uniformly [77]. Laser radiation can be used to dynamically control the internal capillary flow and thus to manipulate the deposition patterns. A related promising finding is that scanning droplets with a 2.9 μm high-power laser beam can guarantee arbitrary particle deposition patterns [78]. Furthermore, a highly uniform material deposition is possible to be obtained by manipulating the diameter and exposure time of the laser beam. The manipulation of particle deposition patterns can be achieved by adjusting the interaction forces between two particles or a particle and a wall. The PH value is found to be an crucial parameter

in governing the DLVO force, where a PH value of 1.4 is found to guarantee an attractive DLVO force while a value of 11.7 tends to turn the attraction into repulsion [79]. Namely, an attractive DLVO force leads to homogeneous deposition but ring-like patterns are found with repulsive DLVO force.

1.2 Motivation and objectives

Tracking trajectories of particles during the drying process of suspension droplets is interesting but also challenging. The scientific problems needed to be addressed involve the droplet evaporation dynamics with contact line pinning, surface tension, particle dynamics, as well as CFD-DEM coupling to account for the fluid drag force acting on particles (shown in Figure 1.5). The CFD-DEM cou-

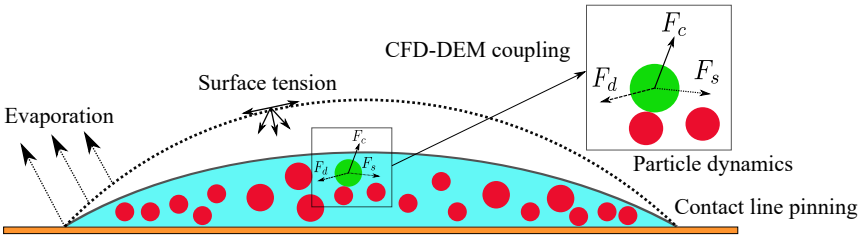


Figure 1.5: Key scientific problems behind a drying suspension droplet with insolvable particles.

pling approach is proposed and developed to model the multi-phase flow with insolvable suspended particles in this thesis. The CFD-DEM coupling model can be used to investigate evaporation-driven particle transport and accumulation to understand many different particle deposition patterns. Potential applications of the numerical tool developed in this thesis involve computational modelling of spray coating, drying, etc. Furthermore, a variation of the particle deposition pattern can be achieved by extensive parameter study on key governing parameters of both droplets and particles. The correlation between particle deposition pattern and the resulting mechanical property of printed structures motivates the

development of a complex coupled numerical simulation tool for surface-tension-dominant multi-phase (liquid-gas-particle) systems with phase change.

In this thesis, the open source DEM code LIGGGHTS (**L**AMMPS **I**mproved for **G**eneral **G**ranular and **G**ranular **H**eat **T**ransfer **S**imulations) [80] and open-source C++ toolbox OpenFOAM [81] are extended to model the movement of micro-sized particles, droplet evaporation and evaporation-driven internal drag force acting on suspended particles. The following research objectives are achieved in this thesis:

- Extension of the conventional DEM approach to model the mechanical behaviour of micro-sized particles and nanoparticles.
- Improvement of CFD modelling to capture the free surface sharply and thereby suppress un-physical spurious velocities with improved surface-tension force modelling.
- Implementation and validation of three different evaporation models.
- Implementation of the contact line pinning model to fix the contact line during the droplet drying/evaporation process.
- Development of a simple yet efficient coupling solver to combine CFD and DEM for modelling the complex interactions between liquid phase and solid particles.
- Validation of newly implemented drag force models and solvers.
- Numerical demonstrations to demonstrate the performance of the CFD-DEM method developed in this thesis in modelling complex particle-fluid interactions with and without evaporation.

1.3 Outline of the thesis

The thesis consists of the following chapters and each chapter is outlined by a general description as follows:

Chapter 2 details the theory of both conventional and extended DEM approaches. The newly incorporated surface force models, i.e. Van der Waals, electrostatic and DLVO are discussed and followed by the random force model accounting for the Brownian-motion-induced random walks of fine particles. Some numerical validations are conducted to validate these surface force and random force models discussed in this chapter.

Chapter 3 gives an overview of free-surface tracking/capturing with both the newly developed improved Coupled Level Set and Volume of Fluid (i-CLSVoF) framework and some other existing methods. The un-physical spurious velocities are then discussed. An improved surface tension force model is proposed and incorporated into i-CLSVoF and some essential numerical methods are detailed in this chapter. The i-CLSVoF framework developed in this chapter guarantees more numerical stability and good sharpness of the interface capturing. Extensive numerical benchmark cases demonstrate the promising performance of the improved model in suppressing the un-physical spurious velocities and also sharp-interface capturing.

Chapter 4 details the theory and numerical methods of models for droplet evaporation (phase change from liquid to vapour). Three different evaporation model (constant mass flux, thermally driven evaporation and evaporation at room temperature) are incorporated into the i-CLSVoF framework. Numerical validations are conducted for the newly implemented evaporation models. Promising agreement between the numerical solutions and corresponding analytical solutions are found for our evaporation models. A simple yet efficient contact line pinning model based on the contact angle hysteresis is developed and implemented to guarantee droplet evaporation with a constant contact radius.

Chapter 5 covers the background, theory and numerical issues of the resolved CFD-DEM approach. Evaporation models are incorporated into the resolved CFD-DEM model for modelling the phase change of the liquid phase. An improved capillary force model has been developed and implemented to model the capillary interactions for partially floating solid particles at a free surface. Two classical benchmark cases are adopted to validate the prediction of the solid-fluid interactions, and results show that the resolved CFD-DEM can accurately calculate the complex interaction forces between the solid and liquid phases. Some numerical demonstration cases are presented in this chapter to show the performance of the resolved CFD-DEM approach in computationally modelling the evaporation-induced particle transport and accumulations for the gas-liquid-solid multiphase system with evaporation.

Chapter 6 presents an extended un-resolved CFD-DEM framework for modelling gas-liquid-solid multiphase free surface flow involving many solid particles. A corrected formula for the drag coefficient has been implemented in this chapter for accurately predicting the drag force acting on the solid phase by the liquid phase. This corrected model applies to a very extensive range of Reynolds number. Numerical validations and demonstrations demonstrate that the extended, un-resolved CFD-DEM approach discussed in this chapter accurately predicts the complex particle-fluid interactions.

Chapter 7 summarizes the main contributions in this thesis and also outlooks some future work and directions.

2 The extended DEM approach for fine particles

The DEM resolves the complex mechanical behaviour of granular systems at particle scale. The computational cost and efficiency are improved with the development of the High Performance Computing (HPC) technology. The conventional DEM can only be used to study the mechanical behaviour of granular media when they contact with their neighbouring particles with non-zero overlap [58, 82]. However, some additional non-contacting forces also known as surface forces come into play when the particle size gets smaller [59–61]. Concerning modelling nano-particles suspended inside a micro-sized droplet, the non-contacting surface forces and Brownian-motion induced random force can not be neglected any more. In this chapter, related extended contact models and forces are implemented in the open-source DEM code LIGGGHTS. It is noted that LIGGGHTS is modified from the open-source molecular dynamics code LAMMPS [83].

2.1 The conventional DEM approach

The conventional DEM accounts for non-zero contact forces between two interacting particles only when their overlap is non-zero. A simple diagram of such contact force F versus the particle overlap h is shown in Figure 2.1. The mechanics and physics behind the DEM approach are governed by the Newton's

[1] Part of this chapter with minor changes has been submitted to arXiv as a preprint [177].

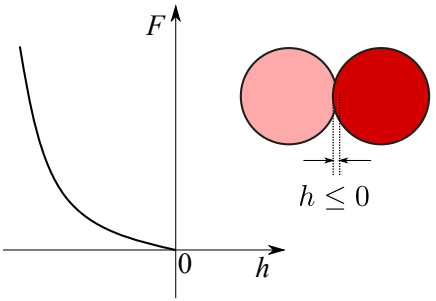


Figure 2.1: The contact force versus particle overlap h of the conventional DEM approach ($h = \delta_n$ for the conventioanl DEM).

second law of motion. In general, the governing equations for the translational and rotational motion, respectively are given as

$$m_i \frac{\partial^2 \mathbf{x}_i}{\partial t^2} = \sum_{i=1}^{N_p} \mathbf{F}_{ij}^{\text{con}} + m_i \mathbf{g}, \tag{2.1}$$

$$I_i \frac{\partial^2 \theta_i}{\partial t^2} = \sum_{i=1}^{N_p} \mathbf{M}_{ij}^{\text{con}}. \tag{2.2}$$

Here, m_i is the mass of the single particle i, N_p the number of particles interacting with the particle i, and $m_i\mathbf{g}$ the gravitational force acting on the particle i as one of the body forces. The conventional contact force $\mathbf{F}_{ij}^{\text{con}}$ between two interacting particles consists of both normal and tangential components where $\mathbf{F}_{ij}^{\text{con}}$ is given by

$$\mathbf{F}_{ij}^{\text{con}} = \mathbf{F}_{ij}^{n} + \mathbf{F}_{ij}^{t}. \tag{2.3}$$

Detailed explanations of parameters and equations related to this contact model are listed in Table 2.1. For the rotational motion of particle i, the angular displacement θ_i is solved and updated from Eqn. 2.2. Here, I_i is the moment of inertia of particle i, and $\mathbf{M}_{ij}^{\text{con}}$ the moment acting on particle i by its neighbouring particles.

Table 2.1: Key parameters and formulas for DEM simulations [84–86]

Parameter [Units]	Formula/Symbol
Coefficient of restitution $[-]$	e_r
Coefficient of friction $[-]$	μ_c
Poisson's ratio $[-]$	ν
Normal relative velocity $[\mathrm{m/s}]$	$\mathbf{u}_n^r$
Tangential relative velocity $[\mathrm{m/s}]$	$\mathbf{u}_t^r$
Unit normal vector $[-]$	$\mathbf{n}_{ij} = \frac{\mathbf{x}_i - \mathbf{x}_j}{\lvert\mathbf{x}_i - \mathbf{x}_j\rvert}$
Normal overlap $[\mathrm{m}]$	$\delta_n = R_i + R_j - \lvert\mathbf{x_i} - \mathbf{x_j}\rvert$
Tangential overlap $[\mathrm{m}]$	$\int_{t_0}^{t_c} \mathbf{u}_t^r dt$
Equivalent radius $[\mathrm{m}]$	$R_{ij}^* = \frac{R_i R_j}{R_i + R_j}$
Equivalent mass $[\mathrm{kg}]$	$m_{ij}^* = \frac{m_i m_j}{m_i + m_j}$
Equivalent Young's modulus $[\mathrm{kg/(m \cdot s^2)}]$	$Y_{ij}^* = \frac{Y_i Y_j}{(1-\upsilon_i{}^2)Y_i + (1-\upsilon_j{}^2)Y_j}$
Equivalent shear modulus $[\mathrm{kg/(m \cdot s^2)}]$	$G_{ij}^* = \frac{Y_i Y_j}{2(2-\upsilon_i)(1+\upsilon_i)Y_j + 2(2-\upsilon_j)(1+\upsilon_j)Y_i}$
Normal contact stiffness $[\mathrm{kg/s^2}]$	$k_n = \frac{4}{3} Y_{ij}^* \sqrt{R_{ij}^* \delta_n}$
Tangential contact stiffness $[\mathrm{kg/s^2}]$	$k_t = 8 G_{ij}^* \sqrt{R_{ij}^* \delta_n}$
Normal damping $[\mathrm{kg/s}]$	$\gamma_n = -2\sqrt{\frac{5}{6}} \frac{\ln(e_r)}{\sqrt{\ln^2(e_r) + \pi^2}} \sqrt{2 Y_{ij}^* \sqrt{R_{ij}^* \delta_{ij}^n m_{ij}^*}}$
Tangential damping $[\mathrm{kg/s}]$	$\gamma_t = -2\sqrt{\frac{5}{6}} \frac{\ln(e_r)}{\sqrt{\ln^2(e_r) + \pi^2}} \sqrt{8 G_{ij}^* \sqrt{R_{ij}^* \delta_{ij}^n m_{ij}^*}}$
Normal contact force $[\mathrm{N}]$	$\mathbf{F}_{ij}^n = (k_n \delta_n - \gamma_n \mathbf{u}_n^r)\mathbf{n}_{ij}$
Tangential contact force $[\mathrm{N}]$	$\mathbf{F}_{ij}^t = \min\left\{ \left\lvert k_t \int_{t_0}^{t_c} \mathbf{u}_t^r dt + \gamma_t \mathbf{u}_t^r \right\rvert, \ \mu_c \mathbf{F}_{ij}^n \right\}$

Particle velocity $\mathbf{u}_p$ can be updated with the two-step Verlet integration scheme once all the forces acting on single particle i are known [87]. The Verlet integration starts with an increment of one half of time-step size Δt and the basic idea is given by

$$\mathbf{u}_p(t+\Delta t/2) = \mathbf{u}_p(t) + \Delta t/2 \frac{\partial \mathbf{u}_p(t)}{\partial t}. \tag{2.4}$$

Then, the particle displacement $\mathbf{x}$ can be integrated with an increment of one time-step as

$$\mathbf{x}(t+\Delta t) = \mathbf{x}(t) + \Delta t \mathbf{u}_p(t+\Delta t/2). \tag{2.5}$$

Finally, the particle velocity is integrated with the increment of the rest half time-step as

$$\mathbf{u}_p(t+\Delta t) = \mathbf{u}_p(t+\Delta t/2) + \Delta t/2 \frac{\partial \mathbf{u}_p(t+\Delta t/2)}{\partial t}. \tag{2.6}$$

The time-step size is crucial for guaranteeing stable DEM simulations, and can be estimated from the following Rayleigh time-step size

$$T_R = \frac{\pi \bar{R} \sqrt{\frac{2\rho(1+\nu)}{Y}}}{0.1631\nu + 0.8766}, \tag{2.7}$$

where $\bar{R}$ is the average particle radius, ρ the particle density, Y the Young's modulus and ν the Poisson's ratio [88]. Normally, the critical time-step size is given by

$$\Delta t = f_s T_R, \tag{2.8}$$

where f_s is a safety factor, and a value ranges from 0.1 to 0.3 is recommended.

2.2 The extended DEM approach

In this section, the extended DEM approach is detailed. As will be detailed below, some additional force terms are incorporated into the governing equations of the

classical DEM approach. The extended governing equations for the extended DEM are given by

$$m_i \frac{\partial^2 \mathbf{x}_i}{\partial t^2} = \sum_{i=1}^{N_p} (\mathbf{F}_{ij}^{\mathrm{con}} + \mathbf{F}_{ij}^{\mathrm{dlvo}}) + m_i \mathbf{g} + \mathbf{F}_{\mathrm{bro}} + \mathbf{F}_{\mathrm{fp}}^{\mathrm{c}}, \tag{2.9}$$

$$I_i \frac{\partial^2 \theta_i}{\partial t^2} = \sum_{i=1}^{N_p} \mathbf{M}_{ij}^{\mathrm{con}} + \mathbf{M}_{\mathrm{fp}}^{\mathrm{c}}. \tag{2.10}$$

Here, $\mathbf{F}_{ij}^{\mathrm{dlvo}}$ is the newly added DLVO force term, and $\mathbf{F}_{\mathrm{bro}}$ the Brownian-motion-induced random force term. The last term $\mathbf{F}_{\mathrm{fp}}^{\mathrm{c}}$ on the right-hand side of Eqn. 2.9 is the fluid-solid interaction force acting from the liquid phase on the solid phase. These force models are detailed in the coming sections. The last term $\mathbf{M}_{\mathrm{fp}}^{\mathrm{c}}$ on the right-hand side of Eqn. 2.10 denotes the fluid-induced moment acting on the solid phase, and is explained in the following sections and chapters, as well.

2.2.1 The DLVO force model

The well-known DLVO theory explains the stability (dispersivity or agglomeration) of colloidal systems [64]. For modelling insolvable particles in suspensions, the DLVO force model based on the DLVO theory becomes crucial for particles with a diameter less than 100 µm. The DLVO force model consists of two different surface force models, i.e. the Van der Waals force and the electrostatic force. The DLVO force is active when the separation distance of two interacting particles is positive (as shown in Figure 2.2), and the sign (attractive or repulsive) of the DLVO force also depends on the separation distance h (detailed in the following sections).

2.2.2 The Van der Waals force

The Van der Waals force serves as an attractive force between two interacting particles or a particle and a wall. The magnitude of the Van der Waals force

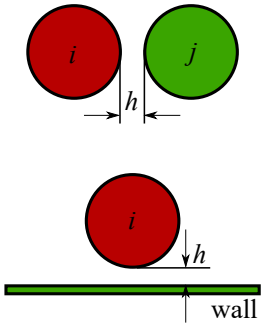


Figure 2.2: The separation distance h between two interacting elements. The top and bottom images demonstrate the pairs of particle-particle and particle-wall, respectively.

between two particles is related to their respective radius R_i and R_j, separation distance h and the material property defined by the Hamaker constant H_a [89]. The formula for calculating the Van der Waals force between two particles is defined by

$$\mathbf{F}_{ij}^{\text{vdw}} = -\frac{H_a}{6}\frac{64R_i^3R_j^3(h+R_i+R_j)}{(h^2+2R_ih+2R_jh)^2(h^2+2R_ih+2R_jh+4R_iR_j)^2}, \tag{2.11}$$

where the minus sign means that the force is attractive [59]. Additionally, the Van der Waals force between a particle and a wall is given by

$$\mathbf{F}_{pw}^{\text{vdw}} = -\frac{H_aR_i}{6h^2}, \tag{2.12}$$

where the magnitude of the force depends on particle radius R_i, the materials property and the separation distance h between the particle and the wall [90]. The crucial material property Hamaker constant H_a is related to the surface energy density γ_s and the cutoff distance h_{min} [91]. Accordingly, H_a can be calculated by

$$H_a = 24\pi\gamma_s h_{\text{min}}^2. \tag{2.13}$$

Furthermore, the separation distance h appears in the denominator, for which a cutoff distance $h_{\min}$ is defined to avoid numerical singularities when h approaches zero. In this thesis, $h = h_{\min}$ when h is smaller than $h_{\min}$.

The Van der Waals force model is implemented in LIGGGHTS-PUBLIC [92] as an extended module (refer to Appendix Code list A.1). The extended version of the LIGGGHTS-PUBLIC is called the LIGGGHTS-XIA in this thesis. Corresponding numerical calculations are conducted to validate the numerical accuracy in computational modelling of the attractive force due to the presence of the Van der Waals force, only. The basic numerical set up is a single particle i approaching another particle j or a fixed wall with an initial separation distance h_0 ($h_0 > 0$) while the Van der Waals forces are recorded for the two cases, respectively. The essential parameters used in the numerical simulations are outlined in Table 2.2. The Van der Waals forces collected from the numerical simulations

Table 2.2: Key parameters for validations of the Van der Waals force model.

Parameter [Units]	Value
R_i [m]	5×10^{-6}
R_j [m]	5×10^{-6}
ρ [kg/m^3]	2500
$h_{\min}$ [m]	1.0×10^{-8}
γ_s [J/m^2]	0.86×10^{-3}
Y [kg/(m$\cdot$s^2)]	1.0×10^{7}
ν [-]	0.29

are compared to the analytical solution given by Eqs. 2.11 and 2.12, respectively. As shown in Figure 2.3, the purple dots are collected from the numerical simulations with LIGGGHTS-XIA whereas the solid lines represent the corresponding analytical solutions. It can be seen from the two figures are that the analytical solution approximates zero when the separation distance becomes larger, and the force maintains a constant value when the separation distance is smaller than the cutoff distance $h_{\min}$. For both Van der Waals interactions between two particles

and a particle with a wall, the newly implemented model shows perfect agreement with the results given by the analytical solutions in Eqs. 2.11 and 2.12.

2.2.3 The electrostatic force

The second part of the DLVO force model is the electrostatic force used to model the repulsion between two interacting particles or a particle-wall pair [61, 93]. The electrostatic force between particle i and j is given by

$$\mathbf{F}_{ij}^{\text{elst}} = \frac{64\pi\varepsilon_0\varepsilon_r k_d}{e^{k_d h}} \left(\frac{R_g T}{zF}\right)^2 \tanh\left(\frac{zF\Psi_i}{4R_g T}\right) \tanh\left(\frac{zF\Psi_j}{4R_g T}\right) R_{ij}^*, \tag{2.14}$$

while the electrostatic force between a particle and a wall is computed by

$$\mathbf{F}_{ij}^{\text{elst}} = \frac{64\pi\varepsilon_0\varepsilon_r k_d}{e^{k_d h}} \left(\frac{R_g T}{zF}\right)^2 \tanh\left(\frac{zF\Psi_i}{4R_g T}\right) \tanh\left(\frac{zF\Psi_j}{4R_g T}\right) R_i. \tag{2.15}$$

Here, R_{ij}^* is the equivalent particle radius, and the other essential parameters are listed in Table 2.3. The validations of the electrostatic force model implemented in LIGGGHTS-XIA is conducted by comparing the electrostatic force acting on a particle when it interacts with another particle or wall. The initial numerical

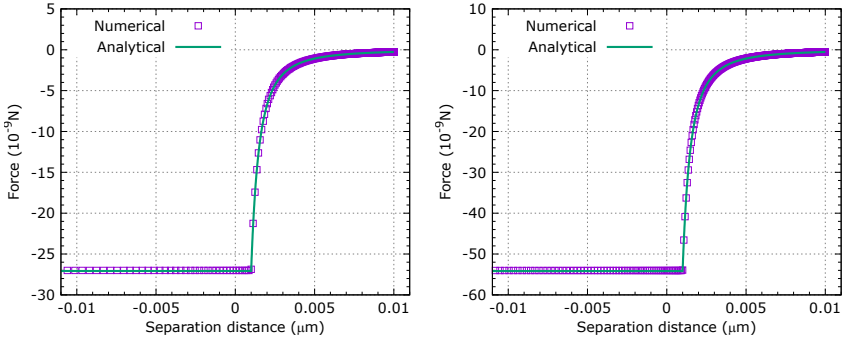


Figure 2.3: Validations of the Van der Waals force model: (a) particle to particle, (b) particle to wall.

Table 2.3: Other parameters of the electrostatic force model.

Symbol [Units]	Parameter
ε_0 [F/m]	Permittivity of vacuum
ε_r [−]	Relative dielectric constant
k_d [1/m]	Debye-Hüeckel parameter
R_g [kg · m^2/(K · mol · s^2)]	Gas constant
T [K]	Temperature
z [−]	Valency of the electrolyte
F [C/mol]	Faraday constant
Ψ [mV]	Zeta potential

configuration is that a tiny overlap exists between two interacting particles or a particle-wall pair. The electrostatic force pushes particles away from each other until the force is small enough and can be neglected. The parameters used in the numerical validations are outlined in Table 2.4. In contrast to the Van der Waals force model, the electrostatic force is repulsive and leads to a positive surface force between two interacting DEM elements. As shown in Figure 2.4, the agreement between numerical solution and the corresponding analytical solutions

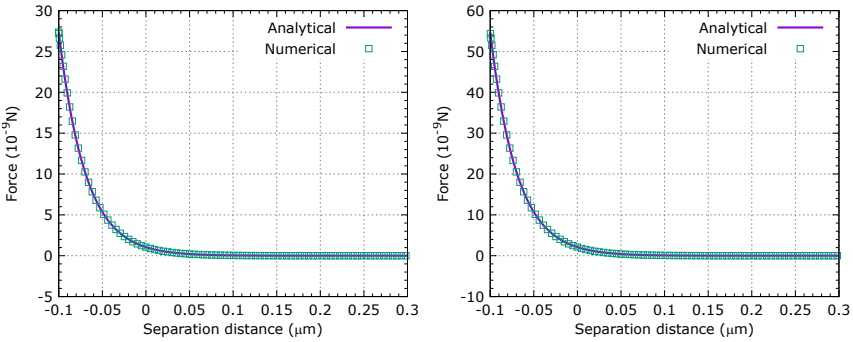


Figure 2.4: Validations of the electrostatic force model: (a) particle to particle, (b) particle to wall.

Table 2.4: Parameters for validations of the electrostatic force model.

Symbol [Units]	Value
R_i [m]	5×10^{-6}
R_j [m]	5×10^{-6}
ρ [kg/m^3]	2500
ε_0 [F/m]	$8.854187817 \times 10^{-9}$
ε_r [−]	80.1
k [1/m]	32.6
R_g [kg · m^2/(K · mol · s^2)]	8.314462×10^{15}
T [K]	293
z [−]	1
F [C/mol]	9.6485×10^{16}
Ψ [mV]	40×10^{-6}

given by Eqs. 2.14 and 2.15 is perfect, as well. As can be seen, the electrostatic force gradually reduces to zero when the separation distance becomes large.

2.2.4 Combination of the Van der Waals and electrostatic forces

The DLVO force consists of the aforementioned two surface force models and is given by

$$\mathbf{F}_{ij}^{\text{dlvo}} = \underbrace{\mathbf{F}_{ij}^{\text{vdw}}}_{\text{attractive}} + \underbrace{\mathbf{F}_{ij}^{\text{elst}}}_{\text{repulsive}} . \tag{2.16}$$

The magnitude and sign (either positive or negative) of the DLVO force depend on the separation distance h. As shown in Figure 2.5, the dotted line below the x axis represents the Van der Waals force and the other dotted line shows the electrostatic force. The combination of the two force models leads to the DLVO force represented by the solid curve. The DLVO force is repulsive when the separation distance is larger than the separation distance threshold h_t, meaning that

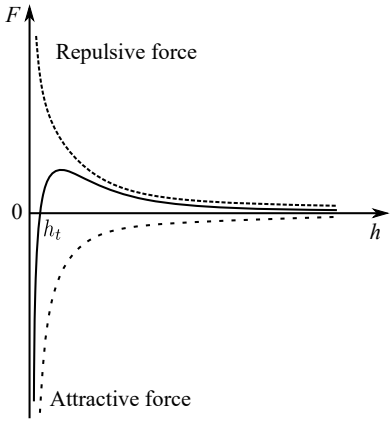


Figure 2.5: Force versus separation distance of the DLVO force model.

the colloidal system is well-dispersed with large separation distances maintained between two particles as shown in Figure 2.6. When the separation distance h is

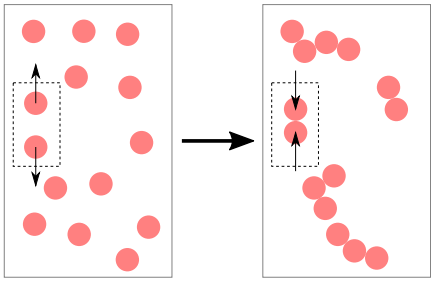


Figure 2.6: The transition from repulsive to attractive DLVO force in the colloidal system.

smaller than h_t, the DLVO force changes from repulsive to attractive. Particles start to agglomerate when the attractive force is dominant. Once these particles are in contact with each other, they do not separate again, unless a third object hits them with strong enough energy [94]. The further particle sedimentation can

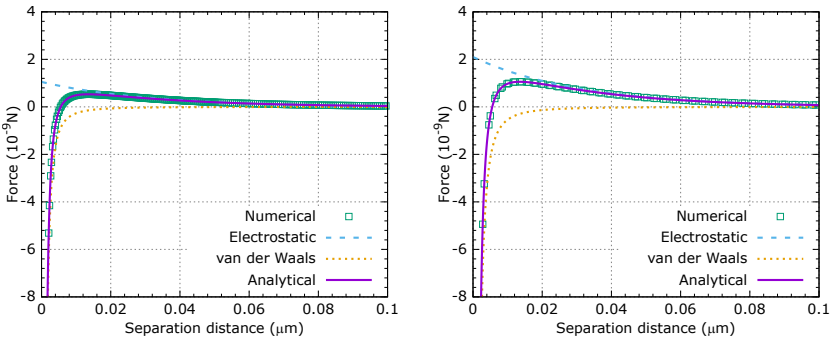


Figure 2.7: Validations of the DLVO force model: (a) particle to particle, (b) particle to wall.

be seen when agglomeration of particles becomes heavy enough to overcome the buoyancy force.

Numerical validations of the DLVO force model are conducted to validate the transition from attraction to repulsion and compared to the analytical solution given by Eqn. 2.16. The numerical set-up is two particles approaching each other with an initial separation distance of 0.1 μm and the DLVO force acting on the tracked particle is recorded. As shown in Figure 2.7, good agreement between numerical and analytical solutions is obtained. They demonstrate that the DLVO force model implemented in LIGGGHTS-XIA can capture the complex interactions between micro-sized particles with good numerical accuracy.

2.3 Brownian motion and its computational modelling

Random motion of fine particles in a colloidal system results from their collisions with fast-moving molecules, and this is the so-called Brownian motion [95]. Modelling Brownian motion with DEM can be achieved by introducing random vectors in x, y and z directions to guarantee that both the direction and

the magnitude of the Brownian force are random. In this thesis, a random vector $\mathbf{G}_r = (\mathbf{G}_r^{\mathbf{x}}, \mathbf{G}_r^{\mathbf{y}}, \mathbf{G}_r^{\mathbf{z}})$ with all three components subject to the Gaussian distribution is incorporated [60]. This Brownian force model is given by

$$\mathbf{F}_{\text{bro}} = \mathbf{G}_r \sqrt{\frac{12\pi R_i \mu_f k_b T}{\Delta t}}, \tag{2.17}$$

where μ_f is the dynamic viscosity of the base fluid, ρ_p the particle density and k_b the Boltzmann constant. Furthermore, Δt is an artificial time-step size which is constant for a given colloidal system calculated by

$$\Delta t = \frac{1}{10} \frac{2\rho_p R_i^2}{9\mu_f}. \tag{2.18}$$

After implementing the Brownian force model in LIGGGHTS-XIA, the numerical validation is conducted to understand the accuracy in predicting the random force acting on single particle i. The numerical set-up is a single particle with diameter of 0.1 µm moving inside the computational domain under the influence of the Brownian force (without other body forces). The simulation was run for 50000 steps, and the random force data was saved for every second step. Thus, it leads to 25000 samples for generating random force distribution, which can be regarded as large enough. According to the Eqs. 2.17 and 2.18, the term $\mathbf{G}_r$ on the right-hand side of Eqn. 2.17 is constant with the exception of $\mathbf{G}_r$. Accordingly, the random force collected from the numerical simulations is divided by the term with constant values such that the resting random components of vector $\mathbf{G}_r$ can be compared to the Gaussian distribution to validate the numerical implementations. The Gaussian fitting is used to find the mean μ and standard deviation σ of the samples, and the fitting results are shown in Figure 2.8. The mean, standard deviation and relative error of the standard deviation are calculated for the three components, respectively, and these values are listed in Table 2.5. It can be seen that the mean and the standard deviation of the simulation data approximate the Gaussian distribution (with mean $\mu = 0$ and standard deviation $\sigma = 1.0$). The

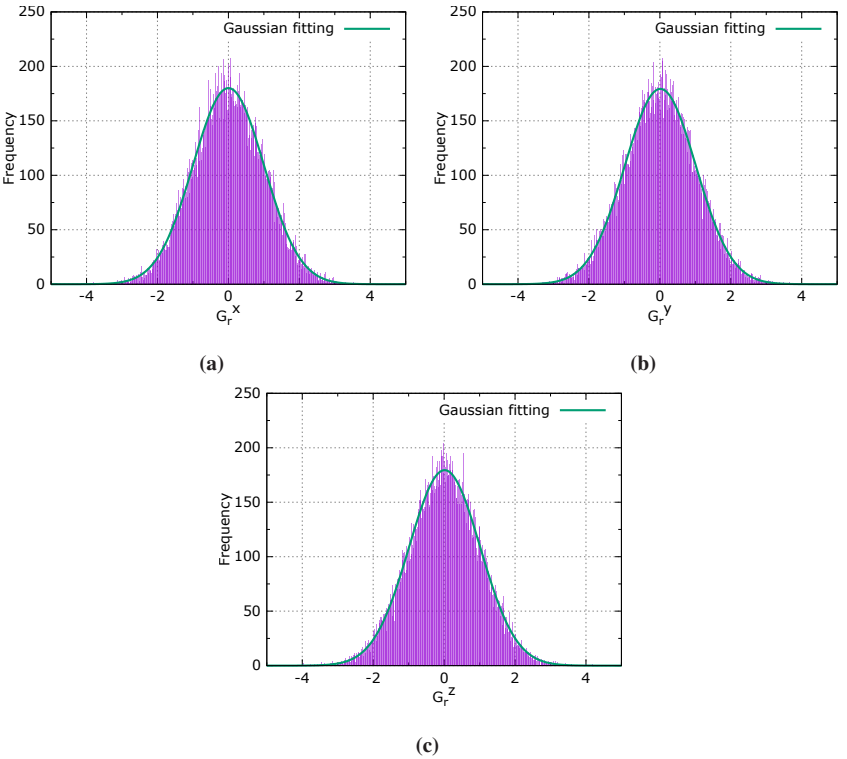


Figure 2.8: Gaussian fitting for random numbers generated in x, y and z components, respectively.

relative errors of the standard deviation in all three directions are smaller than 1%. Overall, the results are best in the x direction and worst in the z direction.

2.4 Summary

In summary, the classical DEM method is extended to model the complex behaviour of micro-sized particles and nano-particles with newly implemented

Table 2.5: The mean, standard deviation and relative error of standard deviation for the random numbers generated with the Brownian force model.

Component	μ	σ	δ_σ
x	0.00192	1.00096	0.096%
y	0.01336	1.00176	0.176%
z	0.01427	1.00209	0.209%

surface-force models and a random body force model. The DLVO model developed in this thesis extends the conventional contact law between two particles (or a particle and a wall) with non-zero overlap to positive separation distance (dashed curve shown in Figure 2.9). A short comparison between

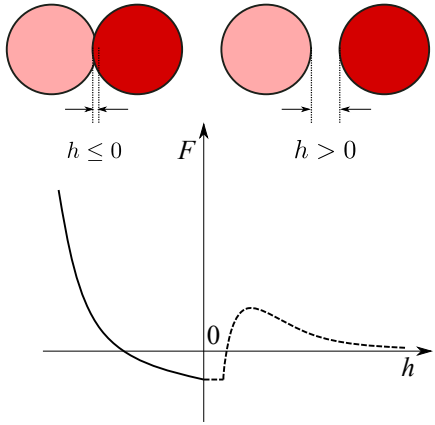


Figure 2.9: The contact model of the extended DEM approach developed in this thesis.

LIGGGHTS public version known as LIGGGHTS-PUBLIC and the extended version LIGGGHTS-XIA detailed in this thesis and developed by the author of the thesis is outlined in Table 2.6. The extended DEM developed and validated in this thesis has potential applications in understanding the physics behind granular media for a wide range of particle size. One of the crucial processes in selective laser sintering/melting is to spread lots of fine particles on substrates and then

use a high-energy laser beam to melt these particles. The surface forces and

Table 2.6: Comparison between LIGGGHTS-PUBLIC and LIGGGHTS-XIA (✓: available; × : not available yet; +✓: newly added)

Model/Module	LIGGGHTS-PUBLIC	LIGGGHTS-XIA
Hertz	✓	✓
Tangential history	✓	✓
Van der Waals	×	+✓
Electrostatic	×	+✓
DLVO	×	+✓
Brownian force	×	+✓

contact forces acting on particles are important for forming a uniform and thin particle layer which is reported to guarantee a good surface quality after melting. The extended DEM developed in this thesis is a candidate for computationally modelling these applications. However, in the first place, the extended DEM is introduced here to be coupled to the improved CFD model (to be discussed in Chapter 3 and Chapter 4) to understand complex physics and phenomena when fine particles start to transport and accumulate inside evaporating droplets.

3 The i-CLSVoF framework for surface-tension-dominant flow

The surface tension force plays a significant role in droplet wetting and evaporation [96, 97]. The water droplet on a leaf is a simple case to demonstrate the role of the surface tension force in forming a given contact angle between the droplet and the leaf surface. The surface tension force is also crucial to maintain the droplet shape under the influence of gravity and other external forces acting on the droplet [98]. The Continuum Surface-tension Force (CSF) model was proposed to model the surface tension force as a volumetric body force [99]. However, this conventional surface-tension model suffers from spurious currents or velocities which appear around the interface. Spurious currents destabilize the simulations and even influence the internal flow inside the droplets when studying droplets numerically [100]. Spurious currents partially result from numerical errors when calculating the interface curvature. Some numerical models have been developed to improve the calculation of the interface curvature and thus suppress the un-physical velocities. The geometric Volume of Fluid (VoF) method represents the interface by a reconstructed thin interface inside each of the interface cells[2] explicitly and is reported to have better performance in interface representations as well as reducing spurious velocities [101]. Some open-source codes or libraries incorporate the geometric VoF, such as PARIS [102], Basilisk [103] , isoAdvector [104], interPlicFoam [105] and VoFLibrary [106].

[1] Part of this chapter with minor changes has been submitted to arXiv as a preprint [176].

[2] Here, interface cell refers to a small computational cell bounded by arbitrary polygonal faces that includes a part of the liquid-gas interface after discretizing the computational domain, numerically.

In contrast to the geometric VoF approach, the algebraic VoF method is relatively simple and easy to implement as the interface is represented implicitly and without explicit interface reconstruction [107, 108]. The algebraic VoF method is mass conserving but was reported to suffer from some interface diffusion [109]. Several methods have been proposed to address the interface diffusion problem, for instance, incorporation of a surface compression term [110], adaptive interface compression [111], and coupling VoF to some other numerical methods, such as the Level Set (LS) method [112]. Concerning the advantages and shortcomings of both the VoF and LS method, the so-called Coupled Level Set and Volume of Fluid (CLSVoF) method was proposed to combine sharp interface representation and mass conservation [113]. The coupled approach improves the suppression of the spurious currents. However, according to experience, relatively large spurious velocities still exist around the interface, especially for micro-sized droplets. A short summary of methods used to track or capture the free surface and their applications are outlined in Table 3.1. This chapter addresses the issues mentioned above related to suppressing of un-physical spurious currents and to capturing the sharp interface. It presents a simple yet efficient

Table 3.1: Summary of numerical methods to track/capture free surface.

Authors (publication year)	Method	Code	Applications
Brackbill et al. (1992) [99]	VoF	In-house code	Interfacial flows
Sussman et al. (1998) [114]	LS	In-house code	Interfacial flows
Popinet (2009) [115]	VoF	Basilisk	Interfacial flows
Raeini et al. (2012) [116]	VoF	OpenFOAM	Porous media
Albadawi et al. (2013) [112]	CLSVoF	OpenFOAM	Bubble dynamics
Yokoi (2014) [117]	CLSVoF	In-house code	Droplet splashing
Roenby et al. (2016) [104]	VoF	isoAvector/OpenFOAM	Interfacial flows
Irfan et al. (2017) [52]	Front tracking	In-house code	Phase change
Dai et al. (2019) [105]	VoF	OpenFOAM	Multiphase flows
Scheufler et al. (2019) [106]	VoF	VoFLibrary/OpenFOAM	Interfacial flows
Jamshidi et al. (2019) [109]	Phase field	OpenFOAM	Microfluids
Aniszewski et al. (2021) [102]	VoF/Front tracking	PARIS	Multiphase flows
Inguva et al. (2022) [118]	Front tracking	In-house code	Two-phase flow

numerical framework to model the dynamics of micro-sized droplets with sharp interface representation and suppressed spurious velocities.

3.1 Mathematical formulation

This section presents the equations governing the physics behind incompressible Newtonian flow based on the one-field formulation. The one-field formulation solves only one set of governing equations for both liquid and gas phases [119]. The indicator function $I(\mathbf{x},t)$

$$I(\mathbf{x},t) = \begin{cases} 1 & \mathbf{x} \text{ in liquid,} \\ 0 & \text{otherwise,} \end{cases} \tag{3.1}$$

is used to identify the tracked phase (liquid phase in this thesis) at time t and position $\mathbf{x}$. The fundamental fluid quantities such as fluid density ρ and dynamic viscosity μ can be represented in the complete multiphase domain by the indicator function $I(\mathbf{x},t)$ using expressions like

$$\begin{aligned} \rho &= I(\mathbf{x},t)\rho_1 + [1 - I(\mathbf{x},t)]\rho_2, \\ \mu &= I(\mathbf{x},t)\mu_1 + [1 - I(\mathbf{x},t)]\mu_2. \end{aligned} \tag{3.2}$$

The physics behind the incompressible Newtonian fluid without phase change is governed by the following Navier-Stokes equations

$$\nabla \cdot \mathbf{U} = 0, \tag{3.3}$$

$$\frac{\partial(\rho\mathbf{U})}{\partial t} + \nabla \cdot (\rho\mathbf{U}\mathbf{U}) = -\nabla p + \nabla \cdot \left[\mu \left(\nabla\mathbf{U} + (\nabla\mathbf{U})^{\mathrm{T}}\right)\right] + \rho\mathbf{g} + \mathbf{F}_{\mathrm{st}}, \tag{3.4}$$

where $\mathbf{U}$ is the velocity field, and the superscript T indicates the transpose of the tensor. Its divergence being equal to zero means that the flow is incompressible. In the momentum equation (Eqn. 3.4), p is the pressure field and $\rho\mathbf{g}$ the gravitational force term. The surface tension force newly introduced in this thesis is

incorporated into the momentum equation as the last term on the right-hand side. The VoF transport equation is solved for capturing the free-surface, by which the volume fraction field can be continuously updated. The material derivative of the volume fraction field α_l defines the VoF transport equation[3]

$$\frac{D\alpha_l}{Dt} = \frac{\partial \alpha_l}{\partial t} + \mathbf{U} \cdot \nabla \alpha_l = 0. \tag{3.5}$$

In OpenFOAM, the conservative VoF equation

$$\frac{\partial \alpha_l}{\partial t} + \nabla \cdot (\alpha_l \mathbf{U}) = 0, \tag{3.6}$$

is solved where the divergence-free condition (Eqn. 3.3) must be satisfied simultaneously to guarantee the incompressibility. The additional so-called interface compression term $\nabla \cdot [\alpha_l(1-\alpha_l)\mathbf{U}_r]$ with $\mathbf{U}_r$ being the artificial compression velocity is generally incorporated into the left-hand side of the VoF equation (Eqn. 3.6) to suppress the interface diffusion [120]. As demonstrated in previous benchmark study and the literature, however, the compression term enhances spurious currents [121, 122].

3.2 Numerical method

Incompressible flow with free surface involves the crucial aspect of tracking or capturing the free surface. There are two widely used methods, namely interface tracking and interface capturing. In the interface tracking approach, the mesh needs to be updated to track the interface as the flow evolves. On the other hand, the interface capturing method is formulated over non-moving meshes by solving an additional advection equation (VoF) or convective Cahn–Hilliard equation (phase-field) to update the fluid volume fraction (VoF) or the order parameter (phase-field) [123]. The fluid volume fraction or the order parameter solved from

[3] The detailed derivations can be found in the Appendix A.1.

the additional equation then marks the location of the moving interface. For interface tracking, the front tracking method is widely used; here, the basic idea is to use so-called marker points to identify the interface's location [52, 118]. Alternatively, as an interface capturing approach, the phase-field method identifies the interface through the value of an order parameter solved from the convective Cahn–Hilliard equation: the order parameter is 1 in the liquid phase, -1 in the gas phase, with a value in between representing the diffuse interface [109, 124]. The Volume of Fluid (VoF) method is another popular interface capturing approach in which the volume fraction field is computed and the interface inferred. The LS method captures the free surface by the signed LS function, with the zero LS at the interface, positive in the liquid and negative in the gas phase. Furthermore, the LS method can guarantee a very sharp interface without interface diffusion [125]. Before introducing the i-CLSVoF framework, two existing popular interface capturing approaches (VoF and LS) are introduced for completeness.

3.2.1 The VoF method

In the VoF method, the volume fraction field α is defined as the volume-averaged volume integral of the phase indicator function $I(\mathbf{x},t)$, denoted as

$$\alpha = \frac{1}{V}\int_V I(\mathbf{x},t)dV. \tag{3.7}$$

The basic idea behind the VoF interface capturing approach is to track the evolution of the volume fraction field for a given phase. Typically, the liquid is selected as the tracked phase, and accordingly, the liquid volume fraction field α_l can be tracked by solving the VoF transport equation given by Eqn. 3.6.

As shown in Figure 3.1, the computational domain has two sub-domains Ω_1 (liquid) and Ω_2 (gas) in the whole computational domain. When the liquid cells are full of liquid, the volume fraction of all liquid cells in Ω_1 is 1. Furthermore, the interface cells are partially filled with the liquid, so that the intermediate values between 0 and 1 are given there. The VoF method is mass-conserving, but as

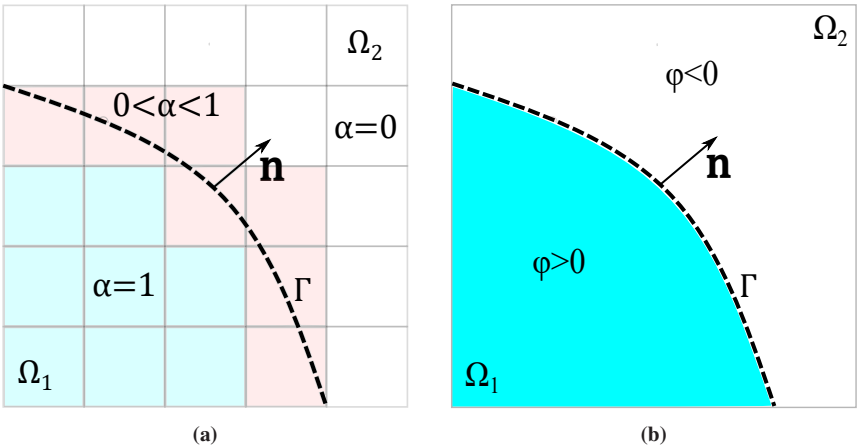


Figure 3.1: (a) the VoF method, (b) the LS method.

mentioned in the literature, the VoF method suffers from some interface diffusion, which can diffuse over several cells around the interface depending on the mesh resolution [126].

3.2.2 The LS method

The LS method is another interface capturing method, and the quantity used is the so-called signed LS function $\psi(\mathbf{x},t)$. The interface can be identified as zero level set, and in the liquid phase (Ω_1), it is given positive values of LS function and negative values hold in the gas phase (Ω_2) [125]. The LS function is solved and updated from the material derivative of ψ given by

$$\frac{\partial \psi}{\partial t} + \mathbf{U} \cdot \nabla \psi = 0. \tag{3.8}$$

A detailed derivation can be found in the Appendix A.1. The signed LS function is continuous and has a smooth transition from liquid to gas phase. The LS method can guarantee a more sharp interface compared to the VoF approach, but the LS method is reported to be not mass-conserving in the literature [127].

Both the VoF and LS methods suffer from un-physical spurious velocities, which destabilize the numerical simulations. Spurious velocities occur due to inaccurate interface curvature calculations. One method that attempted to improve the curvature calculation is to refine the mesh. However, in this thesis, it is found that a finer mesh could not reduce the un-physical velocity but even enhanced it (refer to Figure 3.2), and the same conclusion can also be found in the literature [128].

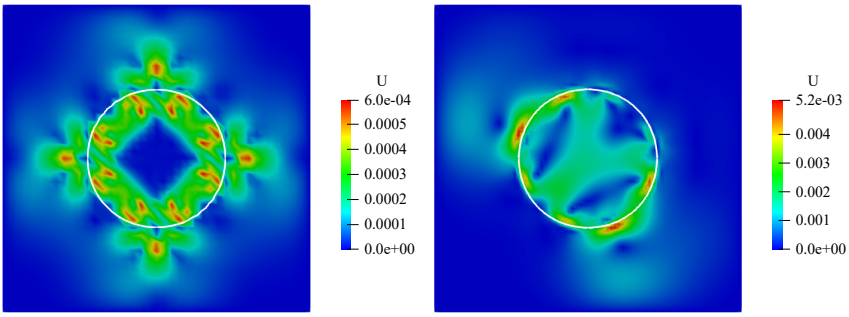


Figure 3.2: Spurious currents around the interface (represents with the solid white line) with two different mesh sizes : (a) coarse mesh (cell number: 40×40), (b) fine mesh (cell number: 80×80).

3.2.3 the i-CLSVoF framework

In this thesis, the advantages of VoF and LS are combined and the so-called Coupled LS and VoF (CLSVoF) approach is extended to capture the free surface. The CLSVoF approach improves the mass-conserving issues of LS method while also guaranteeing a sharp interface [113]. However, although it improves the calculation of the interface curvature [112, 127], the conventional CLSVoF, also known as the simple CLSVoF (s-CLSVoF) approach, still suffers from un-physical spurious velocities. The filtering surface tension model based on the VoF approach

is reported to suppress un-physical spurious currents, especially for droplets interacting with substrates [116]. Accordingly, the improved CLSVoF (i-CLSVoF) method is proposed to suppress the spurious currents further and to improve the numerical stability by extending the filtering method to filter un-physical spurious velocities even more. In contrast to the s-CLSVoF method, the i-CLSVoF framework incorporates an improved surface tension force model to calculate surface-tension forces more accurately and to filter and reduce spurious velocities by additional filtering steps, which are discussed in the following part.

The basic idea behind the i-CLSVoF framework is initializing the initial signed distance function ψ_0 from the liquid volume fraction field α_l. ψ_0 is initialized by the initialization function

$$\psi_0 = (2\alpha_l - 1)\Gamma, \tag{3.9}$$

with the dimensionless quantity $\Gamma = 0.75\Delta x$, where Δx is the minimum mesh size around the interface. It is dimensionless as well (as Δx is artificially divided by a dimensioned quantity with a value of 1 and the dimension of meter). Normally, the LS function gradually loses its property to have a value of zero at the interface and cannot be sufficiently sharp after moving with the convection velocity. Therefore, a re-initialization step is adopted to recover its sharpness. The Hamilton–Jacobi equation

$$\frac{\partial \psi}{\partial \tau} - S(\psi_0)(1 - |\nabla \psi|) = 0, \tag{3.10}$$

is used to re-initialize the LS function with the initial condition $\psi = \psi_0$ [125]. Here, τ is an artificial time step, and the smoothed out sign function $S(\psi_0)$ is defined as

$$S(\psi_0) = \frac{\psi_0}{\sqrt{\psi_0^2 + (\Delta x)^2}}. \tag{3.11}$$

A benchmark case study showed that the smoothed out sign function $S(\psi_0)$ can further reduce un-physical velocities and guarantee more numerical stability than the conventional sign function [114]. The Hamilton–Jacobi equation needs to be solved by numerical iteration until $|\nabla \psi| = 1$, where the number of iterations N_{iter}

of around 15 can be enough [127]. The re-initialization scheme is outlined in Figure 3.3.

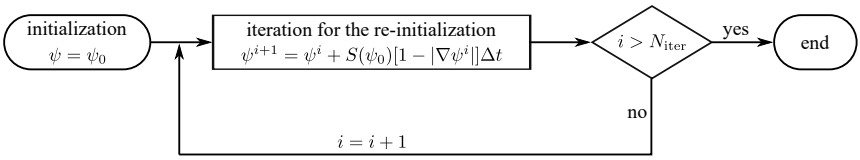


Figure 3.3: Flowchart for solving the Hamilton–Jacobi equation.

Before introducing the new surface tension force model implemented in the i-CLSVoF framework, two widely-used surface tension models are listed below for completeness. The Continuum Surface-tension Force (CSF) model approximates the surface tension with the help of the gradient of α_l [96, 99]. This surface tension force model is given by

$$\sigma K(\alpha_l)\nabla\alpha_l, \tag{3.12}$$

where σ is the surface tension coefficient, and $K(\alpha_l)$ the interface mean curvature defined as

$$K(\alpha_l) = -\nabla \cdot \frac{\nabla\alpha_l}{|\nabla\alpha_l| + \delta_n}. \tag{3.13}$$

Here, δ_n is the stabilization factor to guarantee a non-zero denominator, and δ_n is calculated by

$$\delta_n = \frac{10^{-8}}{\left(\sum_{i=1}^{N} V_i/N\right)^{\frac{1}{3}}}, \tag{3.14}$$

where N is the number of the cells in the computational domain, and V_i is the volume of each cell.

As the liquid volume fraction field α_l is not continuous, the calculation of its gradient in Eqn. 3.13 can not ensure more numerical accuracy. The LS method uses the signed LS function ψ to calculate the interface curvature, which is more

accurate as ψ ensures continuity along the interface normal [125]. The interface curvature with the LS method can then be calculated as

$$K(\psi) = -\nabla \cdot \frac{\nabla \psi}{|\nabla \psi| + \delta_n}. \tag{3.15}$$

The improved curvature calculation method is incorporated into the conventional CLSVoF approach to improve the surface tension calculations [112, 127]. As an alternative to Eqn. 3.12, the surface tension force is then given as

$$\sigma K(\psi) \delta_\psi \nabla \psi, \tag{3.16}$$

where the delta function δ_ψ is given by

$$\delta_\psi = \begin{cases} \frac{1}{2\varepsilon}\left(1 + \cos(\frac{\pi\psi}{\varepsilon})\right) & |\psi| < \varepsilon, \\ 0 & \text{otherwise}, \end{cases} \tag{3.17}$$

with ε as the interface thickness, which usually equals to $1.5\Delta x$ [129]. As discussed in the literature, ε can also range from $1.0\Delta x$ to $1.5\Delta x$ depending on the mesh type [130]. The conventional CLSVoF approach is reported to suppress un-physical spurious currents; however, more can be done to refine the model and reduce the un-physical velocities (these improvements are detailed below). Therefore, the non-symmetrical Heaviside function H_ψ is incorporated into our new surface tension force model. The reason is that the non-symmetrical Heaviside function is reported to improve the numerical stability compared to the symmetrical Heaviside function H_s [117, 127]. The symmetrical Heaviside function

$$H_s = \begin{cases} 0 & \psi < -\varepsilon, \\ \frac{1}{2}\left[1 + \frac{\psi}{\varepsilon} + \frac{1}{\pi}\sin(\frac{\pi\psi}{\varepsilon})\right] & |\psi| \leq \varepsilon, \\ 1 & \psi > \varepsilon, \end{cases} \tag{3.18}$$

is obtained by smoothing out the Heaviside step function (purple line in Figure 3.4). The non-symmetrical Heaviside function H_ψ (yellow dotted curve in Figure 3.4) is given by

$$H_\psi = \begin{cases} 0 & \psi < -\varepsilon, \\ \frac{1}{2}\left[\frac{1}{2} + \frac{\psi}{\varepsilon} + \frac{\psi^2}{2\varepsilon^2} - \frac{1}{4\pi^2}\left(\cos(\frac{2\pi\psi}{\varepsilon}) - 1\right) + \frac{\varepsilon+\psi}{\varepsilon\pi}\sin(\frac{\pi\psi}{\varepsilon})\right] & |\psi| \leq \varepsilon, \\ 1 & \psi > \varepsilon. \end{cases} \tag{3.19}$$

The difference among the three different Heaviside functions is shown in Figure 3.4, and the horizontal dotted line represents the middle line of the vertical axis.

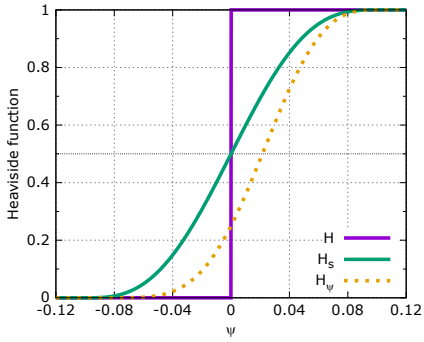


Figure 3.4: Three different Heaviside functions (H: the Heaviside step function, H_s: the symmetrical Heaviside function, H_ψ: the non-symmetrical Heaviside function).

The new and improved surface tension force model is then given accordingly as

$$\mathbf{F}_{\text{st}} = \sigma K(\psi)\nabla H_\psi. \tag{3.20}$$

As inspired by the VoF-based surface tension model, the surface tension related pressure term p_{st} is separated from the total pressure to avoid difficulties in the discretization of the pressure jump [116]. The pressure equation is given by

$$\nabla \cdot \nabla p_{\mathrm{st}} = \nabla \cdot \mathbf{F}_{\mathrm{st}}, \tag{3.21}$$

and the pressure equation can be solved with the prescribed boundary condition

$$\frac{\partial p_{\mathrm{st}}}{\partial n} = 0. \tag{3.22}$$

In order to filter spurious currents, the modified indicator function α_{pc} is introduced for calculating the new Delta function given by Eqn. 3.26, and α_{pc} is defined as

$$\alpha_{\mathrm{pc}} = \frac{1}{1-C_{\mathrm{pc}}}\left[\min\left(\max(\alpha_l, \frac{C_{\mathrm{pc}}}{2}), 1-\frac{C_{\mathrm{pc}}}{2}\right) - \frac{C_{\mathrm{pc}}}{2}\right], \tag{3.23}$$

where C_{pc} is the sharpening coefficient. C_{pc} equal to 0 yields the original indicator function α_l, which is the liquid volume field and defined by Eqn. 3.7. Increasing C_{pc} leads to a sharp representation of the interface and can suppress the spurious velocity but also brings numerical instabilities [116].

Finally, employing p_{st} and α_{pc}, the filtering surface tension force model is introduced to filter un-physical spurious currents parallel to the free surface. It is defined as

$$F^{f}_{\mathrm{st},f} = F_{\mathrm{st},f} - F^{\mathrm{filt}}_{\mathrm{st},f}. \tag{3.24}$$

Here, $F_{\mathrm{st},f}$ is the surface tension force calculated at face center by $F_{\mathrm{st},f} = \mathbf{F}_{\mathrm{st}}\mathbf{n}_f$ with $\mathbf{n}_f$ being the normal vector defined at the face center, and $F^{\mathrm{filt}}_{\mathrm{st},f}$ is a time-dependent term also defined at the face center. It is calculated from

$$F^{\mathrm{filt}}_{\mathrm{st},f} = \frac{\delta_{\mathrm{st}}}{|\delta_{\mathrm{st}}|+\delta_n}\left(R_f(F^{\mathrm{filt}}_{\mathrm{st},f})_{i-1} + C_{\mathrm{fc}}\left\langle \nabla p_{\mathrm{st}} - (\nabla p_{\mathrm{st}} \cdot \mathbf{n}_s)\mathbf{n}_s \right\rangle_f \cdot \mathbf{n}_f\right), \tag{3.25}$$

where R_f is a relaxation factor, and $(F^{\mathrm{filt}}_{\mathrm{st},f})_{i-1}$ the value of $F^{\mathrm{filt}}_{\mathrm{st},f}$ in the previous time step. In addition, $\langle\rangle_f$ denotes the interpolation from the cell center to the

cell face in OpenFOAM, C_{fc} is the coefficient determining how fast the spurious velocity is filtered, and $\mathbf{n}_s$ is the normal vector defined at the cell center ($\mathbf{n}_s = \nabla\alpha_l/|\nabla\alpha_l|$). δ_{st} is a newly defined Delta function based on the previously introduced sharpening indicator function α_{pc}, and its definition is given by

$$\delta_{\mathrm{st}} = \nabla_f^{\Gamma}\alpha_{\mathrm{pc}}. \tag{3.26}$$

Here, ∇_f^{Γ} denotes the gradient normal to the interface.

The final step is to define the threshold for filtering the surface-tension flux, also called the capillary flux defined by $\phi_{\mathrm{cf}} = F_{\mathrm{st},f}|S_f|$ with $|S_f|$ being the magnitude of the face area. The capillary flux is artificially set to zero when the capillary flux is smaller than the threshold, where the filtering capillary flux is defined as

$$\phi_{\mathrm{cf}}^{\mathrm{filt}} = \phi_{\mathrm{cf}} - \min\left(\max(\phi_{\mathrm{cf}}, -\phi_{\mathrm{cf}}^{\mathrm{thre}}), \phi_{\mathrm{cf}}^{\mathrm{thre}}\right). \tag{3.27}$$

Here, the $\phi_{\mathrm{cf}}^{\mathrm{thre}}$ is the threshold value below which the capillary flux is regarded as zero, and it can be calculated by

$$\phi_{\mathrm{cf}}^{\mathrm{thre}} = C_{\mathrm{filt}}|\bar{F_{\mathrm{st},f}}||S_f|, \tag{3.28}$$

where C_{filt} is the filtering coefficient. It is normally set to 0.01, which means that the capillary flux can be regarded as zero when its magnitude is less than 1% of the average capillary flux. Furthermore, $|\bar{F_{\mathrm{st},f}}|$ is the magnitude of the average surface-tension force. The source code for the numerical implementation of the filtering approach can be found in Appendix Code list A.2.

3.3 Numerical implementations and discretization

The aforementioned equations are solved using the Finite Volume Method (FVM). In order to suppress un-physical spurious velocities and to improve numerical

stability, the i-CLSVoF framework is implemented into open-source C++ library OpenFOAM. The corresponding in-house solvers **interDyMFoamX** are accordingly developed.

The maximum time step for solving the governing equation and ensuring spurious currents do not increase over time is estimated from two constraints. The first constraint is

$$\Delta_{t\sigma} < \sqrt{\frac{\rho_{\text{avg}}\Delta x^3}{2\pi\sigma}}, \tag{3.29}$$

where ρ_{avg} is the average density of the phases. It is proposed for the explicit treatment of the surface tension force term [99]. Another more comprehensive time step constraint is given by

$$\Delta_{tc} < \frac{1}{2}\left(C_2\tau_\mu + \sqrt{(C_2\tau_\mu)^2 + 4C_1{\tau_\rho}^2}\right), \tag{3.30}$$

which involves the density and the viscosity of the multiphase system. τ_μ and τ_ρ are given as $\mu_{\text{avg}}\Delta x/\sigma$ and $\sqrt{\rho_{\text{avg}}\Delta x^3/\sigma}$, respectively, with μ_{avg} being the average dynamic viscosity between the phases [131]. Accordingly, the maximum time step size for stable numerical simulations is given as

$$\Delta_t < \min(\Delta_{t\sigma}, \Delta_{tc})C_{\Delta t} \tag{3.31}$$

with $C_{\Delta t}$ being the stabilization factor where a range of $C_{\Delta t}$ between 0.3 and 0.7 is recommended for more stable constraints, especially for cases with phase change [48].

3.3.1 Finite volume discretization of the governing equations

A new pressure field p_{rgh} is defined by

$$p_{\text{rgh}} = p - \rho\mathbf{g}\cdot\mathbf{h}, \tag{3.32}$$

where $\mathbf{h}$ is the position vector. The gradient of p_{rgh} is given by

$$\nabla p_{\text{rgh}} = \nabla p - \rho \mathbf{g} - \mathbf{g} \cdot \mathbf{h} \nabla \rho. \tag{3.33}$$

Substituting $\nabla p = \nabla p_{\text{rgh}} + \rho \mathbf{g} + \mathbf{g} \cdot \mathbf{h} \nabla \rho$ and $\nabla \cdot \left[\mu \left(\nabla \mathbf{U} + (\nabla \mathbf{U})^{\text{T}} \right) \right] = \nabla \cdot (\mu \nabla \mathbf{U}) + \nabla \mathbf{U} \cdot \nabla \mu$ (detailed derivation can be found in [120]) into Eqn. 3.4, it becomes

$$\frac{\partial (\rho \mathbf{U})}{\partial t} + \nabla \cdot (\rho \mathbf{U} \mathbf{U}) = -\nabla p_{\text{rgh}} + \nabla \cdot (\mu \nabla \mathbf{U}) + \nabla \mathbf{U} \cdot \nabla \mu - \mathbf{g} \cdot \mathbf{h} \nabla \rho + \mathbf{F}_{\text{st}}. \tag{3.34}$$

The volume integral of Eqn. 3.34 is given by

$$\begin{aligned} \int_{\Omega_i} \frac{\partial (\rho \mathbf{U})}{\partial t} \, dV + \underbrace{\int_{\Omega_i} \nabla \cdot (\rho \mathbf{U} \mathbf{U}) \, dV}_{= \int_{\partial \Omega_i} (\rho \mathbf{U} \mathbf{U}) \cdot \mathbf{n} \, dS} \\ = - \int_{\Omega_i} \nabla p_{\text{rgh}} \, dV \\ + \underbrace{\int_{\Omega_i} \nabla \cdot (\mu \nabla \mathbf{U}) \, dV}_{= \int_{\partial \Omega_i} (\mu \nabla \mathbf{U}) \cdot \mathbf{n} \, dS} + \int_{\Omega_i} \nabla \mathbf{U} \cdot \nabla \mu \, dV \\ - \int_{\Omega_i} \mathbf{g} \cdot \mathbf{h} \nabla \rho \, dV + \int_{\Omega_i} \mathbf{F}_{\text{st}} \, dV, \end{aligned} \tag{3.35}$$

where the transformation from volume integral to surface integral is done by using the divergence theorem. Besides, $\partial \Omega_i$ is a closed surface bounding the control volume Ω_i, and dS represents an infinitesimal surface element with unit normal $\mathbf{n}$ pointing outwards of the surface $\partial \Omega_i$ ($\mathbf{n} dS = d\mathbf{S}$).

In OpenFOAM, all quantities are solved and stored at cell centers. The cell center is denoted as $\mathbf{x}_P$, and the face center is denoted by $\mathbf{x}_f$. As shown in Figure 3.5,

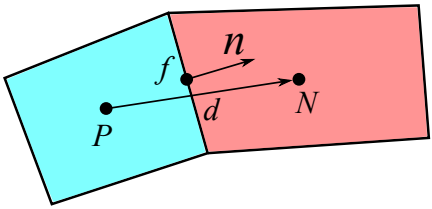


Figure 3.5: Notations for two cells sharing one face.

the owner cell[4] is denoted as P and one of its neighbour cells is denoted as N and the two cells share one face denoted by f.

The temporal derivative term in Eqn. 3.35 is approximated by

$$\int_{\Omega_i} \frac{\partial(\rho \mathbf{U})}{\partial t}\, dV \approx \frac{\rho_P^{n+1}\mathbf{U}_P^{n+1} - \rho_P^n \mathbf{U}_P^n}{\Delta t} V_P, \tag{3.36}$$

where the superscript n represents the current time step and the quantities (e.g. $\mathbf{U}_P^n$) in the current time step are known. The superscript $n+1$ denotes the predicted quantities in the next time step needed to be solved (e.g. $\mathbf{U}_P^{n+1}$). Furthermore, V_P is the volume of the cell.

The convection term in Eqn. 3.35 is approximated by converting the surface integral to a sum over all the faces, and is given by

$$\int_{\partial\Omega_i} (\rho \mathbf{U}\mathbf{U}) \cdot \mathbf{n}\, dS \approx \sum (\rho^n \mathbf{U}^n \mathbf{U}^{n+1})_f \cdot \mathbf{S}_f = \sum F_f^n \mathbf{U}_f^{n+1}, \tag{3.37}$$

where F_f^n is the mass flux, which is calculated by

$$F_f^n = (\rho^n \mathbf{U}^n)_f \cdot \mathbf{S}_f \tag{3.38}$$

[4] In the finite volume method, an owner cell is a three-dimensional region of space that surrounds a point of interest or a discrete point within a computational cell. The owner cell is used to represent a small portion of the computational domain, and is typically defined by the grid cells that surround it.

with $\mathbf{S}_f$ being the face area vector.

The diffusion term is discretized as

$$\int_{\partial\Omega_i} (\mu\nabla\mathbf{U})\cdot\mathbf{n}\,dS \approx \sum(\mu\nabla\mathbf{U}^{n+1})_f\cdot\mathbf{S}_f = \sum\mu_f(\nabla\mathbf{U})_f^{n+1}\cdot\mathbf{S}_f. \tag{3.39}$$

The pressure gradient term is numerically approximated by

$$\int_{\Omega_i}\nabla p_{\mathrm{rgh}}\,dV = \int_{\partial\Omega_i} p_{\mathrm{rgh}}\,d\mathbf{S} \approx \sum p_{\mathrm{rgh},f}^{n}\mathbf{S}_f. \tag{3.40}$$

The explicit expression for the predicted velocity field $\mathbf{U}_P^{n+1}$ is given by

$$\frac{\rho_P^{n+1}\mathbf{U}_P^{n+1}-\rho_P^n\mathbf{U}_P^n}{\Delta t}V_P + \sum F_f^n\mathbf{U}_f^{n+1} = \sum\mu_f(\nabla\mathbf{U})_f^{n+1}\cdot\mathbf{S}_f - \sum p_{\mathrm{rgh},f}^n\mathbf{S}_f. \tag{3.41}$$

The other missing terms in Eqn. 3.35, e.g. gravitational force and surface tension force are incorporated into Eqn. 3.52, as discussed below, which then becomes the discretized form of Eqn. 3.35 with all these terms included.

The velocity field $\mathbf{U}_f^{n+1}$ and pressure field $\mathbf{p}_{\mathrm{rgh},f}^{n+1}$ on faces are calculated by numerical interpolations given by

$$\mathbf{U}_f^{n+1} = \frac{\mathbf{U}_P^{n+1}+\mathbf{U}_N^{n+1}}{2}, \tag{3.42}$$

and

$$p_{\mathrm{rgh},f}^{n} = \frac{p_{\mathrm{rgh},P}^{n}+p_{\mathrm{rgh},N}^{n}}{2}, \tag{3.43}$$

respectively.

The first term $(\nabla\mathbf{U})_f^{n+1}\cdot\mathbf{S}_f$ on the right-hand-side of Eqn. 3.41 is calculated by

$$(\nabla\mathbf{U})_f^{n+1}\cdot\mathbf{S}_f = \left((\nabla\mathbf{U})_f^{n+1}\cdot\frac{\mathbf{S}_f}{|\mathbf{S}_f|}\right)\cdot|\mathbf{S}_f|, \tag{3.44}$$

where $(\nabla \mathbf{U})_f^{n+1} \cdot \frac{\mathbf{S}_f}{|\mathbf{S}_f|}$ is the face normal gradient. It is approximated by

$$(\nabla \mathbf{U})_f^{n+1} \cdot \frac{\mathbf{S}_f}{|\mathbf{S}_f|} \approx \frac{\mathbf{U}_N^{n+1} - \mathbf{U}_P^{n+1}}{|\mathbf{d}|}, \tag{3.45}$$

where $\mathbf{d}$ is the distance vector between the current cell P and the neighbour cell N (shown in Figure 3.5).

Substituting these approximations (Eqs. 3.42, 3.43, and 3.45) into Eqn. 3.41, leads to

$$\begin{aligned} \frac{\rho_P^{n+1}\mathbf{U}_P^{n+1} - \rho_P^n \mathbf{U}_P^n}{\Delta t} V_P + \sum F_f^n \frac{\mathbf{U}_P^{n+1} + \mathbf{U}_N^{n+1}}{2} & \\ = \sum \mu_f \frac{\mathbf{U}_N^{n+1} - \mathbf{U}_P^{n+1}}{|\mathbf{d}|} |\mathbf{S}_f| & \\ - \sum \frac{p_{\mathrm{rgh},P}^n + p_{\mathrm{rgh},N}^n}{2} \mathbf{S}_f. & \end{aligned} \tag{3.46}$$

After some manipulations, Eqn. 3.46 yields

$$\begin{aligned} \left(\frac{\rho_P^{n+1}}{\Delta t} + \frac{1}{V_P} \sum \frac{F_f^n}{2} + \frac{1}{V_P} \sum \frac{\mu_f |\mathbf{S}_f|}{|\mathbf{d}|} \right) \mathbf{U}_P^{n+1} = & \\ - \sum \frac{1}{V_P} \left(\frac{F_f^n}{2} - \frac{\mu_f |\mathbf{S}_f|}{|\mathbf{d}|} \right) \mathbf{U}_N^{n+1} & \\ + \frac{\rho_P^n}{\Delta t} \mathbf{U}_P^n - \frac{1}{V_P} \sum \frac{p_{\mathrm{rgh},P}^n + p_{\mathrm{rgh},N}^n}{2} \mathbf{S}_f. & \end{aligned} \tag{3.47}$$

Eqn. 3.47 can be abbreviated as

$$A_P \mathbf{U}_P^{n+1} = - \sum A_N \mathbf{U}_N^{n+1} + S_P^n - \frac{1}{V_P} \sum \frac{p_{\mathrm{rgh},P}^n + p_{\mathrm{rgh},N}^n}{2} \mathbf{S}_f, \tag{3.48}$$

where

$$A_P = \left(\frac{\rho_P^{n+1}}{\Delta t} + \frac{1}{V_P} \sum \frac{F_f^n}{2} + \frac{1}{V_P} \sum \frac{\mu_f |\mathbf{S}_f|}{|\mathbf{d}|} \right), \tag{3.49}$$

$$A_N = \frac{1}{V_P} \left(\frac{F_f^n}{2} - \frac{\mu_f |\mathbf{S}_f|}{|\mathbf{d}|} \right), \tag{3.50}$$

$$S_P^n = \frac{\rho_P^n}{\Delta t} \mathbf{U}_P^n. \tag{3.51}$$

Eqn. 3.48 can be further changed to

$$A_P \mathbf{U}_P^{n+1} + \sum A_N \mathbf{U}_N^{n+1} = S_P^n - \nabla p_{\mathrm{rgh},P} - \mathbf{g} \cdot \mathbf{h} \nabla \rho_P + \sigma k \nabla \alpha_P, \tag{3.52}$$

if all the other terms in Eqn. 3.34 are incorporated.

Dividing Eqn. 3.52 by A_P, leads to

$$\mathbf{U}_P^{n+1} = \mathbf{HbyA}_P^{n+1} - \frac{1}{A_P} \left(\nabla p_{\mathrm{rgh},P} + \mathbf{g} \cdot \mathbf{h} \nabla \rho_P - \sigma k \nabla \alpha_P \right), \tag{3.53}$$

where $\mathbf{HbyA}_P^{n+1}$ is given by

$$\mathbf{HbyA}_P^{n+1} = \frac{1}{A_P} \left(-\sum A_N \mathbf{U}_N^{n+1} + S_P^n \right). \tag{3.54}$$

Substituting Eqn. 3.53 into the continuity equation Eqn. 3.3, the pressure equation can be derived as

$$\nabla \cdot \left(\frac{1}{A_P} \nabla p_{\mathrm{rgh}}^{n+1} \right) = \nabla \cdot \left(\mathbf{HbyA}_P^{n+1} + \frac{1}{A_P} (\sigma k \nabla \alpha_{n+1} - \mathbf{g} \cdot \mathbf{h} \nabla \rho^{n+1}) \right). \tag{3.55}$$

The pressure field solved from Eqn. 3.55 is used to correct the velocity field, so that the velocity field can satisfy the continuity equation (Eqn. 3.3). The corresponding numerical correction is given by

$$\mathbf{U} = \mathbf{HbyA}_P + \frac{\mathbf{F}_{\mathrm{st}} - \mathbf{g}\cdot\mathbf{h}\nabla\rho - \nabla p_{\mathrm{rgh}}}{A_P}. \tag{3.56}$$

3.3.2 The overall solution procedure

The overall solution procedure is outlined in Figure 3.6. After initializing the essential fields, e.g. liquid volume fraction, velocity and pressure, the free surface is advected by solving the VoF equation (Eqn. 3.6) and then the LS field can also be calculated. Accordingly, the interface curvature and the surface tension can be updated with the liquid volume fraction solved from the VoF equation. The velocity and pressure fields are calculated by solving Eqn. 3.53, and the pressure equation (Eqn. 3.55), respectively. The new pressure field is then used to correct the velocity field (Eqn. 6.38). The simulation runs into the next cycle until the pre-defined total simulation time t_{end} is reached.

3.4 Numerical demonstrations

In this section, some numerical benchmark cases adopted from the literature are reproduced to demonstrate the sharp interface capturing (dam break with obstacle [132, 133]) and the suppression of un-physical velocities (droplet relaxation case and sessile droplets with three different contact angles [116, 134–136]) with the i-CLSVoF framework developed in this thesis.

3.4.1 Dam break with obstacle

The purpose of the current numerical demonstration is to compare the i-CLSVoF model developed in this work against the conventional VoF model in terms of

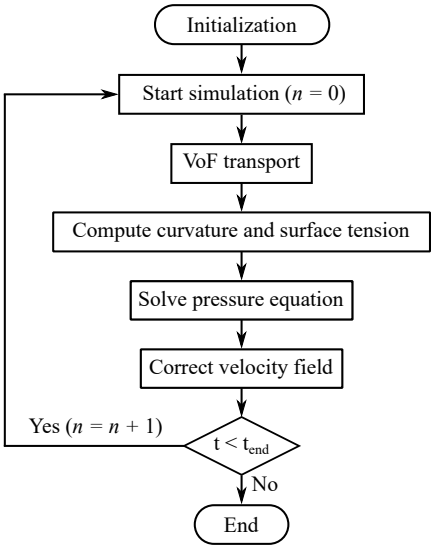


Figure 3.6: Flowchart of the i-CLSVoF framework for surface-tension-dominant flow.

free-surface capturing. The interDyMFoam solver of OpenFOAM 5.x is adopted for conducting simulations with the conventional VoF model. In this case, the surface-tension force is not dominant other than in the droplet cases. The numerical set-up is a water column initialized at the bottom left of the 2D computational domain of which the top is open while the remaining boundaries are regarded as walls (as shown in Figure 3.7a). The essential parameters for the simulations are outlined in Table 3.2. The 2D Adaptive Mesh Refinement (AMR) developed by

Table 3.2: Parameters for 2D dam break simulations

Phase	Density $[\mathrm{kg/m^3}]$	Kinematic viscosity $[\mathrm{m^2/s}]$	Surface tension $[\mathrm{N/m}]$
Water	1000	1×10^{-6}	0.07
Air	1	1.48×10^{-5}	-

Ajit Kumar [137] is incorporated into the numerical simulations to capture the

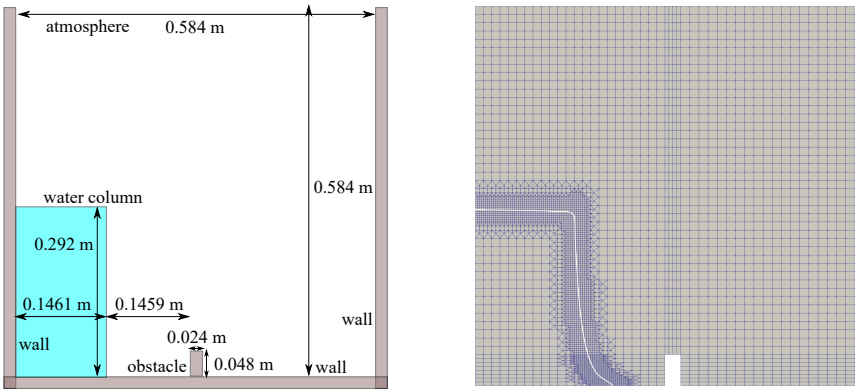


Figure 3.7: (a) Geometry and boundaries of the 2D dam break simulation, (b) Numerical simulations with 2D AMR (the white curve represents the free-surface).

sharp interface with finer computational mesh around the interface region and coarser mesh elsewhere to save computational costs (as shown in Figure 3.7b). The total simulation time is 0.5 s, and it is enough for water to reach both the bottom obstacle and the right wall. As shown in Figure 3.8a and 3.8c are the simulation results with the VoF approach, and Figure 3.8b and 3.8d are with the i-CLSVoF model. When water hit the obstacle and form a wave over the obstacle, the results with VoF approach turn out some interface diffusion (see Figure 3.8a) but the i-CLSVoF captures the sharp interface accurately (see Figure 3.8b). As the simulation progresses, water hits the right wall, and some water is bounced back to the container. Figure 3.8c shows large interface diffusion like rising gas around center of the right wall, whereas Figure 3.8d presents an interface is captured sharply and without interface diffusion.

3.4.2 Droplet relaxation with surface-tension force only

The second case is to demonstrate the suppression of spurious currents by studying the relaxation of a 2D droplet (density: 1000 $\mathrm{kg/m^3}$, viscosity: 10^{-3} $\mathrm{Pa \cdot s}$)

immersed in a base fluid (density: 1000 kg/m^3, viscosity: 10^{-3} Pa$\cdot$s) [116]. A constant surface tension coefficient is assumed as 0.7 N/m. The initial configuration is a 2D square droplet (side length: 40 µm) sitting at the center of a square 2D computational domain (side length: 100 µm). Gravity is absent from the simulation, and the surface tension force is the only external force acting on the droplet. Accordingly, the surface tension deforms the droplet from its initial square shape to its equilibrium shape, i.e. a 2D circle, gradually.

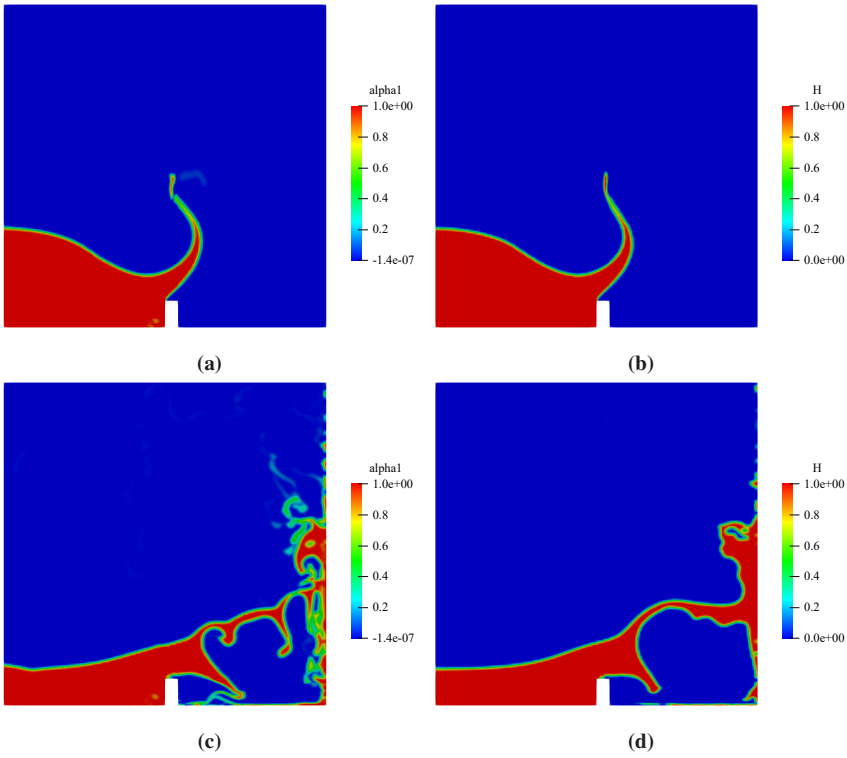


Figure 3.8: Evolutions of the free-surface in the dam break simulations: (a) with the VoF method (t = 0.2 s), (b) with the i-CLSVoF method (t = 0.2 s), (c) with the VoF method (t = 0.5 s), (d) with the i-CLSVoF method (t = 0.5 s).

The maximum velocity of the system is recorded, and the numerical results of our i-CLSVoF framework are compared to simulation results with the conventional VoF and CLSVoF methods. As an alternative to tracking the maximum velocity $|\mathbf{U}_{\max}|$, the dimensionless capillary number $\mathrm{Ca} = \frac{\mu|\mathbf{U}_{\max}|}{\sigma}$ is used to quantify the evolution of spurious velocities in the literature [134]. The convergence of the capillary number with our i-CSLVoF framework is also promising as the ratio between liquid dynamic viscosity $\mu = 10^{-3}$ and surface-tension coefficient $\sigma = 0.07$ is much smaller than unity in our benchmark cases. Concerning the total simulation time for droplet relaxation, 0.001 s is enough to guarantee that the maximum velocity converges to zero numerically (less than 1.0×10^{-8} in our model). As shown in Figure 3.9, the conventional CLSVoF approach can

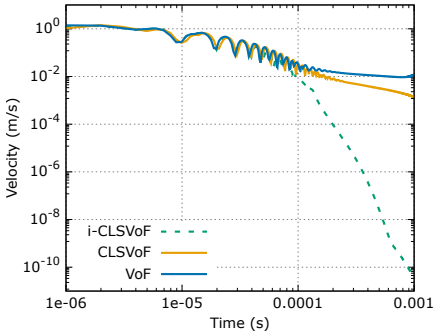


Figure 3.9: Evolution of the un-physical velocities with different surface tension force models.

improve the suppression of un-physical spurious velocity better than the VoF approach; however, the result is still far away from perfect. The dashed line represents the convergence of the velocity within one millisecond by the i-CLSVoF method, and the velocity converges to 10^{-10} which is small enough to eliminate the influence of un-physical spurious currents on the numerical stabilities. Overall, the i-CLSVoF method can reduce the un-physical velocities by at least seven orders of magnitudes.

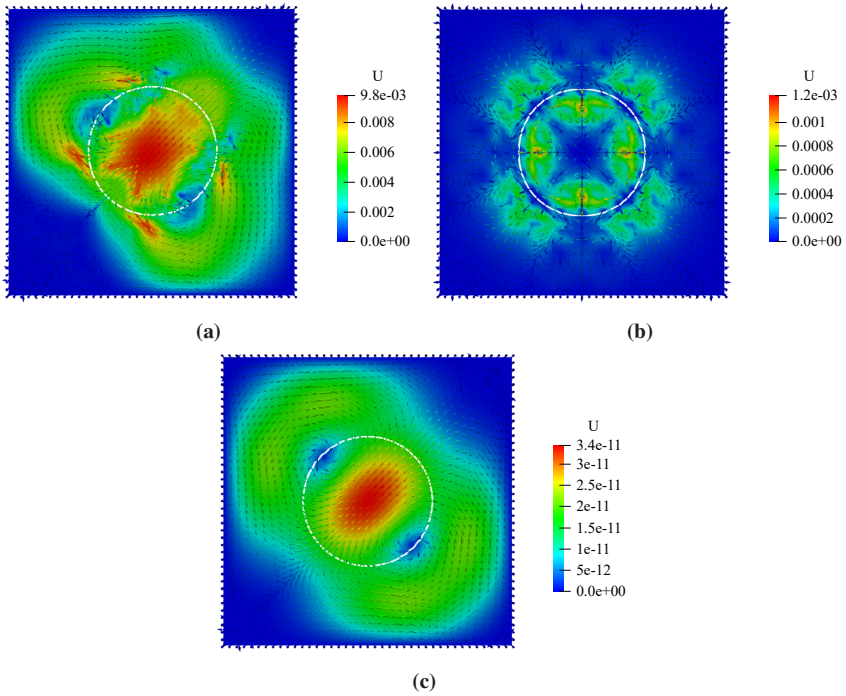


Figure 3.10: Velocity vector contours at 0.001 s with different surface tension models (white circle represents the 0.5 iso-surface for α_l field): (a) VoF, (b) CLSVoF, (c) i-CLSVoF.

The corresponding velocity vector contours at 0.001 s with three different surface tension models, namely the VoF based surface tension force (Eqn. 3.12), the CLSVoF based surface tension force (Eqn. 3.16) and the i-CLSVoF based surface tension force (Eqn. 3.20), are given in Figure 3.10. Here, the white circles represent the 0.5 iso-surface for the liquid volume fraction field α_l. The distributions of spurious vortices for the three different surface-tension models are different. For the normal VoF approach, four large spurious vortices appear around the free surface and point toward four different directions, which leads to the strong spurious velocities that deform the free surface of the droplet and then move the droplet randomly away from its center. Concerning the spurious vortices of the CLSVoF approach, four main vortices point inward but the overall distribution

is symmetrical along the horizontal and vertical directions. For the simulation with the i-CLSVoF method, the maximum velocity is located inside the droplet, however, no large spurious vertices are found around the free surface. The velocity distribution is symmetrical with respect to the diagonal of the computational domain, and the magnitude of the maximum velocity is tiny enough to avoid any influence of spurious currents on the numerical stability.

The analytical solution of the capillary pressure jump across droplets is given by the Young–Laplace equation, where $\Delta p_c = \frac{\sigma}{R} = \frac{0.07\ \text{N/m}}{40/\sqrt{\pi}\times 10^{-6}\ \text{m}} = 3101.8$ Pa is the theoretical solution for 2D droplets with R being the droplet radius [134]. The relative error for the capillary pressure jump is given by

$$E(\Delta p_c) = \frac{|p_c^n - p_c^a|}{p_c^a}, \tag{3.57}$$

where p_c^a and p_c^n are the analytical and numerical capillary pressure, respectively. The comparison among three different surface-tension force models in predicting the capillary pressure is shown in Figure 3.11a. The i-CLSVoF framework

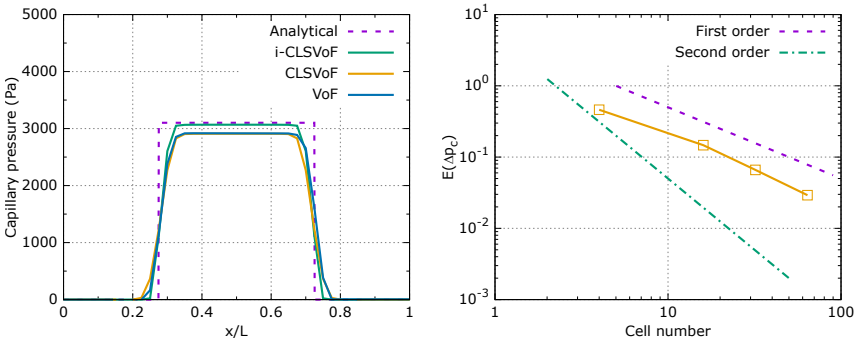


Figure 3.11: (a) Capillary pressure fields of droplets with three different surface tension force models (dotted purple line represents the analytical solution), (b) Relative errors for capillary pressure jump with different mesh resolutions.

developed in this work demonstrates the best agreement between numerical prediction and the analytical solution. The quantitative study on the relative error in predicting the capillary pressure jump with the i-CLSVoF framework is shown in Figure 3.11b, where the first-order convergence is found with our model.

The sharpening coefficient C_{pc} is a key parameter governing the suppression of un-physical spurious currents. An additional parameter study demonstrates the

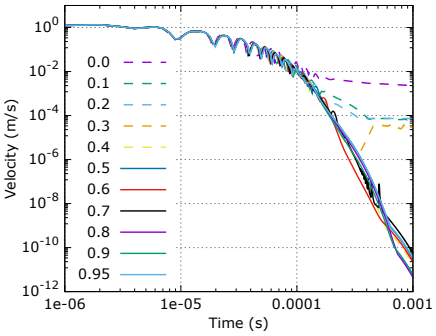


Figure 3.12: Effect of the sharpening coefficient C_{pc} on the suppression of spurious velocity.

effect of the sharpening coefficient on the suppression of the spurious currents. As shown in Figure 3.12, the suppression of un-physical velocities is improved with increasing sharpening coefficient C_{pc}, especially for relatively smaller sharpening coefficient values (0.1 - 0.4). However, for a relatively large sharpening coefficient (0.5 - 0.95), the corresponding results give no major improvement. Note that a large sharpening coefficient may lead to numerical instability.

3.4.3 Spurious currents around sessile droplets

Another challenging problem for surface-tension-dominant flow is to suppress un-physical spurious velocities when a droplet interacts with a substrate with a given contact angle. In this simulation, gravity is absent such that only the surface

tension force dominants the interface deformation. The initial liquid configuration is not an ideal 2D spherical cap due to the numerical errors in initializing the liquid fields, and the error is even larger for a coarse mesh. The surface tension force acting on the droplet deforms the droplet from its initial shape to an ideal 2D spherical cap gradually, and un-physical spurious velocities are in danger to appear around the droplet. In this section, the issue of un-physical spurious velocities around sessile droplets with three different contact angles (60°, 90° and 120°) is investigated. The essential parameters of both liquid and gas phases are listed in Table 3.3. For a relatively small contact angle, i.e. 60°, the comparison

Table 3.3: Parameters for 2D sessile droplet simulations

Phase	Density $[\mathrm{kg/m^3}]$	Kinematic viscosity $[\mathrm{m^2/s}]$	Surface tension $[\mathrm{N/m}]$
Liquid	1000	1×10^{-6}	0.07
Gas	1	1×10^{-5}	-

among three simulations with different surface-tension force models are shown in Figure 3.13. The VoF model shows random un-physical velocities around the interface (represented by a solid white curve) except for the triple contact line region. The CLSVoF and the i-CLSVoF models show relatively large spurious velocities around the triple contact line. Furthermore, the distribution of spurious velocities is axisymmetric along the vertical line in the middle. For the simulation with the i-CLSVoF framework, the magnitude of the spurious velocities is the smallest compared to that of the other two models. The evolution of spurious velocities is shown in Figure 3.14, where the i-CLSVoF framework shows a convergence to 1×10^{-5}, while the VoF and CLSVoF models fail to reach convergence of the velocities. Overall, the i-CLSVoF method can reduce the unphysical velocities around the sessile droplet with a contact angle of $\theta = 60^\circ$ by four orders of magnitudes.

For a contact angle of $\theta = 90^\circ$, all spurious velocities are smaller than those of $\theta = 60^\circ$ as shown in Figure 3.15. The conventional VoF and CLSVoF approaches

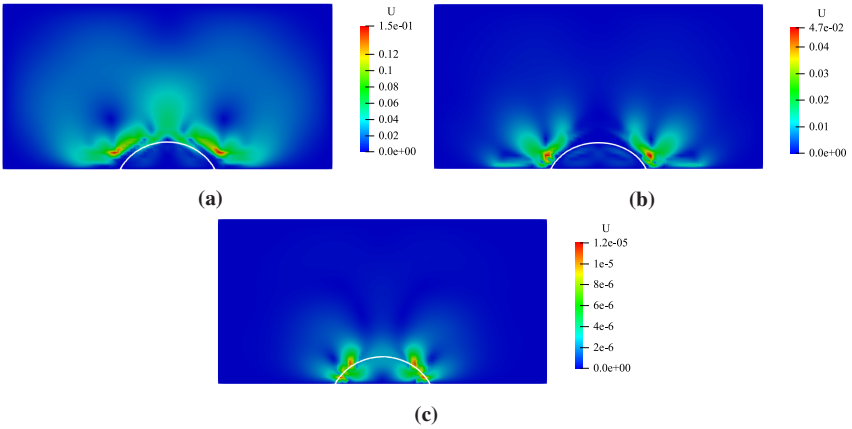


Figure 3.13: Spurious velocities around droplets ($\theta = 60°$) with three different surface-tension force models (the white curve represents the interface): (a) with the VoF method, (b) with the CLSVoF method, (c) with the i-CLSVoF method.

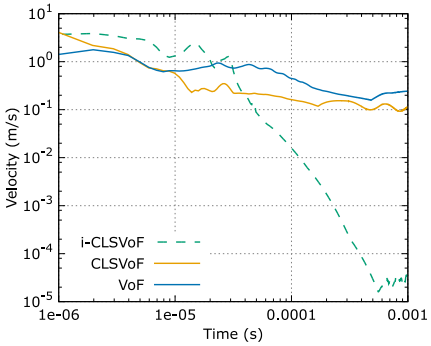


Figure 3.14: Evolution of spurious velocities with different surface-tension models ($\theta = 60°$).

suffer from large un-physical velocities (around 1×10^{-2}). The i-CLSVoF framework reduces the un-physical velocities to 1×10^{-6}, although some relatively large un-physical velocities appear around the peak of the droplet (as shown in Figure 3.15c). The corresponding evolution of the spurious velocities within 0.001 s is shown in Figure 3.16. Similar to the case with a contact angle of

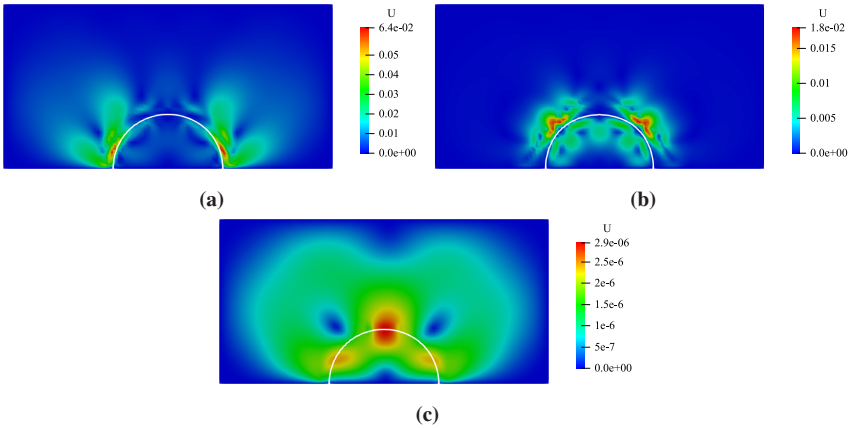


Figure 3.15: Spurious velocities around droplets ($\theta = 90°$) with three different surface-tension force models (the white curve represents the interface): (a) with the VoF method, (b) with the CLSVoF method, (c) with the i-CLSVoF method.

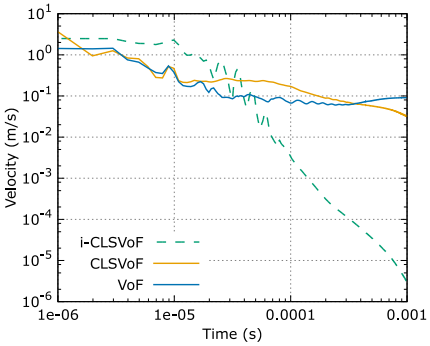


Figure 3.16: Evolution of spurious velocities with different surface-tension models ($\theta = 90°$).

$\theta = 60°$, the i-CLSVoF framework shows best performance in reducing the un-physical velocities much more than the other two methods. Quantitatively, the un-physical spurious velocities are suppressed by four orders of magnitudes by the i-CLSVoF model.

For a large contact angle, e.g. droplet wetting on hydrophobic substrates with a contact angle of $\theta = 120^{\circ}$, the un-physical velocities are much more challenging to suppress even with the i-CLSVoF approach. As shown in Figure 3.17, all

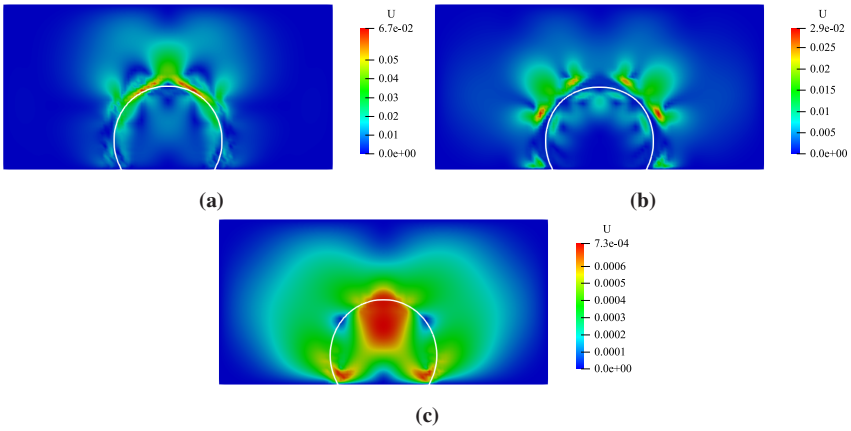


Figure 3.17: Spurious velocities around droplets ($\theta = 120^{\circ}$) with three different surface-tension force models (the white curve represents the interface): (a) with the VoF method, (b) with the CLSVoF method, (c) with the i-CLSVoF method.

three cases suffer from un-physical velocities, while the i-CLSVoF model can suppress the un-physical velocities to 7.0×10^{-4} at least. The largest spurious velocities occur around the peak of the droplet and also at the triple contact line regions for the i-CLSVoF model as shown in Figure 3.17c. Clearly, the i-CLSVoF framework demonstrates the smallest un-physical spurious velocities compared to the other two methods as can be seen in both Figure 3.17 and Figure 3.18, respectively. As shown in Figure 3.18, regarding the reduction of the un-physical spurious velocities, two orders of magnitudes are reached by the i-CLSVoF model compared against the other two models.

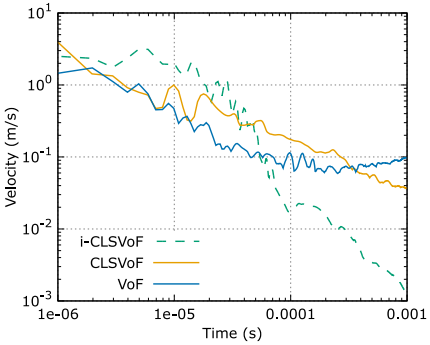


Figure 3.18: Evolution of spurious velocities with different surface-tension models ($\theta = 120°$).

3.5 Summary

In this chapter, the available widely-used interface-capturing approaches in the literature are reviewed and the key issues related to interface diffusion and unphysical spurious velocities when modelling the surface-tension-dominant flow are discussed.

The governing equations were solved based on the one-field formulation using the FVM. The sharp LS method is coupled to the VoF approach to suppress the interface diffusion. Furthermore, the superior performance of the improved interface capturing approach developed in this work is demonstrated by the dam break benchmark case reproduced with both the i-CLSVoF framework and VoF model.

The non-symmetrical Heaviside function was incorporated into the improved surface-tension force model and the filtering approach was further implemented to filter additional non-physical velocities around the interface. Suppression of un-physical velocities can improve the numerical stability and guarantee less influence on the internal flow inside droplets. The droplet relaxation case demonstrates that the i-CLSVoF framework developed in this chapter suppresses unphysical spurious velocities with promising performance compared to the other

two models, namely VoF and CLSVoF. The superiority of the new numerical framework over the conventional VoF and the CLSVoF approaches was demonstrated. A promising amount of suppression of un-physical spurious currents by 2 to 8 orders of magnitude was achieved with our improved numerical framework.

Next, the i-CLSVoF framework is extended to model the evaporation of micro-sized droplets with and without contact line pinning in the following chapter and some numerical issues are detailed there.

4 Modelling droplet evaporation with the i-CLSVoF framework

Modelling droplet evaporation is of great importance for many applications, such as direct inkjet printing [138], spray coating [139], and combustion of fuel droplets [140]. The critical issues in the computational modelling of droplet evaporation are three-fold: free-surface tracking or capturing, accurate calculations of the surface tension force, and the phase change from liquid to vapour [141]. Free-surface capturing and an improved surface-tension force model are discussed in the previous Chapter 3. The current chapter focuses on the computational modelling of phase change or evaporation. Several phase-change models have been developed to model the phase change from liquid to vapour. These include the constant mass flux model [47, 48], the thermally driven model [49–51], and the vapour mass fraction gradient model [48, 52, 53]. The challenging part in modelling phase change or evaporation is to address the velocity jump at the interface, which results in some numerical difficulties. Kunkelmann developed an approach that removes the source terms at the interface cells and defines positive and negative mass sources in the most adjacent liquid and gas cells, respectively [54]. This approach was demonstrated to have good performance in modelling boiling [54]. A similar method is also implemented into the open-source code Gerris for modelling droplet evaporation subject to a large mass transfer rate [55]. Both methods are highly dependent on the mesh resolution at the interface. Normally, Adaptive Mesh Refinement (AMR) is needed to accurately cut the interface region into two different regions with negative

[1] Part of this chapter with minor changes has been submitted to arXiv as a preprint [176].

and positive mass sources. These numerical models are implemented in either in-house or commercial codes. Thus, it is proposed to develop a comprehensive solver in the open-source C++ toolbox OpenFOAM to incorporate simple yet efficient evaporation models to model the micro-sized sessile droplet evaporation with negligible influence of un-physical spurious currents.

4.1 Governing equations and numerical method

The primary governing equations for the incompressible flow without phase change discussed in Chapter 3 are modified to account for the phase change from the liquid phase to the vapour phase occurring at the liquid surface. The velocity field for the incompressible flow with phase change is not divergence-free any more. A volumetric mass source term $\dot{m}$ (mass change rate per unit volume) on the right-hand side of the continuity equation is introduced, and the equation reads

$$\nabla \cdot \mathbf{U} = \dot{m}(\frac{1}{\rho_g} - \frac{1}{\rho_l}). \tag{4.1}$$

Here, $\dot{m}$ means the mass loss of the liquid phase, which reappears at the vapour phase with the same amount. $\dot{m}$ is only non-zero around the free surface and is zero in the region far away from the interface. Correspondingly, ρ_g and ρ_l are the density of the gas and liquid phases, respectively. The momentum equation is the same as for the incompressible flow without phase change.

However, the velocity field in the VoF transport Eqn. 3.6 is replaced by the new interface velocity field $\mathbf{U}_\Gamma$ as

$$\frac{\partial \alpha_l}{\partial t} + \nabla \cdot (\alpha_l \mathbf{U}_\Gamma) = \alpha_l \nabla \cdot \mathbf{U}_\Gamma. \tag{4.2}$$

The reason behind that lies in preliminary numerical simulations, which confirmed that using the default one-field velocity $\mathbf{U}$ to solve the VoF transport equation tends to overestimate the evaporation rate, and the same conclusion can be

found in the literature [47]. The calculations of $\mathbf{U}_\Gamma$ can be derived from the interface mass flux balance. The interface mass flux per unit area J $[\mathrm{kg}/(\mathrm{m}^2 \cdot \mathrm{s})]$ when the phase changes is derived from the mass flux balance across the interface,

$$J = \rho_l(\mathbf{U}_e - \mathbf{U}_\Gamma) \cdot \mathbf{n} = \rho_g(\mathbf{U}_g - \mathbf{U}_\Gamma) \cdot \mathbf{n}, \tag{4.3}$$

where $\mathbf{n}$ is the interface normal vector, and $\mathbf{U}_e$ and $\mathbf{U}_g$ are the fluid velocities in the liquid and gas phase, respectively. The interface velocity $\mathbf{U}_\Gamma$ can be accordingly derived as

$$\mathbf{U}_\Gamma = \mathbf{U}_e - \frac{J}{\rho_l}\mathbf{n}, \tag{4.4}$$

where the second term $J/\rho_l \cdot \mathbf{n}$ is the interface recession velocity and accounts for the interface shrinking during the evaporation process. The first term $\mathbf{U}_e$ is known as the extended divergence-free liquid velocity.

A new method is developed in this chapter to reconstruct a new divergence-free velocity field $\mathbf{U}_e$. The basic idea behind the reconstruction of $\mathbf{U}_e$ is to solve the velocity potential equation

$$\begin{cases} a\phi + \nabla^2\phi = \dot{m}(\frac{1}{\rho_g} - \frac{1}{\rho_l}), \\ \mathbf{U}_s = \nabla\phi, \end{cases} \tag{4.5}$$

in the whole computational domain with the homogeneous Dirichlet boundary condition applied on the boundary to guarantee that the velocity potential at the domain boundary is zero. Here, ϕ is the velocity potential, and $\mathbf{U}_s$ denotes the evaporation-induced Stefan flow velocity (equal to the gradient of the velocity potential). The critical parameter a is used to divide the whole computation domain into two sub-domains (refer to Figure 4.1), where a is zero in the liquid phase (blue circle) and within the three most adjacent cells around the interface (cells in pink in the circle with solid line). For the rest of the computational domain, a can be any arbitrary non-zero value. The square of the time-step size is used in the current study. The new method guarantees more numerical stability than solving Eqn. 4.5 with a equal zero in the whole computational domain as

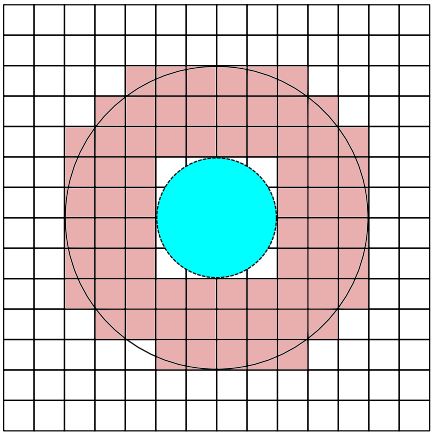


Figure 4.1: The sub-domains for solving the velocity potential equation.

proposed in [48]. As the right-hand side of the velocity potential Eqn. 4.5 has the same source term as the one of the continuity Eqn. 4.1 for the incompressible flow with phase change, the new divergence-free velocity field $\mathbf{U}_e$ is defined by subtracting the evaporation-induced Stefan flow velocity $\mathbf{U}_s$ from the one-field velocity field $\mathbf{U}$ as

$$\mathbf{U}_e = \mathbf{U} - \mathbf{U}_s. \tag{4.6}$$

Typically, the divergence of $\mathbf{U}_e$ should approximate 10^{-8} or even smaller values of 10^{-10}, which can be regarded as zero numerically. As an alternative to solve and update the liquid volume fraction field α_l with the implicit source term as shown in Eqn. 4.2, the divergence-free velocity field $\mathbf{U}_e$ can also be used to advect the free surface with either an explicit

$$\frac{\partial \alpha_l}{\partial t} + \nabla \cdot (\alpha_l \mathbf{U}_e) = -\frac{\dot{m}}{\rho_l} \tag{4.7}$$

or an implicit

$$\frac{\partial \alpha_l}{\partial t} + \nabla \cdot (\alpha_l \mathbf{U}_e) = \alpha_l \frac{-\dot{m}}{(\alpha_l + \delta_s)\rho_l} \tag{4.8}$$

source term[2] accounting for the mass loss in the liquid phase due to the evaporation where δ_s in Eqn. 4.8 is a small number to guarantee a non-zero denominator.

4.2 The i-CLSVoF framework with evaporation

The i-CLSVoF framework developed in Chapter 3 suppresses spurious velocities and guarantees sharp interface capturing. Computational modelling of droplet evaporation, especially for micro-sized droplets, suffers from un-physical spurious velocities, and the i-CLSVoF is a promising tool to model micro-sized droplet evaporation. In this chapter, three different evaporation models are introduced, implemented and validated. These models share the same governing equations except for the calculation of the mass source term per unit area J.

4.2.1 The constant mass flux evaporation model

Starting with the most simple evaporation model, the only parameter needing to be defined is the constant mass flux per unit area J. In contrast to calculating the mass flux J with complex equations, for instance, by temperature difference, a given constant, e.g. $J = 1.25 \times 10^{-2}$ kg/(m^2·s) is specified in this chapter. The constant mass flux evaporation model can be used to validate the implementations of the governing equations before implementing complex approaches to conduct calculations of the source terms.

4.2.2 The thermally driven evaporation model

The thermally driven evaporation drives the phase change from liquid to vapour when the temperature around the liquid interface is higher than its saturation temperature T_{sat}, and the mass flux depends on the temperature difference at

[2] When solving the VoF equation in OpenFOAM, the implicit source term is recommended.

the interface [49, 142]. Tracking the temperature distribution around an evaporating droplet is crucial to model droplet evaporation subjected to the ambient temperature gradient. Conservation of thermal energy is given by the following temperature equation

$$\frac{\partial(\rho c_p T)}{\partial t} + \nabla\cdot(\rho c_p \mathbf{U} T) = \nabla\cdot(k\nabla T) - \dot{m} h_{\mathrm{ev}} + \left[\frac{\partial(\rho c_p)}{\partial t} + \nabla\cdot(\rho c_p \mathbf{U})\right] T, \quad (4.9)$$

where T is the temperature field, c_p the specific heat capacity, k the thermal conductivity and h_{ev} the enthalpy or latent heat of evaporation. The second term on the right-hand side of the temperature equation is due to evaporation-induced cooling, and the last term couples to the mass source term.

The mass flux J of the thermally driven evaporation model is given by

$$J = \frac{T - T_{\mathrm{sat}}}{R_{\mathrm{int}} h_{\mathrm{ev}}}, \quad (4.10)$$

where h_{ev} is the latent heat of evaporation and R_{int} the heat resistance of the liquid-vapour interface. R_{int} is calculated by

$$R_{\mathrm{int}} = \frac{2-\chi_e}{2\chi_e}\frac{\sqrt{2\pi R_{\mathrm{gas}}}}{h_{\mathrm{ev}}^2}\frac{T_{\mathrm{sat}}^{3/2}}{\rho_g}, \quad (4.11)$$

where R_{gas} is the gas constant and χ_e the evaporation coefficient which depends on the density ratio $\lambda = \rho_l/\rho_g$ between liquid and gas phase [143]. The density-ratio-dependent evaporation coefficient

$$\chi_e = \left\{1 - \left(\frac{1}{\lambda}\right)^{\frac{1}{3}}\right\} \exp\left(-\frac{1}{2\lambda^{\frac{1}{3}} - 2}\right), \quad (4.12)$$

is newly introduced in this chapter instead of determining χ_e empirically [144]. The evaporation coefficient χ_e increases gradually with the density ratio λ and tends to reach a plateau at high-density ratios as shown in Figure 4.2.

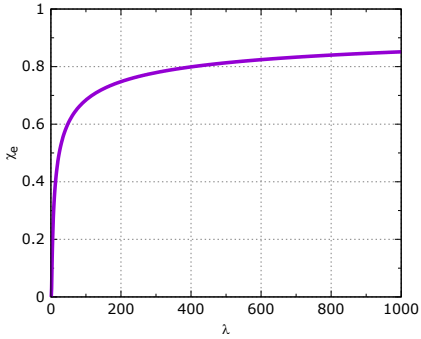


Figure 4.2: The evaporation coefficient χ_e calculated by Eqn. 4.12 versus the density ratio λ in the range up to 1000.

4.2.3 The vapour mass fraction gradient evaporation model

For droplet evaporation at room temperature, the vapour concentration gradient around the evaporating droplet drives the phase change from liquid to vapour phase. Accordingly, the vapour concentration is solved directly to model the phase change from liquid to vapour in the literature [11]. To simplify the characterization of numerical models, in the current work, a dimensionless quantity called the vapour mass fraction field Y is solved for and updated by the convection-diffusion equation given as

$$\frac{\partial Y}{\partial t} + \nabla \cdot (Y\mathbf{U}) = D_v \nabla^2 Y, \tag{4.13}$$

where D_v is the vapour diffusivity coefficient. The convection-diffusion equation is solved on the whole computational domain with prescribed so-called internal boundary conditions. For applying this internal boundary condition, all the liquid cells are assigned the saturation mass fraction which accordingly guarantees that vapour diffuses only from the liquid surface to the gas domain without unphysical diffusion going back to the droplet [145].

The mass flux per unit area J can be calculated from mass balance across the interface with the single-component liquid $J(1-Y^\Gamma) = \rho_g D_v \nabla_\Gamma Y \mathbf{n}^\Gamma$, which leads to the following formula for the mass flux

$$J = \frac{\rho_g D_v \nabla^\Gamma Y \mathbf{n}^\Gamma}{1-Y^\Gamma}, \tag{4.14}$$

where $\mathbf{n}^\Gamma$ is the unit interface normal and ∇^Γ denotes the gradient at the interface. Additionally, Y^Γ is the saturation vapour mass fraction, which is given by the Clausius–Clapeyron relation

$$Y^\Gamma = \frac{X_v M_l}{(1-X_v)M_g + X_v M_l}, \tag{4.15}$$

where X_v is further given by

$$X_v = \exp\left[\frac{-h_{\mathrm{ev}} M_l}{R_{\mathrm{gas}}}\left(\frac{1}{T} - \frac{1}{T_{\mathrm{sat}}}\right)\right] \tag{4.16}$$

with M_l and M_g being the molar mass of liquid and gas, respectively, and T the temperature, T_{sat} the saturation temperature, h_{ev} the latent heat of evaporation and R_{gas} the gas constant for the liquid phase [53].

The mass source term per unit volume $\dot{m}$ can be calculated by

$$\dot{m} = J|\nabla \alpha_l| \tag{4.17}$$

once the mass source term per unit area J is calculated with the three aforementioned models. Normally, the mass source term $\dot{m}$ is only non-zero at a thin layer around the droplet interface (see Figure 4.3a). A preliminary numerical study showed that it leads to numerical instability, especially for evaporation with large mass flux.

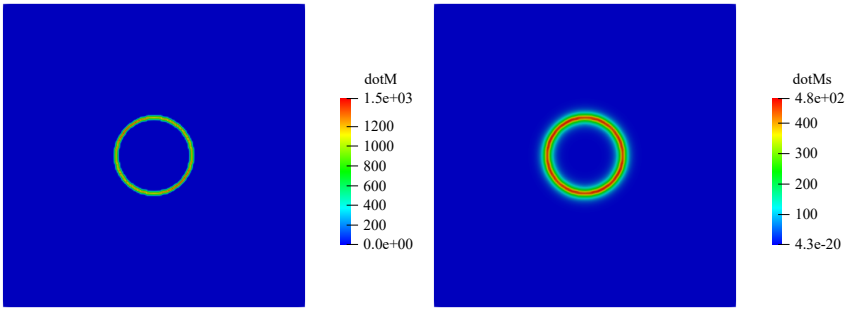


Figure 4.3: The mass source distributions for two different cases: (a) without smearing, (b) with smearing.

The improved solution proposed in this thesis is to extend the distribution of $\dot{m}$ to a wide band by smearing $\dot{m}$ over several adjacent cells near the interface (see Figure 4.3b). The basic idea is to solve the Helmholtz equation

$$\dot{m}_s = \dot{m} + (\Delta x N)^2 \Delta \dot{m}_s \tag{4.18}$$

with the homogeneous Neumann boundary conditions [49]. Here, $\dot{m}_s$ is the

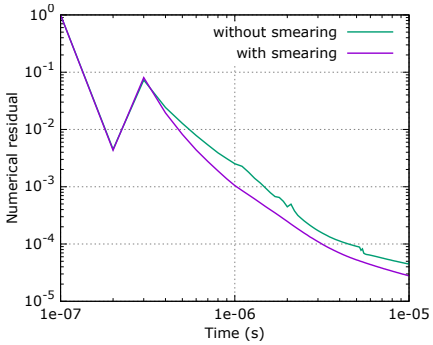


Figure 4.4: Evolution of the numerical residual for two different cases with and without smearing the mass source term.

smeared mass source term, Δx the minimal mesh size, and N is the number of cells over which the mass source term is smeared along the interface normal. Employing smearing of the mass source term, the numerical residual for simple 2D static droplet evaporation cases has been recorded. It turned out that the smeared approach can guarantee a smaller numerical residual as can be seen in Figure 4.4.

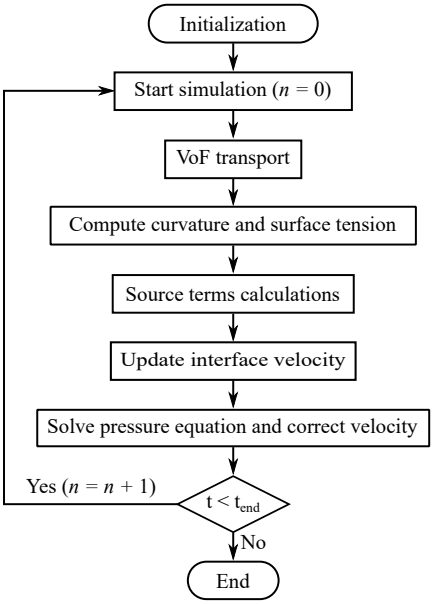


Figure 4.5: Flowchart of solving governing equations with the i-CLSVoF framework when phase changes.

The overall solution procedure is outlined in Figure 4.5. In contrast to the flowchart in Figure 3.6 discussed in Chapter 3, some other additional equations have to be solved to update some essential quantities. In order to calculate the mass flux when modelling the droplet evaporation with three different models,

the vapour mass fraction or temperature distribution needs to be updated by solving the corresponding Eqs. 4.13 and 4.9, respectively. A divergence-free velocity field is reconstructed by solving the velocity potential Eqn. 4.5, and then the interface velocity field can be updated by solving Eqn. 4.4. Either the divergence-free velocity or the interface velocity field can be used to solve the VoF Eqn. 4.7 or Eqn. 4.8 to capture the free surface.

4.3 Contact angle hysteresis and contact line pinning

The relation between the equilibrium contact angle and three different surface tensions is given by Young's equation as

$$\cos\,\theta_e = \frac{\sigma_{sv} - \sigma_{sl}}{\sigma_{lv}}, \tag{4.19}$$

where θ_e is the equilibrium contact angle, σ_{sv} the surface tension between solid and vapour phase, σ_{sl} the surface tension between solid and liquid phase, and σ_{lv} the surface tension between liquid and vapour phase [146]. The equilibrium contact angle is measured between the solid substrate and the gas-liquid interface. It is used to characterize the wetting behaviour of sessile droplets. For the dynamic behaviour of droplets, two additional contact angles, namely, advancing contact angle θ_a and receding contact angle θ_r, depending on material parameters of both the liquid and the solid, are used. The advancing contact angle θ_a is related to an increase in the contact area between the droplet and the substrate. In contrast, the receding contact angle θ_r belongs to a decreasing contact area as shown in the inset of Figure 4.6 [147]. The contact line represents a line or curve at which three different phases, namely gas, liquid and solid meet. The contact line moves with a certain velocity when a droplet spreads on a substrate. The relation between contact line velocity $\mathbf{U}_{cl}$ and contact angle θ is demonstrated in Figure 4.6. The

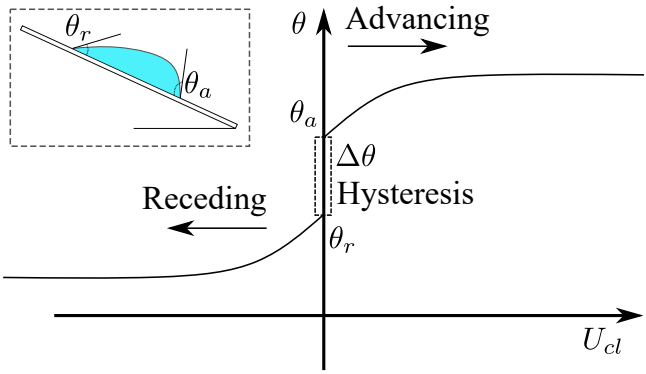


Figure 4.6: Contact line velocity versus contact angle, and the contact angle hysteresis.

contact angle hysteresis is defined by the difference between the advancing and receding contact angles as

$$\Delta\theta = \theta_a - \theta_r. \tag{4.20}$$

The contact angle hysteresis is of great significance in many applications, e.g. coating, floating and painting, etc. A review of classical and modern methods to measure the contact angle hysteresis and applications can be found in [148]. The contact line velocity $\mathbf{U}_{cl}$ is zero within the contact angle hysteresis region, where thus the contact line is fixed. In contrast, the contact line has a non-zero velocity outside the contact angle hysteresis region. The contact line moves outwards and increases its contact angle during the advancing phase, while it moves inwards and decreases its contact angle during the receding phase.

The initial contact radius of an evaporating droplet stays constant when sitting on a substrate with rough surface as shown in Figure 4.7a. This is droplet evaporation with so-called contact line pinning [7, 149]. Gradual accumulations of insolvable particles around the fixed contact line were also found to induce and enhance contact line pinning when drying suspension droplets on substrates. This is known as self-pinning of the contact line as mentioned in the literature [56, 150] and shown in Figure 4.7b.

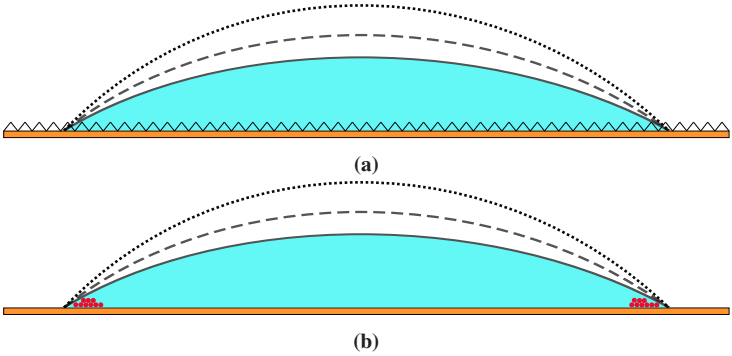


Figure 4.7: Consequences behind the contact line pinning: (a) surface roughness, (b) particle accumulating around the triple contact line region.

The contact line hysteresis can inspire computational modelling of contact line pinning within the VoF framework. The basic idea is to maintain the current contact angle within the hysteresis. Complex wetting of droplets on substrates subjected to a shear flow was investigated numerically with OpenFOAM by Linder [151, 152]. In Linder's work, a simple yet efficient numerical framework was proposed to fix the moving contact line. This numerical framework was further extended to model the forced wetting of droplets in a shear flow with improved numerical stability and incorporated the geometric VoF approach by Rettenmaier [153]. In this thesis, these two models are extended to model contact line pinning in the presence of phase change or evaporation of the liquid phase. The current contact angle θ_0, the advancing contact angle θ_a, the receding contact angle θ_r and the contact line velocity $\mathbf{U}_{cl}$ are essential parameters to fix the moving contact line.

4.3.1 Contact angles and contact line velocity

As shown in Figure 4.8, the current contact angle θ_0 between the droplet and the substrate is calculated by

$$\theta_0 = \arccos\left(\mathbf{n}_w \cdot \mathbf{n}_{i,w,0}\right). \tag{4.21}$$

The interface normal $\mathbf{n}_{i,w,0}$ at the triple contact line point can be calculated by

$$\mathbf{n}_{i,w,0} = \cos\theta_0 \mathbf{n}_w + \sin\theta_0 \mathbf{t}_w, \tag{4.22}$$

where θ_0 is the current contact angle (see Figure 4.8), and $\mathbf{n}_w$ and $\mathbf{t}_w$ are the unit vectors normal and tangential to the wall, respectively [154]. The interface normal $\mathbf{n}_{i,w}$ after the motion and displacement of the interface is calculated by

$$\mathbf{n}_{i,w} = \cos\theta \mathbf{n}_w + \sin\theta \mathbf{t}_w, \tag{4.23}$$

where θ is the target contact angle. However, the current contact angle θ_0 is not equal to the target contact angle θ due to the motion and displacement of the interface. In Eqn. 4.22, the interface normal $\mathbf{n}_{i,w,0}$ and the wall normal $\mathbf{n}_w$ can be

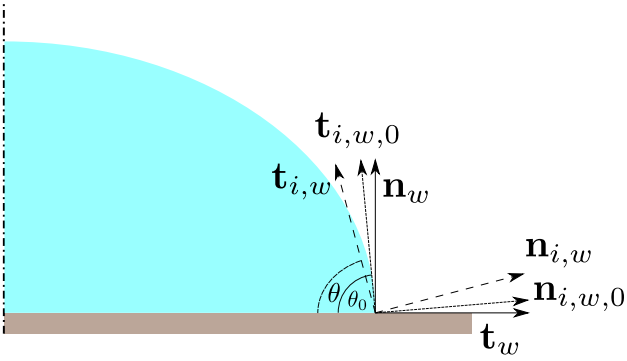


Figure 4.8: The current contact angle of a droplet resting on a substrate.

calculated with the liquid volume fraction field α_l. Thus, Eqn. 4.22 leads to

$$\mathbf{t}_w = \frac{1}{\sin\theta_0}\mathbf{n}_{i,w,0} - \frac{\cos\theta_0}{\sin\theta_0}\mathbf{n}_w. \tag{4.24}$$

Substituting Eqn. 4.24 into Eqn. 4.23, leads to

$$\begin{aligned}
\mathbf{n}_{i,w} &= \cos\theta\mathbf{n}_w + \sin\theta\left(\frac{1}{\sin\theta_0}\mathbf{n}_{i,w,0} - \frac{\cos\theta_0}{\sin\theta_0}\mathbf{n}_w\right) \\
&= \frac{\sin^2\theta_0\cos\theta - \sin\theta_0\sin\theta\cos\theta_0}{\sin^2\theta_0}\mathbf{n}_w + \frac{\sin\theta_0\sin\theta}{\sin^2\theta_0}\mathbf{n}_{i,w,0} \\
&= \frac{\cos\theta - \cos\theta_0\left[\cos\theta_0\cos\theta + \sin\theta_0\sin\theta\right]}{\sin^2\theta_0}\mathbf{n}_w + \frac{\cos(\theta_0-\theta) - \cos\theta_0\cos\theta}{\sin^2\theta_0}\mathbf{n}_{i,w,0} \\
&= \frac{\cos\theta - \cos\theta_0\cos(\theta_0-\theta)}{1-\cos^2\theta_0}\mathbf{n}_w + \frac{\cos(\theta_0-\theta) - \cos\theta_0\cos\theta}{1-\cos^2\theta_0}\mathbf{n}_{i,w,0}.
\end{aligned} \tag{4.25}$$

Eqn. 4.25 can be abbreviated as

$$\mathbf{n}_{i,w} = a\mathbf{n}_w + b\mathbf{n}_{i,w,0}, \tag{4.26}$$

where the coefficients a and b are given by

$$a = \frac{\cos\theta - \cos\theta_0\cos(\theta_0-\theta)}{1-\cos^2\theta_0}, \tag{4.27}$$

$$b = \frac{\cos(\theta_0-\theta) - \cos\theta_0\cos\theta}{1-\cos^2\theta_0}, \tag{4.28}$$

respectively.

The contact angle decreases during the droplet drying/evaporation process, and a relative velocity between the droplet and the substrate is found. This is the so-called contact line velocity U_{cl}. The direction of the contact line velocity indicates an advancing or receding phase. The capillary number $\mathrm{Ca} = \mu U_{cl}/\sigma$ is used in this thesis to distinguish the two different phases by

$$\mathrm{Ca}\begin{cases} > 0 & \Rightarrow \text{ advancing phase,} \\ > 0 & \Rightarrow \text{ receding phase.} \end{cases} \tag{4.29}$$

The numerical singularity around the triple contact line region brings many numerical challenges, and some approaches have been proposed to approximate the contact line velocity [155]. In this chapter, a simple model proposed by Linder and Rettenmaier is adopted to calculate the contact line velocity U_{cl} [151, 153]. As shown in Figure 4.9, the contact line experiences both translational and rotational motion when the contact line moves. Capital T and R indicate in Figure 4.9

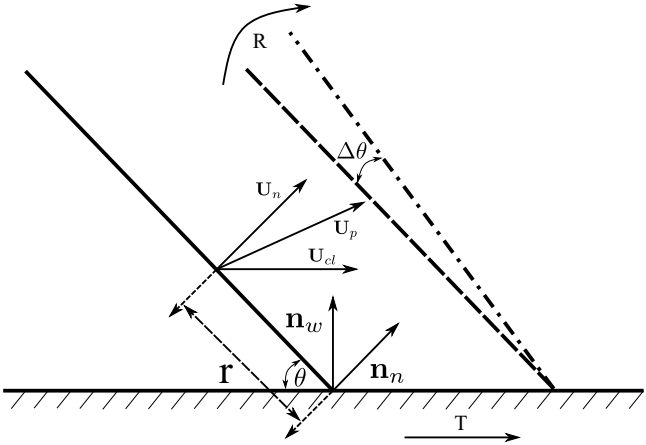


Figure 4.9: Translational and rotational motion of the interface.

the translational and rotational motion, respectively. In this section, one essential assumption is that the interface near the wall is nearly planar. $\mathbf{U}_n$ is the velocity vector normal to the interface, and $\Delta\theta$ is the angular displacement of the interface. According to the geometrical configuration and detailed derivations in the literature [156], it leads to

$$U_n = \mathbf{U}_p \mathbf{n}_n \approx U_{cl} \sin\theta + r\frac{\Delta\theta}{\Delta t}, \tag{4.30}$$

where $\mathbf{U}_p$ and $\mathbf{n}_n$ are the material point velocity and the unit vector normal to the interface, respectively. U_n is the interface velocity field normal to the interface. It is noted that $r\frac{\Delta\theta}{\Delta t}$ can be negligible as r is very small, and thus leading to

$$U_{cl} = \frac{U_n}{\sin\theta}. \tag{4.31}$$

According to the geometric configuration demonstrated in Figure 4.9, it leads to

$$\cos\theta = \mathbf{n}_w \cdot \mathbf{n}_n, \tag{4.32}$$

where $\mathbf{n}_w$ is a vector normal to the wall. Finally, the contact line velocity can be approximated by

$$U_{cl} \approx \frac{\mathbf{U}_p \cdot \mathbf{n}_n}{\sqrt{1-(\mathbf{n}_w \cdot \mathbf{n}_n)^2}}, \tag{4.33}$$

once the material point velocity of the contact line, wall and interface normal vectors are known. The contact line velocity calculated with the formula as mentioned above is crucial in realizing the contact line pinning, and this part is detailed in the next section.

4.3.2 The Robin boundary condition

To realize the contact line pinning within the i-CLSVoF framework developed in this thesis, a Robin boundary condition developed by Linder [151] is adopted to prescribe a mixed boundary condition to fix the moving contact line. This Robin boundary condition is a linear combination of the Neumann and Dirichlet boundary conditions and given by

$$\alpha_w = f\alpha_w + (1-f)\left(\alpha_w + \nabla\alpha_w \cdot \mathbf{d}_w\right), \tag{4.34}$$

where α_w is the volume fraction field at the wall boundary, and $\mathbf{d}_w$ distance vector from the cell midpoint to the face center at the wall boundary. The blending factor f switches between zero and one. $f = 0$ leads to the Neumann boundary

condition, according to which the contact line is free to move, while $f = 1$ ends up with the Dirichlet boundary condition guaranteeing a fixed contact line. The basic algorithm to realize the Robin boundary condition is outlined in **Algorithm 1**.

Algorithm 1 The algorithm for realizing the Robin boundary condition.

for all boundary cells do
calculate the current contact angle θ_{cur}
calculate the capillary number Ca
if $(\theta_{\text{cur}} > \theta_A \;\&\&\; \text{Ca} > 0) \;\|\; (\theta_{\text{cur}} < \theta_R \;\&\&\; \text{Ca} < 0)$ **then**
    Apply the Neumann BC
    $\nabla\alpha \leftarrow$ dynamic contact angle model
else
    Apply the Dirichlet BC
    α_w fixed
end if

To determine which boundary condition should be applied, the current contact angle θ_{cur} is updated and compared against the given values of the hysteresis. Different cases should be taken into account as follows:

- The contact line is forced to move when the current contact angle θ_{cur} exceeds the advancing contact angle θ_A, and the interface velocity near the contact line moves towards the gas phase, namely $\text{Ca} > 0$.
- Similarly, the contact line is forced to move when the current contact angle θ_{cur} is smaller than the receding contact angle θ_R, and the interface velocity near the contact line moves towards the liquid phase, namely $\text{Ca} < 0$.
- In the third case, the current contact angle is within the contact angle hysteresis, and thus the contact line remains pinned.

4.4 Numerical validation and case study

Several benchmark cases are conducted to validate the i-CLSVoF framework with evaporation. An axisymmetric model is adopted in the current study to save computational cost. As shown in Figure 4.10, only a quarter of a 2D droplet with an initial diameter of 250 μm is simulated, and symmetry boundary conditions on the left and bottom sides are applied. Outflow boundary conditions, namely a Dirichlet boundary condition for the pressure field and a Newmann boundary condition for the velocity field, are applied on the other two sides to let the newly generated vapour from the liquid surface leave the domain freely.

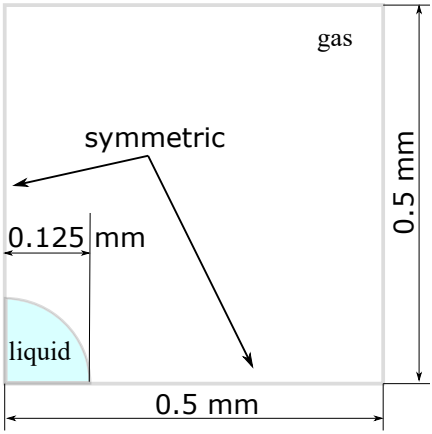


Figure 4.10: The schematic diagram of the numerical setup for 2D droplet evaporation.

The parameters used in the evaporation cases are listed in Table 4.1. In order to save the computational cost, the scaled liquid density 10 is used in the current study. A constant surface-tension coefficient is used here, meaning that the effect of temperature on the surface-tension coefficient is not considered in these simulations.

Table 4.1: Physical properties for liquid and gas phases.

Property	Liquid	Gas	Units
Density ρ	10	1	$[\mathrm{kg}\cdot\mathrm{m}^{-3}]$
Dynamic viscosity μ	1×10^{-3}	1×10^{-5}	$[\mathrm{Pa}\cdot\mathrm{s}]$
Thermal conductivity k	0.1	0.01	$[\mathrm{W}\cdot\mathrm{m}^{-1}\cdot\mathrm{K}^{-1}]$
Specific heat capacity c_p	4181	1900	$[\mathrm{J}\cdot\mathrm{kg}^{-1}\cdot\mathrm{K}^{-1}]$
Molar mass M	0.018	0.029	$[\mathrm{kg/mol}]$
Entropy of evaporation h_{ev}	1×10^{6}	-	$[\mathrm{J}\cdot\mathrm{kg}^{-1}]$
Surface-tension coefficient σ	0.072	-	$[\mathrm{N}\cdot\mathrm{m}^{-1}]$
Vapour diffusivity D_v	-	1×10^{-5}	$[\mathrm{m}^2/\mathrm{s}]$

4.4.1 Droplet evaporation with the constant mass flux evaporation model

First, the evaporation of 2D droplets with the pre-defined constant mass flux J is studied, as this can validate the implementation of the governing equations in simple manner without taking the calculation of source terms into account. The analytical solution for 2D droplet evaporation with constant mass flux is derived in this section. Let R and R_0 be the shrinking and the initial droplet radius, respectively. The droplet shrinks when the evaporation moves the interface inwards with the interface velocity $\mathbf{U}_\Gamma$. For the 2D sessile droplet cases, the interface velocity $\mathbf{U}_\Gamma$ equals the interface recession velocity $\mathbf{U}_{\mathrm{re}}$, leading to

$$R = R_0 - |\mathbf{U}_\Gamma| t = R_0 - \frac{J}{\rho_l} t. \tag{4.35}$$

Let D and D_0 denote the shrinking and the initial droplet diameter, respectively, and t^* the total evaporation time. The dimensionless droplet diameter changes with the dimensionless time during the evaporation process which is given as

$$\frac{D}{D_0} = 1 - \frac{2J}{\rho_l D_0} t = 1 - \frac{2Jt^*}{\rho_l D_0} \frac{t}{t^*}. \tag{4.36}$$

This formula is the analytical solution for 2D static droplet evaporation subject to the constant mass flux and is also valid for 3D static evaporation cases.

The crucial aspect of modelling droplet evaporation with the i-CLSVoF framework lies in reconstructing the new divergence-free velocity field $\mathbf{U}_e$. After implementing the velocity-potential approach (refer to Eqn. 4.5) proposed in this work for the reconstruction of the new velocity field, three different velocity fields are obtained by the simulations: the one-field velocity $\mathbf{U}$, the evaporation-induced Stefan flow velocity $\mathbf{U}_s$, and the newly reconstructed velocity $\mathbf{U}_e$ for a 2D droplet subject to the constant mass flux are shown in Figure 4.11a, 4.11b and 4.11c.

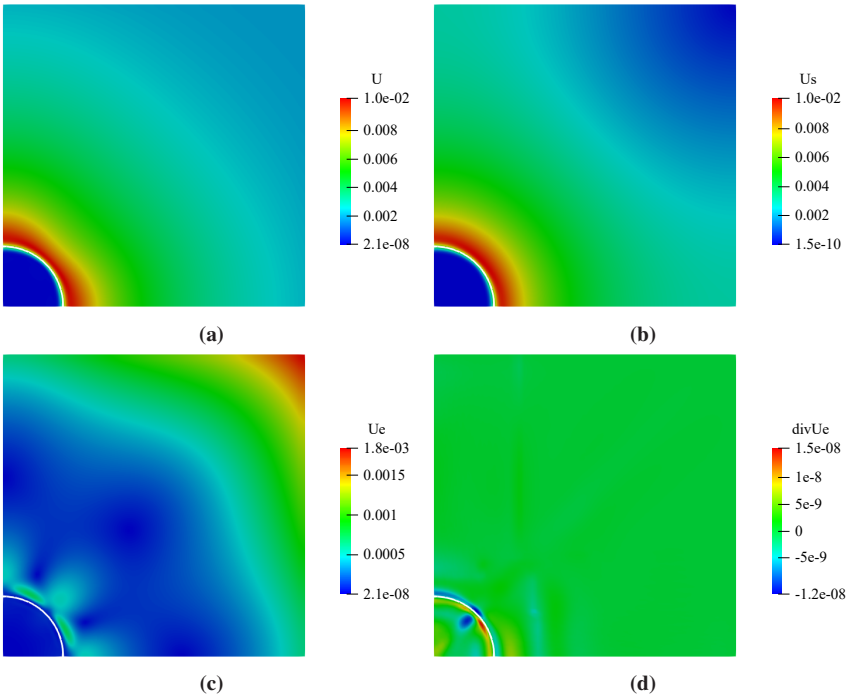


Figure 4.11: Different velocities fields of an evaporating droplet (white line represents the interface); (a): the one-field velocity field, (b): the evaporation-induced Stefan flow velocity field, (c): the divergence-free velocity field, (d): the divergence of $\mathbf{U}_e$.

The maximum magnitudes of $\mathbf{U}$ and $\mathbf{U}_s$ are the same. It shows the contour of the divergence of velocity field $\mathbf{U}_e$ in Figure 4.11d, demonstrating that its magnitude is around 10^{-8} which can be regarded as numerically zero. This means that the divergence-free velocity field $\mathbf{U}_e$ is successfully reconstructed. The divergence-free velocity field $\mathbf{U}_e$ for the sessile droplet evaporation case should be zero, but some velocity vertices can be seen around the interface in Figure 4.11c. The reason lies in both evaporation and surface tension deforming the interface during the evaporation process, where the interplay between evaporation and surface tension produces additional spurious velocities. These additional spurious currents are challenging to reduce, especially with the algebraic VoF approach of OpenFOAM on which the i-CLSVoF framework is based. Furthermore, the distribution of the spurious velocities is symmetrical along the diagonal of the computational domain as shown in Figure 4.11c. A symmetrical distribution of spurious velocities around an evaporating droplet is more stable than the case with random distribution.

Figure 4.12 presents the velocity contour of the one-filed velocity field $\mathbf{U}$. All

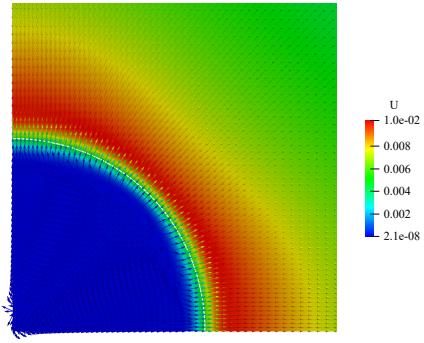


Figure 4.12: Vector contour of the one-field velocity field $\mathbf{U}$ (the white solid line represents the interface).

the vectors are perpendicular to the interface (represented by the white solid line) and point from the liquid phase to the vapour phase. Additionally, a velocity

jump can be also found around the interface. This demonstrates that the influence of spurious currents on the droplet evaporation is negligible. The further numerical validations shown in Figure 4.15a also confirm that the symmetrical spurious velocities never deform the interface in an un-physical way such that perfect interface shapes are predicted with the i-CLSVoF framework.

Three different mesh sizes are considered to study the effect of mesh resolution on the numerical results. As shown in Figure 4.13a and 4.13b, the time evolution

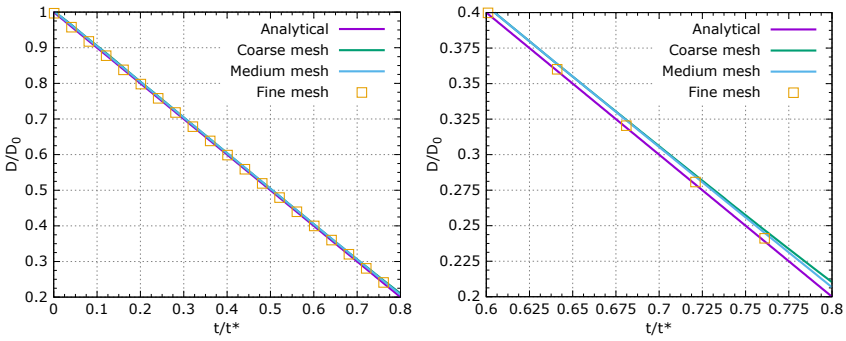


Figure 4.13: The mesh convergence study: (a) the global plot, (b) the local magnified plot.

of dimensionless droplet diameter with the dimensionless time until 80% of the total evaporation time is presented. As presented in the literature [48, 52], 80% of the total evaporation time is enough to validate the performance of numerical models. The mesh resolution for a static mesh is not fine enough to resolve simulations for the remaining 20% of the total evaporation time. The agreement between the numerical solution and the corresponding analytical solution is getting better with finer mesh. For the fine mesh, the analytical curve perfectly goes through all the numerical data points as shown in the locally magnified plot in Figure 4.13b.

In order to validate the evaporation model quantitatively, the relative error of the predictions for the shrinking droplet diameter calculated with different mesh

resolutions are compared in Figure 4.14. The relative error is lower than 1% when

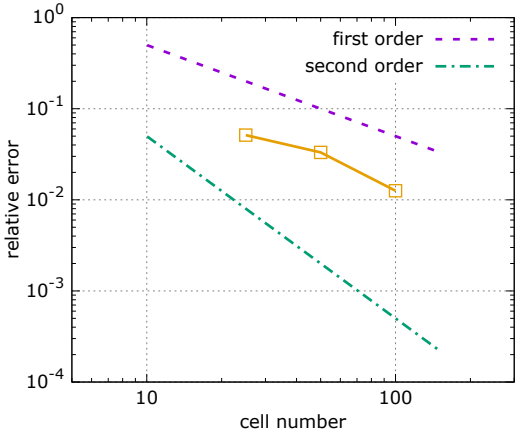
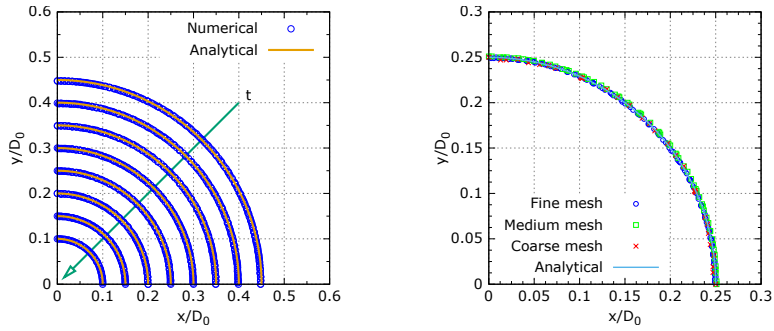


Figure 4.14: The numerical error for different mesh resolutions.

the number of cells in one direction for 2D cases is larger than 100, and first-order convergence is obtained with the evaporation model developed in this work. Additionally, the numerical error for predicting the shrinking droplet diameter is still acceptable (around 5%) even for a coarse mesh with our improved numerical model.

Interface capturing is another crucial aspect in modelling droplet evaporation, and the i-CLSVoF approach developed in this work can capture the free surface in a sharp manner. The corresponding validation is also conducted. In Figure 4.15a and 4.15b, the solid yellow curves are the analytical solution at different evaporation stages, while the blue dots are the numerical data collected on the 0.5 iso-surface for the eight stages. As shown in Figure 4.15a, the agreement between the numerical data and the corresponding analytical solution is perfect. It is also promising to see that the circular droplet shape is maintained, meaning that the influence of spurious currents on destroying the droplet shape is suppressed



(a) **(b)**

Figure 4.15: Numerical validations of the interface capturing: (a) the global plot, (b) the mesh convergence study.

throughout the whole evaporation process. For a given stage, the mesh convergence study indicates no major difference when refining the mesh as shown in Figure 4.15b. The reason is that the numerical data are only collected at the 0.5 iso-surface, which does not depend on mesh resolution.

4.4.2 Droplet evaporation with the thermally driven evaporation model

After validating the droplet evaporation with constant mass flux, the i-CLSVoF framework is extended to incorporate evaporation with more complex evaporation mass flux calculations. The mass flux of the thermally driven evaporation model depends on the interfacial temperature difference, which drives the phase change from liquid to vapour. The analytical solution for the shrinking droplet diameter square D^2 during the evaporation process was derived with the interface energy balance in the literature [157]. The analytical solution is given as the ordinary partial differential equation

$$\frac{dD^2}{dt} = -\frac{8k_g}{\rho_l c_{p_g}} \ln(1+B_q), \tag{4.37}$$

where B_q is the Spalding mass transfer number defined by

$$B_q = \frac{c_{p_g}(T^\infty - T_{\text{sat}})}{h_{\text{ev}}}, \tag{4.38}$$

and T^∞ is the temperature on the boundary. This is known as the D^2 law, and it is derived for the droplet evaporation in an infinite domain. The modified D^2 law takes into account the effect of computational domain size on the evolution of droplet diameter, and this model is more suitable for validating droplet evaporation in a finite computational domain. The modified analytical solution is

$$\frac{dD^2}{dt} = -\frac{8k_g}{\rho_l c_{p_g}} \frac{\ln(1+B_q)}{\ln(D_s/\sqrt{D^2})}, \tag{4.39}$$

where D_s is the diameter of the inscribed circle for the computational domain [52]. However, the unknown D^2 appears in the denominator due to the correction, and consequently Eqn. 4.39 can be solved numerically (refer to Appendix Code list A.3).

The numerical set-up for the 2D simulation of the thermally driven evaporation model is similar to the set-up as shown in Figure 4.10. The only difference is that in the initial configuration the temperature for the droplet is equal to its saturation temperature T_{sat} while the temperature for the rest of the domain is higher than the saturation temperature. Additionally, a Dirichlet boundary condition for the temperature field on the boundaries except the symmetry boundaries must be applied. The numerical validation in this part starts with the temperature difference of 50 K, and the corresponding temperature distribution around the evaporating droplet is shown in Figure 4.16a. The temperature gradient around the droplet drives the phase change from liquid to vapour, and the temperature inside the droplet stays constant and equals the saturation temperature. As shown in Figure 4.16b, the interface velocity $\mathbf{U}_\Gamma$ points towards the droplet center, which is related to the evaporation-induced droplet shrinking. Additionally, the interface velocity $\mathbf{U}_\Gamma$ is dominant only at the droplet interface, which demonstrates that the

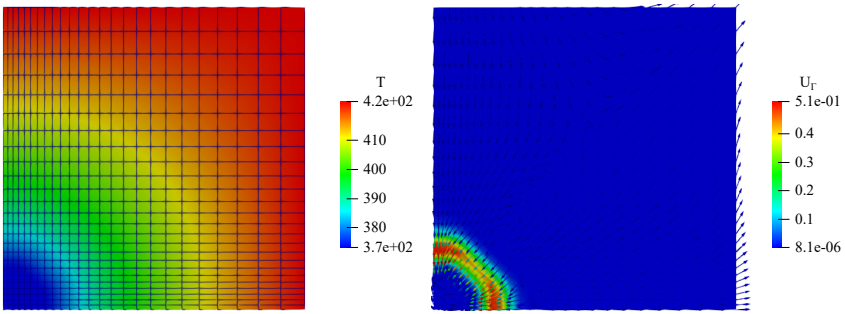


Figure 4.16: (a) The temperature distribution around the droplet, (b) the interface velocity field $\mathbf{U}_\Gamma$.

divergence-free velocity field is successfully reconstructed also for the thermally driven evaporation model.

The quantitative study of the shrinking droplet diameter is compared against the corresponding analytical model according to Eqn. 4.39. The dimensionless droplet diameter and the dimensionless time are adopted. It can be seen from Figure 4.17a, that an accurate solution is achieved for around 65% of the total evaporation time $t^\star$ with the graded mesh shown in Figure 4.16a. The corresponding

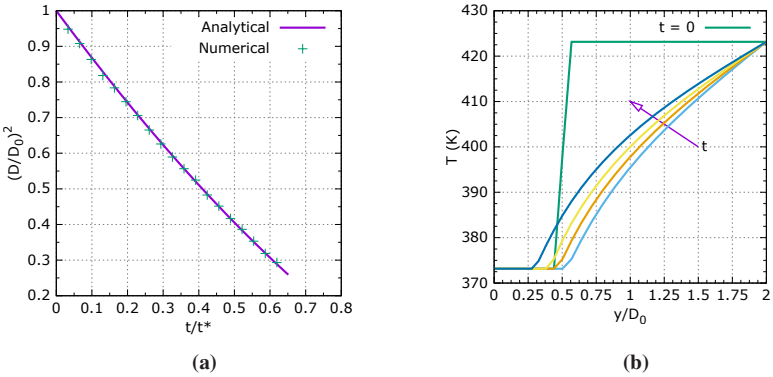


Figure 4.17: (a) The numerical validation for thermally driven evaporation model, (b) the temperature evolutions during the evaporation process.

temperature evolution collected from the bottom-left corner to the bottom-right corner can be seen in Figure 4.17b. The solid green line represents the initial configuration for the temperature field, where the transition band from the saturation temperature to the temperature value corresponding to the boundaries in the initial temperature field is relatively large. Further mesh refinement can shorten this transition band. The other curves in Figure 4.17b show the evolution of temperature (the purple arrow indicates time going on) during the evaporation process. The saturation temperature inside the droplet and the temperature at the domain boundary are strictly maintained constant during the evaporation process.

Additionally, droplet evaporation with different Stefan numbers is further studied to validate our model extensively. The Stefan number is defined as

$$\mathrm{St} = \frac{c_{p_g} \Delta T}{h_{\mathrm{ev}}}, \tag{4.40}$$

where $\Delta T = T_\infty - T_{\mathrm{sat}}$ is the temperature difference between the saturation temperature T_{sat} inside the droplet and the temperature T_∞ at the boundaries, and c_{p_g} the specific heat capacity of the gas/vapour phase. As can be seen from Figure 4.18, four evaporation cases with different Stefan numbers are presented. Cases with a large Stefan number experience faster evaporation, and a good agreement between the numerical and the corresponding solution of the analytical model is observed for all cases.

4.4.3 Droplet evaporation with vapour mass fraction gradient evaporation model

The numerical set-up for modelling droplet evaporation at room temperature is the same as for the two aforementioned evaporation models (refer to Figure 4.10). In addition to prescribing the outflow boundary conditions, a Dirichlet boundary condition for the vapour mass fraction field is applied on the boundaries except for the symmetry boundaries.

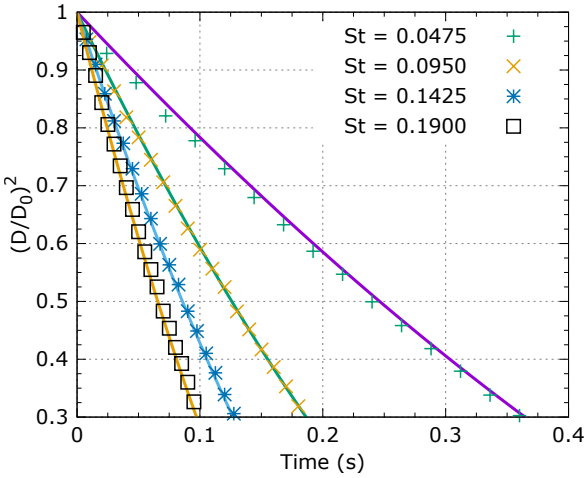


Figure 4.18: Validations for four cases with different Stefan number *St* (solid lines represent the analytical solutions, and points represent the corresponding numerical data).

Essential parameters used in the simulations are listed in Table 4.1. The one-field velocity field during the evaporation process is shown in Figure 4.19a where a velocity jump can be seen around the interface region where the white line represents the interface. The corresponding vapour mass fraction field is shown in Figure 4.19b with the vapour mass faction at the top and right boundaries given as constants prescribed by the Dirichlet boundaries conditions. The vapour mass fraction gradient at the interface drives the droplet evaporation.

The shrinking droplet diameter D is recorded to quantitatively validate the evaporation model by comparing the diameter predicted by the numerical simulations to the analytical solution given by the so-called D^2 law [157]. Again, for the droplet evaporation in a finite domain, the classical D^2 law should be corrected to take the computational domain size into account, and such a modified analytical model is adopted in this work [52]. This modified D^2 law is given by

$$\frac{dD^2}{dt} = -\frac{8\rho_g D_v}{\rho_l} \frac{\ln(1+B_y)}{\ln(D_s/\sqrt{D^2})}, \tag{4.41}$$

where D_s is the diameter of the inscribed circle for the computational domain, and B_y the Spalding mass transfer number defined as

$$B_y = \frac{Y^\Gamma - Y^\infty}{1 - Y^\Gamma} \tag{4.42}$$

with Y^∞ being the vapour mass fraction far way from the droplet [157].

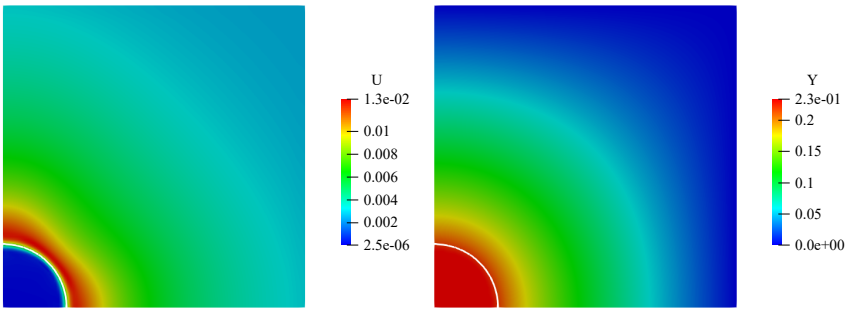


Figure 4.19: (a) The one-field velocity field **U** of droplet evaporation at room temperature, (b) The vapour mass fraction field of droplet evaporation at room temperature (white line represents the interface).

A parameter study on a wide range of evaporation temperatures was conducted to validate the evaporation model. In Figure 4.20, the solid lines represent corresponding analytical solutions given by Eqn. 4.41 while the points represent numerical solutions predicted by the evaporation model. Good agreement between numerical and analytical solutions are found for evaporation at relatively higher temperatures (303.15 K, 313.15 K and 323.15 K). Some minor discrepancy between the numerical and analytical solutions are found, especially for evaporation at relatively low temperatures (283.15 K and 293.15 K). The underlying reason is that evaporation at lower temperatures is more susceptible to the influence of un-physical spurious velocity.

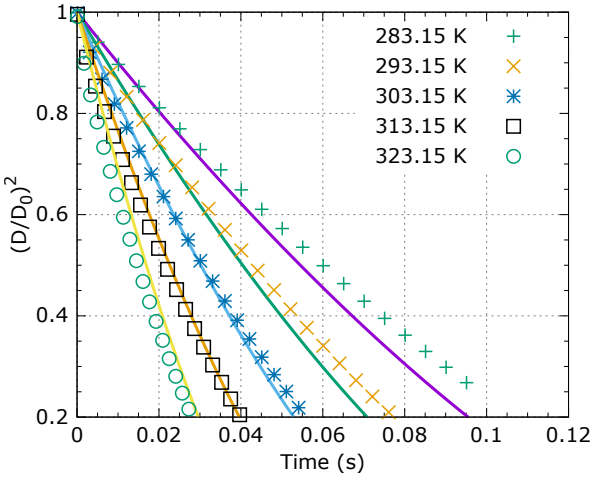


Figure 4.20: Validations for five cases with different evaporation temperatures (solid lines represent the analytical solutions, and points represent the corresponding numerical data).

4.4.4 Droplet evaporation with contact line pinning

A 2D numerical demonstration is presented in this section to demonstrate the performance of the Robin boundary condition implemented in this thesis to pin the contact line. A 2D droplet with initial droplet radius of 50 µm resting on a substrate with a length and a height of the computational domain of 300 µm and 150 µm, respectively is shown in Figure 4.21. A outflow boundary condition is applied on the boundaries except for the bottom wall on which the no-slip boundary condition is applied. Additionally, the Robin boundary condition is also applied at the bottom wall. The initial contact angle for the 2D droplet is 60°. Adaptive mesh refinement is applied to refine the mesh around the free surface to save computational cost as well as to capture free surface accurately when the droplet height gets thin at the late stage of its evaporation.

The numerical results are shown in Figure 4.21, where the liquid volume fraction field of four different stages is presented. The phase in red indicates the droplet,

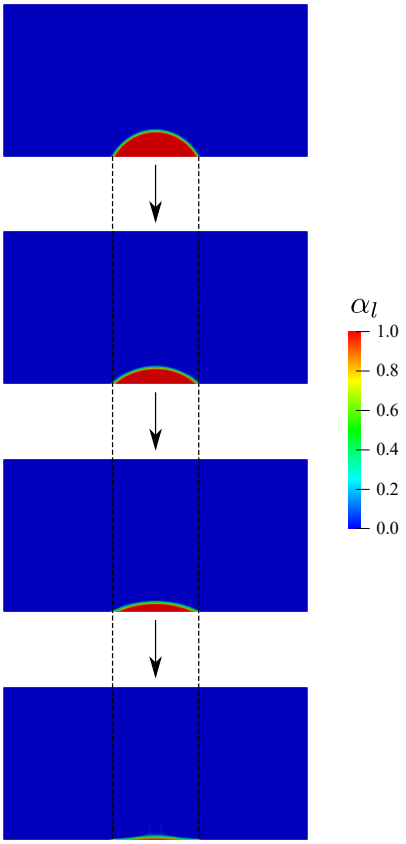


Figure 4.21: Four different stages of the liquid volume fraction fields during the evaporation process.

while the other phase in blue is the gas phase. Two vertical lines are drawn to demonstrate the most initial contact points between the droplet and the bottom wall. As can be seen from the image, the initial contact radius is firmly pinned during the evaporation process. Furthermore, the spherical cap of the droplet is maintained during its evaporating process and the free surface is accurately captured even in the final stage of the evaporation without interface diffusion. It

proves that the Robin boundary condition incorporated in the i-CLSVoF framework performs good in modelling droplet evaporation with contact line pinning, and also captures free surface in a sharp way.

Furthermore, the velocity fields for droplet evaporation with contact line pinning are shown in Figure 4.22. The velocity vector in the gas phase points out of

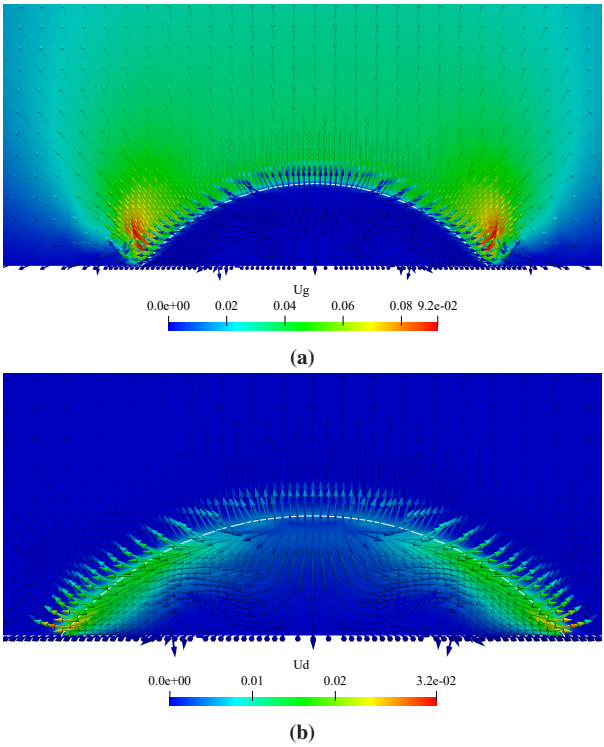


Figure 4.22: Velocity fields of droplet evaporation with contact line pinning (white line represents the interface): (a) velocity in the gas phase, (b) velocity inside the droplet.

the droplet, which indicates the velocity field of the vapour generated from the droplet surface where the phase change occurs. The magnitude of the velocity field around the triple contact line region is larger than that of elsewhere. This

indicates that the evaporation flux is larger around the triple contact line region. The same conclusion can also be found in the literature for droplet evaporation with contact line pinning [7, 8]. The internal velocity field inside evaporating suspension droplet is of great importance in affecting the motion of particles inside. From the numerical point of view, the velocity field inside the evaporating droplet is demonstrated in Figure 4.22b. An internal capillary flow from droplet center to its edge is found. The reason behind the lateral capillary flow is due to the interplay between faster evaporation around the triple contact line region and the surface tension. In an evaporating droplet with contact line pinning, the mass loss around the triple contact line region must be replenished by the lateral capillary flow induced by the surface tension acting on the droplet to maintain its shape of a spherical cap during its evaporation process. The velocity fields inside both liquid and gas phases of the numerical simulations further prove the correct implementation of the Robin boundary condition to pin the contact line for evaporating droplets.

4.5 Summary

The i-CLSVoF framework was further extended to model droplet evaporation, especially for the micro-scale phenomena. A simple approach was proposed and implemented to reconstruct a divergence-free velocity by removing the evaporation-induced irrotational velocity from the one-field velocity field to predict the evaporation rate accurately. The constant mass flux evaporation model was first incorporated into the i-CLSVoF framework to check the implementations of the basic equations. The numerical validations proved the successful construction of the divergence-free velocity field. Besides, the model accurately predicted the droplet evaporation as shown by comparing the shrinking dimensionless droplet diameter to an analytical solution derived in this work. The interface capturing of the i-CLSVoF was also demonstrated being accurate enough by comparing the numerically captured free surface to the analytical solution, where no spurious velocity induced interface deformation was found during the

evaporation process. The thermally driven evaporation model was implemented to account for the phase change induced by a temperature gradient. A novel density-ratio dependent evaporation coefficient was used to calculate the mass flux at the interface. The modified D^2 law was used to validate the numerical model, and a good agreement between the numerical solution and the analytical solution was achieved. Additionally, parameter studies were conducted to demonstrate the accuracy of the thermally driven evaporation model for a wide range of Stefan numbers. The third evaporation model incorporated in this chapter describes the sessile droplet evaporation at room temperature. This model can predict the evaporation rate accurately for different evaporation temperatures while some minor discrepancy between numerical and analytical solutions is found for relatively small evaporation temperature, which is considered acceptable.

A Robin boundary condition based on the contact angle hysteresis is implemented in i-CLSVoF framework to pin the mobile contact line when a droplet evaporates. A 2D droplet evaporation case is presented in this chapter. The contact line is successfully pinned for an evaporating droplet with initial contact angle of 60°. Large evaporation flux around the triple contact line region and a lateral capillary flow are found with this newly implemented boundary condition. The numerical model developed in this chapter is applicable for modelling evaporation not only with but also without contact line pinning.

As the next step, the i-CLSVoF framework is coupled to the DEM approach to understand transport of particles inside evaporating suspension droplets. This is detailed in the coming chapters, namely, Chapters 5 and 6.

5 The improved resolved CFD-DEM framework for multiphase system with and without evaporation

Granular materials widely exist in our daily life (e.g. sugar, salt and coffee bins) and industry (e.g. coal, sand and bearing balls). However, some granular materials are surrounded by fluids such as gases or liquids (e.g. pneumatic conveying [158], fluidized beds [159], mudflow [160], capillary suspensions [161] and more [38, 162, 163]). Computational modelling of these kinds of gas-liquid-solid multiphase systems is of great importance for better understanding the complex interactions among solid particles and between the solid phase (particles) and the fluid phase (liquids or gases).

The DEM is capable of modelling the complex mechanical behaviour of solid particles and the interactions between a particle and a wall. An extended DEM approach, detailed in Chapter 2, has been developed in this thesis. The motion of solid particles in different scales, namely macro-, meso- and micro-scale, can be tracked by solving Newton's second law of motion. CFD is generally used to model the motion of fluids by solving the Navier-Stokes equations. The so-called coupling approach, namely coupling CFD to DEM (CFD-DEM), is widely used to model the complex interaction between solid and fluid phases, as will be discussed now.

[1] Part of this chapter with minor changes has been submitted to arXiv as a preprint [177].

In this chapter, the resolved CFD-DEM approach is discussed. In the resolved CFD-DEM approach, the motions of solid and fluid phases are governed by DEM and CFD, respectively [46]. Modelling enables us to understand multiphase systems at different scales, and extensive parameter studies can be conducted to investigate the influence of operating parameters at a lower cost. During the past decades, the CFD-DEM approach has been widely used to computationally model either compressible or incompressible particle-laden flow [164–166]. Hager et al. developed a simple method for smooth representation of the void fraction field for multi-scale resolved CFD-DEM simulations [164, 167]. Podlozhnyuk implemented the superquadric particles into the resolved CFD-DEM [165]. Davydzenka et al. developed a resolved CFD-DEM model accounting for the wettability of complex geometry in multiphase flow [168]. Blood flow with irregular red blood cell particles was investigated within the resolved CFD-DEM framework by Balachandran et al. [46]. Free surface flow with capillary interactions was studied by Nguyen et al., where a capillary force model was developed [169]. Flow with irregular particles constructed with multi-sphere clumps in an incompressible free surface flow was investigated by Shen et al. [170]. Melting of solid particles for selective laser melting was computationally modelled by an improved resolved CFD-DEM approach developed by Yu et al. [45]. Schnorr Filho et al. investigated the hydraulic conveying of solid particles through a narrow elbow with a resolved CFD-DEM model [39]. Free surface flow with superquadric particles was investigated by Washino et al., with the capillary force incorporated into the resolved CFD-DEM model [171].

A summary of numerical simulations and applications of existing resolved CFD-DEM approaches is listed in Table 5.1. However, what can be seen from the table is that a limited number of publications can be found related to model solid particles immersed in an incompressible flow that undergoes phase change or evaporation. Direct inkjet printing and spray coating involve complex solid-liquid interactions, the phase change from liquid to vapour, surface tension and beyond [26, 173]. Accordingly, developing such a numerical model accounting for additional phase change and surface tension is of great significance.

Table 5.1: Summary and comparison among these applications of resolved CFD-DEM formulations.

Authors (publication year)	Applications	Surface tension	Phase change
Hager et al. (2014) [164, 167]	Multi-scale modelling	No	No
Podlozhnyuk et al. (2017) [172]	Flow with superquadric particles	No	No
Davydzenka et al. (2020) [168]	Multiphase flow in porous media	Yes	No
Balachandran Nair et al. (2021) [46]	Blood flow in microfluidic devices	No	No
Nguyen et al. (2021) [169]	Flow with capillary interactions	Yes	No
Shen et al. (2022) [170]	Flow with irregular particles	Yes	No
Yu et al. (2021) [45]	Selective laser melting	Yes	Yes
Schnorr Filho et al. (2022) [39]	Hydraulic conveying	No	No
Washino et al. (2023) [171]	Flow with non-spherical particles	Yes	No

In this chapter, an improved resolved CFD-DEM framework incorporating free surface capturing, surface tension and phase change is developed by extending the resolved CFD-DEM model developed by Hager et al. [164, 167]. The new resolved CFD-DEM framework is implemented in the open-source framework **CFDEMcoupling-PUBLIC** [174] bridging the open-source DEM code **LIGGGHTS** [92] and the open-source Finite Volume Method (FVM) based C++ library **OpenFOAM** [175]. The large-scale parallel computation and data exchange between the two codes are realized using the Message Passing Interface (MPI) software [178].

5.1 Mathematical formulation

In this section, the theory and mathematical formulation of the resolved CFD-DEM approach are introduced in detail. For the resolved CFD-DEM approach, one solid particle usually occupies several CFD cells (see the inset on the top-left of Figure 5.1). The void fraction field ε_f is used to quantify how much volume is not occupied by a solid particle in each CFD cell, because of which $\varepsilon_f = 0$ means that a solid particle fully covers the current CFD cell. The free surface is captured by the VoF method incorporated in the i-CLSVoF framework [176]. The Fictitious Domain Method developed by Patankar et al. [179] was extended to

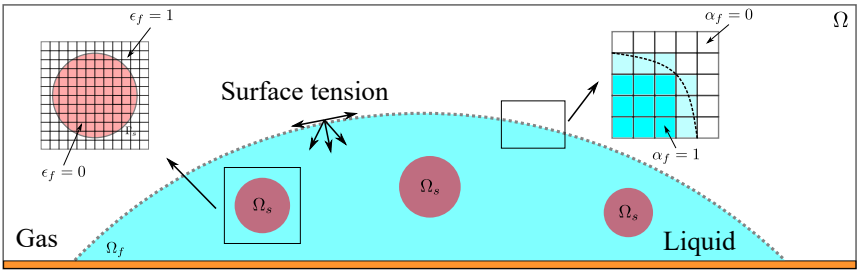


Figure 5.1: The schematic diagram of a surface-tension-dominant gas-liquid-solid multiphase system with free surface.

incorporate the free-surface capturing, surface tension and evaporation for modelling the complex multiphase system as demonstrated by Figure 5.1.

As shown in Figure 5.2, Ω is the whole computational domain, and Ω_f and Ω_s are the liquid and solid phases, respectively. Γ and Γ_s denote the boundaries of the whole computational domain and the solid particles immersed in the liquid, respectively. The governing equations for the surface-tension-dominant incom-

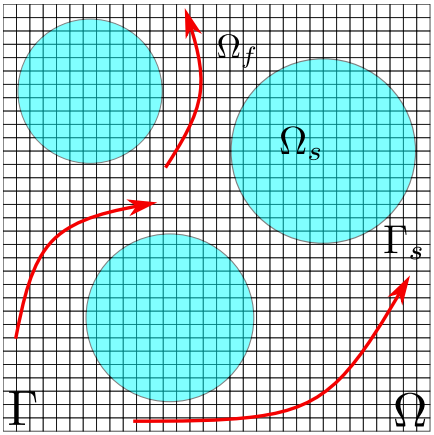


Figure 5.2: The schematic diagram of the resolved CFD-DEM method. Arrows represent the vectors of fluid velocity around particles, and circles in blue represent solid particles.

pressible Newtonian flow are given by

$$\nabla \cdot \mathbf{U} = \varepsilon_f \dot{m}(\frac{1}{\rho_g} - \frac{1}{\rho_l}) \text{ in } \Omega, \tag{5.1}$$

$$\frac{\partial(\rho \mathbf{U})}{\partial t} + \nabla \cdot (\rho \mathbf{U}\mathbf{U}) = -\nabla p + \nabla \cdot \left[\mu \left(\nabla \mathbf{U} + (\nabla \mathbf{U})^T\right)\right] + \rho \mathbf{g} + \mathbf{F}_{\text{st}} \text{ in } \Omega, \tag{5.2}$$

where ε_f is the void fraction field. The incorporation of this term is to guarantee that phase change does not occur inside solid particles. $\dot{m}$ in the continuity equation (Eqn. 5.1) is the mass source term per unit volume, where $\dot{m} = 0$ holds for the case without evaporation or phase change only. Furthermore, some other terms, e.g. pressure gradient ∇p and surface tension force $\mathbf{F}_{\text{st}}$ in the momentum equation (Eqn. 5.2) are detailed in Chapter 3.

The governing equations and numerical method for the computational modelling of droplet evaporation are detailed in Chapter 4. This section only mentions some basic equations for modelling evaporation. The vapour mass fraction gradient model introduced in Sect. 4.2.3 is used in this chapter. The vapour mass fraction Y is solved from

$$\frac{\partial Y}{\partial t} + \nabla \cdot (Y \mathbf{U}) = D_v \nabla^2 Y, \tag{5.3}$$

where D_v is the vapour diffusion coefficient. The mass source term $\dot{m}$ in the continuity equation (Eqn. 5.1) is calculated by

$$\dot{m} = \frac{\rho_g D_v \nabla^\Gamma Y \mathbf{n}^\Gamma}{1 - Y^\Gamma} |\nabla \alpha_l|, \tag{5.4}$$

where α_l is the liquid volume fraction field solved from the VoF transport equation (Eqn. 4.2). Some other quantities (e.g. $\mathbf{n}^\Gamma$ and Y^Γ) and the numerical method for the evaporation model are detailed in the previous chapters, e.g. Chapter 3 and Chapter 4.

In the Fictitious Domain Method, additional boundary and interface conditions are needed to be applied due to the presence of solid particles in the liquid phase [180]. These additional conditions are given by

$$\begin{cases} \mathbf{U} = \mathbf{U}_\Gamma \text{ on } \Gamma, \\ \mathbf{U} = \mathbf{U}_i \text{ on } \Omega_s, \\ \boldsymbol{\Sigma} \cdot \mathbf{n} = \mathbf{t}_{\Gamma_s} \text{ on } \Gamma_s, \\ \mathbf{U}(\mathbf{x},\, t = 0) = \mathbf{U}_0(\mathbf{x}) \text{ in } \Omega_f. \end{cases} \tag{5.5}$$

Here, the subscript i indicates the i^{th} particle and $\boldsymbol{\Sigma}$ is the total stress tensor as discussed below, and $\mathbf{n}$ is the outward normal vector to Γ_s. $\mathbf{t}_{\Gamma_s}$ is the traction vector acting from the liquid phase on the surface of solid particles. The second and third equations of Eqn. 5.5 are responsible for the coupling between liquid and solid phases. Additionally, the second equation ensures the transfer of the particle velocity $\mathbf{U}_i$ to the liquid velocity of CFD cells covered by the solid particle i. The third equation represents the force acting on the boundaries of the solid phase.

Newton's second law of motion governs the motion of solid particles. The equation for the translational motion is given by

$$m_i \frac{\partial^2 \mathbf{x}_i}{\partial t^2} = \sum_{i=1}^{N_p} \mathbf{F}_{ij} + m_i \mathbf{g} + \mathbf{F}_{\text{fp}}^{\text{c}} + \mathbf{F}_{\text{cp}}, \tag{5.6}$$

where m_i and $\mathbf{x}_i$ are the mass and position vector of the i^{th} particle, respectively, $\mathbf{F}_{ij}$ is the contact force between two DEM elements (particle-particle or particle-wall) and $\mathbf{F}_{\text{fp}}^{\text{c}}$ the CFD-DEM coupling force acting on the solid particles by the liquid phase. The last term on the right-hand side of Eqn. 5.6 is the capillary force acting on solid particles, which is discussed in the next section in detail. Besides, the rotational motion of solid particles is governed by

$$I_i \frac{\partial^2 \theta_i}{\partial t^2} = \sum_{i=1}^{N_p} \mathbf{M}_{ij} + \mathbf{M}_{\text{fp}}^{\text{c}} + \mathbf{M}_{\text{cp}}, \tag{5.7}$$

where I_i and θ_i are the moment of inertia and angular displacement of the particle i, respectively. $\mathbf{M}_{ij}$ is the torque acting on particle i by some other particles interacting with it. $\mathbf{M}_{\mathrm{fp}}^{\mathrm{c}}$ is the coupling term accounting for the torque acting on the solid phase by the liquid phase. The last term $\mathbf{M}_{\mathrm{cp}}$ on the right-hand side of Eqn. 5.7 is the torque acting on the particle i due to capillary interactions. Calculations of these terms are introduced in the forthcoming sections. Some other contact models of the extended DEM approach are detailed in Chapter 2.

5.2 Numerical method

Calculations of interaction forces acting on the solid phase by the liquid phase are of great importance for realizing the resolved CFD-DEM.

5.2.1 Calculations of the interaction forces and torque

The force acting on the solid phase by the liquid phase can be calculated by integrating the third equation of Eqn. 5.5 over the whole solid surface as

$$\mathbf{F}_{\mathrm{fp}}^{\mathrm{c}} = \int_{\Gamma_s} \mathbf{t}_{\Gamma_s}\, dS. \tag{5.8}$$

The surface integral can be transformed to a volume integral using the divergence theorem which leads to

$$\int_{\Gamma_s} \mathbf{t}_{\Gamma_s}\, dS = \int_{\Gamma_s} \boldsymbol{\Sigma}\cdot\mathbf{n}\, dS = \int_{\Omega_s} \nabla\cdot\boldsymbol{\Sigma}\, dV. \tag{5.9}$$

The total stress tensor $\boldsymbol{\Sigma}$ in Eqn. 5.9 consists of two terms given by

$$\boldsymbol{\Sigma} = -p\mathbf{I} + \boldsymbol{\tau}, \tag{5.10}$$

where p is the pressure and $\mathbf{I}$ the identity tensor of size 3×3, while τ is the viscous stress tensor. For incompressible Newtonian fluids, τ is deviatoric and given by

$$\tau = \mu \left(\nabla \mathbf{U} + (\nabla \mathbf{U})^{\mathrm{T}} \right). \tag{5.11}$$

Substituting Eqs. 5.9, 5.10 and 5.11 into Eqn. 5.8, the interaction force acting on the solid phase can be calculated by

$$\begin{aligned} \mathbf{F}_{\mathrm{fp}}^{\mathrm{c}} &= \int_{\Omega_s} \nabla \cdot \mathbf{\Sigma} \, dV \\ &= \int_{\Omega_s} \nabla \cdot \left[-p\mathbf{I} + \mu \left(\nabla \mathbf{U} + (\nabla \mathbf{U})^{\mathrm{T}} \right) \right] dV \\ &= \int_{\Omega_s} \underbrace{\nabla \cdot (-p\mathbf{I})}_{=-\nabla p} + \nabla \cdot \left[\mu \left(\nabla \mathbf{U} + (\nabla \mathbf{U})^{\mathrm{T}} \right) \right] dV. \end{aligned} \tag{5.12}$$

As derived in the literature [120], for incompressible Newtonian fluids, the divergence of the viscous stress tensor is given by

$$\nabla \cdot \tau = \nabla \cdot \left[\mu \left(\nabla \mathbf{U} + (\nabla \mathbf{U})^{\mathrm{T}} \right) \right] = \nabla \cdot (\mu \nabla \mathbf{U}) + \nabla \mathbf{U} \cdot \nabla \mu. \tag{5.13}$$

In this chapter, one assumption is that the dynamic viscosity μ given by Eqn. 3.2 is constant, and thus Eqn. 5.13 leads to

$$\nabla \cdot \tau = \nabla \cdot \left[\mu \left(\nabla \mathbf{U} + (\nabla \mathbf{U})^{\mathrm{T}} \right) \right] = \mu \nabla^2 \mathbf{U}. \tag{5.14}$$

Substituting Eqn. 5.14 into Eqn. 5.12, leads to

$$\mathbf{F}_{\mathrm{fp}}^{\mathrm{c}} = \int_{\Omega_s} \nabla \cdot \left[-p\mathbf{I} + \mu \left(\nabla \mathbf{U} + (\nabla \mathbf{U})^{\mathrm{T}} \right) \right] dV = \int_{\Omega_s} \left(-\nabla p + \mu \nabla^2 \mathbf{U} \right) dV. \tag{5.15}$$

The volume integral in Eqn. 5.15 can be approximated by summarizing all the cells either partially or fully covered by a solid particle as

$$\mathbf{F}_{\mathrm{fp}}^{\mathrm{c}} \approx \sum_{i=1}^{N_c} (-\nabla p + \mu \nabla^2 \mathbf{U}) V_c, \tag{5.16}$$

where N_c is the total number of cells, and V_c is the cell volume.

Similarly, the torque $\mathbf{M}_{\mathrm{fp}}^{\mathrm{c}}$ acting on particles by the liquid phase can be calculated by

$$\mathbf{M}_{\mathrm{fp}}^{\mathrm{c}} = \int_{\Gamma_s} \mathbf{r} \times \mathbf{t}_{\Gamma_s} \, dS = \int_{\Omega_s} \mathbf{r} \times (-\nabla p + \nabla \cdot \tau) \, dV, \tag{5.17}$$

where $\mathbf{r}$ is the position vector. $\mathbf{M}_{\mathrm{fp}}^{\mathrm{c}}$ is approximated by summarizing all these quantities as

$$\mathbf{M}_{\mathrm{fp}}^{\mathrm{c}} \approx \sum_{i=1}^{N_c} \mathbf{r} \times \left(-\nabla p + \mu \nabla^2 \mathbf{U}\right) V_c. \tag{5.18}$$

The last terms on the right-hand side of Eqs. 5.6 and 5.7 are due to the capillary interactions among particles protruding from the free surface of liquids. The capillary force is of great importance for surface-tension-dominant flow because capillary interactions govern the motions, e.g. leading to self-assembly or self-organization of particles which appear at a free surface [182]. The capillary force is a long-range attractive force which is more dominant than other forces, e.g. inertial force and gravitational force for surface-tension-dominant cases. Thus, this force is accounted for in this chapter.

As proven by Fujita et al. in the literature [182], the sum of the surface tension force along the three-phase contact line ∂s is equal to the sum of the surface tension force over the virtual free surface fully immersed inside the solid particle in three dimensions as shown in Figure 5.3.

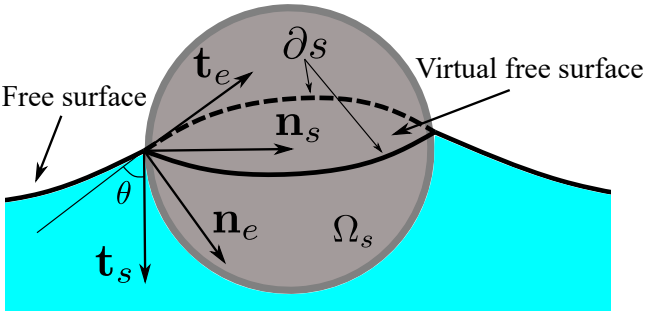


Figure 5.3: The schematic diagram of the capillary force model. The virtual free surface is bounded by the solid and dashed three-phase contact lines ∂s.

The Immersed Free Surface model proposed in the literature [169, 171, 183] is extended to account for the wettability of solid particles at a free surface in this chapter. The basic idea is to solve the extrapolation equation

$$\frac{\partial \alpha_l}{\partial t} + \mathbf{t}_e \cdot \nabla \alpha_l = 0 \tag{5.19}$$

in the true particle domain, namely, a domain with $\varepsilon_f < 0.5$ to extrapolate the free surface from the liquid phase inside solid particles [169]. In Eqn. 5.19, $\mathbf{t}_e$ is the tangent vector to the liquid surface pointing towards the particle. It is used to extend the liquid volume fraction and is defined by

$$\mathbf{t}_e = \frac{\mathbf{n}_s - (\mathbf{n}_e \cdot \mathbf{n}_s)\mathbf{n}_e}{|\mathbf{n}_s - (\mathbf{n}_e \cdot \mathbf{n}_s)\mathbf{n}_e|} \tag{5.20}$$

with $\mathbf{n}_e$ being the normal vector to the liquid pointing inside the liquid, which is given by

$$\mathbf{n}_e = \mathbf{n}_s \cos\theta + \mathbf{t}_s \sin\theta. \tag{5.21}$$

Here θ is the contact angle between the surface of the solid particle and the free surface. The normal vector to the particle surface pointing inwards $\mathbf{n}_s$ and the unit vector $\mathbf{t}_s$ perpendicular to $\mathbf{n}_s$ are defined by

$$\mathbf{n}_s = \frac{\nabla \varepsilon_s}{|\nabla \varepsilon_s|} \tag{5.22}$$

and

$$\mathbf{t}_s = \frac{\nabla \alpha_l - (\mathbf{n}_s \cdot \nabla \alpha_l)\mathbf{n}_s}{|\nabla \alpha_l - (\mathbf{n}_s \cdot \nabla \alpha_l)\mathbf{n}_s|}, \tag{5.23}$$

respectively, where ε_s is the solid fraction defined by $1.0 - \varepsilon_f$.

In addition to the extrapolation equation given by Eqn. 5.19, another model proposed by Fujita et al. [183] is also implemented in this chapter. The first model given by Eqn. 5.19 is named as *Model A* and the other model is called *Model B*. The extrapolation equation of *Model B* is given by

$$\frac{\partial \alpha_l}{\partial t} + \varepsilon_s \mathbf{n}_s \cdot \nabla \alpha_l = \varepsilon_s |\nabla \alpha_l| \cos\theta. \tag{5.24}$$

Solving a diffusion equation and then an anti-diffusion equation to smoothen the liquid volume fraction field α_l and simultaneously suppressing the interface diffusion after solving Eqn. 5.19 or Eqn. 5.24 to guarantee more numerical stability was proposed by Nguyen et al. [169]. In this chapter, a simple approach without interface diffusion is applied, namely, the Laplacian filter approach proposed by Lafaurie et al. [184] is adopted to transform $\mathbf{t}_e \cdot \nabla \alpha_l$ in Eqn. 5.19 or $\mathbf{n}_s \cdot \nabla \alpha_l$ in Eqn. 5.24 into a smoother function $\widetilde{\mathbf{t}_e \cdot \nabla \alpha_l}$ or $\widetilde{\mathbf{n}_s \cdot \nabla \alpha_l}$. The transformations for $\mathbf{t}_e \cdot \nabla \alpha_l$ and $\mathbf{n}_s \cdot \nabla \alpha_l$ are given by

$$\widetilde{\mathbf{t}_e \cdot \nabla \alpha_l} = \frac{\sum_{f=1}^{n} (\mathbf{t}_e \cdot \nabla \alpha_l)_f S_f}{\sum_{f=1}^{n} S_f}, \tag{5.25}$$

$$\widetilde{\mathbf{n}_s \cdot \nabla \alpha_l} = \frac{\sum_{f=1}^{n} (\mathbf{n}_s \cdot \nabla \alpha_l)_f S_f}{\sum_{f=1}^{n} S_f}, \tag{5.26}$$

respectively, where S_f is the magnitude of the f^{th} face area of the computational cell which is bounded by n faces, and f denotes the face index. The value of $(\mathbf{t}_e \cdot \nabla\alpha_l)_f$ or $(\mathbf{n}_s \cdot \nabla\alpha_l)_f$ at the face center is calculated using linear interpolation over the interface region. The smooth function is then used to solve either Eqn. 5.19 or Eqn. 5.24 to construct a smooth virtual free surface. An artificial correction of the liquid volume fraction field α_l given by

$$\alpha_l = \max\big(0, \min(1, \alpha_l)\big) \tag{5.27}$$

after solving Eqn. 5.19 or Eqn. 5.24 is conducted to guarantee the boundedness of α_l. Artificial boundedness of the liquid volume fraction field inside the solid phase by Eqn. 5.27 leads to mass conservation issues, namely, the total liquid volume in the whole computational domain is not conserved. A numerical correction step is then adopted to correct the liquid volume fraction in the liquid phase. First, the total liquid volume of the pure liquid phase before solving Eqn. 5.19 or Eqn. 5.24 denoted as V_{before} is calculated by

$$V_{\text{before}} = \int_{\Omega} \varepsilon_f \alpha_l \, dV. \tag{5.28}$$

Second, the total liquid volume of the pure liquid phase after solving Eqn. 5.19 or Eqn. 5.24 is denoted V_{after} is calculated by

$$V_{\text{after}} = \int_{\Omega} \varepsilon_f \alpha_l \, dV. \tag{5.29}$$

The volume loss of the pure liquid phase due to the evaporation V_{evap} is calculated by

$$V_{\text{evap}} = \int_{\Omega} \frac{\dot{m}}{\rho_l} \Delta t \, dV, \tag{5.30}$$

where Δt is the time step, and ρ_l is the density of the liquid phase. Accordingly, the volume change ΔV is computed by

$$\Delta V = V_{\text{after}} - V_{\text{before}} - V_{\text{evap}}. \tag{5.31}$$

The liquid volume fraction field α_l in the liquid phase ($\varepsilon_f > 0.5$) can be corrected by

$$\alpha_l = \alpha_l + \frac{\varepsilon_f \Delta V |\nabla \alpha_l|}{S_s}. \tag{5.32}$$

Here, S_s is the total surface area of the pure liquid phase calculated by

$$S_s = \int_\Omega \varepsilon_f |\nabla \alpha_l| \, dV. \tag{5.33}$$

The numerical procedure to construct the virtual free surface inside solid particles using the Immersed Free Surface model is outlined as follows:

- Smoothening either $\mathbf{t}_e \cdot \nabla \alpha_l$ with Eqn. 5.25 or $\mathbf{n}_s \cdot \nabla \alpha_l$ with Eqn. 5.26 before solving Eqn. 5.19 for *Model A* and Eqn. 5.24 for *Model B*, respectively.
- Solving either Eqn. 5.19 or Eqn. 5.24 to construct a virtual free surface inside solid particles.
- Using Eqn. 5.27 to correct and constrain the liquid volume fraction field α_l, artificially.
- Correcting the liquid volume fraction in the liquid phase with Eqn. 5.32.

Once constructing the virtual free surface inside solid particles as shown in Figure 5.3, the capillary force can be calculated by

$$\mathbf{F}_{\text{cp}} = \oint_{\partial s} \sigma K \mathbf{n} \, ds, \tag{5.34}$$

where ds is the infinitesimal integration area, and σ and K are the surface tension coefficient and mean interface curvature, respectively. As derived in Appendix B of [185], the surface integral over the immersed free surface S in Eqn. 5.34 can be transformed into a volume integral enclosing the immersed free surface S shown in Figure 5.3 given by

$$\mathbf{F}_{\text{cp}} = \int_{\Omega_s} \sigma K \mathbf{n} \delta_s \, dV, \tag{5.35}$$

where δ_s is a Dirac function discussed in Chapter 3. The term $\sigma K \mathbf{n} \delta_s$ can be summarized by the surface tension force $\mathbf{F}_{\text{st}}$. As suggested in the literature [169], the capillary force can then be calculated over the true particle domain, namely domain with $\varepsilon_f < 0.5$ (see Figure 5.4) by

$$\mathbf{F}_{\text{cp}} = \int_{\Omega_s^{\varepsilon_f<0.5}} \mathbf{F}_{\text{st}} \, dV. \tag{5.36}$$

Similarly, the torque resulting from the capillary force can be calculated by

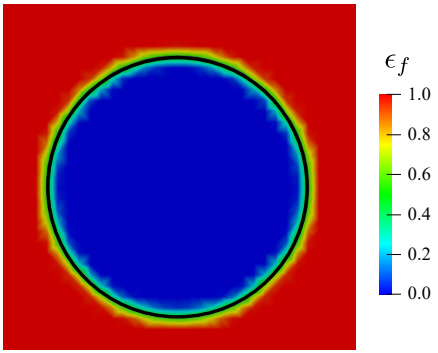


Figure 5.4: The void fraction field of a solid particle and the solid black curve represents the iso-surface with $\varepsilon_f = 0.5$.

$$\mathbf{M}_{\text{cp}} = \int_{\Omega_s^{\varepsilon_f<0.5}} \mathbf{r} \times \mathbf{F}_{\text{st}} \, dV, \tag{5.37}$$

once the surface tension force $\mathbf{F}_{\text{st}}$ is known.

However, some preliminary numerical simulations demonstrate that this approach suffers from un-physical spurious velocities inside solid particles, which is also discussed in the literature [171]. Therefore, the filtered surface tension

force model implemented and discussed in Chapter 3 is adopted here to calculate the capillary force and the torque as

$$\mathbf{F}_{\mathrm{cp}} = \int_{\Omega_s \varepsilon_f < 0.5} \mathbf{F}^f_{\mathrm{st},f}\, dV \tag{5.38}$$

and

$$\mathbf{M}_{\mathrm{cp}} = \int_{\Omega_s \varepsilon_f < 0.5} \mathbf{r} \times \mathbf{F}^f_{\mathrm{st},f}\, dV, \tag{5.39}$$

respectively, where $\mathbf{F}^f_{\mathrm{st},f}$ is the filtered surface tension force. Its numerical details can be found in Chapter 3.

Alternatively, the term $\mathbf{F}_{\mathrm{st}}$ in Eqs. 5.36 and 5.37 can be replaced by $\mathbf{F}_{\mathrm{ccf}}$ calculated by the Continuous Capillary Force (CCF) model [186] which is given by

$$\mathbf{F}_{\mathrm{ccf}} = \sigma \mathbf{t}_c (\nabla \alpha_l \cdot \mathbf{t}_s)(\nabla \varepsilon_s \cdot \mathbf{n}_s), \tag{5.40}$$

where $\mathbf{t}_c$ is given by

$$\mathbf{t}_c = -\frac{\mathbf{n}_s - (\mathbf{n}_c \cdot \mathbf{n}_s)\mathbf{n}_c}{|\mathbf{n}_s - (\mathbf{n}_c \cdot \mathbf{n}_s)\mathbf{n}_c|}. \tag{5.41}$$

In the resolved CFD-DEM approach, one solid particle covers several CFD cells, and thus the void fraction ε_f is of great importance in calculating $\mathbf{F}^c_{\mathrm{fp}}$, $\mathbf{M}^c_{\mathrm{fp}}$ and some other quantities, accurately. The smooth representation algorithm proposed by Hager [167] is used in this chapter to create a smooth transition of the void fraction around the particle surface. It is proven that this algorithm is more stable and guarantees reasonable numerical accuracy.

5.2.2 The numerical procedure

In literature, an additional force term is incorporated to the right-hand side of the momentum Eqn. 5.2 to account for the interaction force acting on the fluid phase by the solid phase, which is known as the direct forcing approach [46, 171, 187, 188]. Instead of this, the numerical correction approach proposed in the

literature as well [164, 165, 167] is used in the current chapter. The numerical procedure to solve these equations mentioned above within the resolved CFD-DEM framework and the numerical correction step to guarantee the divergence-free condition of the velocity field are detailed below:

- First, an intermediate velocity field $\hat{\mathbf{U}}$ is solved from the Navier-Stokes equations (Eqs. 5.1 and 5.2) over the whole computational domain. In this step, the presence of solid particles in the CFD domain is not considered.
- Second, the intermediate velocity field $\hat{\mathbf{U}}$ in the CFD cells covered by a solid particle is overwritten by imposing the particle velocity calculated from the DEM side, explicitly. This leads to a new velocity field $\tilde{\mathbf{U}}$.
- In general, the new velocity field $\tilde{\mathbf{U}}$ is not divergence-free. Thus, a numerical correction step is further needed to correct this velocity. A Poisson equation given by

$$\nabla^2 \phi_r = \nabla \cdot \tilde{\mathbf{U}} - \varepsilon_f \dot{m} \left(\frac{1}{\rho_g} - \frac{1}{\rho_l} \right) \tag{5.42}$$

  is solved to update the velocity potential field ϕ_r. Another new velocity after the numerical correction is defined as $\bar{\mathbf{U}}$ given by

$$\bar{\mathbf{U}} = \tilde{\mathbf{U}} - \nabla \phi_r. \tag{5.43}$$

  It can be proven that the new velocity field $\bar{\mathbf{U}}$ is divergence-free:

$$\begin{aligned} \nabla \cdot \bar{\mathbf{U}} = \nabla \cdot \left(\tilde{\mathbf{U}} - \nabla \phi_r \right) &= \nabla \cdot \tilde{\mathbf{U}} - \underbrace{\nabla \cdot \nabla \phi_r}_{=\nabla^2 \phi_r} \\ &= \nabla \cdot \tilde{\mathbf{U}} - \nabla \cdot \tilde{\mathbf{U}} + \varepsilon_f \dot{m} \left(\frac{1}{\rho_g} - \frac{1}{\rho_l} \right) \\ &= \varepsilon_f \dot{m} \left(\frac{1}{\rho_g} - \frac{1}{\rho_l} \right). \end{aligned} \tag{5.44}$$

Here the last term $\varepsilon_f \dot{m}(\frac{1}{\rho_g} - \frac{1}{\rho_l})$ in Eqs. 5.42 and 5.44 results from the phase change of the liquid phase, and this term is equal to zero when there is no phase change.

- The velocity potential field ϕ_r is then used to correct the pressure field by

$$p = \hat{p} + \rho \frac{\phi_r}{\Delta t}, \tag{5.45}$$

where $\hat{p}$ is the pressure field solved from the Navier-Stokes equations (Eqs. 5.1 and 5.2), ρ is the density field defined by Eqn. 3.2 and Δt the time step.

5.2.3 Coupling algorithm

The open-source framework CFDEMcoupling-PUBLIC [174] is extended to implement the variable-density-based multiphase framework coupling CFD to DEM. A new coupling solver named **cfdemSolverVoFIB** is developed in this thesis. This new solver is capable of realizing the following functionality:

- Get particle data, e.g. particle coordinates, velocity and particle radius from DEM.
- Identify CFD cells covered by solid particles and calculate the void fraction ε_f of each CFD cell.
- Correct the velocity field when solid particles are present in the liquid phase.
- Calculate the fluid-solid interaction force, capillary force, etc.
- Give essential data to DEM, e.g. buoyancy, capillary force, fluid-solid interaction force, etc.
- Repeat these steps mentioned above until the simulation ends.

A detailed coupling algorithm between DEM and CFD for the resolved CFD-DEM approach is shown in Figure 5.5. The solver **cfdemSolverVoFIB** consists of three modules, namely, the CFD module, the DEM module, and the data exchange and processing module as shown in Figure 5.5. OpenFOAM and

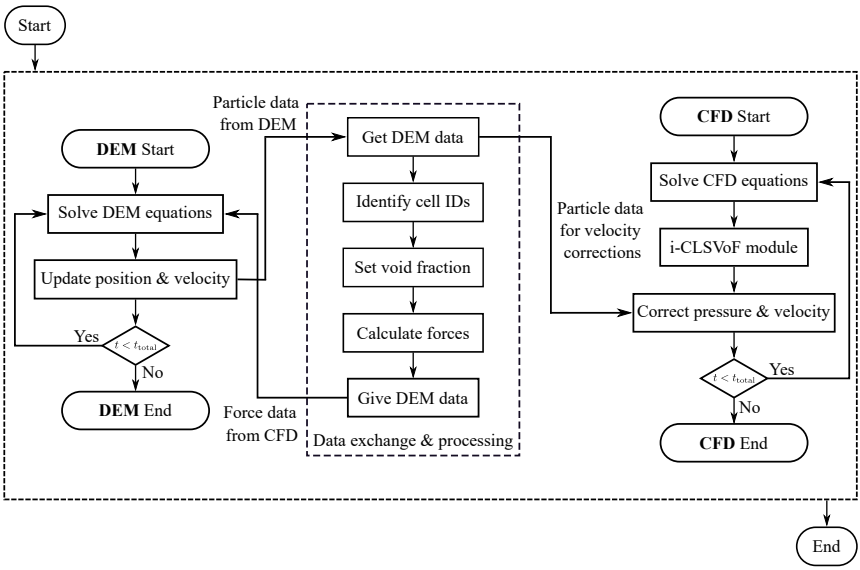


Figure 5.5: The coupling algorithm for the resolved CFD-DEM approach.

LIGGGHTS are employed to conduct the CFD and DEM simulations, respectively. In the resolved CFD-DEM simulations, DEM and CFD conduct their simulations, separately, once the CFD-DEM simulation starts. In DEM, the governing equations (Eqs. 5.6 and 5.7) are solved to update the velocity, position and other information for solid particles. These information is transferred to the data exchange and processing module for further calculations as detailed below. In CFD, the governing equations (Eqs. 5.1 and 5.2) are first solved to update an intermediate velocity field, ignoring solid particles present in the liquid phase.

The i-CLSVoF module is used to capture the moving free surface and calculate the surface tension force.

The data exchange between DEM and CFD is crucial to realize the resolved CFD-DEM coupling. The essential data, e.g. particle positions and velocities calculated on the DEM side, are transferred to the data exchange and processing module. Then, the CFD cells covered by solid particles can be identified, and the void fraction for these CFD cells can be obtained. Furthermore, the particle-fluid interaction forces can be calculated in the data processing module and then transferred to the DEM side to update particle data in the next cycle. Particle velocity data from DEM is transferred to the CFD side, and the numerical correction step is then used to correct the velocity field to satisfy the divergence-free condition as discussed in Section 5.2.2.

The time-step size for stable DEM simulations is given by the Rayleigh time-step given by Eqn. 2.7 in Chapter 2. The maximum time step for guaranteeing a stable CFD simulation is given by Eqn. 3.31 in Chapter 3. The minimal coupling interval for data exchange between DEM and CFD is defined by

$$i_c = \frac{\Delta t^{\mathrm{CFD}}}{\Delta t^{\mathrm{DEM}}}, \tag{5.46}$$

which must be an integer. Increasing the coupling interval requires less computational cost; however, the coupled simulations may then not be accurate enough as the latest data are not exchanged between DEM and CFD in time. The particle data from DEM is used to calculate the void fraction, fluid-structure interaction force, etc. These interaction forces are given back to DEM, and thus the interaction forces acting on the solid phase by the fluid phase can be considered. DEM and CFD go to the next loop once one data exchange is completed, and the whole simulation ends until the prescribed total simulation time is reached.

5.3 Numerical validation

In this section, two benchmark cases are used to validate the resolved CFD-DEM solver **cfdemSolverVoFIB** developed in this thesis. The first case compares the drag coefficient calculated with the resolved CFD-DEM approach against a formula. The other case is to compare the settling velocities and particle position against the corresponding experimental results when a spherical particle settles in liquids.

5.3.1 Validation of calculations of the drag coefficient

The schematic diagram for calculating the drag coefficient is shown in Figure 5.6. A sphere falls down under the influence of gravity. The sphere is fully immersed

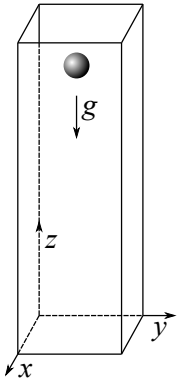


Figure 5.6: The schematic diagram of the numerical set-up for calculations of the drag coefficient.

in the liquid of a container, and the essential parameters for the numerical simulations are listed in Table 5.2. The dynamic viscosity of the liquid in the container significantly influences the sphere's motion, and a wide range of Reynolds numbers can be achieved by varying the liquid dynamic viscosity. Seven numerical

benchmark cases with different dynamic viscosities (refer to Table 5.3) are conducted in this section.

The motion of a single sphere inside the container is governed by Newton's second law of motion given by

$$m\frac{d\mathbf{U}}{dt} = \mathbf{F}_g - \mathbf{F}_b - \mathbf{F}_d, \tag{5.47}$$

where $\mathbf{F}_g$, $\mathbf{F}_b$ and $\mathbf{F}_d$ are the gravitational force, buoyancy and drag force acting on the sphere, respectively [86]. Substituting expressions of these force terms into Eqn. 5.47, leads to

$$\frac{\pi\rho_p D_p^3}{6}\frac{d\mathbf{U}}{dt} = \frac{\pi(\rho_p - \rho_f)D_p^3\mathbf{g}}{6} - \frac{1}{8}C_d\pi D_p^2\rho_f\mathbf{U}^2, \tag{5.48}$$

where D_p is the particle diameter, and ρ_p and ρ_f are the density for the solid and liquid phases, respectively.

Thus, the rate of change of particle velocity $\mathbf{U}$ with respect to time is given by

$$\frac{d\mathbf{U}}{dt} = -\frac{3\rho_f C_d}{4\rho_p D_p}\mathbf{U}^2 + \frac{\rho_p - \rho_f}{\rho_p}\mathbf{g}. \tag{5.49}$$

Table 5.2: Parameters for numerical simulations used to validate calculations of the drag coefficient.

Parameter	Value [units]
Domain size (length, width, height)	$(20, 20, 60)$ [mm]
Particle diameter	2 [mm]
Particle density	3000 [kg/m^3]
Liquid density	1000 [kg/m^3]
Initial particle position (x, y, z)	$(10, 10, 50.5)$ [mm]

Table 5.3: Dynamic viscosities for these seven different cases.

No.	1	2	3	4	5	6	7
μ [kg·m$^{-1}\cdot s^{-1}$]	2.5	5×10^{-1}	1×10^{-1}	5×10^{-2}	1×10^{-2}	5×10^{-3}	3×10^{-3}

Typically, the particle velocity increases gradually and reaches a steady velocity, known as the terminal velocity, when a particle settles in a fluid. Thus, the drag coefficient C_d in Eqn. 5.49 can be calculated by

$$C_d = \frac{4}{3}\frac{\rho_p - \rho_f}{\rho_f}\frac{|\mathbf{g}|D_p}{|\mathbf{U}_t|^2}, \tag{5.50}$$

where $|\mathbf{U}_t|$ is the magnitude of the terminal velocity. Additionally, the particle Reynolds number is given by

$$\mathrm{Re} = \frac{|\mathbf{U}_t|D_p}{\nu}, \tag{5.51}$$

where ν is the kinematic viscosity of the liquid.

Brown et al. corrected the drag coefficient by comparing the corrected formula

$$C_d = \frac{24}{\mathrm{Re}}(1.0+0.15\mathrm{Re}^{0.681}) + \frac{0.407}{1+\frac{8710}{\mathrm{Re}}} \tag{5.52}$$

against extensive experimental data [189]. This corrected drag coefficient (denoted as $C_d-\mathrm{Exp}$) is used to validate the drag coefficient calculations in this section, while the drag coefficient calculated from the numerical simulations (with Eqn. 5.50) is denoted as $C_d-\mathrm{Num}$.

The comparison between the numerical drag coefficient and the corrected drag coefficient given by the formula Eqn. 5.52 is shown in Figure 5.7. Good agreement is obtained for a wide range of Reynolds numbers ranging from 10^{-3} to 10^3. The relative errors of calculating the drag coefficient are listed in Table 5.4 for the quantitative comparison. Calculations of the drag coefficient are more accurate for the Reynolds number between 0.1 and 100. However, the accuracy

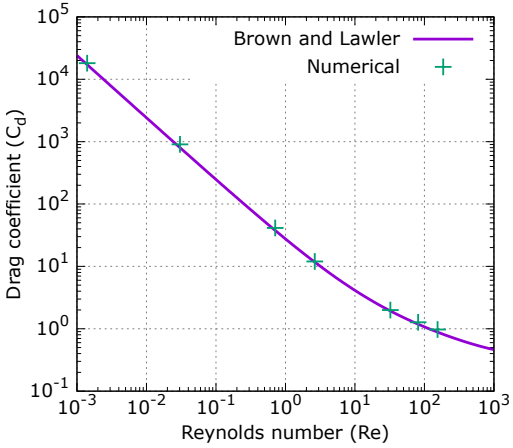


Figure 5.7: Validations for calculations of the drag coefficient C_d.

Table 5.4: Terminal velocity and relative error for these seven validation cases.

Re [-]	Terminal velocity [m/s]	C_d – Num [-]	C_d – Exp [-]	Relative error [-]
0.0014	0.0017	18103.8062	17172.1450	0.0543
0.0304	0.0076	905.8172	800.4452	0.1316
0.7120	0.0356	41.2827	37.7199	0.0945
2.6400	0.0660	12.0110	11.7323	0.0238
32.3600	0.1618	1.9985	1.9306	0.0352
81.2000	0.2030	1.2696	1.1848	0.0716
154.4667	0.2317	0.9746	0.8837	0.1029

of the model needed to be improved for high Reynolds numbers, namely when Re is larger than 100. This relatively large discrepancy for simulations with high Reynolds numbers was reported in the literature as well [39, 170].

5.3.2 Single particle settling

The experimental study and corresponding numerical simulations of single particle settling in viscous liquids were conducted by Ten Cate et al. [190]. The

experimental set-up for the single particle settling is a spherical bearing ball with a diameter of 15 mm, and a density of 1120 kg/m^3 which settles in a container (length $\times$ width $\times$ height= $100 \times 100 \times 160$ mm) under the influence of gravity ($g = 9.81$ m/s^2 in the vertical direction). The sphere is fully immersed in the liquid before it starts to fall, and the initial separation distance between the sphere center and the bottom wall of the container is 120 mm. The sphere experiences acceleration at the beginning and then decelerates when it approaches the bottom wall. This scenario is suitable for validating the numerical implementations for computing the fluid-solid interaction forces and the trajectory of the spherical particle.

In this section, four cases with different liquid densities and dynamic viscosities are considered to validate the numerical model developed in this chapter. These essential parameters are detailed in Table 5.5. The number of cells of the base

Table 5.5: Parameters for the single particle settling simulations (data adopted from [190]).

Case No.	Re [-]	Liquid density [kg/m^3]	Dynamic viscosity [$kg \cdot m^{-1} \cdot s^{-1}$]
1	1.5	970	0.373
2	4.1	965	0.212
3	11.6	962	0.113
4	31.9	960	0.058

mesh resolution is $40 \times 40 \times 64$. Adaptive mesh refinement is used to guarantee fine mesh resolution around the sphere and to allow a relatively coarse mesh elsewhere to reduce computational cost while guaranteeing reasonable numerical accuracy. The no-slip boundary condition is applied to the boundary of the container. The time-step size for both DEM and CFD is 1.0×10^{-5}, and a coupling interval of one is used to exchange data between DEM and CFD.

The liquid velocity field for $\mathrm{Re} = 11.6$ is shown in Figure 5.8. An elongated wake can be seen from the simulation. The dimensionless gap height H/D_p between the sphere and the bottom wall and the magnitude of the particle settling velocity

in the vertical direction are recorded and compared against the corresponding experimental results. This comparison is shown in Figure 5.9. Figure 5.9a and Fig-

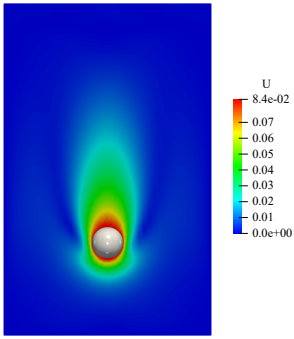
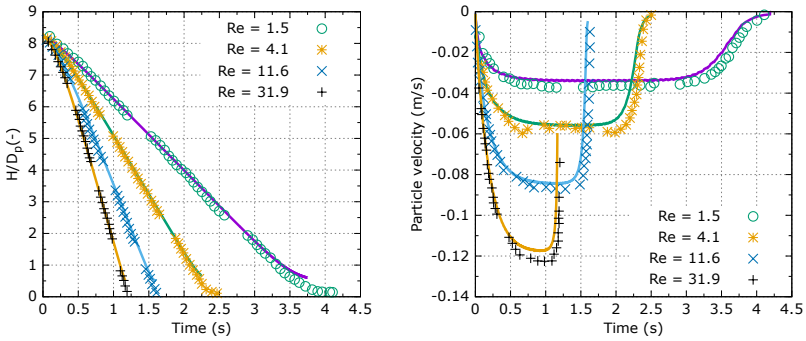


Figure 5.8: The liquid velocity field of single particle settling simulation (Re = 11.6).



(a) **(b)**

Figure 5.9: Validations of the single sphere settling in liquids: (a) dimensionless sphere height, (b) sphere settling velocity.

ure 5.9b are the dimensionless gap height and particle settling velocity, respectively. The solid curves represent numerical results, while the points represent experimental results adopted from the literature [190]. As shown in Figure 5.9b,

the spherical particle first undergoes an acceleration phase and then decelerates due to squeezing liquid between the sphere and the bottom wall when the particle approaches the bottom wall. A good agreement can be found for these four different cases. Some minor discrepancies can be seen from Figure 5.9a, especially when $\mathrm{Re} = 1.5$. The reason is that the lubrication force is not negligible when the gap between the particle and the bottom wall is small. Incorporating the lubrication force can be subject to future work which is not included in the current chapter. Overall, these simulations demonstrate that the numerical implementation for the resolved CFD-DEM method is correct, and that the model is accurate enough to capture the complex fluid-solid interaction and to predict the trajectory of solid particles.

5.4 Application

The improved resolved CFD-DEM method developed in this chapter can be used to model capillary-force-induced or evaporation-induced transport and agglomeration of particles. In this section, some numerical benchmark cases are presented to demonstrate the performance of the resolved CFD-DEM method with the capillary interactions developed in this chapter.

5.4.1 Two particles moving along a free surface with evaporation

The first simulation is to model the evaporation-induced deformation of the meniscus between two spherical particles sitting on a substrate. The numerical set-up for the 3D simulation is shown in Figure 5.10, namely, two spherical particles resting on a wettable substrate. The initial distance between centers of the two particle is $1.3D_p$ (particle diameter). Periodic boundary conditions are applied in x and y directions. The initial liquid height is higher than the particle diameter. The outflow boundary condition is applied at the top to let vapour leave

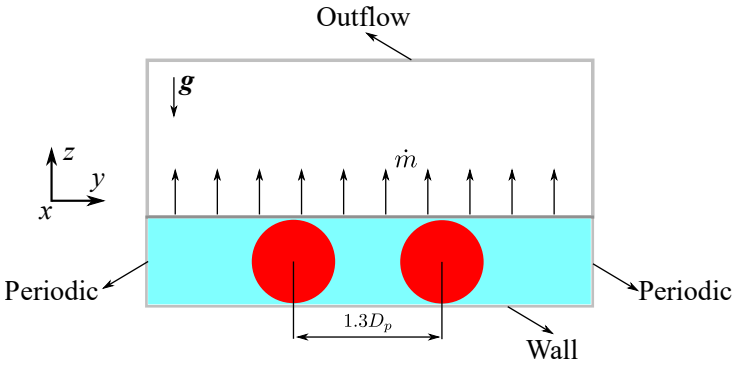
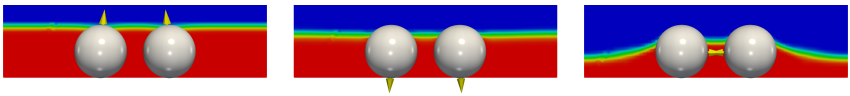


Figure 5.10: 2D schematic diagram of the numerical set-up for two particles moving along a free surface with evaporation.

the domain freely. The constant contact angle and slip boundary conditions are applied on the bottom wall. The specified contact angle between the spherical particle and the liquid surface is 30°, and a constant contact angle specified at the bottom wall is 0°. The constant mass flux evaporation model discussed in Chapter 4 is adopted to model the evaporation of the liquid phase into the gas phase with a constant evaporation rate in this section. To demonstrate the performance of the capillary force model extended in this chapter, only the gravitational force, buoyancy and capillary force are effective, while the particle-liquid interaction force given by Eqn. 5.15 is not considered in the numerical simulation. The essential parameters for this simulation are listed in Table 5.6.

Two particles gradually protrude from the liquid surface after evaporating some liquid from the liquid surface, as shown in Figure 5.11. The vectors of the



(a) (b) (c)

Figure 5.11: Vectors of the particle velocity are represented by the yellow arrows during the evaporation process: (a) pointing upwards, (b) pointing downwards, (c) pointing towards each other.

Table 5.6: Parameters for two particles moving along a free surface with evaporation.

Parameter [Units]	Value
Liquid density $[\mathrm{kg/m^3}]$	10
Gas density $[\mathrm{kg/m^3}]$	1
Particle density $[\mathrm{kg/m^3}]$	25
Particle diameter [m]	1.0×10^{-6}
Liquid viscosity $[\mathrm{Pa \cdot s}]$	1.0×10^{-3}
Gas viscosity $[\mathrm{Pa \cdot s}]$	1.0×10^{-5}
Surface tension $[\mathrm{N/m}]$	0.072
CFD time step [s]	1.0×10^{-9}
DEM time step [s]	1.0×10^{-9}
Coupling interval $[-]$	1
Restitution coefficient $[-]$	0.5
Friction coefficient $[-]$	0.3
Contact angle (particle-interface) $[-]$	30°, 45°, 60°
Contact angle (particle-wall) $[-]$	0°

particle velocity point upwards during the early stage of evaporation (see Figure 5.11a). This is due to the upward capillary force acting on the two particles. Then, the vectors of the particle velocity point downwards (see Figure 5.11b) after evaporating more liquids. A concave meniscus between the two particles gradually forms, which leads to attractive interactions between them, as shown in Figure 5.11c. This attractive capillary force acting on the two particles makes moving toward each other.

Three simulations with different contact angles, namely, 30°, 45° and 60° are presented in Figure 5.12. The key parameters for the solid and liquid phases can be found in Table 5.6. For the very early stage of evaporation, the free surface gradually decreases due to the mass loss and then contacts with the top of the spheres. The virtual free surface inside the solid particles is concave upward, as shown in the top row of Figure 5.12. More mass loss can be found around the two particles when the contact angle increases from 30° to 60°, as shown in

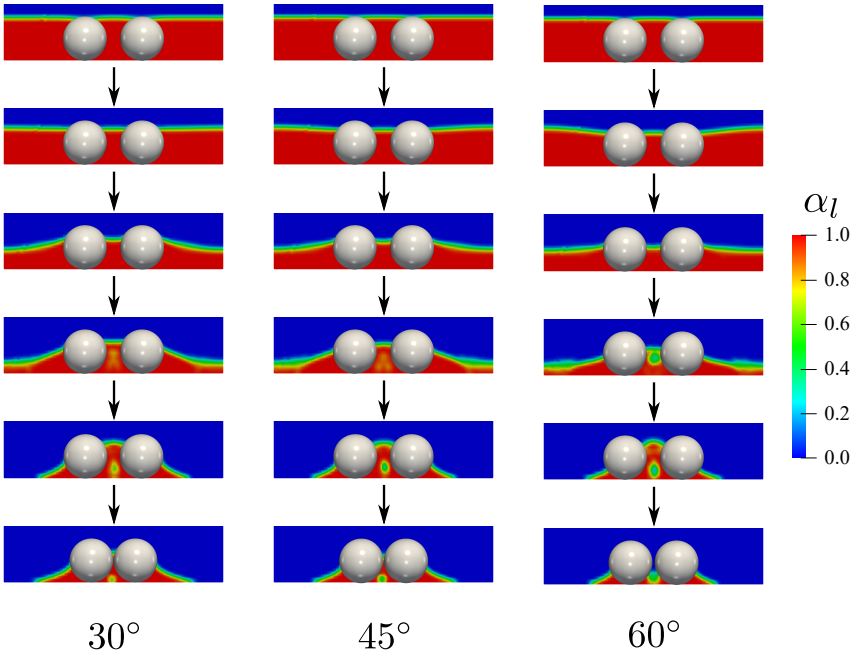


Figure 5.12: Numerical simulations of two particles moving along the free surface for three different specified contact angles.

the second row of Figure 5.12. Concave meniscus and a liquid bridge gradually form around every two particles for the cases with contact angle $\theta = 30^\circ$ and $\theta = 45^\circ$, while a flat meniscus is found for the case with a contact angle of 60°, as shown in the third row of Figure 5.12. These concave meniscuses lead to attractive capillary forces and force each pair of particles to come closer to each other. As the simulation continues, more liquid evaporates around two sides of the computational domain, and the shape of the meniscus changes from concave to convex, as demonstrated by the fourth and fifth rows of Figure 5.12. In all three cases, particles gradually move towards each other during the evaporation process. The liquid phase evaporates faster when the contact angle increases

from 30° to 60°. The same conclusion is also shown in Fig. 4 presented in the literature [191].

5.4.2 Particle transport and accumulation in an evaporating droplet with contact line pinning

In contrast to the aforementioned numerical simulations, the current numerical benchmark case involves many particles inside an evaporating droplet with contact line pinning. In principle, droplet evaporation with contact line pinning results in an internal capillary flow from the droplet center to the edge, as discussed in Section 4.4.4. The radial capillary flow carries some suspended particles from the droplet center to its edge and finally leads to some different particle deposition patterns. Accordingly, the internal flow field is of great significance in affecting particle transport and accumulation during the evaporation process, while in the first numerical demonstration case as discussed in Section 5.4.1, the capillary force is more dominant, and the internal velocity field does not play a significant role.

In order to save computational cost and visualize the evaporation-induced particle transport during the evaporation process, A 2D numerical simulation is adopted in this section, namely, all the particles inside the evaporating droplet can only move along the x and y directions. The numerical set-up is shown in Figure 5.13. 300 micro-sized spherical particles with a diameter of 1 μm are generated ran-

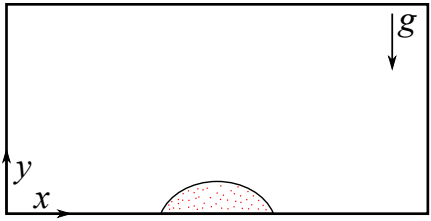


Figure 5.13: The schematic diagram of the numerical setup for 2D droplet evaporation with suspended particles.

domly inside a spherical cap with an initial contact angle of 45° and a radius of 50 µm. The length and height of the computational domain are 300 µm and 150 µm, respectively. As the particle size is small, the Van der Waals force model discussed in Section 2.2.2 is incorporated in this section to account for the non-contacting attractive force acting on the solid particles during the evaporation process. The parameters used in the simulation are listed in Table 5.7.

Table 5.7: Essential parameters for modelling particle transport inside an evaporating droplet.

Parameter [Units]	Value
Liquid density $[\mathrm{kg/m^3}]$	10
Gas density $[\mathrm{kg/m^3}]$	1
Particle density $[\mathrm{kg/m^3}]$	250
Particle diameter [m]	1.0×10^{-6}
Liquid viscosity $[\mathrm{Pa \cdot s}]$	1.0×10^{-3}
Gas viscosity $[\mathrm{Pa \cdot s}]$	1.0×10^{-5}
Surface tension $[\mathrm{N/m}]$	0.072
Surface energy density $[\mathrm{J/m^2}]$	0.86×10^{-3}
CFD time step [s]	1.0×10^{-12}
DEM time step [s]	1.0×10^{-12}
Coupling interval [−]	1
Restitution coefficient [−]	0.5
Friction coefficient [−]	0.3
Contact angle (particle-interface) [−]	30°
Contact angle (particle-wall) [−]	0°

The no-slip boundary condition is applied at the bottom wall to fix the contact line during the evaporation process, and the outflow boundary condition is applied at the top to let vapour leave the domain freely. In order to speed up the numerical simulations and mitigate the influence of un-physical spurious velocity on the internal flow field inside the evaporating droplet, the density of the

liquid phase is scaled by 0.01. As mentioned in Chapter 4, the density-scaled approach is also adopted in the literature [35]. A 2D axisymmetrical model was used in the literature [35]; however, the non-symmetrical numerical configuration, as shown in Figure 5.13, is used in this section. The reason is that the initial particle packing for the DEM simulations is not axisymmetric.

Figure 5.14 shows snapshots of the numerical simulations. The surface in green represents the free surface of the evaporating droplet. As the evaporation pro-

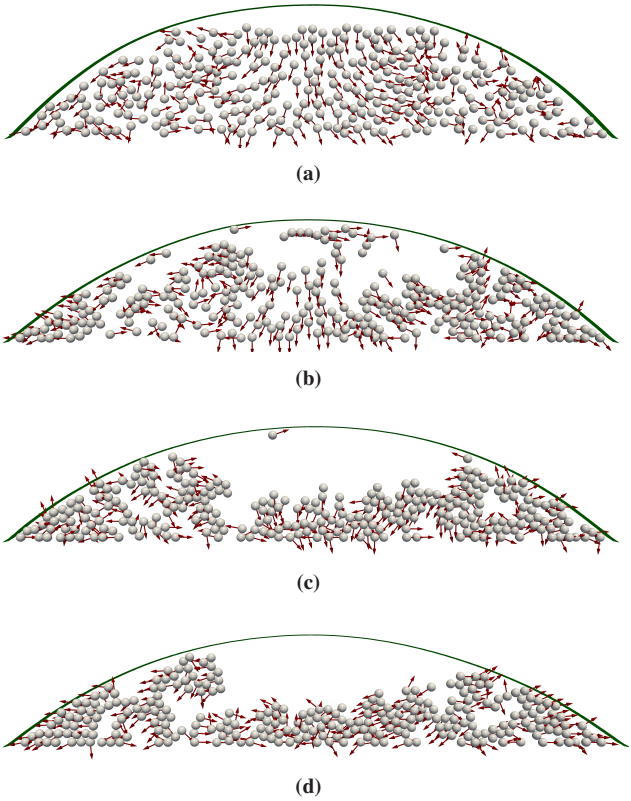


Figure 5.14: Snapshots of the numerical simulations of the four stages of particle deposition patterns inside an evaporating droplet.

ceeds, the initial contact radius between the sessile droplet and the substrate is constant, and the droplet height decreases gradually. The red arrows indicate the velocity vectors of particles. In the initial stage of the evaporation as shown in Figure 5.14a, particles in the middle of the droplet tend to move downwards, while particles around the two corners tend to move towards the triple contact line region due to the radial capillary flow. As shown in Figure 5.14b, some particles tend to agglomerate with their neighbouring particles around the free surface. More and more particles are dragged towards the two corners of the evaporating sessile droplet as the evaporation proceeds, as shown in Figure 5.14c. Figure 5.14d shows agglomerations of particles around the triple contact line region, while fewer particles are deposited in the middle of the droplet.

In this thesis, a simple yet helpful approach has been developed to calculate the local packing fraction with the open-source Voronoi tessellation code Voro++ (refer to Appendix A.3). This approach is adopted to calculate the local packing structure and packing fraction for the particle assembly in this section. Figure 5.15 shows the local packing fraction for the particle deposition pattern shown in Figure 5.14d. It demonstrates that a higher packing fraction can be found

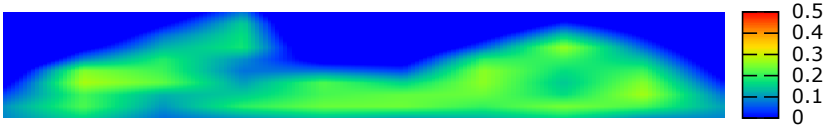


Figure 5.15: The local packing fraction for the particle deposition pattern shown in Figure 5.14d.

around the two corners where much more particles are agglomerated.

This numerical benchmark case demonstrates that the resolved CFD-DEM model can capture the complex particle-fluid, particle-particle and particle-wall interactions when the liquid phase undergoes phase change from liquid to vapour. Changing liquid parameters, e.g. the liquid viscosity, was found to alter particle deposition patterns as discussed in the experimental work [5]. Thus, extensive

numerical simulations with this resolved CFD-DEM model can be conducted to understand the effect of some parameters on the resulting particle deposition patterns.

5.5 Summary

In this chapter, the theory and numerical issues of the improved resolved CFD-DEM approach are discussed. A variable-density resolved model is developed, implemented and validated. An improved capillary force model has been implemented into the open-source CFDEMcoupling-PUBLIC library. A corresponding improved resolved CFD-DEM solver cfdemSolverVoFIB has been thus developed. The main contribution and improvements in this chapter are as follows:

- A variable-density resolved CFD-DEM model with free-surface capturing has been developed.
- The evaporation models discussed in Chapter 4 have been incorporated into the cfdemSolverVoFIB solver to model the gas-liquid-solid multiphase system with evaporation of the liquid phase.
- An improved capillary force model with numerical smoothing has been implemented to improve numerical stability issues when modelling capillary interactions for solid particles moving at a free surface.

A brief comparison between the standard resolved CFD-DEM solver cfdemSolverIB implemented in the CFDEMcoupling-PUBLIC library and the cfdemSolverVoFIB developed in this chapter is outlined in Table 5.8. It turns out that the new solver cfdemSolverVoFIB developed in this chapter extends the applications of the standard resolved CFD-DEM solver cfdemSolverIB.

Two numerical validation cases have been conducted to validate the resolved CFD-DEM solver developed in this chapter. It is proven that the resolved CFD-DEM solver predicts complex particle-fluid interactions with reasonable numerical accuracy. Two numerical benchmark cases, e.g. two particles moving along

Table 5.8: Comparison between the standard cfdemSolverIB solver and the improved solver cfdemSolverVoFIB.

Model/Module	cfdemSolverIB	cfdemSolverVoFIB
Incompressible flow solver	Yes	Yes
Variable density	No	Yes
Free surface capturing	No	Yes
Surface tension	No	Yes
Capillary force	No	Yes
Evaporation of the liquid phase	No	Yes

a free surface with evaporation and evaporation-induced agglomerations of many particles inside an evaporating droplet, have been presented in this chapter. It demonstrates that the performance of the improved resolved CFD-DEM solver is reasonably good in modelling gas-liquid-solid multiphase systems.

6 The improved un-resolved CFD-DEM framework for gas-liquid-solid multiphase flow

The resolved CFD-DEM approach detailed in Chapter 5 can resolve the fluid-solid interactions much more accurately. However, this approach is computationally expensive and thus it is incapable of dealing with a large number of particles. In engineering applications, e.g. civil engineering [160], pharmaceutical industry [38], chemical engineering [43, 132] and so on, a large number of solid particles are processed. Understanding the flow and complex interactions between the solid and fluid phases is of great importance to further the development and control of these applications in industry, as well. In contrast to experimental studies, computational modelling of these solid-fluid systems is capable of capturing and visualizing detailed interactions between solid and fluid phases at lower expense, which would be beyond the capability of experimental devices. The un-resolved CFD-DEM approach was developed to understand the global behaviour of solid-fluid systems with many particles [40]. This approach is not computationally as costly as compared to the resolved model. Similar to the resolved CFD-DEM method, the motion of the solid phase is calculated by the DEM, while locally averaged Navier-Stokes equations govern the flow of the liquid phase. As shown in Figure 6.1, the main feature is that a CFD cell contains several solid particles for the un-resolved CFD-DEM approach. In contrast, a solid particle covers several CFD cells for the re-solved CFD-DEM method, as shown in Figure 5.2. Thus, different numerical methods are needed to calculate solid-fluid interaction forces for the un-resolved CFD-DEM approach.

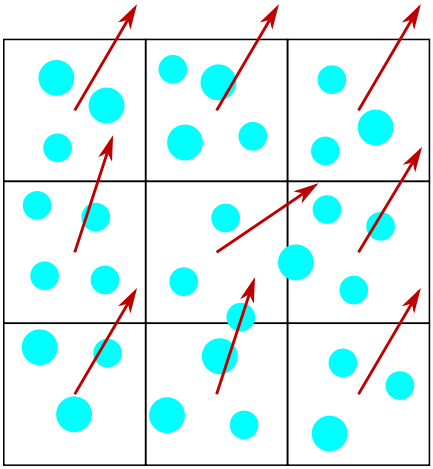


Figure 6.1: The schematic diagram of the un-resolved CFD-DEM approach. Arrows represent the vectors of the local fluid velocity, and blue circles represent solid particles.

During the past few years, the coupled CFD-DEM approach was employed to model some classical geotechnical problems, e.g. one-dimensional consolidation and single particle settling in liquids [44]. Jing et al. developed a VoF-DEM model for modelling free surface flow involving many particles with a large size ratio of CFD cell size to particle size [36]. The effect of rolling friction on a single spout fluidized bed was investigated using a CFD-DEM model by Goniva et al. [84]. The flood fill method was proposed by Lu et al. to model bubble dynamics in a fluidized bed, and various properties of bubbles can be obtained with this novel method, simultaneously [192]. The un-resoled CFD-DEM was adopted to study the flow behaviour of particulate suspensions. DLVO theory was incorporated into an un-resolved CFD-DEM to understand the influence of particle parameters on the property of suspension rheology [43]. An extended CFD-DEM framework with a sub-grid viscosity model was developed by Blais et al. to model the viscous solid-liquid mixing [85]. The adhesion of solid particles on the surface of microchannels was modelled with a CFD-DEM model, which was extended to model in this way the fouling process [193]. The effect

of wettability on the deformation of porous media was investigated using a CFD-DEM model with free surface flow [194]. An improved interface smoothness was developed to model gas-liquid-solid three-phase systems with free surface capturing [195]. A new implicit algorithm was developed for improving stability issues when calculating drag forces acting on solid particles, and the capability of this new algorithm was proven by some numerical simulations related to fluidized beds [196]. A unified finite element CFD-DEM solver was developed, proving that the new solver is faster and cheaper and can mitigate limitations of conventional CFD-DEM models [197]. A summary of these un-resolved CFD-DEM formulations and their applications are listed in Table 6.1. Little literature

Table 6.1: Summary and comparison among these applications of un-resolved CFD-DEM models.

Authors (publication year)	Applications	Surface tension
Goniva et al. (2012) [84]	Fluidized bed	No
Zhao et al. (2013) [44]	Geotechnical problems	No
Lu et al. (2015) [192]	Bubble dynamics	No
Smuts (2015) [43]	Suspension rheology	No
Jing et al. (2016) [36]	Free surface flow with particles	Yes
Blais et al. (2016) [85]	Solid-fluid mixing	No
Trofa et al. (2019) [193]	Fouling in microchannels	No
Davydzenka et al. (2020) [194]	Multiphase flow in porous media	Yes
Washino et al. (2021) [195]	Three-phase flow	Yes
Mori et al. (2021) [196]	New algorithm for stable simulations	Yes
El Geitani et al. (2023) [197]	High-order CFD-DEM solver	No

is found to be related to un-resolved CFD-DEM simulations with the incorporation of an efficient and accurate surface tension force model. However, modelling gas-liquid-solid multiphase systems with surface tension is of great significance to engineering applications, such as direct inkjet printing and spray coating.

In this chapter, the un-resolved CFD-DEM approach is extended for modelling a surface-tension-dominant multiphase system with a large number of particles. In such a gas-solid-liquid system, it involves the following complex interactions:

- **Fluid-fluid interactions**

The interaction between two immiscible fluids is known as the two-phase flow, and a moving interface separates the two fluids, typically known as the gas and liquid phases. Some numerical methods, e.g. LS, VoF, and phase field method, are proposed to capture the evolution and movement of the free surface in the literature as discussed in Chapter 3. In this chapter, the i-CLSVoF framework detailed in in Chapter 3 is adopted to capture the free surface and compute the surface tension. The motion of fluid phases is updated by solving the Navier-Stokes equations.

- **Solid-solid interactions**

  The solid particles interact with their neighbouring particles when the solid phase moves inside the liquid phase. The extended DEM approach detailed in Chapter 2 is adopted in this chapter to model both contact and non-contacting interaction forces between two interacting solid particles or one particle interacting with a wall.

- **Fluid-solid interactions**

  The movement of solid particles immersed in the fluid phase is driven by the complex interplay among fluid drag force, pressure gradient force, viscous force, capillary force, etc. Calculations of these fluid-solid interaction forces are realized by coupling DEM to CFD. The algorithm of coupling DEM to CFD, data exchange and numerical details are detailed in the coming sections.

6.1 Mathematical formulation

In this section, some essential equations for realizing the un-resolved CFD-DEM approach are discussed. The fluid phase is considered as a continuum phase that is governed by the locally averaged Navier–Stokes equations, while the solid phase is modelled as a discrete phase, which is governed by Newton's second law

of motion [86]. Accordingly, in DEM and CFD, two different sets of equations are solved.

6.1.1 Governing equations of the fluid phase

In contrast to the governing equations of the immiscible two-phase flow discussed in Chapter 3, for the un-resolved CFD-DEM approach, the governing equations for the fluid phase are given by

$$\frac{\partial(\varepsilon_f \rho_f)}{\partial t} + \nabla \cdot (\varepsilon_f \rho_f \mathbf{U}) = \dot{m}, \tag{6.1}$$

$$\frac{\partial(\varepsilon_f \rho_f \mathbf{U})}{\partial t} + \nabla \cdot (\varepsilon_f \rho_f \mathbf{U}\mathbf{U}) = \varepsilon_f \nabla p + \varepsilon_f \nabla \cdot \tau + \varepsilon_f \rho_f \mathbf{g} + \varepsilon_f \mathbf{F}_{\text{st}} + \mathbf{F}^{\text{c}}_{\text{pf}}. \tag{6.2}$$

Here, ρ_f is the one-field fluid density. $\dot{m}$ on the right-hand side of Eqn. 6.1 is only non-zero for the system with phase change. Additionally, ε_f is the local fluid volume fraction, and its calculation is detailed in the next section. p is the pressure, τ the viscous stress tensor, $\mathbf{F}_{\text{st}}$ the surface tension, and $\mathbf{F}^{\text{c}}_{\text{pf}}$ the fluid-solid momentum exchange term. The detailed expressions and formulations of these aforementioned force models are discussed in the following sections. Derivation of Eqs. 6.1 and 6.2 is detailed in the Appendix A.2.

For free surface capturing in the un-resolved CFD-DEM, the i-CLSVoF framework detailed in Chapter 3 is extended to capture the sharp free surface. The solid particle phase is present in the liquid phase, and thus the fluid void fraction ε_f is incorporated to account for the motion of the pure liquid phase by excluding the solid phase. As modification of Eqn. 3.6, the transport equation for the un-resolved CFD-DEM model is given by

$$\frac{\partial \varepsilon_f \alpha_f}{\partial t} + \nabla \cdot (\varepsilon_f \alpha_f \mathbf{U}) = 0, \tag{6.3}$$

where α_f is the liquid volume fraction field.

6.1.2 Governing equations of the solid phase

The translational and rotational motions of solid particles are updated by solving the equations

$$m_i \frac{\partial^2 \mathbf{x}_i}{\partial t^2} = \sum_{i=1}^{N_p} (\mathbf{F}_{ij}^{\text{con}} + \mathbf{F}_{ij}^{\text{dlvo}}) + m_i \mathbf{g} + \mathbf{F}_{\text{bro}} + \mathbf{F}_{\text{cap}} + \mathbf{F}_{\text{fp}}^{\text{c}}, \tag{6.4}$$

$$I_i \frac{\partial^2 \theta_i}{\partial t^2} = \sum_{i=1}^{N_p} \mathbf{M}_{ij}^{\text{con}} + \mathbf{M}_{\text{cap}} + \mathbf{M}_{\text{fp}}^{\text{c}}, \tag{6.5}$$

respectively. Here, $\mathbf{F}_{\text{cap}}$ and $\mathbf{M}_{\text{cap}}$ are force and torque resulting from the capillary force, respectively. Besides, $\mathbf{F}_{\text{fp}}^{\text{c}}$ is the fluid-particle interaction force acting on solid particles by the fluid phase. Detailed discussion related to $\mathbf{F}_{\text{fp}}^{\text{c}}$ is presented in the next section. Formulas and details of some other force terms in Eqs. 6.4 and 6.5 have been discussed in Chapters 2 (Section 2.2) and Chapter 5 (Section 5.2.1).

6.2 Numerical method

The numerical implementations for the improved un-resolved CFD-DEM model are based on the CFDEMcoupling-PUBLIC library [92]. In addition, a new un-resolved CFD-DEM solver, i.e. **cfdemSolverVoFX** is developed in this chapter. The numerical method and algorithm to calculate fluid-particle interaction forces and data exchange between DEM and CFD are detailed in this section.

6.2.1 Calculation of the interaction forces

In contrast to numerical calculations of fluid-particle interaction forces for the resolved CFD-DEM model, the fluid cell is larger than the particle size for the un-resolved CFD-DEM approach. Thus, numerical integration over the surface

of a solid particle used by the resolved CFD-DEM approach to calculate these fluid-particle interaction forces is no longer possible. In the un-resolved CFD-DEM approach, the fluid-particle interaction force, the volume fraction of the solid phase, and the void fraction of the fluid phase are calculated by the so-called volume-averaged method. Two sets of formulations, known as model A and model B are widely used to describe the motion of the fluid phase. If the pressure is shared between the solid and fluid phases, it is referred to be model A. Otherwise, it is model B [40].

The fluid-particle interaction force $\mathbf{F}_{\mathrm{fp}}^{\mathrm{c}}$ in Eqn. 6.4 consists of the following force terms given by

$$\mathbf{F}_{\mathrm{fp}}^{\mathrm{c}} = \mathbf{F}_{\mathrm{drag}} + \mathbf{F}_{\mathrm{gradP}} + \mathbf{F}_{\mathrm{viscF}} + \mathbf{F}_{\mathrm{cap}}, \tag{6.6}$$

where $\mathbf{F}_{\mathrm{drag}}$, $\mathbf{F}_{\mathrm{gradP}}$, $\mathbf{F}_{\mathrm{viscF}}$, $\mathbf{F}_{\mathrm{cap}}$ are the drag force, pressure gradient force, viscous force, capillary force term, respectively. Further force term, such as the Basset force, the added mass force, and the lift force which can also be found in the literature [86] are not considered here as they are considered negligible for the numerical simulations presented in this chapter.

The pressure gradient force is calculated by

$$\mathbf{F}_{\mathrm{gradP}} = -V_i \nabla p, \tag{6.7}$$

where V_i is the volume of particle i, and p the fluid pressure. Similarly, the fluid viscous force $\mathbf{F}_{\mathrm{viscF}}$ resulting from the fluid shear stress is calculated by

$$\mathbf{F}_{\mathrm{viscF}} = -V_i \nabla \cdot \tau, \tag{6.8}$$

where τ is the viscous stress tensor. The capillary force acting on solid particles is active when solid particles are immersed in interface cells. It is calculated by

$$\mathbf{F}_{\mathrm{cap}} = \mathbf{F}_{\mathrm{st}} V_i, \tag{6.9}$$

where $\mathbf{F}_{\text{st}}$ is the surface tension force discussed in Chapter 3 (Section 3.2.3). The fluid drag force $\mathbf{F}_{\text{drag}}$ is crucial in realizing the un-resolved CFD-DEM coupling. The formula of drag force $\mathbf{F}_{\text{drag}}$ is given by

$$\mathbf{F}_{\text{drag}} = \frac{1}{8} C_d \pi d^2 \rho_f |\mathbf{U}_f - \mathbf{U}_p| (\mathbf{U}_f - \mathbf{U}_p), \tag{6.10}$$

where $\mathbf{U}_f$ and $\mathbf{U}_p$ are the fluid and particle velocity, respectively and d and ρ_f are the particle diameter and fluid density, respectively. Besides, C_d is the drag coefficient for which several different corrections have been proposed in the literature [189, 198–201] as listed in Table 6.2. The drag coefficient is related to the particle Reynolds number Re.

Table 6.2: Difference formulas for the corrections of the drag coefficient C_d

Model	Formula ($C_d =$)
Stokes [198]	$\frac{24}{\text{Re}}$
Dallavalle [199]	$\left[0.63 + \frac{4.8}{\sqrt{\text{Re}}}\right]^2$
Wen and Yu [200]	$\begin{cases} \frac{24}{\text{Re}}(1.0 + 0.15\text{Re}^{0.687}) & \text{Re} \leq 1000 \\ 0.44 & \text{Re} > 1000 \end{cases}$
Brown and Lawler [189]	$\frac{24}{\text{Re}}(1.0 + 0.15\text{Re}^{0.681}) + \frac{0.407}{1 + \frac{8710}{\text{Re}}}$
Haider and Levenspiel [201]	$\frac{24}{\text{Re}}(1.0 + 0.1806\text{Re}^{0.6459}) + \frac{0.4251}{1 + \frac{6880.95}{\text{Re}}}$

Extensive experimental data was collected by Brown et al., to compare it against the five different corrections of the drag coefficient [189]. As shown in Figure 6.2, the Stokes drag correction only works well in the lower Reynolds number range as shown in Figure 6.2b, while underestimating the drag coefficient when the Reynolds number is larger than 1. Accordingly, Stokes' correction is only

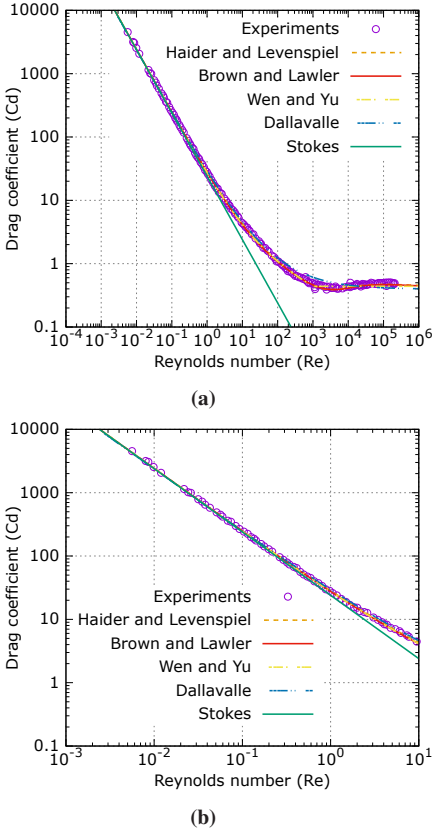


Figure 6.2: Comparison between corrections of drag coefficient against the experimental data: (a) global plot , (b) local plot.

applicable to very small Reynolds numbers. The correction of the drag coefficient proposed by Haider and Levenspiel is in best agreement with the extensive experimental data adopted from the literature [189] for the whole range of the Reynolds number ($10^{-4} - 10^{6}$). Thus, the drag coefficient corrected by Haider et al. is adopted in this chapter to calculate the fluid drag force $\mathbf{F}_{\text{drag}}$.

According to Newton's third law of motion, the momentum exchange term $\mathbf{F}_{\mathrm{pf}}^{\mathrm{c}}$ in Eqn. 6.2 can be calculated by summarizing the reaction forces acting on all particles by the fluid phase inside one CFD cell:

$$\mathbf{F}_{\mathrm{pf}}^{\mathrm{c}} = -\frac{\sum_{i}^{N_p} \mathbf{F}_{\mathrm{fp}}^{\mathrm{c},i}}{V_{\mathrm{cell}}}. \tag{6.11}$$

Here N_p is the number of particles inside the CFD cell and V_{cell} is the volume of the CFD cell.

6.2.2 Calculation of the local fluid volume fraction

An accurate estimation of the fluid volume fraction is of great significance in obtaining accurate results of the un-resolved CFD-DEM simulations, because the fluid volume fraction appears in several equations, e.g. Eqn. 6.1, Eqn. 6.2 and Eqn. 6.3 of the un-resolved CFD-DEM approach. The local fluid volume fraction ε_f is defined as the volume of pure fluids divided by the volume of the fluid cell and is given by

$$\varepsilon_f = \frac{V_{\mathrm{cell}} - V_p}{V_{\mathrm{cell}}}, \tag{6.12}$$

where V_p is the total volume for all the particles inside a fluid cell as shown in Figure 6.3. Two methods have been implemented in the CFDEMcoupling-PUBLIC library, namely *center* and *divided* volume fraction models to calculate the local fluid volume fractions [165]. In the *center* volume fraction model, only those particles are accounted for in calculating the liquid volume fraction the centers $\mathbf{X}_{p,i}$ of which are located in the fluid cell j. The local fluid volume fraction is then calculated by

$$\varepsilon_f = 1 - \frac{\sum_{i=1}^{N_p} V_{p,i}}{V_{\mathrm{cell},j}}. \tag{6.13}$$

Here, N_p is the number of particles either fully or partially located inside a fluid cell. This method is easy to understand and implement but is not accurate enough.

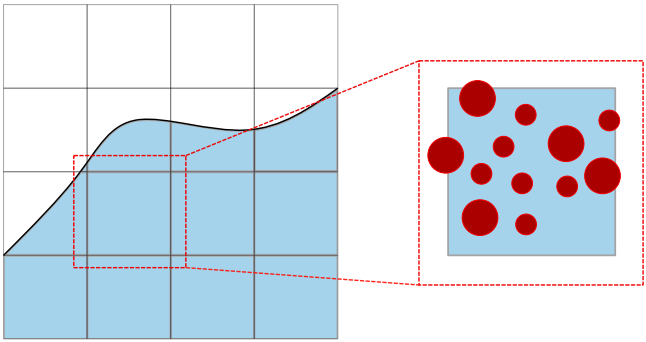


Figure 6.3: The schematic diagram of some solid particles immersed in an internal fluid cell.

As shown in Figure 6.4, four particles partially cover other fluid cells. The particle $P1$ is fully immersed in fluid cell $C1$, and the fluid volume fraction of cell $C1$ is correctly calculated. However, in fluid cells, $C2$ and $C3$, only some partition of particles are detected. The partition in yellow is out of fluid cells where the center of these particles locate. Thus, the *center* volume fraction model overestimates the volume fractions in both cells $C2$ and $C3$. Two partitions of particle $P3$ and

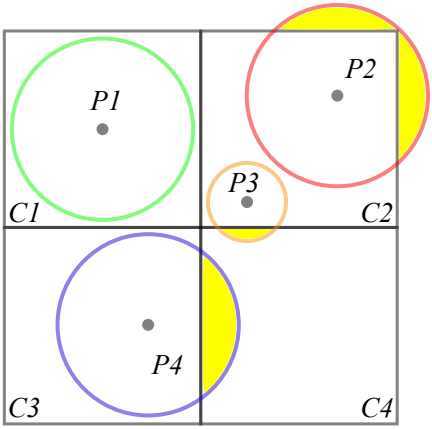


Figure 6.4: The schematic diagram of the *center* volume fraction model.

$P4$ locate in cell $C4$; however, the volume fraction of cell $C4$ is underestimated as zero by the *center* volume fraction model.

The *divided* fluid volume fraction model is more accurate as an additional parameter known as the particle volume fraction $\alpha_{p,i} \in [0,1]$ is used to calculate the volume of particles located at boundaries of fluid cells as shown in Figure 6.4. The liquid volume fraction calculated by the *divided* model is given by

$$\varepsilon_f = 1 - \frac{\sum_{i=1}^{N_p}(\alpha_{p,i} V_{p,i})}{V_{\text{cell}}}, \tag{6.14}$$

where $\alpha_{p,i}$ is one when the particle i is fully immersed in a fluid cell, e.g. particle $P1$ as shown in Figure 6.4, and lower than one when particles are immersed in a fluid cell, partially.

In order to calculate the particle volume fraction $\alpha_{p,i}$ for particle i, the volume of particle i is divided into $N_{\text{sd}} = 29$ non-overlapping sub-domains with the same volume [165]. The center of each sub-domains is checked if it is inside the fluid cell j, and then the particle volume fraction can be computed by

$$\alpha_{p,i} = \frac{N_{\text{sum}}}{N_{\text{sd}}}, \tag{6.15}$$

where N_{sum} is the total number of sub-domain whose centers are inside the fluid cell j. Thus, the *divided* volume fraction model predicts the liquid volume fraction more accurately than the *center* model.

6.2.3 Finite volume discretization of the governing equations

Substituting $\nabla p = \nabla p_{\text{rgh}} + \rho\mathbf{g} + \mathbf{g}\cdot\mathbf{h}\nabla\rho$ and $\nabla\cdot\left[\mu\left(\nabla\mathbf{U} + (\nabla\mathbf{U})^T\right)\right] = \nabla\cdot(\mu\nabla\mathbf{U}) + \nabla\mathbf{U}\cdot\nabla\mu$ (detailed derivation can be found in [120]) into Eqn. 6.2, leads to

$$\begin{aligned}\frac{\partial(\varepsilon_f\rho\mathbf{U})}{\partial t} + \nabla\cdot(\varepsilon_f\rho\mathbf{U}\mathbf{U}) = \\ &-\varepsilon_f(\nabla p_{\text{rgh}} + \mathbf{g}\cdot\mathbf{h}\nabla\rho) \\ &+\varepsilon_f\nabla\cdot(\mu\nabla\mathbf{U}) + \varepsilon_f\nabla\mathbf{U}\cdot\nabla\mu \\ &+\varepsilon_f\mathbf{F}_{\text{st}} + \mathbf{F}^{\text{c}}_{\text{pf}}.\end{aligned} \tag{6.16}$$

The volume integral of Eqn. 6.16 is given by

$$\begin{aligned}\int_{\Omega_i}\frac{\partial(\varepsilon_f\rho\mathbf{U})}{\partial t}\,dV + \underbrace{\int_{\Omega_i}\nabla\cdot(\varepsilon_f\rho\mathbf{U}\mathbf{U})\,dV}_{=\int_{\partial\Omega_i}(\varepsilon_f\rho\mathbf{U}\mathbf{U})\cdot\mathbf{n}\,dS} \\ &= -\int_{\Omega_i}\varepsilon_f\nabla p_{\text{rgh}}\,dV - \int_{\Omega_i}\varepsilon_f\mathbf{g}\cdot\mathbf{h}\nabla\rho\,dV \\ &+ \underbrace{\int_{\Omega_i}\varepsilon_f\nabla\cdot(\mu\nabla\mathbf{U})\,dV}_{=\int_{\partial\Omega_i}(\varepsilon_f\mu\nabla\mathbf{U})\cdot\mathbf{n}\,dS} + \int_{\Omega_i}\varepsilon_f\nabla\mathbf{U}\cdot\nabla\mu\,dV \\ &+ \int_{\Omega_i}\varepsilon_f\mathbf{F}_{\text{st}}\,dV + \int_{\Omega_i}\mathbf{F}^{\text{c}}_{\text{pf}}\,dV,\end{aligned} \tag{6.17}$$

where the transformation from volume integral to surface integral is done by using the divergence theorem.

As mentioned in Chapter 3, $\partial\Omega_i$ is a closed surface bounding the control volume Ω_i, and dS represents an infinitesimal surface element with unit normal $\mathbf{n}$ pointing outwards of the surface $\partial\Omega_i$ ($\mathbf{n}dS = d\mathbf{S}$).

The temporal derivative term in Eqn. 6.17 is approximated by

$$\int_{\Omega_i} \frac{\partial(\varepsilon_f \rho \mathbf{U})}{\partial t}\, dV \approx \frac{\varepsilon_{f,P}^{n+1}\rho_P^{n+1}\mathbf{U}_P^{n+1} - \varepsilon_{f,P}^{n}\rho_P^{n}\mathbf{U}_P^{n}}{\Delta t} V_P, \tag{6.18}$$

where the superscript n represents the current time step at which the quantities (e.g. $\mathbf{U}_P^n$) are known. The superscript $n+1$ denotes the quantities in the next time step needed to be solved (e.g. $\mathbf{U}_P^{n+1}$). Furthermore, V_P is the volume of the cell.

The convection term in Eqn. 6.17 is approximated by converting the surface integral to a sum over all the faces given by

$$\int_{\partial\Omega_i} (\varepsilon_f \rho \mathbf{U}\mathbf{U}) \cdot \mathbf{n}\, dS \approx \sum (\varepsilon_f^n \rho^n \mathbf{U}^n \mathbf{U}^{n+1})_f \cdot \mathbf{S}_f = \sum \varepsilon_f^n F_f^n \mathbf{U}_f^{n+1}, \tag{6.19}$$

where F_f^n is the mass flux calculated by

$$F_f^n = (\rho^n \mathbf{U}^n)_f \cdot \mathbf{S}_f \tag{6.20}$$

with $\mathbf{S}_f$ being the face area vector.

The diffusion term is discretized as

$$\int_{\partial\Omega_i} (\varepsilon_f \mu \nabla \mathbf{U}) \cdot \mathbf{n}\, dS \approx \sum (\varepsilon_f \mu \nabla \mathbf{U}^{n+1})_f \cdot \mathbf{S}_f = \sum \varepsilon_f \mu_f (\nabla \mathbf{U})_f^{n+1} \cdot \mathbf{S}_f. \tag{6.21}$$

The pressure gradient term is numerically approximated by

$$\int_{\Omega_i} \varepsilon_f \nabla p_{\mathrm{rgh}}\, dV = \int_{\partial\Omega_i} \varepsilon_f p_{\mathrm{rgh}}\, d\mathbf{S} \approx \sum \varepsilon_{f,f}^n p_{\mathrm{rgh},f}^n \mathbf{S}_f. \tag{6.22}$$

The explicit expression for the predicted velocity field $\mathbf{U}_P^{n+1}$ is given by

$$\begin{aligned}\frac{\varepsilon_{f,P}^{n+1}\rho_P^{n+1}\mathbf{U}_P^{n+1}-\varepsilon_{f,P}^{n}\rho_P^{n}\mathbf{U}_P^{n}}{\Delta t}V_P \\ +\sum\varepsilon_f^n F_f^n\mathbf{U}_f^{n+1}= \\ \sum\varepsilon_f^n\mu_f(\nabla\mathbf{U})_f^{n+1}\cdot\mathbf{S}_f-\sum\varepsilon_f^n p_{\mathrm{rgh},f}^n\mathbf{S}_f.\end{aligned} \tag{6.23}$$

The other missing terms in Eqn. 6.17, e.g. surface tension, gravitational force and particle-fluid interaction force are incorporated into Eqn. 6.34, as discussed below, which then becomes the discretized form of Eqn. 6.17 with all these terms included.

The velocity field $\mathbf{U}_f^{n+1}$ and pressure field $\mathbf{p}_{\mathrm{rgh},f}^{n+1}$ on faces are calculated by the numerical interpolations

$$\mathbf{U}_f^{n+1}=\frac{\mathbf{U}_P^{n+1}+\mathbf{U}_N^{n+1}}{2}, \tag{6.24}$$

and

$$p_{\mathrm{rgh},f}^{n}=\frac{p_{\mathrm{rgh},P}^{n}+p_{\mathrm{rgh},N}^{n}}{2}, \tag{6.25}$$

respectively.

The first term $\varepsilon_f^n\mu_f(\nabla\mathbf{U})_f^{n+1}\cdot\mathbf{S}_f$ on the right-hand side of Eqn. 6.23 is calculated by

$$\varepsilon_f^n\mu_f(\nabla\mathbf{U})_f^{n+1}\cdot\mathbf{S}_f=\varepsilon_f^n\mu_f\left((\nabla\mathbf{U})_f^{n+1}\cdot\frac{\mathbf{S}_f}{|\mathbf{S}_f|}\right)\cdot|\mathbf{S}_f|, \tag{6.26}$$

where $(\nabla\mathbf{U})_f^{n+1}\cdot\frac{\mathbf{S}_f}{|\mathbf{S}_f|}$ is the face normal gradient, and is approximated by

$$(\nabla\mathbf{U})_f^{n+1}\cdot\frac{\mathbf{S}_f}{|\mathbf{S}_f|}\approx\frac{\mathbf{U}_N^{n+1}-\mathbf{U}_P^{n+1}}{|\mathbf{d}|}, \tag{6.27}$$

where $\mathbf{d}$ is the distance vector between the current cell P and the neighbour cell N.

Substituting these approximations (Eqs. 6.24, 6.25, and 6.27) into Eqn. 6.23, leads to

$$\begin{aligned}\frac{\varepsilon_{f,P}^{n+1}\rho_P^{n+1}\mathbf{U}_P^{n+1}-\varepsilon_{f,P}^{n}\rho_P^{n}\mathbf{U}_P^{n}}{\Delta t}V_P & \\ +\sum\varepsilon_f^n F_f^n\frac{\mathbf{U}_P^{n+1}+\mathbf{U}_N^{n+1}}{2} & \\ =\sum\varepsilon_f^n\mu_f\frac{\mathbf{U}_N^{n+1}-\mathbf{U}_P^{n+1}}{|\mathbf{d}|}|\mathbf{S}_f| & \\ -\sum\varepsilon_f^n\frac{p_{\mathrm{rgh},P}^n+p_{\mathrm{rgh},N}^n}{2}\mathbf{S}_f. &\end{aligned} \tag{6.28}$$

After some manipulations, Eqn. 6.28 leads to

$$\begin{aligned}&\left(\frac{\varepsilon_{f,P}^{n+1}\rho_P^{n+1}}{\Delta t}+\frac{1}{V_P}\sum\frac{\varepsilon_f^n F_f^n}{2}+\frac{1}{V_P}\sum\varepsilon_f^n\frac{\mu_f|\mathbf{S}_f|}{|\mathbf{d}|}\right)\mathbf{U}_P^{n+1}= \\ &-\sum\frac{1}{V_P}\left(\varepsilon_f^n\frac{F_f^n}{2}-\varepsilon_f^n\frac{\mu_f|\mathbf{S}_f|}{|\mathbf{d}|}\right)\mathbf{U}_N^{n+1} \\ &+\varepsilon_{f,P}^n\frac{\rho_P^n}{\Delta t}\mathbf{U}_P^n-\frac{1}{V_P}\sum\varepsilon_f^n\frac{p_{\mathrm{rgh},P}^n+p_{\mathrm{rgh},N}^n}{2}\mathbf{S}_f.\end{aligned} \tag{6.29}$$

Eqn. 6.29 can be abbreviated as

$$A_P\mathbf{U}_P^{n+1}=-\sum A_N\mathbf{U}_N^{n+1}+S_P^n-\frac{1}{V_P}\sum\varepsilon_f^n\frac{p_{\mathrm{rgh},P}^n+p_{\mathrm{rgh},N}^n}{2}\mathbf{S}_f, \tag{6.30}$$

where

$$A_P=\left(\frac{\varepsilon_{f,P}^{n+1}\rho_P^{n+1}}{\Delta t}+\frac{1}{V_P}\sum\frac{\varepsilon_f^n F_f^n}{2}+\frac{1}{V_P}\sum\varepsilon_f^n\frac{\mu_f|\mathbf{S}_f|}{|\mathbf{d}|}\right), \tag{6.31}$$

$$A_N=\frac{1}{V_P}\left(\varepsilon_f^n\frac{F_f^n}{2}-\varepsilon_f^n\frac{\mu_f|\mathbf{S}_f|}{|\mathbf{d}|}\right), \tag{6.32}$$

$$S_P^n=\varepsilon_{f,P}^n\frac{\rho_P^n}{\Delta t}\mathbf{U}_P^n. \tag{6.33}$$

Eqn. 6.30 can be further changed to

$$A_P\mathbf{U}_P^{n+1} + \sum A_N\mathbf{U}_N^{n+1} = S_P^n - \varepsilon_f \nabla p_{\mathrm{rgh},P} - \varepsilon_f \mathbf{g}\cdot\mathbf{h}\nabla\rho_P + \varepsilon_f \sigma k \nabla \alpha_P + \mathbf{F}_{\mathrm{pf}}^c, \quad (6.34)$$

if all the other terms in Eqn. 6.16 are incorporated.

Dividing Eqn. 6.34 by A_P, leads to

$$\mathbf{U}_P^{n+1} = \mathbf{HbyA}_P^{n+1} - \frac{1}{A_P}\left(\varepsilon_f \nabla p_{\mathrm{rgh},P} + \varepsilon_f \mathbf{g}\cdot\mathbf{h}\nabla\rho_P - \varepsilon_f \sigma k \nabla \alpha_P + \mathbf{F}_{\mathrm{pf}}^{\mathrm{c}}\right), \quad (6.35)$$

where $\mathbf{HbyA}_P^{n+1}$ is given by

$$\mathbf{HbyA}_P^{n+1} = \frac{1}{A_P}\left(-\sum A_N\mathbf{U}_N^{n+1} + S_P^n\right). \quad (6.36)$$

Substituting Eqn. 6.35 to the continuity equation (Eqn. 6.1), the pressure equation can be thus derived as

$$\nabla\cdot\left(\frac{\varepsilon_f}{A_P}\nabla p_{\mathrm{rgh}}^{n+1}\right) = \nabla\cdot\left(\mathbf{HbyA}_P^{n+1} + \frac{1}{A_P}(\varepsilon_f \sigma k \nabla\alpha_{n+1} - \varepsilon_f \mathbf{g}\cdot\mathbf{h}\nabla\rho^{n+1} + \mathbf{F}_{\mathrm{pf}}^{\mathrm{c}})\right). \quad (6.37)$$

The pressure field solved from the Eqn. 6.37 is used to correct the velocity field, so that the velocity field can satisfy the continuity equation (Eqn. 6.1). The corresponding numerical correction is given by

$$\mathbf{U} = \mathbf{HbyA}_P + \frac{\varepsilon_f\left(\mathbf{F}_{\mathrm{st}} - \mathbf{g}\cdot\mathbf{h}\nabla\rho - \nabla p_{\mathrm{rgh}}\right) + \mathbf{F}_{\mathrm{pf}}}{A_P}. \quad (6.38)$$

6.2.4 Coupling algorithm

Similar to the resolved CFD-DEM method, the basic idea behind the un-resolved CFD-DEM coupling is also to run DEM and CFD simulations in parallel and exchange essential data between CFD and DEM for every coupling interval. However, numerical methods used to calculate the particle-fluid interaction forces for

the un-resolved CFD-DEM method are different from that of the resolved CFD-DEM method. The essential numerical steps for realizing the un-resolved CFD-DEM coupling are outlined in Figure 6.5. Once the un-resolved simulation starts,

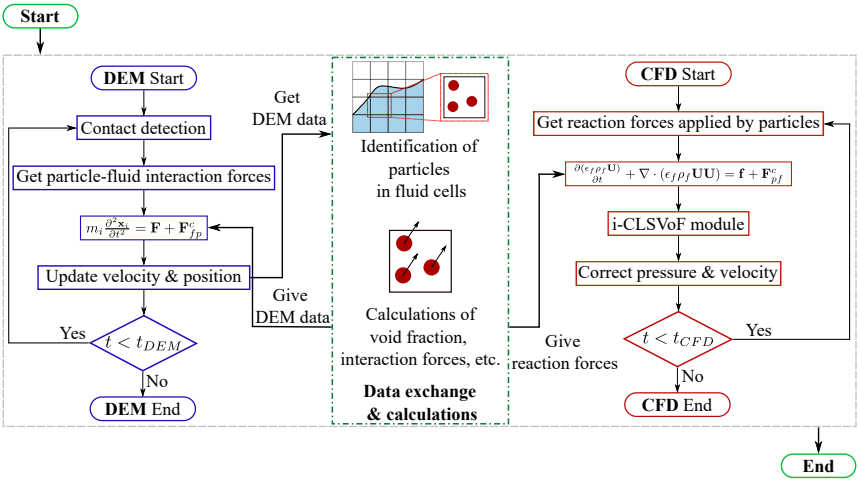


Figure 6.5: The coupling algorithm of the un-resolved CFD-DEM model.

the DEM and CFD simulations are conducted by LIGGGHTS and OpenFOAM, respectively. Some detailed numerical steps are listed below:

- First, calculate essential particle data, e.g. particle coordinates, particle velocity and particle radius in DEM and then transfer these data to the data exchange and calculations module, as demonstrated by the arrow with text "Get DEM data" shown in Figure 6.5.
- Second, identify particles in each CFD cell and then calculate the void fraction for each CFD cell, once receiving these essential data from DEM. This is conducted by the data exchange and calculations module.

- Third, calculate the fluid-particle interaction forces, e.g. drag force, pressure gradient force, viscous force and capillary force also in the data exchange and calculations module.
- Fourth, receive these fluid-particle interaction forces for DEM and then solve the governing equations of DEM to update particle data for the next computational loop. This is known as "give DEM data" as shown by the arrow with text "Give DEM data".
- Fifth, calculate the reaction forces acting on the fluid phase by particles according to Eqn. 6.11. Incorporation of the reaction forces into the governing equations of CFD is realized by the "Give reaction forces" module as shown in Figure 6.5 and then solve these equations to update the motion of the fluid phase.
- Finally, repeat these steps mentioned above until the un-resolved CFD-DEM simulation ends.

6.3 Numerical validation and application

Three classical benchmark cases, e.g. single particle settling in viscous liquids [36, 44], one sphere settling from the air phase to the liquid phase [36, 44, 132] and dam break with particles [36, 132] widely used in the literature are adopted to validate the extended un-resolved CFD-DEM method developed in this chapter.

6.3.1 Single particle settling in viscous liquids

A spherical solid particle moving downwards in a viscous liquid under the influence of gravity is widely adopted to validate the implementation for calculating the fluid-solid interaction force in the literature [36, 132, 202]. According to the

derivation in Chapter 5, Eqn. 5.49, the rate of change of particle velocity versus time is given by

$$\frac{d\mathbf{U}}{dt} = -\frac{3\rho_f C_d}{4\rho_p D_p}\mathbf{U}^2 + \frac{\rho_p - \rho_f}{\rho_p}\mathbf{g}. \tag{6.39}$$

Here, C_d is the drag coefficient, and the corrected formula by Haider et al. [201] is used in this section. Thus, Eqn. 6.39 can only be solved numerically to update the particle velocity and position. A python script attached in the Appendix Code list A.5 is used to solve this ordinary differential equation numerically.

Six different numerical simulations were conducted to validate the capability of the un-resolved CFD-DEM model for a wide range of Reynolds numbers. The varying quantity for the six different cases is the fluid dynamic viscosity μ_f which leads to different terminal velocities $|\mathbf{U}_t|$. Fluid dynamic viscosities, terminal velocities and Reynolds numbers are listed in Table 6.3. The comparison be-

Table 6.3: Parameters for these six different cases.

Case No.	1	2	3	4	5	6
μ_f [Pa·s]	1×10^{-2}	5×10^{-3}	2.5×10^{-3}	1×10^{-3}	5×10^{-4}	1×10^{-4}
$\lvert\mathbf{U}_t\rvert$ [m/s]	0.0673	0.0977	0.1308	0.1764	0.2102	0.2577
Re [−]	6.73	19.54	52.32	176.40	420.40	2577.00

tween numerical results and the corresponding analytical solutions are shown in Figure 6.6. The predicted particle velocities are in good agreement with their corresponding analytical results from Eqn. 6.39 as shown in Figure 6.6a. The minor discrepancy for the smallest Reynolds number is found before reaching the plateau. The discrepancy is due to an assumption underlying the analytical result given by Eqn. 6.39, namely, that the fluid velocity $\mathbf{U}$ is zero during the whole process. However, the four-way coupling, namely, accounting for interactions between particle-particle, particle-wall, particle-fluid and fluid-particle is adopted in the un-resolved CFD-DEM model implemented in this chapter. Thus the movement of the sphere in the liquid phase leads to non-zero velocity fields $\mathbf{U}$ for the liquid phase in these numerical simulations. On the other hand, the influence of particle movement on the liquid velocity can be negligible for simulations

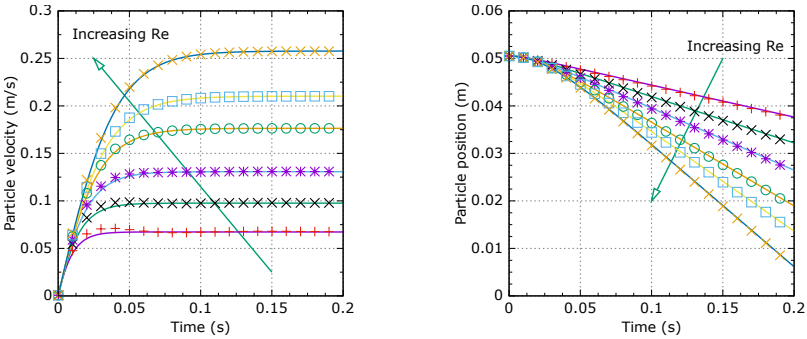


Figure 6.6: Validations of the single sphere settling in viscous liquids: (a) particle velocity, (b) particle position (solid curves: solution from Eqn. 6.39, dots: numerical solution from simulations).

with larger Reynolds number, as can be seen from the very good match between the analytical result from Eqn. 6.39 and the CFD-DEM solution in Figure 6.6a. The particle positions are in perfect agreement with the numerical solution of Eqn. 6.39 for all six cases, as demonstrated in Figure 6.6b.

6.3.2 Single particle settling from air to liquid

In the second benchmark case, a spherical sphere with its diameter of 1 mm and a density of 2500 kg/m^3 starts to fall down from a height of 0.02 m over the liquid surface as shown in Figure 6.7. This is a classical benchmark case widely used in the literature [36, 44, 202]. The sphere experiences free falling, then goes through the liquid surface, and finally hits the bottom of the container. The evolution of the particle velocity when it moves downwards in both gas and liquid phases can be updated by solving Eqn. 6.39. In the first stage, i.e. the particle falls down in the gas phase, the gas density and viscosity are used to solve Eqn. 6.39, while the liquid density and viscosity are used when the particle moves in the liquid phase for the second stage. As shown in Figure 6.8, the particle accelerates in the gas phase, and then its velocity decreases gradually until it reaches its terminal

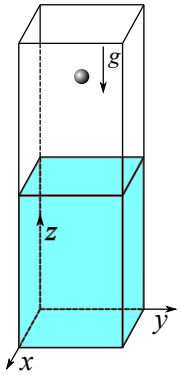


Figure 6.7: The schematic diagram of a single particle settling from air to liquid.

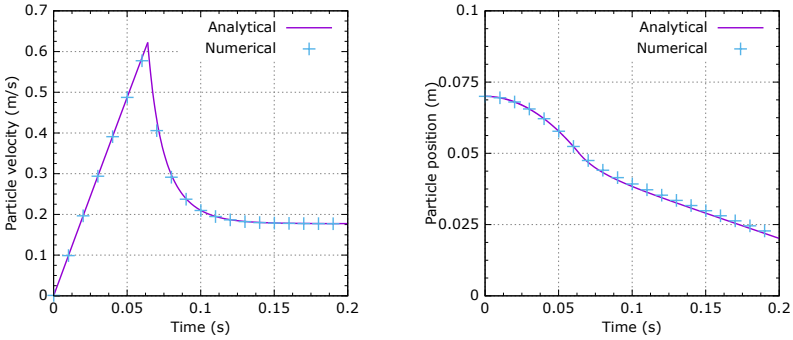


Figure 6.8: Validations of the single sphere settling from the gas phase to the liquid phase: (a) particle velocity, (b) particle position (solid curves: analytical solution, dots: numerical solution).

velocity when it moves downwards in the liquid phase. Both particle settling velocity as shown in Figure 6.8a and position as shown in Figure 6.8b are in perfect agreement with the corresponding solution of Eqn. 6.39 for both the first and second stages. It proves that the un-resolved CFD-DEM method presented in this chapter can predict complex solid-particle interactions accurately when the solid particle moves from one phase to the other.

6.3.3 3D dam break simulations with particles

The third benchmark case involves 3883 spherical particles with the mean diameter of 2.7 mm. The numerical set-up of the 3D dam break simulation is shown in Figure 6.9, and the size of the computational domain is 0.1 m $\times$ 0.2 m $\times$ 0.3 m

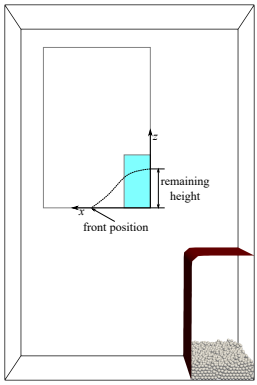


Figure 6.9: The schematic diagram of the 3D dam break simulation with particles and the 3D surface in red indicates the initial free surface of the water column. The inset demonstrates a 2D sketch of the evolution of the free surface.

(width $\times$ length $\times$ height). No-slip boundary conditions are applied at the walls. A water tank with a width of $a = 0.05$ m and a height of 0.1 m is located at the bottom-right corner of the computational domain. The particles are generated randomly at the bottom-right corner of the computational domain as well. The water column collapses when releasing the water tank, and the remaining height (shown in the inset of Figure 6.9) of the water column in the z direction and the front position in the x direction are recorded to compare against the experimental results reported in the literature [132].

These essential parameters used in the 3D un-resolved CFD-DEM simulation are listed in Table 6.4. Additionally, three dimensionless quantities are defined in this section, namely, the dimensionless time $t^* = t\sqrt{\frac{2g}{a}}$, dimensionless remaining

Table 6.4: Parameters for 3D dam break simulations.

Parameter [Units]	Value
Liquid density [kg/m^3]	1000
Gas density [kg/m^3]	1
Liquid viscosity [Pa$\cdot$s]	1.0×10^{-3}
Gas viscosity [Pa$\cdot$s]	1.0×10^{-5}
Surface tension [N/m]	0.07
Size of fluid cell [m]	0.005
CFD time step [s]	5.0×10^{-5}
DEM time step [s]	1.0×10^{-5}
Restitution coefficient [-]	0.9
Friction coefficient [-]	0.3
Coupling interval [-]	5

height $Z^* = \frac{Z}{a}$ and dimensionless front position $X^* = \frac{X}{a}$ to compare the numerical results against the corresponding experimental measurements [132].

The evolution of the liquid column and movement of solid particles for four different stages are shown in Figure 6.10. The left column of Figure 6.10 demonstrates the experimental results at $t = 0.1$ s, 0.2 s, 0.3 s, and 0.4 s, respectively. The right column shows the corresponding numerical results. The color bar shows the particle velocity, and the surface in red indicates the iso-surface of the liquid volume fraction field α_l. For these four stages, numerical simulations are in good agreement with experimental validations. The solid phase moves slower than the front wave of the liquid phase during the early stage of the simulation, namely, $t < 0.2$ s. Then, the front wave of the liquid phase hits the left wall in the second stage of the simulation. The solid particles spread over the bottom wall under the influence of liquid-driven forces. In the third stage, the front wave of the liquid phase at the left wall increases gradually, and some solid particles are lifted over the bottom wall. Finally, the front wave of the liquid phase starts to decrease when it reaches the highest position at the left wall, as shown in the fourth stage of the simulation.

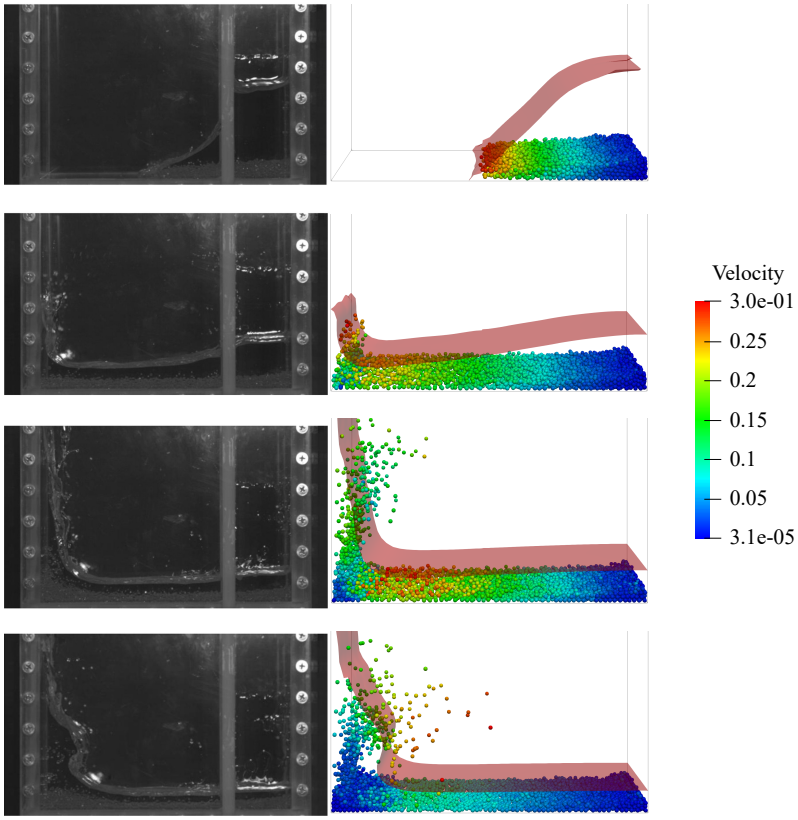


Figure 6.10: Comparison between experimental (left column) and numerical (right column) results at $t = 0.1$ s, 0.2 s, 0.3 s and 0.4 s, respectively. The color of the particles indicates the magnitude of particle velocity, and red indicates high velocity, while blue indicates low velocity. The 3D surface in red indicates the free surface of the liquid phase.

For the quantitative comparison, the normalized positions of the front wave in X (at the bottom wall) and Z (at the right wall) directions as shown in the inset of Figure 6.9 are recorded. The position of the front liquid wave is in good agreement with the experimental results reported in the literature [132] as shown in Figure 6.11a. A minor discrepancy is found in the Z direction and at a late stage

in the X direction, namely, the numerical model overestimates the movement of the front wave. The reason is that a gate in the experimental study to release the

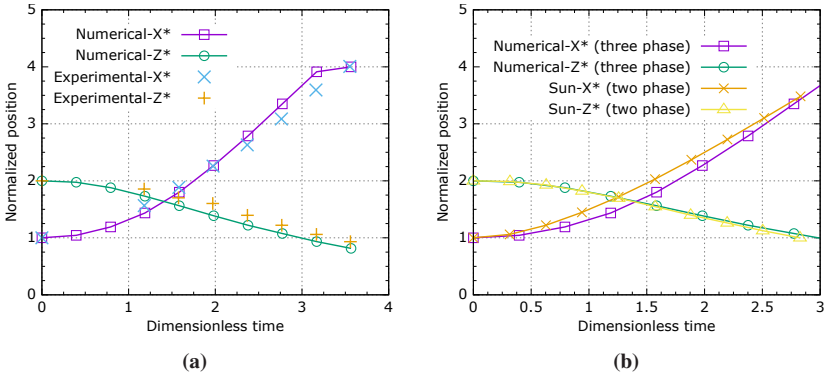


Figure 6.11: Validation of the un-resolved CFD-DEM model: (a) comparison between numerical and experimental data, (b) comparison between three-phase (gas-solid-liquid) and two-phase (gas-liquid) simulations (data of two-phase simulations are adopted from [132]).

water tank before starting the experiment is missing in the numerical simulation. Instead of such a gate, sudden release is assumed in the numerical simulation for simplicity, lead to a faster decrease at the right wall (Z direction) and faster movement at the bottom wall (X direction) of the liquid wave.

Furthermore, two-phase simulations of a 3D dam break simulation without particles are adopted from the literature [132] for comparison to the three phase simulations calculated here. This comparison is to understand the influence of the solid phase on the motion of the liquid phase. A good agreement is found in the Z direction, however, an discrepancy is found in the X direction, as shown in Figure 6.11b. The reason behind the discrepancy is that the movement of solid particles on the bottom wall in the multiphase system hinders the movement of the front wave compared to the two-phase system without solid particles.

6.4 Summary

In this chapter, an improved, un-resolved CFD-DEM solver **cfdemSolverVoFX** has been developed and implemented in the open-source code CFDEMcoupling-PUBLIC. The theory, equations, and numerical method for the un-resolved CFD-DEM approach are detailed in this chapter. The main contribution of this chapter to improve the un-resolved CFD-DEM model is as follows: The improved surface tension force model and the sharp free surface capturing approach developed in Chapter 3 have been incorporated into this un-resolved CFD-DEM model. An improved drag force model has been implemented to calculate the drag force for a very wide range of Reynolds numbers, accurately.

Numerical validations have been conducted to validate the un-resolved CFD-DEM model, e.g. single particle settling in viscous liquids and one particle settling from the gas to the liquid phase. Good agreement can be found in these numerical validations, demonstrating that the un-resolved CFD-DEM model extended in this chapter can accurately calculate solid-particle interaction forces. A numerical demonstration presented in this chapter is modelling a classical 3D dam break simulation widely adopted in the literature. Comparison between numerical and experimental results proves that the un-resolved CFD-DEM approach can predict the solid-particle interactions for a multiphase free surface flow with around 4000 solid particles well.

Incorporating the evaporation models for the liquid phase discussed in Chapter 4 into the un-resolved CFD-DEM model is one of the future topics. Computational modelling of multiphase gas-liquid-solid systems with the evaporation of the liquid phase and also involving many micro-sized particles is fairly challenging. However, developing such a numerical model is of great significance in many engineering applications. Potential applications of such an un-resolved CFD-DEM method with evaporation are, e.g. direct inkjet printing, thin film evaporation of suspensions and spray coating.

7 Summary and outlook

The main goal of the present thesis is to develop an efficient CFD-DEM framework to computationally model the surface-tension-dominant gas-liquid-solid multiphase free surface flow with and without evaporation of the liquid phase. In the present thesis, two improved CFD-DEM coupling methods, namely, resolved and un-resolved CFD-DEM methods, have been developed and validated. The theory, numerical method and numerical issues and details have been discussed in this thesis. Some numerical benchmark cases are conducted to demonstrate the applications of CFD-DEM methods. The main contributions of this thesis are listed below:

- **The extended DEM**

  Non-contacting surface forces, e.g. Van der Waals, electrostatic and DLVO forces, and one random force called Brownian force, have been detailed in the second chapter of this thesis. These force models have been implemented into the open-source DEM code **LIGGGHTS-PUBLIC**. Thus, an in-house extended DEM library named **LIGGGHTS-XIA** has been developed for modelling granular materials in micro-scale or nano-scale. Some essential numerical validations have been conducted. It proves that the extended DEM can accurately predict these non-contacting forces (Van der Waals, electrostatic and DLVO forces) and the random force.

- **The i-CLSVoF framework with and without evaporation**

  The i-CLSVoF framework has been developed in this thesis to model the surface-tension-dominant flow with and without evaporation. In this i-CLSVoF framework, the VoF approach is coupled to the LS method for

capturing the sharp interface and improving calculations of the interface curvature. Additional numerical filtering steps have been incorporated to suppress un-physical velocities, which are challenging to reduce, especially, in micro-scale. The i-CLSVoF framework is implemented into the open-source C++ library OpenFOAM. Numerical demonstrations, e.g. un-physical velocities around spherical droplets and sessile droplets sitting on a substrate with prescribed contact angles, demonstrate that the i-CLSVoF framework has good performance in capturing a sharp free surface and in suppressing un-physical velocities for a large density ratio, i.e. around 1000 between the liquid and gas phases.

Three different evaporation models have been implemented into the i-CLSVoF framework for modelling phase change of the liquid phase. A velocity-potential-based approach has been implemented to reconstruct a divergence-free velocity field for advecting the free surface when there is a phase change. Extensive numerical validations prove that these evaporation models can predict the evaporation rate accurately. A Robin boundary condition has been incorporated to model the evaporation of sessile droplets with contact line pinning. The numerical benchmark case shows that an internal capillary flow from the droplet centre to its edge can be seen when fixing the contact line during the evaporation process of a sessile droplet.

- **The resolved CFD-DEM method**

  An extended resolved CFD-DEM method has been developed to resolve flow fields around solid particles accurately. One contribution to the resolved CFD-DEM method is incorporating the aforementioned evaporation models into the resolved CFD-DEM method. The other one is modelling the capillary interactions for particles floating around the free surface with a newly extended capillary force model. Numerical validations have been conducted to validate the resolved CFD-DEM method. The results prove that the resolved CFD-DEM method developed in this thesis can predict

particle-fluid interactions accurately. Some additional numerical demonstrations have been investigated to show the performance of the resolved CFD-DEM method in modelling the evaporation-induced particle transport and accumulations of solid particles.

- **The un-resolved CFD-DEM method**

  For modelling large-scale numerical simulations with many particles, the un-resolved CFD-DEM method has been extended in this thesis. The i-CSLVoF framework has been adopted in this part to capture the free surface and to calculate the surface tension force. An improved drag force model, which is applicable to a very wide range of Reynolds numbers, has been implemented. The single particle settling benchmark case has been adopted to validate this newly incorporated drag force model. It proves that the new drag force model accurately predicts the particle-fluid interaction force. An additional 3D dam break benchmark case shows the performance of the un-resolved CFD-DEM method developed in this thesis in realizing the so-called four-way coupling between the solid and liquid phases. The 3D dam break simulation shows good agreement with the corresponding experimental studies. It proves that the un-resolved CFD-DEM method extended in this thesis has good performance in capturing complex particle-fluid interactions with reasonable numerical accuracy and lower computational cost.

Although the numerical models and framework developed in the present thesis demonstrate good performance, there are some numerical limitations which can be fixed in future work. Accordingly, some future work and directions are addressed below:

- **The geometric VoF for the free surface capturing**

  The algebraic VoF approach is adopted in this thesis. As an alternative, a relatively new interface capturing method, i.e. the geometric VoF method,

is suggested for capturing the free surface with explicit interface reconstruction and without interface diffusion. Some promising results produced with the geometric VoF approach can be found in the literature [106, 203, 204]. Consequently, incorporation of the geometric VoF into the i-CLSVoF framework can be further work. Un-physical spurious velocities become large when using adaptive mesh refinement with the i-CLSVoF framework, especially when the contact angle of a droplet is larger than 90°. Further numerical developments and implementations of improved numerical methods could be conducted to fix the numerical issues of this part.

- **CFD-DEM simulations with superquadric particles**

  In this thesis, ideally spherical particles are used for the CFD-DEM simulations; however, the majority of particles are non-spherical in the real world. The superquadric particles were implemented in LIGGGHTS by Podlozhnyuk et al. [172]. Accordingly, developing a CFD-DEM framework accounting for superquadric particles and incorporating free-surface flow, surface tension, and phase change is significant for understanding many other engineering applications.

- **The un-resolved CFD-DEM method with evaporation**

  The un-resolved CFD-DEM framework discussed in Chapter 6 only accounts for particle-fluid interactions without the evaporation or phase change of the liquid phase. Further numerical implementations can be conducted to incorporate the evaporation models detailed in Chapter 4 into the un-resolved CFD-DEM framework. Direct inkjet printing is one promising additive manufacturing technique, where many micro-sized solid particles are involved in this application. The evaporation of suspension droplets with and without contact line pinning leads to different particle deposition patterns. Developing such an un-resolved CFD-DEM framework with efficient evaporation models helps further developments of the direct inkjet printing technology.

- **The hybrid resolved and un-resolved CFD-DEM method**

  Combining resolved and un-resolved CFD-DEM methods has potential applications in modelling granular systems with large size ratios. The basic idea is to calculate the flow around larger particles with the resolved CFD-DEM method while to simulate the flow around smaller particles with the un-resolved CFD-DEM method to save computational cost. Adaptive mesh refinement is needed to refine the mesh around the big particles. Developing such a hybrid numerical model is challenging but of great importance to computationally model some engineering applications with lower computational cost and better numerical accuracy.

Bibliography

[1] B. Derby, "Inkjet printing ceramics: From drops to solid," *Journal of the European Ceramic Society*, vol. 31, no. 14, pp. 2543–2550, 2011.

[2] D. Lohse, "Fundamental fluid dynamics challenges in inkjet printing," *Annual Review of Fluid Mechanics*, vol. 54, pp. 349–382, 2022.

[3] B. Derby, "Printing and prototyping of tissues and scaffolds," *Science*, vol. 338, no. 6109, pp. 921–926, 2012.

[4] M. Mikolajek, A. Friederich, C. Kohler, M. Rosen, A. Rathjen, K. Krüger, and J. R. Binder, "Direct inkjet printing of dielectric ceramic/polymer composite thick films," *Advanced Engineering Materials*, vol. 17, no. 9, pp. 1294–1301, 2015.

[5] M. Mikolajek, T. Reinheimer, M. Muth, P. Hohwieler, M. J. Hoffmann, and J. R. Binder, "Control of the surface morphology of ceramic/polymer composite inks for inkjet printing," *Advanced Engineering Materials*, vol. 20, no. 9, p. 1800318, 2018.

[6] B. Derby, "Inkjet printing of functional and structural materials: fluid property requirements, feature stability, and resolution," *Annual Review of Materials Research*, vol. 40, pp. 395–414, 2010.

[7] R. D. Deegan, O. Bakajin, T. F. Dupont, G. Huber, S. R. Nagel, and T. A. Witten, "Capillary flow as the cause of ring stains from dried liquid drops," *Nature*, vol. 389, no. 6653, pp. 827–829, 1997.

[8] R. D. Deegan, "Pattern formation in drying drops," *Physical Review E*, vol. 61, no. 1, p. 475, 2000.

[9] A. F. Routh, "Drying of thin colloidal films," *Reports on Progress in Physics*, vol. 76, no. 4, p. 046603, 2013.

[10] R. D. Deegan, O. Bakajin, T. F. Dupont, G. Huber, S. R. Nagel, and T. A. Witten, "Contact line deposits in an evaporating drop," *Physical Review E*, vol. 62, no. 1, p. 756, 2000.

[11] H. Hu and R. G. Larson, "Evaporation of a sessile droplet on a substrate," *The Journal of Physical Chemistry B*, vol. 106, no. 6, pp. 1334–1344, 2002.

[12] X. Shen, C.-M. Ho, and T.-S. Wong, "Minimal size of coffee ring structure," *The Journal of Physical Chemistry B*, vol. 114, no. 16, pp. 5269–5274, 2010.

[13] M. J. Hertaeg, C. Rees-Zimmerman, R. F. Tabor, A. F. Routh, and G. Garnier, "Predicting coffee ring formation upon drying in droplets of particle suspensions," *Journal of Colloid and Interface Science*, vol. 591, pp. 52–57, 2021.

[14] A. D. Eales, A. F. Routh, N. Dartnell, and G. Simon, "Evaporation of pinned droplets containing polymer–an examination of the important groups controlling final shape," *AIChE Journal*, vol. 61, no. 5, pp. 1759–1767, 2015.

[15] J. Wilkinson, C. Tam, A. Askounis, and S. Qi, "Suppression of the coffee-ring effect by tailoring the viscosity of pharmaceutical sessile drops," *Colloids and Surfaces A: Physicochemical and Engineering Aspects*, vol. 614, p. 126144, 2021.

[16] P. J. Yunker, T. Still, M. A. Lohr, and A. Yodh, "Suppression of the coffee-ring effect by shape-dependent capillary interactions," *Nature*, vol. 476, no. 7360, pp. 308–311, 2011.

[17] P. J. Yunker, M. A. Lohr, T. Still, A. Borodin, D. J. Durian, and A. G. Yodh, "Effects of particle shape on growth dynamics at edges of evaporating

drops of colloidal suspensions," *Physical Review Letters*, vol. 110, no. 3, p. 035501, 2013.

[18] A. K. Thokchom, S. K. Majumder, and A. Singh, "Internal fluid motion and particle transport in externally heated sessile droplets," *AIChE Journal*, vol. 62, no. 4, pp. 1308–1321, 2016.

[19] M. Majumder, C. S. Rendall, J. A. Eukel, J. Y. Wang, N. Behabtu, C. L. Pint, T. Y. Liu, A. W. Orbaek, F. Mirri, J. Nam *et al.*, "Overcoming the "coffee-stain" effect by compositional marangoni-flow-assisted drop-drying," *The Journal of Physical Chemistry B*, vol. 116, no. 22, pp. 6536–6542, 2012.

[20] J. Farhadi and V. Bazargan, "Marangoni flow and surfactant transport in evaporating sessile droplets: A lattice Boltzmann study," *Physics of Fluids*, vol. 34, no. 3, p. 032115, 2022.

[21] A. W. Wray, D. T. Papageorgiou, R. V. Craster, K. Sefiane, and O. K. Matar, "Electrostatic suppression of the "coffee stain effect"," *Langmuir*, vol. 30, no. 20, pp. 5849–5858, 2014.

[22] H. Jin, J. Qian, L. Zhou, J. Yuan, H. Huang, Y. Wang, W. M. Tang, and H. L. W. Chan, "Suppressing the coffee-ring effect in semitransparent MnO2 film for a high-performance solar-powered energy storage window," *ACS Applied Materials & Interfaces*, vol. 8, no. 14, pp. 9088–9096, 2016.

[23] D. Parthasarathy, S. Chandragiri, S. P. Thampi, P. Ravindran, and M. G. Basavaraj, "An experimental and theoretical study of the inward particle drift in contact line deposits," *Soft Matter*, vol. 18, no. 12, pp. 2414–2421, 2022.

[24] P. Sáenz, A. Wray, Z. Che, O. Matar, P. Valluri, J. Kim, and K. Sefiane, "Dynamics and universal scaling law in geometrically-controlled sessile drop evaporation," *Nature Communications*, vol. 8, no. 1, pp. 1–9, 2017.

[25] M. R. Moore, D. Vella, and J. M. Oliver, "The nascent coffee ring with arbitrary droplet contact set: an asymptotic analysis," *Journal of Fluid Mechanics*, vol. 940, 2022.

[26] B. Derby, "Additive manufacture of ceramics components by inkjet printing," *Engineering*, vol. 1, no. 1, pp. 113–123, 2015.

[27] B. H. Wu and C. Chung, "Modeling and simulation of solid-containing droplet drying and different-structure particle formation," *International Journal of Heat and Mass Transfer*, vol. 152, p. 119469, 2020.

[28] C. N. Kaplan and L. Mahadevan, "Evaporation-driven ring and film deposition from colloidal droplets," *Journal of Fluid Mechanics*, vol. 781, 2015.

[29] X. Du and R. Deegan, "Ring formation on an inclined surface," *Journal of Fluid Mechanics*, vol. 775, 2015.

[30] T. K. Pradhan and P. K. Panigrahi, "Deposition pattern of interacting droplets," *Colloids and Surfaces A: Physicochemical and Engineering Aspects*, vol. 482, pp. 562–567, 2015.

[31] V. Charitatos, T. Pham, and S. Kumar, "Droplet evaporation on inclined substrates," *Physical Review Fluids*, vol. 6, no. 8, p. 084001, 2021.

[32] A. G. Marin, H. Gelderblom, D. Lohse, and J. H. Snoeijer, "Order-to-disorder transition in ring-shaped colloidal stains," *Physical Review Letters*, vol. 107, no. 8, p. 085502, 2011.

[33] T. Breinlinger and T. Kraft, "A simple method for simulating the coffee stain effect," *Powder Technology*, vol. 256, pp. 279–284, 2014.

[34] P. A. Zolotarev and K. S. Kolegov, "Monte Carlo simulation of particle size separation in evaporating bi-dispersed colloidal droplets on hydrophilic substrates," *Physics of Fluids*, vol. 34, no. 1, p. 017107, 2022.

[35] C. Zhang, H. Zhang, Y. Zhao, and C. Yang, "An immersed boundary-lattice Boltzmann model for simulation of deposited particle patterns in an evaporating sessile droplet with dispersed particles," *International Journal of Heat and Mass Transfer*, vol. 181, p. 121905, 2021.

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

[37] Z. Xie, S. Wang, and Y. Shen, "CFD-DEM modelling of the migration of fines in suspension flow through a solid packed bed," *Chemical Engineering Science*, vol. 231, p. 116261, 2021.

[38] S. Golshan, R. Sotudeh-Gharebagh, R. Zarghami, N. Mostoufi, B. Blais, and J. Kuipers, "Review and implementation of CFD-DEM applied to chemical process systems," *Chemical Engineering Science*, vol. 221, p. 115646, 2020.

[39] E. A. Schnorr Filho, N. C. Lima, and E. M. Franklin, "Resolved CFD-DEM simulations of the hydraulic conveying of coarse grains through a very-narrow elbow," *Powder Technology*, vol. 395, pp. 811–821, 2022.

[40] Z. Zhou, S. Kuang, K. Chu, and A. Yu, "Discrete particle simulation of particle–fluid flow: model formulations and their applicability," *Journal of Fluid Mechanics*, vol. 661, pp. 482–510, 2010.

[41] Z. Wang, Y. Teng, and M. Liu, "A semi-resolved CFD-DEM approach for particulate flows with kernel based approximation and Hilbert curve based searching strategy," *Journal of Computational Physics*, vol. 384, pp. 151–169, 2019.

[42] K. Cheng, Y. Wang, and Q. Yang, "A semi-resolved CFD-DEM model for seepage-induced fine particle migration in gap-graded soils," *Computers and Geotechnics*, vol. 100, pp. 30–51, 2018.

[43] E. M. Smuts, "A methodology for coupled CFD-DEM modeling of particulate suspension rheology," Dissertation, University of Cape Town, 2015.

[44] J. Zhao and T. Shan, "Coupled CFD-DEM simulation of fluid–particle interaction in geomechanics," *Powder Technology*, vol. 239, pp. 248–258, 2013.

[45] T. Yu and J. Zhao, "Semi-coupled resolved CFD-DEM simulation of powder-based selective laser melting for additive manufacturing," *Computer Methods in Applied Mechanics and Engineering*, vol. 377, p. 113707, 2021.

[46] A. N. Balachandran Nair, S. Pirker, and M. Saeedipour, "Resolved CFD-DEM simulation of blood flow with a reduced-order RBC model," *Computational Particle Mechanics*, pp. 1–16, 2021.

[47] S. Tanguy, T. Ménard, and A. Berlemont, "A level set method for vaporizing two-phase flows," *Journal of Computational Physics*, vol. 221, no. 2, pp. 837–853, 2007.

[48] N. Scapin, P. Costa, and L. Brandt, "A volume-of-fluid method for interface-resolved simulations of phase-changing two-fluid flows," *Journal of Computational Physics*, vol. 407, p. 109251, 2020.

[49] S. Hardt and F. Wondra, "Evaporation model for interfacial flows based on a continuum-field representation of the source terms," *Journal of Computational Physics*, vol. 227, no. 11, pp. 5871–5895, 2008.

[50] M. Nabil and A. S. Rattner, "interThermalPhaseChangeFoam—A framework for two-phase flow simulations with thermally driven phase change," *SoftwareX*, vol. 5, pp. 216–226, 2016.

[51] Y. Sato and B. Ničeno, "A sharp-interface phase change model for a mass-conservative interface tracking method," *Journal of Computational Physics*, vol. 249, pp. 127–161, 2013.

[52] M. Irfan and M. Muradoglu, "A front tracking method for direct numerical simulation of evaporation process in a multiphase system," *Journal of Computational Physics*, vol. 337, pp. 132–153, 2017.

[53] J. Palmore Jr and O. Desjardins, "A volume of fluid framework for interface-resolved simulations of vaporizing liquid-gas flows," *Journal of Computational Physics*, vol. 399, p. 108954, 2019.

[54] C. Kunkelmann, "Numerical modeling and investigation of boiling phenomena," Dissertation, TU Darmstadt, 2011.

[55] Y. Wang and V. Yang, "Vaporization of liquid droplet with large deformation and high mass transfer rate, I: Constant-density, constant-property case," *Journal of Computational Physics*, vol. 392, pp. 56–70, 2019.

[56] B. M. Weon and J. H. Je, "Self-pinning by colloids confined at a contact line," *Physical Review Letters*, vol. 110, no. 2, p. 028303, 2013.

[57] X. Yang, C. Y. Li, and Y. Sun, "From multi-ring to spider web and radial spoke: competition between the receding contact line and particle deposition in a drying colloidal drop," *Soft Matter*, vol. 10, no. 25, pp. 4458–4463, 2014.

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

[59] R. Yang, R. Zou, and A. Yu, "Computer simulation of the packing of fine particles," *Physical Review E*, vol. 62, no. 3, p. 3900, 2000.

[60] E. E. Michaelides, "Brownian movement and thermophoresis of nanoparticles in liquids," *International Journal of Heat and Mass Transfer*, vol. 81, pp. 179–187, 2015.

[61] Z. Peng, E. Doroodchi, and G. Evans, "DEM simulation of aggregation of suspended nanoparticles," *Powder Technology*, vol. 204, no. 1, pp. 91–102, 2010.

[62] L. Wang, K. Dong, C. Wang, R. Zou, Z. Zhou, and A. Yu, "Computer simulation of the packing of nanoparticles," *Powder Technology*, p. 117317, 2022.

[63] S. Mandal, M. Nicolas, and O. Pouliquen, "Insights into the rheology of cohesive granular media," *Proceedings of the National Academy of Sciences*, vol. 117, no. 15, pp. 8366–8373, 2020.

[64] N. Kovalchuk and V. Starov, "Aggregation in colloidal suspensions: Effect of colloidal forces and hydrodynamic interactions," *Advances in Colloid and Interface Science*, vol. 179, pp. 99–106, 2012.

[65] C. W. Hong, "New concept for simulating particle packing in colloidal forming processes," *Journal of the American Ceramic Society*, vol. 80, no. 10, pp. 2517–2524, 1997.

[66] M. m. Kim and A. L. Zydney, "Effect of electrostatic, hydrodynamic, and Brownian forces on particle trajectories and sieving in normal flow filtration," *Journal of Colloid and Interface Science*, vol. 269, no. 2, pp. 425–431, 2004.

[67] A. Matavž, U. Uršič, J. Močivnik, D. Richter, M. Humar, S. Čopar, B. Malič, and V. Bobnar, "From coffee stains to uniform deposits: Significance of the contact-line mobility," *Journal of Colloid and Interface Science*, vol. 608, pp. 1718–1727, 2022.

[68] J. Freed-Brown, "Evaporative deposition in receding drops," *Soft Matter*, vol. 10, no. 47, pp. 9506–9510, 2014.

[69] Y. F. Li, Y. J. Sheng, and H. K. Tsao, "Evaporation stains: suppressing the coffee-ring effect by contact angle hysteresis," *Langmuir*, vol. 29, no. 25, pp. 7802–7811, 2013.

[70] S. P. Thampi and M. G. Basavaraj, "Beyond coffee rings: drying drops of colloidal dispersions on inclined substrates," *ACS Omega*, vol. 5, no. 20, pp. 11 262–11 270, 2020.

[71] P. L. Kumar, S. P. Thampi, and M. G. Basavaraj, "Patterns from drops drying on inclined substrates," *Soft Matter*, vol. 17, no. 33, pp. 7670–7681, 2021.

[72] X. Man and M. Doi, "Ring to mountain transition in deposition pattern of drying droplets," *Physical Review Letters*, vol. 116, no. 6, p. 066101, 2016.

[73] Y. Li, C. Lv, Z. Li, D. Quéré, and Q. Zheng, "From coffee rings to coffee eyes," *Soft Matter*, vol. 11, no. 23, pp. 4669–4673, 2015.

[74] B. M. Weon and J. H. Je, "Fingering inside the coffee ring," *Physical Review E*, vol. 87, no. 1, p. 013003, 2013.

[75] F. Du, L. Zhang, and W. Shen, "Controllable dried patterns of colloidal drops," *Journal of Colloid and Interface Science*, vol. 606, pp. 758–767, 2022.

[76] J. Gerber, T. M. Schutzius, and D. Poulikakos, "Patterning of colloidal droplet deposits on soft materials," *Journal of Fluid Mechanics*, vol. 907, 2021.

[77] J. Pyeon and H. Kim, "Controlling uniform patterns by evaporation of multi-component liquid droplets in a confined geometry," *Soft Matter*, vol. 17, no. 13, pp. 3578–3585, 2021.

[78] V. Ta, R. Carter, E. Esenturk, C. Connaughton, T. J. Wasley, J. Li, R. W. Kay, J. Stringer, P. Smith, and J. D. Shephard, "Dynamically controlled deposition of colloidal nanoparticle suspension in evaporating drops using laser radiation," *Soft Matter*, vol. 12, no. 20, pp. 4530–4536, 2016.

[79] M. Anyfantakis and D. Baigl, "Manipulating the coffee-ring effect: interactions at work," *ChemPhysChem*, vol. 16, no. 13, pp. 2726–2734, 2015.

[80] C. Kloss, C. Goniva, A. Hager, S. Amberger, and S. Pirker, "Models, algorithms and validation for opensource DEM and CFD–DEM," *Progress*

in Computational Fluid Dynamics, an International Journal, vol. 12, no. 2-3, pp. 140–152, 2012.

[81] H. Jasak, "OpenFOAM: open source CFD in research and industry," *International Journal of Naval Architecture and Ocean Engineering*, vol. 1, no. 2, pp. 89–94, 2009.

[82] H. Xia, Z. Li, and X. Tong, "Modelling continuous materials using discrete element modelling: investigations on the effect of particle packing," *Computational Particle Mechanics*, vol. 6, no. 4, pp. 823–836, 2019.

[83] A. P. Thompson, H. M. Aktulga, R. Berger, D. S. Bolintineanu, W. M. Brown, P. S. Crozier, P. J. in't Veld, A. Kohlmeyer, S. G. Moore, T. D. Nguyen *et al.*, "LAMMPS-a flexible simulation tool for particle-based materials modeling at the atomic, meso, and continuum scales," *Computer Physics Communications*, vol. 271, p. 108171, 2022.

[84] C. Goniva, C. Kloss, N. G. Deen, J. A. Kuipers, and S. Pirker, "Influence of rolling friction on single spout fluidized bed simulation," *Particuology*, vol. 10, no. 5, pp. 582–591, 2012.

[85] B. Blais, M. Lassaigne, C. Goniva, L. Fradette, and F. Bertrand, "Development of an unresolved CFD–DEM model for the flow of viscous suspensions and its application to solid–liquid mixing," *Journal of Computational Physics*, vol. 318, pp. 201–221, 2016.

[86] H. R. Norouzi, R. Zarghami, R. Sotudeh-Gharebagh, and N. Mostoufi, *Coupled CFD-DEM modeling: formulation, implementation and application to multiphase flows*. John Wiley & Sons, Hoboken, 2016.

[87] L. Verlet, "Computer "experiments" on classical fluids. I. Thermodynamical properties of lennard-jones molecules," *Physical Review*, vol. 159, no. 1, p. 98, 1967.

[88] Y. Li, Y. Xu, and C. Thornton, "A comparison of discrete element simulations and experiments for 'sandpiles' composed of spherical particles," *Powder Technology*, vol. 160, no. 3, pp. 219–228, 2005.

[89] H. C. Hamaker, "The London—van der Waals attraction between spherical particles," *Physica*, vol. 4, no. 10, pp. 1058–1072, 1937.

[90] H. Abbasfard, G. Evans, and R. Moreno-Atanasio, "Effect of van der Waals force cut-off distance on adhesive collision parameters in DEM simulation," *Powder Technology*, vol. 299, pp. 9–18, 2016.

[91] M. Götzinger and W. Peukert, "Dispersive forces of particle–surface interactions: direct AFM measurements and modelling," *Powder Technology*, vol. 130, no. 1-3, pp. 102–109, 2003.

[92] D. Computing, "LIGGGHTS-DEM simulation engine," 2015. [Online]. Retrieved on May 20th, 2023, from: https://github.com/CFDEMproject/LIGGGHTS-PUBLIC

[93] C. W. Hong, "From long-range interaction to solid-body contact between colloidal surfaces during forming," *Journal of the European Ceramic Society*, vol. 18, no. 14, pp. 2159–2167, 1998.

[94] H. J. Butt, K. Graf, and M. Kappl, *Physics and chemistry of interfaces*. John Wiley & Sons, Hoboken, 2013.

[95] A. Einstein, "Über die von der molekularkinetischen Theorie der Wärme geforderte Bewegung von in ruhenden Flüssigkeiten suspendierten Teilchen," *Annalen der Physik*, vol. 322, no. 8, pp. 549–560, 1905.

[96] S. Popinet, "Numerical models of surface tension," *Annual Review of Fluid Mechanics*, vol. 50, pp. 49–75, 2018.

[97] N. Takada, J. Matsumoto, S. Matsumoto, and K. Kurihara, "Phase-field model-based simulation of two-phase fluid motion on partially wetted and textured solid surface," *Journal of Computational Science*, vol. 17, pp. 315–324, 2016.

[98] N. Ioannou, H. Liu, and Y. Zhang, "Droplet dynamics in confinement," *Journal of Computational Science*, vol. 17, pp. 463–474, 2016.

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

[100] S. Mohammadrezaei, M. Siavashi, and S. Asiaei, "Surface topography effects on dynamic behavior of water droplet over a micro-structured surface using an improved-VOF based lattice Boltzmann method," *Journal of Molecular Liquids*, p. 118509, 2022.

[101] L. Malan, A. Malan, S. Zaleski, and P. Rousseau, "A geometric VOF method for interface resolved phase change and conservative thermal energy advection," *Journal of Computational Physics*, vol. 426, p. 109920, 2021.

[102] W. Aniszewski, T. Arrufat, M. Crialesi-Esposito, S. Dabiri, D. Fuster, Y. Ling, J. Lu, L. Malan, S. Pal, R. Scardovelli *et al.*, "Parallel, robust, interface simulator (PARIS)," *Computer Physics Communications*, vol. 263, p. 107849, 2021.

[103] S. Popinet, "A quadtree-adaptive multigrid solver for the serre–green–naghdi equations," *Journal of Computational Physics*, vol. 302, pp. 336–358, 2015.

[104] J. Roenby, H. Bredmose, and H. Jasak, "A computational method for sharp interface advection," *Royal Society Open Science*, vol. 3, no. 11, p. 160405, 2016.

[105] D. Dai and A. Y. Tong, "Analytical interface reconstruction algorithms in the PLIC-VOF method for 3D polyhedral unstructured meshes," *International Journal for Numerical Methods in Fluids*, vol. 91, no. 5, pp. 213–227, 2019.

[106] H. Scheufler and J. Roenby, "Accurate and efficient surface reconstruction from volume fraction data on general meshes," *Journal of Computational Physics*, vol. 383, pp. 1–23, 2019.

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

[108] X. Yin, I. Zarikos, N. Karadimitriou, A. Raoof, and S. Hassanizadeh, "Direct simulations of two-phase flow experiments of different geometry complexities using Volume-of-Fluid (VOF) method," *Chemical Engineering Science*, vol. 195, pp. 820–827, 2019.

[109] F. Jamshidi, H. Heimel, M. Hasert, X. Cai, O. Deutschmann, H. Marschall, and M. Wörner, "On suitability of phase-field and algebraic volume-of-fluid OpenFOAM® solvers for gas–liquid microfluidic applications," *Computer Physics Communications*, vol. 236, pp. 72–85, 2019.

[110] J. Klostermann, K. Schaake, and R. Schwarze, "Numerical simulation of a single rising bubble by VOF with surface compression," *International Journal for Numerical Methods in Fluids*, vol. 71, no. 8, pp. 960–982, 2013.

[111] M. Aboukhedr, A. Georgoulas, M. Marengo, M. Gavaises, and K. Vogiatzaki, "Simulation of micro-flow dynamics at low capillary numbers using adaptive interface compression," *Computers & Fluids*, vol. 165, pp. 13–32, 2018.

[112] A. Albadawi, D. Donoghue, A. Robinson, D. Murray, and Y. Delauré, "Influence of surface tension implementation in volume of fluid and coupled volume of fluid with level set methods for bubble growth and detachment," *International Journal of Multiphase Flow*, vol. 53, pp. 11–28, 2013.

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

[114] M. Sussman, E. Fatemi, P. Smereka, and S. Osher, "An improved level set method for incompressible two-phase flows," *Computers & Fluids*, vol. 27, no. 5-6, pp. 663–680, 1998.

[115] S. Popinet, "An accurate adaptive solver for surface-tension-driven interfacial flows," *Journal of Computational Physics*, vol. 228, no. 16, pp. 5838–5866, 2009.

[116] A. Q. Raeini, M. J. Blunt, and B. Bijeljic, "Modelling two-phase flow in porous media at the pore scale using the volume-of-fluid method," *Journal of Computational Physics*, vol. 231, no. 17, pp. 5653–5668, 2012.

[117] K. Yokoi, "A density-scaled continuum surface force model within a balanced force formulation," *Journal of Computational Physics*, vol. 278, pp. 221–228, 2014.

[118] V. Inguva, E. Y. Kenig, and J. B. Perot, "A front-tracking method for two-phase flow simulation with no spurious currents," *Journal of Computational Physics*, vol. 456, p. 111006, 2022.

[119] S. S. Deshpande, L. Anumolu, and M. F. Trujillo, "Evaluating the performance of the two-phase flow solver interFoam," *Computational Science & Discovery*, vol. 5, no. 1, p. 014016, 2012.

[120] H. Rusche, "Computational fluid dynamics of dispersed two-phase flows at high phase fractions," Dissertation, Imperial College London, 2003.

[121] D. A. Hoang, V. van Steijn, L. M. Portela, M. T. Kreutzer, and C. R. Kleijn, "Benchmark numerical simulations of segmented two-phase flows in microchannels using the Volume of Fluid method," *Computers & Fluids*, vol. 86, pp. 28–36, 2013.

[122] M. Nekouei and S. A. Vanapalli, "Volume-of-fluid simulations in microfluidic T-junction devices: Influence of viscosity ratio on droplet size," *Physics of Fluids*, vol. 29, no. 3, p. 032007, 2017.

[123] T. E. Tezduyar, "Interface-tracking and interface-capturing techniques for finite element computation of moving boundaries and interfaces," *Computer Methods in Applied Mechanics and Engineering*, vol. 195, no. 23-24, pp. 2983–3000, 2006.

[124] M. Reder, P. W. Hoffrogge, D. Schneider, and B. Nestler, "A phase-field based model for coupling two-phase flow with the motion of immersed rigid bodies," *International Journal for Numerical Methods in Engineering*, vol. 123, no. 16, pp. 3757–3780, 2022.

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

[126] V. T. Nguyen and W. G. Park, "A volume-of-fluid (VOF) interface-sharpening method for two-phase incompressible flows," *Computers & Fluids*, vol.152, pp. 104–119, 2017.

[127] T. Yamamoto, Y. Okano, and S. Dost, "Validation of the S-CLSVOF method with the density-scaled balanced continuum surface force model in multiphase systems coupled with thermocapillary flows," *International Journal for Numerical Methods in Fluids*, vol. 83, no. 3, pp. 223–244, 2017.

[128] Z. Guo, D. F. Fletcher, and B. S. Haynes, "Implementation of a height function method to alleviate spurious currents in CFD modelling of annular flow in microchannels," *Applied Mathematical Modelling*, vol. 39, no. 16, pp. 4665–4686, 2015.

[129] H. Wen, C. H. Yu, and T. W. Sheu, "On the development of LS-assisted VOF method for incompressible interfacial flows," *Journal of Computational Physics*, vol. 406, p. 109188, 2020.

[130] N. K. Singh and B. Premachandran, "A coupled level set and volume of fluid method on unstructured grids for the direct numerical simulations of

two-phase flows including phase change," *International Journal of Heat and Mass Transfer*, vol. 122, pp. 182–203, 2018.

[131] C. Galusinski and P. Vigneaux, "On stability condition for bifluid flows with surface tension: Application to microfluidics," *Journal of Computational Physics*, vol. 227, no. 12, pp. 6140–6164, 2008.

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

[133] F. Garoosi and T. F. Mahdi, "New benchmark problems for validation and verification of incompressible multi-fluid flows based on the improved Volume-Of-Fluid (VOF) method," *Colloids and Surfaces A: Physicochemical and Engineering Aspects*, vol. 648, p. 129313, 2022.

[134] M. Shams, A. Q. Raeini, M. J. Blunt, and B. Bijeljic, "A numerical model of two-phase flow at the micro-scale using the volume-of-fluid method," *Journal of Computational Physics*, vol. 357, pp. 159–182, 2018.

[135] L. Chirco, R. Da Vià, and S. Manservisi, "VOF evaluation of the surface tension by using variational representation and Galerkin interpolation projection," *Journal of Computational Physics*, vol. 395, pp. 537–562, 2019.

[136] F. Denner, F. Evrard, and B. van Wachem, "Breaching the capillary time-step constraint using a coupled VOF method with implicit surface tension," *Journal of Computational Physics*, vol. 459, p. 111128, 2022.

[137] A. Kumar, "Adaptive mesh refinement in OpenFOAM-v1812 for 2-dimensional problems," 2019. [Online]. Retrieved on May 20th, 2023, from: https://github.com/krajit/dynamicRefine2DFvMesh

[138] H. Wijshoff, "Drop dynamics in the inkjet printing process," *Current Opinion in Colloid & Interface Science*, vol. 36, pp. 20–27, 2018.

[139] C. Christodoulou, E. Sorensen, A. Khair, S. García-Muñoz, and L. Mazzei, "A model for the fluid dynamic behavior of a film coating suspension during tablet coating," *Chemical Engineering Research and Design*, vol. 160, pp. 301–320, 2020.

[140] A. Saufi, A. Frassoldati, T. Faravelli, and A. Cuoci, "Interface-resolved simulation of the evaporation and combustion of a fuel droplet suspended in normal gravity," *Fuel*, vol. 287, p. 119413, 2021.

[141] D. Zang, S. Tarafdar, Y. Y. Tarasevich, M. D. Choudhury, and T. Dutta, "Evaporation of a droplet: From physics to applications," *Physics Reports*, vol. 804, pp. 1–56, 2019.

[142] K. Montazeri, H. Lee, and Y. Won, "Microscopic analysis of thin-film evaporation on spherical pore surfaces," *International Journal of Heat and Mass Transfer*, vol. 122, pp. 59–68, 2018.

[143] P. Das and H. Udaykumar, "A sharp-interface method for the simulation of shock-induced vaporization of droplets," *Journal of Computational Physics*, vol. 405, p. 109005, 2020.

[144] C. R. Kharangate and I. Mudawar, "Review of computational studies on boiling and condensation," *International Journal of Heat and Mass Transfer*, vol. 108, pp. 1164–1196, 2017.

[145] A. Hassanvand and S. Hashemabadi, "Direct numerical simulation of interphase mass transfer in gas–liquid multiphase systems," *International Communications in Heat and Mass Transfer*, vol. 38, no. 7, pp. 943–950, 2011.

[146] T. Young, "An essay on the cohesion of fluids," *Philosophical transactions of the royal society of London*, no. 95, pp. 65–87, 1805.

[147] D. Brutin, *Droplet wetting and evaporation: from pure to complex fluids*. Academic Press, London, 2015.

[148] H. J. Butt, J. Liu, K. Koynov, B. Straub, C. Hinduja, I. Roismann, R. Berger, X. Li, D. Vollmer, W. Steffen *et al.*, "Contact angle hysteresis," *Current Opinion in Colloid & Interface Science*, p. 101574, 2022.

[149] J. Zhang, F. Leroy, and F. Müller-Plathe, "Influence of contact-line curvature on the evaporation of nanodroplets from solid substrates," *Physical Review Letters*, vol. 113, no. 4, p. 046101, 2014.

[150] R. van Gaalen, C. Diddens, H. Wijshoff, and J. Kuerten, "The evaporation of surfactant-laden droplets: A comparison between contact line models," *Journal of Colloid and Interface Science*, vol. 579, pp. 888–897, 2020.

[151] N. Linder, "Numerical simulation of complex wetting," Dissertation, TU Darmstadt, 2015.

[152] N. Linder, A. Criscione, I. V. Roisman, H. Marschall, and C. Tropea, "3D computation of an incipient motion of a sessile drop on a rigid surface with contact angle hysteresis," *Theoretical and Computational Fluid Dynamics*, vol. 29, no. 5, pp. 373–390, 2015.

[153] D. Rettenmaier, "Numerical simulation of shear driven wetting," Dissertation, TU Darmstadt, 2019.

[154] E. Berberovic, "Investigation of free-surface flow associated with drop impact: numerical simulations and theoretical modeling," Dissertation, TU Darmstadt, 2010.

[155] J. H. Snoeijer and B. Andreotti, "Moving contact lines: Scales, regimes, and dynamical transitions," *Annual Review of Fluid Mechanics*, vol. 45, no. 1, pp. 269–292, 2013.

[156] I. Roisman, L. Opfer, C. Tropea, M. Raessi, J. Mostaghimi, and S. Chandra, "Drop impact onto a dry surface: Role of the dynamic contact angle," *Colloids and Surfaces A: Physicochemical and Engineering Aspects*, vol. 322, no. 1-3, pp. 183–191, 2008.

[157] S. Turns. “An introduction to combustion: concepts and applications,” *Mechanical Engineering Series. McGraw Hill*, New York, 2000.

[158] S. Kuang, M. Zhou, and A. Yu, “CFD-DEM modelling and simulation of pneumatic conveying: A review,” *Powder Technology*, vol. 365, pp. 186–207, 2020.

[159] Y. Zhang, J. Xu, Q. Chang, P. Zhao, J. Wang, and W. Ge, “Numerical simulation of fluidization: Driven by challenges,” *Powder Technology*, vol. 414, p. 118092, 2023.

[160] J. Fang, L. Wang, Y. Hong, and J. Zhao, “Influence of solid–fluid interaction on impact dynamics against rigid barrier: CFD–DEM modelling,” *Géotechnique*, vol. 72, no. 5, pp. 391–406, 2022.

[161] E. Koos, “Capillary suspensions: Particle networks formed through the capillary force,” *Current Opinion in Colloid & Interface Science*, vol. 19, no. 6, pp. 575–584, 2014.

[162] L. Lu, X. Gao, J. F. Dietiker, M. Shahnam, and W. A. Rogers, “MFiX based multi-scale CFD simulations of biomass fast pyrolysis: A review,” *Chemical Engineering Science*, vol. 248, p. 117131, 2022.

[163] H. Ma, L. Zhou, Z. Liu, M. Chen, X. Xia, and Y. Zhao, “A review of recent development for the CFD-DEM investigations of non-spherical particles,” *Powder Technology*, p. 117972, 2022.

[164] A. Hager, C. Kloss, S. Pirker, and C. Goniva, “Parallel resolved open source CFD-DEM: method, validation and application,” *The Journal of Computational Multiphase Flows*, vol. 6, no. 1, pp. 13–27, 2014.

[165] A. Podlozhnyuk, “Modelling superquadric particles in DEM and CFD-DEM: implementation, validation and application in an open-source framework,” Dissertation, Johannes Kepler University Linz, 2018.

[166] T. Zhao, “Investigation of landslide-induced debris flows by the DEM and CFD.” Dissertation, University of Oxford, 2014.

[167] A. Hager, "CFD-DEM on multiple scales: An extensive investigation of particle–fluid interactions," Dissertation, Johannes Kepler University Linz, 2014.

[168] T. Davydzenka, S. Fagbemi, and P. Tahmasebi, "Coupled fine-scale modeling of the wettability effects: Deformation and fracturing," *Physics of Fluids*, vol. 32, no. 8, p. 083308, 2020.

[169] G. T. Nguyen, E. L. Chan, T. Tsuji, T. Tanaka, and K. Washino, "Interface control for resolved CFD-DEM with capillary interactions," *Advanced Powder Technology*, vol. 32, no. 5, pp. 1410–1425, 2021.

[170] Z. Shen, G. Wang, D. Huang, and F. Jin, "A resolved CFD-DEM coupling model for modeling two-phase fluids interaction with irregularly shaped particles," *Journal of Computational Physics*, vol. 448, p. 110695, 2022.

[171] K. Washino, E. L. Chan, T. Tsujimoto, T. Tsuji, and T. Tanaka, "Development of resolved CFD–DEM coupling model for three-phase flows with non-spherical particles," *Chemical Engineering Science*, vol. 267, p. 118335, 2023.

[172] A. Podlozhnyuk, S. Pirker, and C. Kloss, "Efficient implementation of superquadric particles in discrete element method within an open-source framework," *Computational Particle Mechanics*, vol. 4, no. 1, pp. 101–118, 2017.

[173] P. Kieckhefen, T. Lichtenegger, S. Pietsch, S. Pirker, and S. Heinrich, "Simulation of spray coating in a spouted bed using recurrence CFD," *Particuology*, vol. 42, pp. 92–103, 2019.

[174] D. Computing, "CFDEMcoupling - Open Source CFD-DEM coupling," 2017. [Online]. Retrieved on May 20th, 2023, from: https://github.com/CFDEMproject/CFDEMcoupling-PUBLIC.git

[175] O. Foundation, "OpenFOAM-5.x," 2018. [Online]. Retrieved on May 20th, 2023, from: https://github.com/OpenFOAM/OpenFOAM-5.x

[176] H. Xia and M. Kamlah, "An improved coupled level set and volume of fluid (i-CLSVoF) framework for droplet evaporation," *arXiv preprint arXiv:2202.01019*, 2022.

[177] H. Xia and M. Kamlah, "Computational modelling of gas-liquid-solid multiphase free surface flow with and without evaporation," *arXiv preprint arXiv:2307.09313*, 2023.

[178] E. Gabriel, G. E. Fagg, G. Bosilca, T. Angskun, J. J. Dongarra, J. M. Squyres, V. Sahay, P. Kambadur, B. Barrett, A. Lumsdaine *et al.*, "Open MPI: Goals, concept, and design of a next generation MPI implementation," in *Recent Advances in Parallel Virtual Machine and Message Passing Interface: 11th European PVM/MPI Users' Group Meeting Budapest, Hungary, September 19-22, 2004. Proceedings 11*. Springer, 2004, pp. 97–104.

[179] N. A. Patankar, P. Singh, D. D. Joseph, R. Glowinski, and T. W. Pan, "A new formulation of the distributed Lagrange multiplier/fictitious domain method for particulate flows," *International Journal of Multiphase Flow*, vol. 26, no. 9, pp. 1509–1524, 2000.

[180] A. A. Shirgaonkar, M. A. MacIver, and N. A. Patankar, "A new mathematical formulation and fast algorithm for fully resolved simulation of self-propulsion," *Journal of Computational Physics*, vol. 228, no. 7, pp. 2366–2390, 2009.

[181] M. Darwish and F. Moukalled, *The finite volume method in computational fluid dynamics: an advanced introduction with OpenFOAM® and Matlab®*. Springer, Cham, 2016.

[182] M. Fujita, O. Koike, and Y. Yamaguchi, "Computation of capillary interactions among many particles at free surface," *Applied Physics Express*, vol. 6, no. 3, p. 036501, 2013.

[183] M. Fujita, O. Koike, and Y. Yamaguchi, "Direct simulation of drying colloidal suspension on substrate using immersed free surface model," *Journal of Computational Physics*, vol. 281, pp. 421–448, 2015.

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

[185] G. Tryggvason, R. Scardovelli, and S. Zaleski, *Direct numerical simulations of gas–liquid multiphase flows*. Cambridge University Press, Cambridge, 2011.

[186] K. Washino, H. Tan, M. Hounslow, and A. Salman, "A new capillary force model implemented in micro-scale CFD–DEM coupling for wet granulation," *Chemical Engineering Science*, vol. 93, pp. 197–205, 2013.

[187] M. Uhlmann, "An immersed boundary method with direct forcing for the simulation of particulate flows," *Journal of Computational Physics*, vol. 209, no. 2, pp. 448–476, 2005.

[188] M. Wu, B. Peters, T. Rosemann, and H. Kruggel-Emden, "A forcing fictitious domain method to simulate fluid-particle interaction of particles with super-quadric shape," *Powder Technology*, vol. 360, pp. 264–277, 2020.

[189] P. P. Brown and D. F. Lawler, "Sphere drag and settling velocity revisited," *Journal of Environmental Engineering*, vol. 129, no. 3, pp. 222–231, 2003.

[190] A. Ten Cate, C. Nieuwstad, J. Derksen, and H. Van den Akker, "Particle imaging velocimetry experiments and lattice-boltzmann simulations on a single sphere settling under gravity," *Physics of Fluids*, vol. 14, no. 11, pp. 4012–4025, 2002.

[191] Y. Mino, C. Tanaka, H. Tanaka, K. Nakaso, and K. Gotoh, "Numerical simulation of a drying colloidal suspension on a wettable substrate using the lattice Boltzmann method," *Chemical Engineering Science*, vol. 263, p. 118050, 2022.

[192] Y. Lu, J. Huang, and P. Zheng, "A CFD–DEM study of bubble dynamics in fluidized bed using flood fill method," *Chemical Engineering Journal*, vol. 274, pp. 123–131, 2015.

[193] M. Trofa, G. D'Avino, L. Sicignano, G. Tomaiuolo, F. Greco, P. L. Maffettone, and S. Guido, "CFD-DEM simulations of particulate fouling in microchannels," *Chemical Engineering Journal*, vol. 358, pp. 91–100, 2019.

[194] T. Davydzenka, S. Fagbemi, and P. Tahmasebi, "Wettability control on deformation: Coupled multiphase fluid and granular systems," *Physical Review E*, vol. 102, no. 1, p. 013301, 2020.

[195] K. Washino, E. L. Chan, T. Kaji, Y. Matsuno, and T. Tanaka, "On large scale CFD–DEM simulation for gas–liquid–solid three-phase flows," *Particuology*, vol. 59, pp. 2–15, 2021.

[196] Y. Mori and M. Sakai, "Development of a robust Eulerian–Lagrangian model for the simulation of an industrial solid–fluid system," *Chemical Engineering Journal*, vol. 406, p. 126841, 2021.

[197] T. El Geitani, S. Golshan, and B. Blais, "Toward high-order CFD-DEM: Development and validation," *Industrial & Engineering Chemistry Research*, vol. 62, no. 2, pp. 1141–1159, 2023.

[198] G. G. Stokes, *Mathematical and physical papers*. Cambridge university press, Cambridge, 1901.

[199] J. M. DallaValle, *Micromeritics: the technology of fine particles*. Pitman Pub. Corp., New York, 1948.

[200] C. Y. Wen, "Mechanics of fluidization," *Chem. Eng. Progress. Symposium Series*, vol. 62, 1966, pp. 100–111.

[201] A. Haider and O. Levenspiel, "Drag coefficient and terminal velocity of spherical and nonspherical particles," *Powder Technology*, vol. 58, no. 1, pp. 63–70, 1989.

[202] M. Vangö, “CFD-DEM modeling of multiphase fluid-granular systems and its application to blast furnace tapping,” Dissertation, Johannes Kepler University Linz, 2019.

[203] J. López and J. Hernández, “gVOF: An open-source package for unsplit geometric volume of fluid methods on arbitrary grids,” *Computer Physics Communications*, vol. 277, p. 108400, 2022.

[204] A. Esteban, J. López, P. Gómez, C. Zanzi, J. Roenby, and J. Hernández, “A comparative study of two open-source state-of-the-art geometric VOF methods,” *Computers & Fluids*, vol. 250, p. 105725, 2023.

List of Figures

1.1 Schematic diagram of the 3D direct inkjet printing process. 1
1.2 Ring-like material deposition patterns found in the structures fabricated by the direct inkjet printing [5]. 2
1.3 Evaporation flux around evaporating droplets: (a) single droplet, (b) two interacting droplets sitting near each other [30]. . . 6
1.4 Different deposition patterns and some crucial governing parameters [67]. 11
1.5 Key scientific problems behind a drying suspension droplet with insolvable particles. 13
2.1 The contact force versus particle overlap h of the conventional DEM approach ($h = \delta_n$ for the conventioanl DEM). . 18
2.2 The separation distance h between two interacting elements. The top and bottom images demonstrate the pairs of particle-particle and particle-wall, respectively. 22
2.3 Validations of the Van der Waals force model: (a) particle to particle, (b) particle to wall. 24
2.4 Validations of the electrostatic force model: (a) particle to particle, (b) particle to wall. 25
2.5 Force versus separation distance of the DLVO force model. 27
2.6 The transition from repulsive to attractive DLVO force in the colloidal system. 27
2.7 Validations of the DLVO force model: (a) particle to particle, (b) particle to wall. 28
2.8 Gaussian fitting for random numbers generated in x, y and z components, respectively. 30
2.9 The contact model of the extended DEM approach developed in this thesis. 31

3.1 (a) the VoF method, (b) the LS method. 38
3.2 Spurious currents around the interface (represents with the solid white line) with two different mesh sizes : (a) coarse mesh (cell number: 40×40), (b) fine mesh (cell number: 80×80). . 39
3.3 Flowchart for solving the Hamilton–Jacobi equation. 41
3.4 Three different Heaviside functions (H: the Heaviside step function, H_s: the symmetrical Heaviside function, H_ψ: the non-symmetrical Heaviside function). 43
3.5 Notations for two cells sharing one face. 48
3.6 Flowchart of the i-CLSVoF framework for surface-tension-dominant flow. 53
3.7 (a) Geometry and boundaries of the 2D dam break simulation, (b) Numerical simulations with 2D AMR (the white curve represents the free-surface). 54
3.8 Evolutions of the free-surface in the dam break simulations: (a) with the VoF method (t = 0.2 s), (b) with the i-CLSVoF method (t = 0.2 s), (c) with the VoF method (t = 0.5 s), (d) with the i-CLSVoF method (t = 0.5 s). 55
3.9 Evolution of the un-physical velocities with different surface tension force models. 56
3.10 Velocity vector contours at 0.001 s with different surface tension models (white circle represents the 0.5 iso-surface for α_l field): (a) VoF, (b) CLSVoF, (c) i-CLSVoF. 57
3.11 (a) Capillary pressure fields of droplets with three different surface tension force models (dotted purple line represents the analytical solution), (b) Relative errors for capillary pressure jump with different mesh resolutions. 58
3.12 Effect of the sharpening coefficient C_{pc} on the suppression of spurious velocity. 59
3.13 Spurious velocities around droplets ($\theta = 60^\circ$) with three different surface-tension force models (the white curve represents the interface): (a) with the VoF method, (b) with the CLSVoF method, (c) with the i-CLSVoF method. 61
3.14 Evolution of spurious velocities with different surface-tension models ($\theta = 60^\circ$). 61

3.15 Spurious velocities around droplets ($\theta = 90^\circ$) with three different surface-tension force models (the white curve represents the interface): (a) with the VoF method, (b) with the CLSVoF method, (c) with the i-CLSVoF method. 62
3.16 Evolution of spurious velocities with different surface-tension models ($\theta = 90^\circ$). 62
3.17 Spurious velocities around droplets ($\theta = 120^\circ$) with three different surface-tension force models (the white curve represents the interface): (a) with the VoF method, (b) with the CLSVoF method, (c) with the i-CLSVoF method. 63
3.18 Evolution of spurious velocities with different surface-tension models ($\theta = 120^\circ$). 64
4.1 The sub-domains for solving the velocity potential equation. 70
4.2 The evaporation coefficient χ_e calculated by Eqn. 4.12 versus the density ratio λ in the range up to 1000. 73
4.3 The mass source distributions for two different cases: (a) without smearing, (b) with smearing. 75
4.4 Evolution of the numerical residual for two different cases with and without smearing the mass source term. 75
4.5 Flowchart of solving governing equations with the i-CLSVoF framework when phase changes. 76
4.6 Contact line velocity versus contact angle, and the contact angle hysteresis. 78
4.7 Consequences behind the contact line pinning: (a) surface roughness, (b) particle accumulating around the triple contact line region. 79
4.8 The current contact angle of a droplet resting on a substrate. 80
4.9 Translational and rotational motion of the interface. 82
4.10 The schematic diagram of the numerical setup for 2D droplet evaporation. 85
4.11 Different velocities fields of an evaporating droplet (white line represents the interface); (a): the one-field velocity field, (b): the evaporation-induced Stefan flow velocity field, (c): the divergence-free velocity field, (d): the divergence of $\mathbf{U}_e$. 87

4.12 Vector contour of the one-field velocity field **U** (the white solid line represents the interface). 88
4.13 The mesh convergence study: (a) the global plot, (b) the local magnified plot. 89
4.14 The numerical error for different mesh resolutions. 90
4.15 Numerical validations of the interface capturing: (a) the global plot, (b) the mesh convergence study. 91
4.16 (a) The temperature distribution around the droplet, (b) the interface velocity field $\mathbf{U}_\Gamma$. 93
4.17 (a) The numerical validation for thermally driven evaporation model, (b) the temperature evolutions during the evaporation process. 93
4.18 Validations for four cases with different Stefan number *St* (solid lines represent the analytical solutions, and points represent the corresponding numerical data). 95
4.19 (a) The one-field velocity field **U** of droplet evaporation at room temperature, (b) The vapour mass fraction field of droplet evaporation at room temperature (white line represents the interface). 96
4.20 Validations for five cases with different evaporation temperatures (solid lines represent the analytical solutions, and points represent the corresponding numerical data). 97
4.21 Four different stages of the liquid volume fraction fields during the evaporation process. 98
4.22 Velocity fields of droplet evaporation with contact line pinning (white line represents the interface): (a) velocity in the gas phase, (b) velocity inside the droplet. 99
5.1 The schematic diagram of a surface-tension-dominant gas-liquid-solid multiphase system with free surface. 106
5.2 The schematic diagram of the resolved CFD-DEM method. Arrows represent the vectors of fluid velocity around particles, and circles in blue represent solid particles. 106
5.3 The schematic diagram of the capillary force model. The virtual free surface is bounded by the solid and dashed three-phase contact lines ∂s. 112

5.4 The void fraction field of a solid particle and the solid black curve represents the iso-surface with $\varepsilon_f = 0.5$. 116
5.5 The coupling algorithm for the resolved CFD-DEM approach. . . . 120
5.6 The schematic diagram of the numerical set-up for calculations of the drag coefficient. 122
5.7 Validations for calculations of the drag coefficient C_d. 125
5.8 The liquid velocity field of single particle settling simulation (Re = 11.6). 127
5.9 Validations of the single sphere settling in liquids: (a) dimensionless sphere height, (b) sphere settling velocity. 127
5.10 2D schematic diagram of the numerical set-up for two particles moving along a free surface with evaporation. 129
5.11 Vectors of the particle velocity are represented by the yellow arrows during the evaporation process: (a) pointing upwards, (b) pointing downwards, (c) pointing towards each other. 129
5.12 Numerical simulations of two particles moving along the free surface for three different specified contact angles. 131
5.13 The schematic diagram of the numerical setup for 2D droplet evaporation with suspended particles. 132
5.14 Snapshots of the numerical simulations of the four stages of particle deposition patterns inside an evaporating droplet. 134
5.15 The local packing fraction for the particle deposition pattern shown in Figure 5.14d. 135
6.1 The schematic diagram of the un-resolved CFD-DEM approach. Arrows represent the vectors of the local fluid velocity, and blue circles represent solid particles. 140
6.2 Comparison between corrections of drag coefficient against the experimental data: (a) global plot , (b) local plot. 147
6.3 The schematic diagram of some solid particles immersed in an internal fluid cell. 149
6.4 The schematic diagram of the *center* volume fraction model. 149
6.5 The coupling algorithm of the un-resolved CFD-DEM model. . . . 156
6.6 Validations of the single sphere settling in viscous liquids: (a) particle velocity, (b) particle position (solid curves: solution from Eqn. 6.39, dots: numerical solution from simulations). 159

6.7 The schematic diagram of a single particle settling from air to liquid. 160
6.8 Validations of the single sphere settling from the gas phase to the liquid phase: (a) particle velocity, (b) particle position (solid curves: analytical solution, dots: numerical solution). 160
6.9 The schematic diagram of the 3D dam break simulation with particles and the 3D surface in red indicates the initial free surface of the water column. The inset demonstrates a 2D sketch of the evolution of the free surface. 161
6.10 Comparison between experimental (left column) and numerical (right column) results at $t = 0.1$ s, 0.2 s, 0.3 s and 0.4 s, respectively. The color of the particles indicates the magnitude of particle velocity, and red indicates high velocity, while blue indicates low velocity. The 3D surface in red indicates the free surface of the liquid phase. 163
6.11 Validation of the un-resolved CFD-DEM model: (a) comparison between numerical and experimental data, (b) comparison between three-phase (gas-solid-liquid) and two-phase (gas-liquid) simulations (data of two-phase simulations are adopted from [132]). 164
A.1 Schematic diagram of a sphere enclosed by its voronoi cell. 214
A.2 (a) voronoi cells of the SCP, (b) local packing fraction of SCP. . . . 215
A.3 (a) voronoi cells of the HCP, (b) local packing fraction of HCP. . . 216

List of Tables

2.1 Key parameters and formulas for DEM simulations [84–86] 19
2.2 Key parameters for validations of the Van der Waals force model. . 23
2.3 Other parameters of the electrostatic force model. 25
2.4 Parameters for validations of the electrostatic force model. 26
2.5 The mean, standard deviation and relative error of standard deviation for the random numbers generated with the Brownian force model. 31
2.6 Comparison between LIGGGHTS-PUBLIC and LIGGGHTS-XIA (✓: available; × : not available yet; +✓: newly added) 32
3.1 Summary of numerical methods to track/capture free surface. . . . 34
3.2 Parameters for 2D dam break simulations 53
3.3 Parameters for 2D sessile droplet simulations 60
4.1 Physical properties for liquid and gas phases. 86
5.1 Summary and comparison among these applications of resolved CFD-DEM formulations. 105
5.2 Parameters for numerical simulations used to validate calculations of the drag coefficient. 123
5.3 Dynamic viscosities for these seven different cases. 124
5.4 Terminal velocity and relative error for these seven validation cases. 125
5.5 Parameters for the single particle settling simulations (data adopted from [190]). 126
5.6 Parameters for two particles moving along a free surface with evaporation. 130
5.7 Essential parameters for modelling particle transport inside an evaporating droplet. 133

5.8 Comparison between the standard cfdemSolverIB solver and the improved solver cfdemSolverVoFIB. 137
6.1 Summary and comparison among these applications of un-resolved CFD-DEM models. 141
6.2 Difference formulas for the corrections of the drag coefficient C_d . . 146
6.3 Parameters for these six different cases. 158
6.4 Parameters for 3D dam break simulations. 162
A.1 Derived quantities used in the CFD-DEM simulations with OpenFOAM & LIGGGHTS 235
A.2 Basic quantities used in OpenFOAM for different unit systems . . . 236

List of symbols and abbreviations

Abbreviations

2D	Two-dimensional
3D	Three-dimensional
AMR	Adaptive Mesh Refinement
CCA	Constant Contact Angle
CCR	Constant Contact Radius
CFD	Computational Fluid Dynamics
CLSVoF	Coupled Level Set and Volume of Fluid
CSF	Continuum Surface-tension Force
DEM	Discrete Element Method
DLVO	Derjaguin-Landau-Verwey-Overbeek
FVM	Finite Volume Method
i-CLSVoF	improved Coupled Level Set and Volume of Fluid
LBM	Lattice Boltzmann Method
LIGGGHTS	LAMMPS Improved for General Granular and Granular Heat Transfer Simulations
LS	Level Set
OpenFOAM	Open-source Field Operation and Manipulation
RH	Relative Humidity
s-CLSVoF	simple Coupled Level Set and Volume of Fluid

VoF	Volume of Fluid
Constants	
$\mathbf{g}$	Gravitational acceleration constant $[\mathrm{m/s^2}]$
c_p	Specific heat capacity $[\mathrm{J/(kg \cdot K)}]$
h_{ev}	Enthalpy/latent heat of evaporation $[\mathrm{J/kg}]$
k	Thermal conductivity $[\mathrm{W/(m \cdot K)}]$
T_{sat}	Saturation temperature $[\mathrm{K}]$
Greek symbols	
α	Volume fraction $[-]$
δ	Normal overlap $[\mathrm{m}]$
ε	Void fraction $[-]$
γ	Damping coefficient $[\mathrm{(N \cdot s)/m}]$
λ	Density ratio $[-]$
μ	Dynamic viscosity $[\mathrm{kg/(m \cdot s)}]$
ν	Poisson's ratio $[-]$
Φ	Zeta potential $[\mathrm{kg \cdot m^2/(A \cdot s^{-3})}]$
ϕ	Velocity potential $[\mathrm{m^2/s}]$
ψ	Signed level-set function $[-]$
ρ	Density $[\mathrm{kg/m^3}]$
σ	Surface tension coefficient $[\mathrm{N/m}]$
Others	
$\boldsymbol{n}_f$	Normal vector at face centre
$\boldsymbol{n}_s$	Normal vector at cell centre
χ_e	Evaporation coefficient $[-]$
Δ	Laplace operator
$\dot{m}$	Mass source per unit volume $[\mathrm{kg/(m^3 \cdot s)}]$
ε_0	Permittivity of vacuum $[\mathrm{F/m}]$

ε_r	Relative dielectric constant $[-]$
γ_s	Surface energy density $[\mathrm{J/m^2}]$
θ	Particle angular displacement [rad]
$\mathbf{U}$	Velocity $[\mathrm{m/s}]$
$\mathbf{x}$	Particle displacement [m]
μ_c	Coefficient of friction $[-]$
∇	Gradient operator
e_r	Restitution coefficient $[-]$
F	Faraday constant $[\mathrm{C/mol}]$
G	Shear modulus $[\mathrm{kg/(m \cdot s^2)}]$
h	Separation distance [m]
$I(\boldsymbol{x},t)$	Indicator function $[-]$
J	Mass source per unit area $[\mathrm{kg/(m^2 \cdot s)}]$
K	Interface curvature $[1/\mathrm{m}]$
k_d	Debye-Hüeckel parameter $[1/\mathrm{m}]$
k_n	Normal contact stiffness $[\mathrm{kg/s^2}]$
k_t	Tangential contact stiffness $[\mathrm{kg/s^2}]$
m	Mass [kg]
p	Pressure [Pa]
R	Particle radius [m]
R_g	Gas constant $[\mathrm{kg \cdot m^2/(K \cdot mol \cdot s^2)}]$
T	Temperature [K]
Y	Young's modulus $[\mathrm{kg/(m \cdot s^2)}]$
z	Valency of the electrolyte $[-]$

Subscripts/superscripts

∞	Infinity
filt	Filtering

st	Surface tension
g	Gas phase
i	Particle *i*
j	Particle *j*
l	Liquid phase
n	Normal direction
t	Tangential direction

A Appendix

A.1 Derivation of the VoF/LS equation

Derivation of the VoF/LS transport equation is detailed below. Here, the volume fraction field α is taken as the tracked field. At time t_1 and t_2, the Eulerian descriptions of α are denoted as $\alpha_1(x_1, y_1, z_1, t_1)$ and $\alpha_2(x_2, y_2, z_2, t_2)$, respectively. The Taylor's series of α_2 is given by

$$\alpha_2 = \alpha_1 + \frac{\partial \alpha}{\partial x}(x_2 - x_1) + \frac{\partial \alpha}{\partial y}(y_2 - y_1) + \frac{\partial \alpha}{\partial z}(z_2 - z_1) + \frac{\partial \alpha}{\partial t}(t_2 - t_1), \quad \text{(A.1)}$$

Eqn. A.1 is now divided by $t_2 - t_1$, and then becomes

$$\frac{\alpha_2 - \alpha_1}{t_2 - t_1} = \frac{\partial \alpha}{\partial x}\frac{x_2 - x_1}{t_2 - t_1} + \frac{\partial \alpha}{\partial y}\frac{y_2 - y_1}{t_2 - t_1} + \frac{\partial \alpha}{\partial z}\frac{z_2 - z_1}{t_2 - t_1} + \frac{\partial \alpha}{\partial t}\frac{t_2 - t_1}{t_2 - t_1}. \quad \text{(A.2)}$$

Let u, v and w replace the following limits

$$u = \lim_{t_2 \to t_1} \frac{x_2 - x_1}{t_2 - t_1}, v = \lim_{t_2 \to t_1} \frac{y_2 - y_1}{t_2 - t_1}, w = \lim_{t_2 \to t_1} \frac{z_2 - z_1}{t_2 - t_1}, \quad \text{(A.3)}$$

respectively, when t_2 approximates t_1. Eqn. A.2 then becomes

$$\frac{d\alpha}{dt} = \lim_{t_2 \to t_1} \frac{\alpha_2 - \alpha_1}{t_2 - t_1} = u\frac{\partial \alpha}{\partial x} + v\frac{\partial \alpha}{\partial y} + w\frac{\partial \alpha}{\partial z} + \frac{\partial \alpha}{\partial t} = \frac{\partial \alpha}{\partial t} + \mathbf{U} \cdot \nabla \alpha, \quad \text{(A.4)}$$

Eqn. A.4 is also the definition of the material derivative of α.

The liquid volume fraction α_f can be thus solved from the equation

$$\frac{\partial \alpha_f}{\partial t} + \mathbf{U} \cdot \nabla \alpha_f = 0. \tag{A.5}$$

Normally, the conservative form of Eqn. A.5 is widely used in the literature. It is derived by adding $\alpha_f \nabla \cdot \mathbf{U}$ to both sides of Eqn. A.5, then leading to

$$\frac{\partial \alpha_f}{\partial t} + \underbrace{\mathbf{U} \cdot \nabla \alpha_f + \alpha_f \nabla \cdot \mathbf{U}}_{=\nabla \cdot (\alpha_f \mathbf{U})} = \alpha_f \nabla \cdot \mathbf{U}. \tag{A.6}$$

The term on the right-hand side of Eqn. A.6 becomes zero only for incompressible flow without phase change.

A.2 Derivation of equations for the un-resolved CFD-DEM

A.2.1 Derivation of the continuity equation

In principle, the conservation of mass indicates that a region conserves its mass locally in the absence of mass sources or sinks. The rate of mass change rate $\dot{m}$ is given by

$$\dot{m} = \frac{1}{V} \frac{d}{dt} \int_V (\varepsilon_f \rho)\, dV, \tag{A.7}$$

where ε_f is the fluid void fraction field and ρ is the liquid density.

According to the Reynolds transport theorem, $\frac{d}{dt} \int_V (\varepsilon_f \rho)\, dV$ can be given as a sum of two integrals as

$$\frac{d}{dt} \int_V (\varepsilon_f \rho)\, dV = \int_V \frac{\partial (\varepsilon_f \rho)}{\partial t}\, dV + \int_S (\varepsilon_f \rho \mathbf{U} \cdot \mathbf{n})\, dS. \tag{A.8}$$

The surface integral in Eqn. A.8 can be transformed into a volume integral as

$$\int_S (\varepsilon_f \rho \mathbf{U} \cdot \mathbf{n})\, dS = \int_V \nabla \cdot (\varepsilon_f \rho \mathbf{U})\, dV \tag{A.9}$$

by applying the divergence theorem.

Substituting the volume integral of Eqn. A.9 into Eqn. A.8 leads to

$$\int_V \frac{\partial(\varepsilon_f \rho)}{\partial t}\, dV + \int_V \nabla \cdot (\varepsilon_f \rho \mathbf{U})\, dV = \int_V [\frac{\partial(\varepsilon_f \rho)}{\partial t} + \nabla \cdot (\varepsilon_f \rho \mathbf{U})]\, dV = \dot{m} V. \tag{A.10}$$

Assuming validity of Eqn. A.10 for arbitrary volumes V, the continuity equation for the un-resolved CFD-DEM is given by

$$\frac{\partial(\varepsilon_f \rho)}{\partial t} + \nabla \cdot (\varepsilon_f \rho \mathbf{U}) = \dot{m}, \tag{A.11}$$

where $\dot{m}$ is only zero for the system without phase change.

A.2.2 Derivation of the momentum equation

The physics behind the momentum equation is governed by Newton's second law of motion given by

$$\mathbf{F} = \frac{d(m\mathbf{U})}{dt}. \tag{A.12}$$

Multiplying the terms on both sides of Eqn. A.12 by the void fraction ε_f, Eqn. A.12 then becomes

$$\varepsilon_f \mathbf{F} = \frac{d(\varepsilon_f m \mathbf{U})}{dt} = \frac{d}{dt} \int_V (\varepsilon_f \rho \mathbf{U})\, dV. \tag{A.13}$$

According to the Reynolds transport theorem, $\frac{d}{dt}\int_V(\varepsilon_f\rho\mathbf{U})\,dV$ can be given as a sum of two integrals as

$$\frac{d}{dt}\int_V(\varepsilon_f\rho\mathbf{U})\,dV=\int_V\frac{\partial(\varepsilon_f\rho\mathbf{U})}{\partial t}\,dV+\int_S(\varepsilon_f\rho\mathbf{U}\mathbf{U}\cdot\mathbf{n})\,dS. \tag{A.14}$$

The surface integral in Eqn. A.14 can be transformed into a volume integral as

$$\int_S(\varepsilon_f\rho\mathbf{U}\mathbf{U}\cdot\mathbf{n})\,dS=\int_V\nabla\cdot(\varepsilon_f\rho\mathbf{U}\mathbf{U})\,dV. \tag{A.15}$$

Substituting the volume integral in Eqn. A.15 into Eqn. A.14 leads to

$$\int_V\left[\frac{\partial(\varepsilon_f\rho\mathbf{U})}{\partial t}+\nabla\cdot(\varepsilon_f\rho\mathbf{U}\mathbf{U})\right]dV=\varepsilon_f\mathbf{F}=\int_V\left(\varepsilon_f\frac{\mathbf{F}}{V}\right)dV=\int_V(\varepsilon_f\mathbf{f})\,dV, \tag{A.16}$$

where $\mathbf{f}$ is the volumetric body force term.

Assuming validity for arbitrary volumes V, Eqn. A.16 becomes

$$\frac{\partial(\varepsilon_f\rho\mathbf{U})}{\partial t}+\nabla\cdot(\varepsilon_f\rho\mathbf{U}\mathbf{U})=\varepsilon_f\mathbf{f}. \tag{A.17}$$

In principle, for the un-resolved CFD-DEM approach, the volumetric body force term consists of several force terms such as pressure gradient force, gravity, etc. Here, $\mathbf{f}$ is given by

$$\mathbf{f}=-\nabla p+\nabla\cdot\tau-\rho\mathbf{g}+\mathbf{F}_{\mathrm{st}}, \tag{A.18}$$

where $\mathbf{F}_{\mathrm{st}}$ is the surface-tension force.

Finally, the complete form of the momentum equation for the un-resolved CFD-DEM model is given by

$$\frac{\partial(\varepsilon_f\rho\mathbf{U})}{\partial t}+\nabla\cdot(\varepsilon_f\rho\mathbf{U}\mathbf{U})=-\varepsilon_f\nabla p+\varepsilon_f\nabla\cdot\tau-\varepsilon_f\rho\mathbf{g}+\varepsilon_f\mathbf{F}_{\mathrm{st}}. \tag{A.19}$$

A.2.3 Derivation of the VoF equation

In order to derive the VoF equation for the un-resolved CFD-DEM model, it starts with the material derivative of α_f given by

$$\frac{D\alpha_f}{Dt} = 0. \tag{A.20}$$

Replacing α_f in Eqn. A.20 with $\varepsilon_f \alpha_f$, leads to

$$\frac{D\varepsilon_f \alpha_f}{Dt} = \frac{\partial \varepsilon_f \alpha_f}{\partial t} + \mathbf{U} \cdot \nabla \varepsilon_f \alpha_f = 0. \tag{A.21}$$

The conservative form of Eqn. A.21 is given by

$$\frac{\partial \varepsilon_f \alpha_f}{\partial t} + \nabla \cdot (\varepsilon_f \alpha_f \mathbf{U}) = \varepsilon_f \alpha_f \nabla \cdot \mathbf{U}, \tag{A.22}$$

where the term on the right-hand side of Eqn. A.22 becomes zero only for incompressible flow without phase change.

A.3 Voronoi tessellation for granular media: calculations of the local packing fraction with Voro++

Voronoi tessellation is used to calculate the local packing properties, e.g. the local packing fraction and packing structures of either mono-disperse or poly-disperse granular systems in this thesis. The local packing fraction ϕ_f is defined by

$$\phi_f = \frac{V_s}{V_c}, \tag{A.23}$$

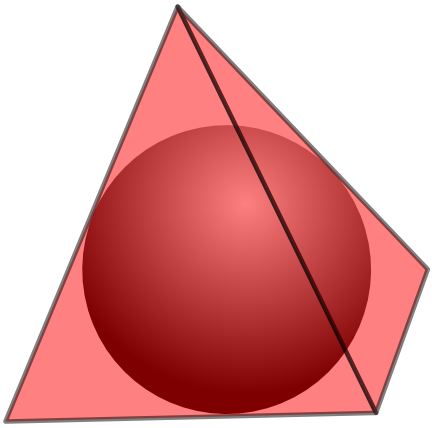


Figure A.1: Schematic diagram of a sphere enclosed by its voronoi cell.

where V_s and V_c are the volume of the sphere and its voronoi cell, namely the tetrahedron as shown in Figure A.1, respectively. The open-source Voronoi tessellation code Voro++ is used to generate voronoi cells and calculate the volume of each voronoi cell.

Two different cases are presented to demonstrate the performance of the Voro++ code in calculating the local packing fractions for the Simple Cubic Packing (SCP) and Hexagonal Close Packing (HCP). The single layer of the SCP is shown in Figure A.2, and the analytical solution of the packing fraction is given by

$$\phi_f{}^{\mathrm{SCP}} = \frac{V_p}{V_c} = \frac{\frac{4\pi R^3}{3}}{(2R)^3} = \frac{\pi}{6} = 0.5236. \tag{A.24}$$

The image on the right-hand side of Figure A.2 is the color map of the local packing fraction of the SCP. It proves that the local packing fraction calculated with Voro++ agrees well with the corresponding analytical solution.

Similarly, single layer of the HCP is shown in Figure A.3, and the periodic boundary condition is applied along the horizontal direction. The analytical solution

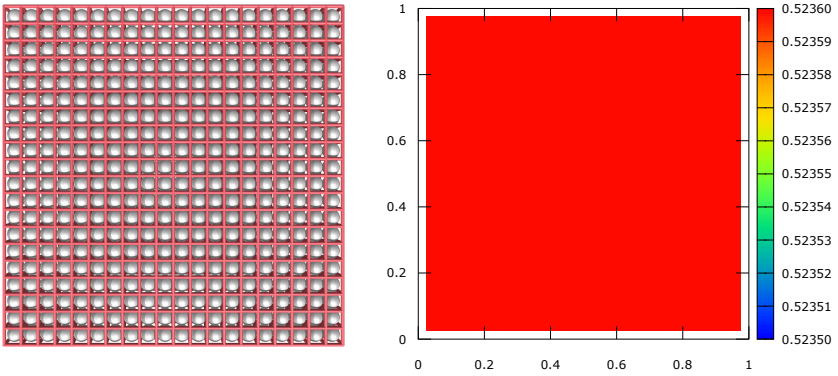


Figure A.2: (a) voronoi cells of the SCP, (b) local packing fraction of SCP.

for the single layer HCP except for the local packing fraction on the boundary is given by

$$\phi_f{}^{\mathrm{HCP}} = \frac{V_p}{V_c} = \frac{\frac{4\pi R^3}{3}}{\frac{3\sqrt{3}(\frac{2R}{\sqrt{3}})^2 2R}{2}} = \frac{\pi}{3\sqrt{3}} = 0.6046. \tag{A.25}$$

The local packing fraction calculated by Voro++ is shown in the right-hand side of Figure A.3, and a good agreement is found between the local packing fraction calculated by Voro++ and the analytical solution.

A.4 Code list

```
/*-------------------------------------------
    Coding by
    Huihuang Xia, PhD student working at
    Karlsruhe Institute of Technology (KIT)
    E-mail: huihuang.xia@kit.edu
-------------------------------------------*/

#ifdef COHESION_MODEL
```

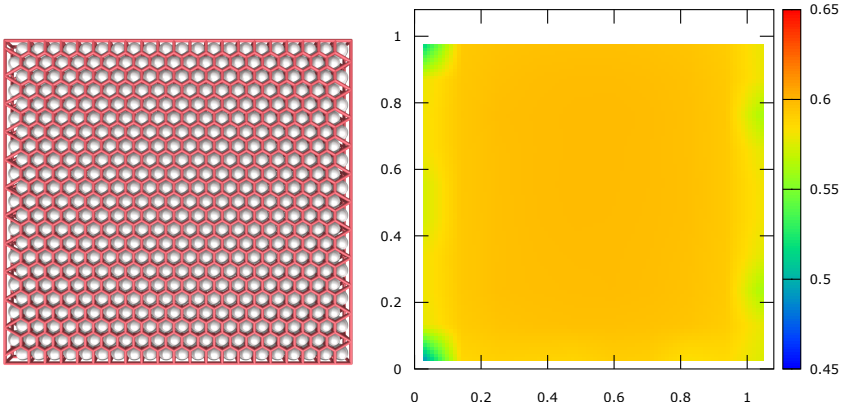


Figure A.3: (a) voronoi cells of the HCP, (b) local packing fraction of HCP.

```
9  COHESION_MODEL(COHESION_VDW, vdw, 3)  // the id of vdw in
       cohesion model
10 #else
11
12 #ifndef COHESION_MODEL_VDW_H_
13 #define COHESION_MODEL_VDW_H_
14
15 #include <cmath>
16 #include <cstring>
17 #include <algorithm>
18 #include "pointers.h"
19 #include "neighbor.h"
20 #include "contact_models.h"
21 #include "global_properties.h"
22 #include "cohesion_model_base.h"
23
24 #define CST 10.6666666666667 // 64/6.0 or 32/3.0
25
26 namespace LIGGGHTS {
27 namespace ContactModels {
28 using namespace std;
29 using namespace LAMMPS_NS;
30
```

```
31  template<>
32  class CohesionModel<COHESION_VDW> : public
      CohesionModelBase {
33  public:
34  CohesionModel(LAMMPS * lmp, IContactHistorySetup * hsetup,
      class ContactModelBase * c) :
35  CohesionModelBase(lmp, hsetup, c),
36  surfaceEnergyDens(NULL), // key parameters in Van Der Waals
       model
37  minSeparaDistRat(0.0), // minSeparaDistRat(0.0) ==
      minSeparaDistRat = 0.0
38  maxSeparaDistRat(0.0),
39  history_offset(0)
40  //
41    {
42      history_offset = hsetup->add_history_value("contflag","
      0");
43    }
44
45    void registerSettings(Settings& settings)
46    {
47      settings.registerOnOff("tangential_reduce",
      tangentialReduce_,false);
48    }
49
50    inline void postSettings(IContactHistorySetup * hsetup,
      ContactModelBase *cmb) {}
51
52    void connectToProperties(PropertyRegistry & registry) //
      defining new parameters
53    {
54      registry.registerProperty("surfaceEnergyDens", &
      MODEL_PARAMS::createSurfaceEnergyDensity);
55      registry.registerProperty("minSeparaDistRat", &
      MODEL_PARAMS::createMinSeparationDistanceRatio);
56      registry.registerProperty("maxSeparaDistRat", &
      MODEL_PARAMS::createMaxSeparationDistanceRatio);
57
58      registry.connect("surfaceEnergyDens", surfaceEnergyDens
      ,"cohesion_model vdw");
```

```
59        registry.connect("minSeparaDistRat", minSeparaDistRat,"
      cohesion_model vdw");
60        registry.connect("maxSeparaDistRat", maxSeparaDistRat,"
      cohesion_model vdw");
61
62        // error checks on coarsegraining
63        if(force->cg_active())
64            error->cg(FLERR,"cohesion model vdw");
65
66        neighbor->register_contact_dist_factor(maxSeparaDistRat
      *1.1); //defining neighboring list
67
68        if (maxSeparaDistRat < 1.0)
69            error->one(FLERR,"\n\ncohesion model vdw requires
      the maxSeparaDistRat >= 1.0");
70        if (minSeparaDistRat >= 1.0)
71            error->one(FLERR,"\n\ncohesion model vdw requires
      the minSeparaDistRat < 1.0");
72      }
73
74      inline void endSurfacesIntersect(SurfacesIntersectData &
      sidata, ForceData&, ForceData&) {}
75      void beginPass(SurfacesIntersectData&, ForceData&,
      ForceData&){}
76      void endPass(SurfacesIntersectData&, ForceData&,
      ForceData&){}
77
78      // when separation distance h is not positive, we need
        define the calculation within the "surfacesIntersect"
        function
79      void surfacesIntersect(SurfacesIntersectData & sidata,
      ForceData & i_forces, ForceData & j_forces)
80      {
81        //r is the distance between the sphere's centers
82        const double r = sidata.r;
83        const double ri = sidata.radi; // radius of particle i
84        const double rj = (sidata.is_wall) ? ri : sidata.radj;
      // radius of particle j
85        const double rMax = (ri >= rj) ? ri : rj; // maximum
      radius between two particles
```

```
86        double hMin = rMax*minSeparaDistRat; // hMin = the
      minimum separation distance between two particles
87        double Ha = 24.0*M_PI*surfaceEnergyDens[sidata.itype][
      sidata.jtype]*hMin*hMin; // Ha = the Hamaker constant
88        double h = (sidata.is_wall) ? r-ri : r-(ri+rj); // h =
      the separation distance between two particles or particle
       - wall
89        double Acont, Acont_num, Acont_den;  //Acont = other
      expressions in the equation excepting the Ha
90
91        if(sidata.contact_flags) *sidata.contact_flags |=
      CONTACT_COHESION_MODEL;
92        double * const contflag = &sidata.contact_history[
      history_offset];
93        // for negitavie separation distance
94        contflag[0] = 1.0;
95
96        //interaction between sphere and wall
97        //equations adopted from "Abbasfard, H., et al. (2016).
       Powder Technology 299: 9-18."
98        if (sidata.is_wall)
99        {
100         if (h <= 0)
101         {
102           Acont = ri/(6.0*hMin*hMin);
103         }
104         else return;
105       }
106       //interaction between sphere and particle
107       // equations adopted from "Yang, R., et al. (2000).
      Physical review E 62(3): 3900"
108       else
109       {
110         if (h <= 0)
111           {
112             Acont_num = ri*ri*ri*rj*rj*rj*(hMin+ri+rj);
113             Acont_den = (hMin*hMin+2.0*ri*hMin+2.0*rj*hMin)*(
      hMin*hMin+2.0*ri*hMin+2.0*rj*hMin)*(hMin*hMin+2.0*ri*hMin
      +2.0*rj*hMin+4.0*ri*rj)*(hMin*hMin+2.0*ri*hMin+2.0*rj*
      hMin+4.0*ri*rj);
```

```
114             Acont = CST * Acont_num/Acont_den;  //if
    separation distance h is quite small, h should be hMin
115           }
116         else return;
117       }
118       // Van Der Waals force is an attractive force; - means
    it is one of the attractive forces
119       const double Fn_vdw = - Ha*Acont;
120       // Fn_vdw = Van Der Waals force component
121       if(tangentialReduce_) sidata.Fn += Fn_vdw;
122
123       // apply normal force
124       if(sidata.is_wall)
125       {
126         const double Fn_ = Fn_vdw * sidata.area_ratio;
127         i_forces.delta_F[0] += Fn_ * sidata.en[0];
128         i_forces.delta_F[1] += Fn_ * sidata.en[1];
129         i_forces.delta_F[2] += Fn_ * sidata.en[2];
130       }
131       else
132        {
133         const double fx = Fn_vdw * sidata.en[0];
134         const double fy = Fn_vdw * sidata.en[1];
135         const double fz = Fn_vdw * sidata.en[2];
136
137         i_forces.delta_F[0] += fx;
138         i_forces.delta_F[1] += fy;
139         i_forces.delta_F[2] += fz;
140
141         j_forces.delta_F[0] -= fx;
142         j_forces.delta_F[1] -= fy;
143         j_forces.delta_F[2] -= fz;
144       }
145     }
146
147   // when separation distance h is positive, we need define
      the calculation within the "surfacesClose" function
148   void surfacesClose(SurfacesCloseData& scdata, ForceData &
      i_forces, ForceData & j_forces)
149   {
```

```
150       if(scdata.contact_flags) *scdata.contact_flags |=
      CONTACT_COHESION_MODEL;
151       double * const contflag = &scdata.contact_history[
      history_offset];
152       contflag[0] = 1.0;
153
154       //r is the distance between the sphere's centers
155       const double r = sqrt(scdata.rsq);
156       const double ri = scdata.radi; // radius of particle i
157       const double rj = (scdata.is_wall) ? ri : scdata.radj;
      // radius of particle j
158       const double rMax = (ri >= rj) ? ri : rj; // maximum
      radius between two particles
159       double hMin = rMax*minSeparaDistRat; // hMin = the
      minimum separation distance between two particles
160       double Ha = 24.0*M_PI*surfaceEnergyDens[scdata.itype][
      scdata.jtype]*hMin*hMin; // Ha = the Hamaker constant
161       double h = (scdata.is_wall) ? r-ri : r-(ri+rj); // h =
      the separation distance between two particles or particle
       - wall
162       double Acont, Acont_num, Acont_den;  //Acont = other
      expressions in the equation excepting the Ha
163       //interaction between sphere and wall
164       //equations adopted from "Abbasfard, H., et al. (2016).
       Powder Technology 299: 9-18."
165       if (scdata.is_wall)
166       {
167         if (h > hMin)
168         {
169           Acont = ri/(6.0*h*h);
170         }
171         else if ((h > 0) && (h <= hMin))
172         {
173           Acont = ri/(6.0*hMin*hMin);
174         }
175         else return;
176       }
177       //interaction between sphere and particle
178       // equations adopted from "Yang, R., et al. (2000).
      Physical review E 62(3): 3900"
179       else
```

```
180       {
181         if (h > hMin)
182           {
183             Acont_num = ri*ri*ri*rj*rj*rj*(h+ri+rj);
184             Acont_den = (h*h+2.0*ri*h+2.0*rj*h)*(h*h+2.0*ri*h
    +2.0*rj*h)*(h*h+2.0*ri*h+2.0*rj*h+4.0*ri*rj)*(h*h+2.0*ri*
    h+2.0*rj*h+4.0*ri*rj);
185             Acont = CST * Acont_num/Acont_den;  //if
    separation distance h is quite small, h should be hMin
186           }
187         else if ((h > 0) && (h <= hMin))
188           {
189             Acont_num = ri*ri*ri*rj*rj*rj*(hMin+ri+rj);
190             Acont_den = (hMin*hMin+2.0*ri*hMin+2.0*rj*hMin)*(
    hMin*hMin+2.0*ri*hMin+2.0*rj*hMin)*(hMin*hMin+2.0*ri*hMin
    +2.0*rj*hMin+4.0*ri*rj)*(hMin*hMin+2.0*ri*hMin+2.0*rj*
    hMin+4.0*ri*rj);
191             Acont = CST * Acont_num/Acont_den;  //if
    separation distance h is quite small, h should be hMin
192           }
193         else return;
194       }
195         // Van Der Waals force is an attractive force; -
    means it is one of the attractive forces
196         const double Fn_vdw = - Ha*Acont;
197         // Fn_vdw = Van Der Waals force component
198
199         const double rinv = 1.0 / r;
200         const double dx = scdata.delta[0];
201         const double dy = scdata.delta[1];
202         const double dz = scdata.delta[2];
203         const double enx = dx * rinv; // vector in x
204         const double eny = dy * rinv; // vector in y
205         const double enz = dz * rinv; // vector in z
206
207        // apply normal and tangential force
208         const double fx = Fn_vdw * enx; // component of Fn_vdw
     in x
209         const double fy = Fn_vdw * eny; // component of Fn_vdw
     in y
```

```
210         const double fz = Fn_vdw * enz; // component of Fn_vdw
        in z
211
212         scdata.has_force_update = true;
213
214          // apply normal force
215          if(scdata.is_wall)
216          {
217            const double area_ratio = scdata.area_ratio;
218            i_forces.delta_F[0] += fx * area_ratio;
219            i_forces.delta_F[1] += fy * area_ratio;
220            i_forces.delta_F[2] += fz * area_ratio;
221          }
222          else
223           {
224            i_forces.delta_F[0] += fx;
225            i_forces.delta_F[1] += fy;
226            i_forces.delta_F[2] += fz;
227
228            j_forces.delta_F[0] -= fx;
229            j_forces.delta_F[1] -= fy;
230            j_forces.delta_F[2] -= fz;
231       }
232   }
233
234   private:
235     double ** surfaceEnergyDens; // surface energy density
236     double minSeparaDistRat; // minimal separation distance
      ratio
237     double maxSeparaDistRat; // maximum separation distance
      ratio
238     bool tangentialReduce_;
239     int history_offset;
240   };
241 }
242 }
243
244 #endif // COHESION_MODEL_VDW_H_
245 #endif
```

Code list A.1: Source code for the implementation of the Van der Waals force model.

```
// Huihuang Xia; the additional coding to further reduce the spurious
// currents/velocities with optional filtering methods to filter the spurious
// currents/velocities
/*######################
References:
[1]. Raeini, A. Q., et al. (2012). J Comput Phys 231(17): 5653-5668.
######################*/

// the Sharp Surface-tension Force (SSF) model without filtering (Ref.01)
if (filteringSC == false)
{

  // to access the surface-tension force
  surfaceScalarField Fst = interfaceNew.surfaceTensionForce();

  phiFst = Fst*mesh.magSf();

  // compute the surface-tension pressure here
  while (pimple.correctNonOrthogonal())
  {
    fvScalarMatrix pcEqn
    (
        fvm::laplacian(dimensionedScalar("1", dimless, 1), pc) == fvc::div(phiFst)
    );

    pcEqn.setReference(pcRefCell, pcRefValue);
    pcEqn.solve();

    if (pimple.finalNonOrthogonalIter())
    {
      phiFst -= pcEqn.flux();
    }

    pc.correctBoundaryConditions();
  }
```

```
36 }
37
38 // the Filtering Surface-tension Force (FSF) model
39 // to filter the SC parallel to the interface further (Ref
       .01)
40 else
41 {
42   // Cpc: the sharping coefficient
43   scalar Cpc =
44         readScalar(alpha1.mesh().solutionDict().subDict("
       PIMPLE").lookup("Cpc"));
45
46   // Cfc: the coefficient for how fast the SC is filtered
47   scalar Cfc =
48         readScalar(alpha1.mesh().solutionDict().subDict("
       PIMPLE").lookup("Cfc"));
49
50   // RelaxF: the relaxation factor
51   scalar RelaxF =
52         readScalar(alpha1.mesh().solutionDict().subDict("
       PIMPLE").lookup("RelaxF"));
53
54   // thresholdF: the threshold for Fst
55   scalar thresholdF =
56     readScalar(alpha1.mesh().solutionDict().subDict("PIMPLE")
       .lookup("thresholdF"));
57
58   // the delta function for filtering SC (Ref.01)
59   volScalarField alphaPc = (1.0/(1.0 - Cpc))*(min(max(alpha1,
       Cpc/2.0),
60                             (1.0 - Cpc/2.0)) - Cpc/2.0);
61
62   surfaceScalarField deltaF = fvc::snGrad(alphaPc);
63
64   surfaceScalarField deltaFNor = mag(deltaF)/(mag(deltaF) +
       1.0E-4*mixture.deltaN());
65
66   // to access the surface-tension force
67   surfaceScalarField Fst = interfaceNew.surfaceTensionForce()
       ;
68
```

```
69  phiFst = Fst*mesh.magSf();
70
71  surfaceScalarField phiCaFilter1st
72  (
73      IOobject
74      (
75          "phiCaFilter1st",
76          runTime.timeName(),
77          mesh,
78          IOobject::NO_READ,
79          IOobject::NO_WRITE
80      ),
81      mesh,
82      dimensionedScalar("phiCaFilter1st", dimPressure/
    dimLength*dimArea, 0.0)
83  );
84
85  // the filtering surface tension force flux
86  phiFst -= phiCaFilter1st;
87
88  // compute the surface-tension pressure here
89  while (pimple.correctNonOrthogonal())
90  {
91    fvScalarMatrix pcEqn
92    (
93        fvm::laplacian(dimensionedScalar("1", dimless, 1), pc
    ) == fvc::div(phiFst)
94    );
95
96    pcEqn.setReference(pcRefCell, pcRefValue);
97    pcEqn.solve();
98
99    if (pimple.finalNonOrthogonalIter())
100   {
101     phiFst -= pcEqn.flux();
102   }
103
104   pc.correctBoundaryConditions();
105 }
106
107 // reconstruct the surface-tension field
```

```
108   volVectorField gPc = fvc::reconstruct(phiFst);
109   gPc.correctBoundaryConditions();
110
111   // nHat: the unit surface normal defined at cell centre
112   volVectorField ns = interfaceNew.nHat();
113
114   volVectorField vgPcllInterface = (gPc) - ((gPc) & (ns))*(ns
        );
115
116   surfaceVectorField gPcllInterface = fvc::interpolate(
        vgPcllInterface);
117
118   phiCaFilter1st = deltaFNor*(RelaxF*phiCaFilter1st.oldTime()
119                  + Cfc*(gPcllInterface & mesh.Sf()));
120
121   phiCaFilter1st.storeOldTime();    // save the data of the
        previous time step
122
123   // to make sure |phiCaFilter| < |phiFst|
124   forAll(phiCaFilter1st, i)
125   {
126     phiCaFilter1st[i] -= (phiFst[i]*phiCaFilter1st[i] < 0) ?
127           (max(min(phiCaFilter1st[i], mag(phiFst[i])), -mag(
        phiFst[i]))) : 0.0;
128   }
129
130   const fvBoundaryMesh& boundary = mesh.boundary();
131
132   forAll(boundary, pI)
133   {
134     scalarField phiCaFilter1stBf = phiCaFilter1st.
        boundaryField()[pI];
135     scalarField phiFstBf = phiFst.boundaryField()[pI];
136
137     forAll(phiCaFilter1stBf, i)
138     {
139       phiCaFilter1stBf[i] -=
140         (phiFstBf[i]*phiCaFilter1stBf[i] < 0.0) ?
141         max(min(phiCaFilter1stBf[i], mag(phiFstBf[i])), -mag(
        phiFstBf[i])) : 0.0;
142     }
```

```
143   }
144
145   dimensionedScalar FstAvg = dimensionedScalar("FstAvg",
        dimPressure/dimLength, 0.0);
146   FstAvg = average(mag(Fst)*deltaFNor)/average(deltaFNor +
        1.0E-9);
147
148   surfaceScalarField phiCaThreshold = thresholdF*FstAvg*mesh.
        magSf();
149
150   // the 2nd filtering
151   surfaceScalarField phiCaFilter2nd = (min(max(phiFst, -
        phiCaThreshold), phiCaThreshold));
152   phiFst -= phiCaFilter2nd;
153
154   // reconstruct the capillary field
155   gPc = fvc::reconstruct(phiFst);
156   gPc.correctBoundaryConditions();
157 }
```

Code list A.2: Source code for the numerical implementation to filter the spurious currents.

```
1  % Huihuang Xia; solve ODE with ODE45 solver
2  % deltaT = 100 K (t_total = 0.15 s)
3  % solve the ODE only within 0~0.1 s
4  format long;
5  kg = 0.01;
6  cpg = 1900;
7  Tinf = 473.15;
8  Tsat = 373.15;
9  hev = 1.0e6;
10 B = cpg*(Tinf - Tsat)/hev;
11 D0 = 2.5E-04;
12 DIns = 4.0*D0;
13 rho1 = 10;
14 k = 8.0*kg*log(1.0+B)/(rho1*cpg);
15
16 % dt = 0.001; t_total = 0.15
17 tspan = 0:0.001:0.15;
18 y0 = D0^2;
19 [t,y] = ode45(@(t,y) -k/log(DIns/sqrt(abs(y))), tspan, y0);
20 plot(t,y,'-o');
```

```
savefig('ode45.fig');
f=open('ode45.fig');
D=get(gca,'Children');
XData=get(D,'XData');
YData=get(D,'YData');
Data=[XData' YData']
```

Code list A.3: Matlab code for solving the ordinary differential equation.

```
/*
#####################################################
New feature I: coupling to open-source DEM code - LIGGGHTS
    with the resolved
               CFD-DEM approach

New feature II: un-physical spurious currents/velocities are
    suppressed

New feature III: 2D & 3D Adaptive Mesh Refinement (AMR)
#####################################################

Developer
    Huihuang Xia, PhD student working at
  the Institute of Applied Materials (IAM),
    Karlsruhe Institute of Technology (KIT), Germany
    E-mail: huihuang.xia@kit.edu
\*---------------------------------------------------------*/

#include "fvCFD.H"
#include "CMULES.H"
#include "EulerDdtScheme.H"
#include "localEulerDdtScheme.H"
#include "CrankNicolsonDdtScheme.H"
#include "subCycle.H"
#include "immiscibleIncompressibleTwoPhaseMixture.H"
#include "turbulentTransportModel.H"
#include "pimpleControl.H"
#include "fvOptions.H"
#include "CorrectPhi.H"
#include "fvcSmooth.H"
#include "interface.H"    // Huihuang Xia; new interface
    properties
```

```
31
32 // Huihuang Xia - comments begin
33 // DEM related stuff for the resolved CFD-DEM
34 #include "cfdemCloudIBVoF.H"    // new class for VoF-based
       coupling
35 #include "implicitCouple.H"
36 #include "averagingModel.H"
37 #include "voidFractionModel.H"
38 #include "dynamicFvMesh.H"
39 #include "cellSet.H"
40 // Huihuang Xia - comments end
41
42 // * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
       * * * * * * * * * //
43
44 int main(int argc, char *argv[])
45 {
46     #include "postProcess.H"
47
48     #include "setRootCase.H"
49     #include "createTime.H"
50     // #include "createMesh.H"    // Huihuang Xia
51     #include "createDynamicFvMesh.H"    // Huihuang Xia
52     #include "createControl.H"
53     #include "createTimeControls.H"
54     #include "initContinuityErrs.H"
55     #include "createFields.H"
56     #include "createAlphaFluxes.H"
57     #include "createFvOptions.H"
58     #include "correctPhi.H"
59     #include "createSolverSettings.H"   // Huihuang Xia
60
61     turbulence->validate();
62
63     // Huihuang Xia - comments begin
64     #include "readTimeControls.H"
65     #include "CourantNo.H"
66     #include "setInitialDeltaT.H"
67
68     // create cfdemCloud
69     cfdemCloudIBVoF particleCloud(mesh);
```

```
    // Huihuang Xia - comments end

    // * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //

    Info<< "\nStarting time loop\n" << endl;

    while (runTime.loop())   // Huihuang Xia; run() -> loop()
    {

        // Huihuang Xia - comments begin
        Info<< "Time = " << runTime.timeName() << nl << endl;

        // DyM
        interFace = mag(mesh.lookupObject<volScalarField>("voidfractionNext"));
        particleCloud.setMeshHasUpdatedFlag(mesh.update());
        // Huihuang Xia - comments end

        #include "readTimeControls.H"

        #include "CourantNo.H"
        #include "alphaCourantNo.H"
        #include "setDeltaT.H"

        // Huihuang Xia - comments begin
        // runTime++;
        //
        // Info<< "Time = " << runTime.timeName() << nl << endl;

        // DEM related stuff
        Info << "- evolve()" << endl;
        particleCloud.evolve(voidfraction, interFace);
        // Huihuang Xia - comments end

        // Huihuang Xia - comments begin
        if (particleCloud.solveFlow())
        {
            // --- Pressure-velocity PIMPLE corrector loop
            while (pimple.loop())
```

```
108             {
109                 #include "alphaControls.H"
110                 #include "alphaEqnSubCycle.H"
111 
112                 mixture.correct();
113 
114                 #include "filteringSC.H"  // Huihuang Xia
115 
116                 #include "UEqn.H"
117 
118                 // --- Pressure corrector loop
119                 while (pimple.correct())
120                 {
121                     #include "pEqn.H"
122                 }
123 
124                 if (pimple.turbCorr())
125                 {
126                     turbulence->correct();
127                 }
128             }
129         } // Huihuang Xia - comments end
130 
131         // Huihuang Xia - comments begin
132         Info << "particleCloud.calcVelocityCorrection() " <<
    endl;
133         volScalarField voidfractionNext
134                        = mesh.lookupObject<volScalarField>("
    voidfractionNext");
135 
136         particleCloud.calcVelocityCorrection(p, U, phiIB,
    voidfractionNext, rho);
137         // Huihuang Xia - comments end
138 
139         runTime.write();
140 
141         Info<< "ExecutionTime = " << runTime.elapsedCpuTime()
     << " s"
142             << "  ClockTime = " << runTime.elapsedClockTime()
     << " s"
143             << nl << endl;
```

```
144     }
145
146     Info<< "End\n" << endl;
147
148     return 0;
149 }
150
151
152 //
    ************************************************************************
     //
```

Code list A.4: Source code for the resolved CFD-DEM approach.

```
1  # Coding by Huihuang Xia
2  # simple python script to solve the ODE and save as *xlsx
3  import xlsxwriter
4  import numpy as np
5  from scipy.integrate import odeint
6  import matplotlib.pyplot as plt
7
8  # define the ODE
9  def model(U,t):
10     # some key parameters
11     rhoF = 1000
12     rhoP = 2000
13     g = 9.81
14     D = 0.001
15     muF = 1.0e-3
16     Re = rhoF*D*U/muF
17     Cd = 24.0/Re*(1.0 + 0.15*pow(Re, 0.681)) + 0.407/(1.0 +
       8710.0/Re)
18     dUdt = (1.0 - rhoF/rhoP)*g - 3.0*rhoF/(4.0*rhoP*D)*Cd*pow
       (U, 2)
19     return dUdt
20
21 # initial condition
22 U0 = 1.0e-12     # U0 should be 0.0 but it leads to Re = 0.0 (
       Re is in the denominator)
23
24 # time step
25 t = np.linspace(0,0.3,1000) # 1000 points
```

```
26
27 # solve the ODE
28 U = odeint(model, U0, t)
29
30 # plot results
31 plt.plot(t, U)
32 plt.xlabel('t')
33 plt.ylabel('U')
34 plt.grid()
35 plt.show()
36 #plt.savefig("plot.png")
37 #plt.savefig("plot.pdf")
38
39 # save numerical data into *xlsx
40 workbook = xlsxwriter.Workbook("numerical.xlsx")
41 worksheet = workbook.add_worksheet()
42 format = workbook.add_format()
43 format.set_align("center")
44 worksheet.set_column("A:B", 20)
45 heading = ("t", "U")
46 worksheet.write_row("A1", heading, format)
47 worksheet.write_column("A2", t, format)
48 worksheet.write_column("B2", U, format)
49 workbook.close()
```

Code list A.5: Python code for solving the ordinary differential equation.

A.5 Quantities in different unit systems

In this thesis, some numerical simulations are conducted with "cgs" or "micro" unit system. These derived and basic quantities are listed in Tables A.1 and A.2, respectively.

Table A.1: Derived quantities used in the CFD-DEM simulations with OpenFOAM & LIGGGHTS

Quantity	SI	SI → CGS	SI → MICRO
Velocity	$[\frac{\mathrm{m}}{\mathrm{s}}]$	$[\frac{\mathrm{cm}}{\mathrm{s}}] \times 10^{2}$	$[\frac{\mu\mathrm{m}}{\mu\mathrm{s}}] \times 1$
Young's modulus	$[\frac{\mathrm{kg}}{\mathrm{m\,s^2}}]$	$[\frac{\mathrm{g}}{\mathrm{cm\,s^2}}] \times 10$	$[\frac{\mathrm{pg}}{\mu\mathrm{m}\,\mu\mathrm{s}^2}] \times 10^{-3}$
Pressure	$[\frac{\mathrm{kg}}{\mathrm{m\,s^2}}]$	$[\frac{\mathrm{g}}{\mathrm{cm\,s^2}}] \times 10$	$[\frac{\mathrm{pg}}{\mu\mathrm{m}\,\mu\mathrm{s}^2}] \times 10^{-3}$
Temperature	$[\mathrm{K}]$	$[\mathrm{K}] \times 1$	$[\mathrm{K}] \times 1$
Mass flux per unit area	$[\frac{\mathrm{kg}}{\mathrm{m^2\,s}}]$	$[\frac{\mathrm{g}}{\mathrm{cm^2\,s}}] \times 10^{-1}$	$[\frac{\mathrm{pg}}{\mu\mathrm{m}^2\,\mu\mathrm{s}}] \times 10^{-3}$
Mass flux per unit volume	$[\frac{\mathrm{kg}}{\mathrm{m^3\,s}}]$	$[\frac{\mathrm{g}}{\mathrm{cm^3\,s}}] \times 10^{-3}$	$[\frac{\mathrm{pg}}{\mu\mathrm{m}^3\,\mu\mathrm{s}}] \times 10^{-9}$
Kinematic viscosity	$[\frac{\mathrm{m}^2}{\mathrm{s}}]$	$[\frac{\mathrm{cm}^2}{\mathrm{s}}] \times 10^{4}$	$[\frac{\mu\mathrm{m}^2}{\mu\mathrm{s}}] \times 10^{6}$
Dynamic viscosity	$[\frac{\mathrm{kg}}{\mathrm{m\,s}}]$	$[\frac{\mathrm{g}}{\mathrm{cm\,s}}] \times 10$	$[\frac{\mathrm{pg}}{\mu\mathrm{m}\,\mu\mathrm{s}}] \times 10^{3}$
Density	$[\frac{\mathrm{kg}}{\mathrm{m}^3}]$	$[\frac{\mathrm{g}}{\mathrm{cm}^3}] \times 10^{-3}$	$[\frac{\mathrm{pg}}{\mu\mathrm{m}^3}] \times 10^{-3}$
Thermal conductivity	$[\frac{\mathrm{kg\,m}}{\mathrm{s}^3\,\mathrm{K}}]$	$[\frac{\mathrm{g\,cm}}{\mathrm{s}^3\,\mathrm{K}}] \times 10^{5}$	$[\frac{\mathrm{pg}\,\mu\mathrm{m}}{\mu\mathrm{s}^3\,\mathrm{K}}] \times 10^{3}$
Specific heat capacity	$[\frac{\mathrm{m}^2}{\mathrm{s}^2\,\mathrm{K}}]$	$[\frac{\mathrm{cm}^2}{\mathrm{s}^2\,\mathrm{K}}] \times 10^{4}$	$[\frac{\mu\mathrm{m}^2}{\mu\mathrm{s}^2\,\mathrm{K}}] \times 1$
Velocity potential	$[\frac{\mathrm{m}^2}{\mathrm{s}}]$	$[\frac{\mathrm{cm}^2}{\mathrm{s}}] \times 10^{4}$	$[\frac{\mu\mathrm{m}^2}{\mu\mathrm{s}}] \times 10^{6}$
Surface tension coefficient	$[\frac{\mathrm{kg}}{\mathrm{s}^2}]$	$[\frac{\mathrm{g}}{\mathrm{s}^2}] \times 10^{3}$	$[\frac{\mathrm{pg}}{\mu\mathrm{s}^2}] \times 10^{3}$
Gravitational constant	$[\frac{\mathrm{m}}{\mathrm{s}^2}]$	$[\frac{\mathrm{cm}}{\mathrm{s}^2}] \times 10^{2}$	$[\frac{\mu\mathrm{m}}{\mu\mathrm{s}^2}] \times 10^{-6}$
Enthalpy of evaporation	$[\frac{\mathrm{m}^2}{\mathrm{s}^2}]$	$[\frac{\mathrm{cm}^2}{\mathrm{s}^2}] \times 10^{4}$	$[\frac{\mu\mathrm{m}^2}{\mu\mathrm{s}^2}] \times 1$
Diffusion coefficient	$[\frac{\mathrm{m}^2}{\mathrm{s}}]$	$[\frac{\mathrm{cm}^2}{\mathrm{s}}] \times 10^{4}$	$[\frac{\mu\mathrm{m}^2}{\mu\mathrm{s}}] \times 10^{6}$
Gas constant	$[\frac{\mathrm{kg\,m}^2}{\mathrm{K\,mol\,s}^2}]$	$[\frac{\mathrm{g\,cm}^2}{\mathrm{K\,mol\,s}^2}] \times 10^{7}$	$[\frac{\mathrm{pg}\,\mu\mathrm{m}^2}{\mathrm{K\,mol}\,\mu\mathrm{s}^2}] \times 10^{15}$
Molar mass	$[\frac{\mathrm{kg}}{\mathrm{mol}}]$	$[\frac{\mathrm{g}}{\mathrm{mol}}] \times 10^{3}$	$[\frac{\mathrm{pg}}{\mathrm{mol}}] \times 10^{15}$
Surface energy density	$[\frac{\mathrm{kg}}{\mathrm{s}^2}]$	$[\frac{\mathrm{g}}{\mathrm{s}^2}] \times 10^{3}$	$[\frac{\mathrm{pg}}{\mu\mathrm{s}^2}] \times 10^{3}$

Table A.2: Basic quantities used in OpenFOAM for different unit systems

Quantity	SI	CGS	MICRO
Speed of light in vacuum	2.99792×10^{8} $[\frac{\mathrm{m}}{\mathrm{s}}]$	2.99792×10^{10} $[\frac{\mathrm{cm}}{\mathrm{s}}]$	2.99792×10^{8} $[\frac{\mu\mathrm{m}}{\mu\mathrm{s}}]$
Gravitational constant	6.67429×10^{-11} $[\frac{\mathrm{m}^3}{\mathrm{kg\ s}^2}]$	6.67429×10^{-8} $[\frac{\mathrm{cm}^3}{\mathrm{g\ s}^2}]$	6.67429×10^{-20} $[\frac{\mu\mathrm{m}^3}{\mathrm{pg}\ \mu\mathrm{s}^2}]$
Planck's constant	6.62607×10^{-34} $[\frac{\mathrm{kg\ m}^2}{\mathrm{s}}]$	6.62607×10^{-27} $[\frac{\mathrm{g\ cm}^2}{\mathrm{s}}]$	6.62607×10^{-13} $[\frac{\mathrm{pg}\ \mu\mathrm{m}^2}{\mu\mathrm{s}}]$
Elementary charge	1.60218×10^{-19} [C]	4.803204×10^{-10} [statC]	1.60218×10^{-7} [pC]
Electron mass	9.10938×10^{-31} [kg]	9.10938×10^{-28} [g]	9.10938×10^{-16} [pg]
Proton mass	1.67262×10^{-27} [kg]	1.67262×10^{-24} [g]	1.67262×10^{-12} [pg]
Atomic mass	1.66054×10^{-27} [kg]	1.66054×10^{-24} [g]	1.66054×10^{-12} [pg]
Boltzmann constant	1.38065×10^{-23} $[\frac{\mathrm{kg\ m}^2}{\mathrm{K\ s}^2}]$	1.38065×10^{-16} $[\frac{\mathrm{g\ cm}^2}{\mathrm{K\ s}^2}]$	1.38065×10^{-8} $[\frac{\mathrm{pg}\ \mu\mathrm{m}^2}{\mathrm{K}\ \mu\mathrm{s}^2}]$
Standard pressure	1.0×10^{5} $[\frac{\mathrm{kg}}{\mathrm{m\ s}^2}]$	1.0×10^{6} $[\frac{\mathrm{g}}{\mathrm{cm\ s}^2}]$	1.0×10^{2} $[\frac{\mathrm{pg}}{\mu\mathrm{m}\ \mu\mathrm{s}^2}]$
Standard temperature	298.15 [K]	298.15 [K]	298.15 [K]

B Publication list

Peer-reviewed journal articles (6)

- Guo, Q. and **Xia, H.***, 2023. A review of the Discrete Element Method/-Modelling (DEM) in agricultural engineering. Journal of Agricultural Engineering (Accepted).
- Chen, Z., Li, Z., **Xia, H.** and Tong, X.*, 2021. Performance optimization of the elliptically vibrating screen with a hybrid MACO-GBDT algorithm. Particuology, 56, pp.193-206.
- **Xia, H.***, Li, Z. and Tong, X., 2019. Modelling continuous materials using discrete element modelling: investigations on the effect of particle packing. Computational Particle Mechanics, 6(4), pp.823-836.
- Wu, X., Li, Z., **Xia, H.** and Tong, X.*, 2018. Vibration Parameter Optimization of a Linear Vibrating Banana Screen Using DEM 3D Simulation. Journal of Engineering & Technological Sciences, 50(3).
- **Xia, H.**, Tong, X.*, Li, Z. and Wu, X., 2017. DEM-FEM coupling simulations of the interactions between particles and screen surface of vibrating screen. International Journal of Mining and Mineral Engineering, 8(3), pp.250-263.
- Li, Z., Tong, X.*, **Xia, H.** and Yu, L., 2016. A study of particles looseness in screening process of a linear vibrating screen. Journal of Vibroengineering, 18(2), pp.671-681.

Conference paper (1)

- **Xia, H.*** and Kamlah M., Modelling Droplet Evaporation with an Improved Coupled Level Set and Volume of Fluid (i-CLSVoF) Framework, in Proceedings of the 8th World Congress on Mechanical, Chemical, and Material Engineering (HTFF'22). Prague, Czech Republic.

Paper in review (2)

- **Xia, H.*** and Kamlah, M., 2022. An improved Coupled Level Set and Volume of Fluid (i-CLSVoF) framework for droplet evaporation. arXiv preprint arXiv:2202.01019.
- **Xia, H.*** and Kamlah, M., 2023. Computational modelling of gas-liquid-solid multiphase free surface flow with and without evaporation. arXiv preprint arXiv:2307.09313.

Paper in preparation (1)

- **Xia, H.*** and Kamlah, M., An improved un-resolved CFD-DEM framework for gas-liquid-solid multiphase free surface flow with and without evaporation.