UNIVERSITY OF CALIFORNIA
Merced

Particle Methods for Mesoscale Flow Simulations and Mesh Generation

A Thesis submitted in partial satisfaction
of the requirements for the degree of

Doctor of Philosophy

in

Mechanical Engineering

by

Anupam Mishra

Committee in Charge:

Dr. Venkattraman Ayyaswamy, Chair

Dr. Jingxiao Xu

Dr. Sachin Goyal

Dr. Yanbao Ma

August 2022

The Thesis of
Anupam Mishra is approved:

Dr. Jingxiao Xu

Dr. Sachin Goyal

Dr. Yanbao Ma

Dr. Venkattraman Ayyaswamy, Committee Chairperson

August 2022

Particle Methods for Mesoscale Flow Simulations and Mesh Generation

To My Grandfather

Acknowledgements

This journey was possible with immense support from many people in my life. I am eternally grateful to my advisor Dr. Yanbao Ma and Dr. Sachin Goyal for considering my research interests and taking me under their wings. I will also be grateful to Dr. Arvind Gopinath and Dr. Venkattraman Ayyaswamy for their continuous guidance. I have had the opportunity to work in an interdisciplinary department and I am grateful to Dr. Jian-Qiao Sun and Dr. Ashlie Martini for that. Throughout my journey, I have grown as a numerical simulation researcher and I am thankful to Dr. Ahmed Hemeda, Dr. Mohsen Torabi, Dr. Krishna Shah, Dr. Arun K Raj, Dr. Mandeep Sekhon, Dr. Rajaseenivasan, and Dr. Jingxiao Xu who have helped me when needed.

This journey would not have been possible without all the people who have given me constant support throughout including friends, family, and loved ones and I am grateful to them.

Curriculum Vitæ
Anupam Mishra

Education

2012-2016 **Bachelor of Technology** Major in Chemical Engineering and Minor in Mechanical Engineering, Indian Institute of Technology (IIT) - Guwahati, Guwahati.

2016-2021 **Master in Science** Major in Mechanical Engineering, University of California Merced.

Experience

2016-2022 Teaching Assistant, University of California, Merced.

2019 Visiting Scholar University of Notre Dame, Notre Dame

2019 Visiting Researcher Los Alamos National Lab, Los Alamos

Publications

Journal Articles

1. **A Mishra**, A Hemeda, S Goyal, L Fu, Y Ma, Isotropic Unstructured Mesh Generation UsingNon-linear Spring Force Approach. *in preparation 2022.*

2. *AA Hemeda, **A Mishra**, J Palko, Y Ma, Effects on Bonding Strength from Thermal Pretreatment of Metallic Powders in Cold Spray. in preparation 2022.*

3. *AA Hemeda, **A Mishra**, J Xu, CT Wu, D Cote, M Siopis, IM Nault, VK Champagne, S-W Lee, M Aindow, A Nardi, JW Palko, Y Ma, Heterogeneous Distribution of Mechanical Properties of Single-Particle Cold Spray Impacts. Journal of Thermal Spray Technology 31 498-507, 2022.*

4. *J Tamayo, **A Mishra**, A Gopinath Ambient Fluid Rheology Modulates Oscillatory Instabilities in Filament-Motor Systems. Frontiers in Physics 478, 2022.*

5. ***A Mishra**, A Hemeda, M Torabi, J Palko, S Goyal, D Li, Y Ma, A simple analytical model of complex wall in multibody dissipative particle dynamics. Journal of Computational Physics 396 416-426, 2019.*

6. *A Hemeda, S Pal, **A Mishra**, M Torabi, M Ahmadlouydarab, Z Li, J Palko, Y Ma Effect of Wetting and Dewetting on the*

Dynamics of Atomic Force Microscopy Measurements. Langmuir 2019.

Conferences

1. ***A Mishra**, Y Ma, A Gopinath, Rebound and Sticking Dynamics of Droplets Impinging on Wettable Surfaces. Bulletin of the American Physical Society, American Physical Society, 23-26 November 2019, Seattle, WA, USA.*

2. *M Torabi, A A Hemeda, **A Mishra**, T Liu, Y Ma, Liquid Bridge Eruption for Mesoscale Gravure Printing using Multi-body Dissipative Paricle Dynamics Thermal Fluid Engineering Conference, 14-17 April 2019, Las Vegas, NV, USA.*

3. *T Liu, **A Mishra**, A A Hemeda, J Palko, Y Ma, A many-body Dissipative Particle Dynamics study of Coalescence induced JumpingThermal Fluid Engineering Conference, 2020, New Orleans, LA, USA.*

3. *T Liu, **A Mishra**, A A Hemeda, J Palko, Y Ma, Electrowetting-induced Droplet Jumping Simulation using Many-body Dissipative Particle Dynamics. Thermal Fluid Engineering Conference, 14-17 April 2019, Las Vegas, NV, USA.*

Presentations

1. ***Anupam Mishra**, Yanbao Ma, and Arvind Gopinath. Rebound and Sticking Dynamics of Droplets Impinging on Wettable Surfaces. APS Division of Fluid Dynamics Meeting Abstracts. 2019.*

2. ***Anupam Mishra**, Ahmed A Hemeda, and Yanbao Ma On the effect of surface curvature on droplet contact angle: Dissipative particle dynamics study. Thermal Fluid Engineering Conference, 4-7 March 2018, Fort Lauderdale, FL, USA.*

3. *Sachin Goyal, Ghazaale Mehrabadi, **Anupam Mishra** and Soheil Fatehiboroujeni. Modeling Nonlinear Dynamics of Biological Filaments in Continuum Limit. ASME-International Design Engineering Technical Conferences & Computers and Information in Engineering Conference, 6-9 August 2017,*
Cleveland, OH, USA ***Posters***

1. *Deniz Akpinaroglu, **Anupam Mishra**, Arvind Gopinath, Spatiotemporal dynamics of microswimmers using Brownian Dynamics. 20th Annual UC Systemwide Bioengineering Symposium, 2019*

Anupam Mishra

Continuum methods have been used for numerical simulation of solid and fluid for a long time but they have several constraints. For complex fluids, numerical simulations using continuum assumptions are very challenging especially when the phenomena occurring in the system are smaller than the continuum — limit below which fluid cannot be considered as a continuum. Thermal fluctuation (due to Brownian motion) become significant at micro and nanoscale. As a result, continuum methods are not directly applicable at these scales. On the other hand, when the continuum simulations work for simulations at other scales, they have an additional challenge(especially in structural dynamics simulations). When the dynamics of the solid simulations are extreme and high deformations occur in the system, the continuum methods that are based on the idea of small element/mesh fail. Generating a mesh, that can be adequate to the dynamics is very difficult and still one of the big challenges of the continuum simulations. Even if a high-quality mesh is generated, a severe mesh distortion may occur for certain dynamics. In those cases, it is important to remesh the system. Thus, having a good quality mesh is not enough, the meshing process should also be extremely fast. Furthermore, the frequency of remeshing depends on the quality of the mesh, so there needs to be a balance of quality and efficiency.

In fluid simulations, most continuum methods are based on the Eulerian approach of the flow field, i.e. system is studied by dividing into small elements/mesh (typically fixed) and fluid motion is studied through those points. In the Lagrangian specification of the flow field, fluid is assumed to be made of small particles. These particles are then tracked throughout the simulation to study the system behavior. These second methods of simulating fluid/solid fall into a general class called Particle methods. There are several kinds of particle methods such as Brownian Dynamics, Smoothed Particle Hydrodynamics, and Dissipative Particle Dynamics(DPD) that vary from each other on length scale, time scale, and model but have some similarities. Particle methods are very powerful and can be used to solve the challenges that are faced in the continuum method. In this dissertation, two problems that are encountered in simulating fluid and solid are tackled and solutions are proposed.

In simulating fluid using particle methods such as DPD, two major problems occur. First, particle density fluctuations near the solid-liquid interface, and second high computational cost due to solid wall. Particle density in DPD is defined as the local density of the particle. Density fluctuations occur due to the low solid wall particle density. One way to tackle it is increasing the solid wall particle density but that results in even higher computational cost. In this dissertation, first a modified version of DPD — multibody Dissipative Particle Dynamics(MDPD)— is chosen due to its capability of handling the free liquid surface. In the context of MDPD, a closed-form mathematical framework is developed to model the solid wall and the original particle wall is replaced by this proposed model. The model is derived using a modified conservative force and a combination of analytical and numerical integration. The wall model is simpler and computationally efficient than traditional approaches of modeling the wall. The proposed wall model is also capable of mimicking the high-density computational wall that results into extremely low (less than 1%) density fluctuations. Furthermore, another challenge in MDPD and particle method community, a discrepancy in wall density is also solved by normalizing the wall density with fluid density. Several examples of fluid-solid interaction are used and the superiority of the wall boundary model is demonstrated. Computational time using wall model is only $30\% - 50\%$ of the computational time using the traditional particle wall. A dynamic case of solid sphere transport inside a water droplet due to wettability gradient is also demonstrated. In that example, several parametric studies such as the effect of wettability gradient, droplet radius and solid sphere radius is studied. The proposed wall model can easily be extended to other particle-based methods such as SPH and density fluctuations can be reduced. Furthermore, lower computational time will enable to simulate larger systems for longer times.

In solid mechanics and structural dynamics simulation, there are three steps, pre-processing(creating the geometry and mesh), numerical simulation, and post-processing (analyzing the results from obtained data). Mesh generation is one of the critical stages in numerical simulation and it is still very challenging depending on the geometry and mesh requirement. The process of mesh generation becomes even more challenging when there is high deformation physics. In those cases, the mesh needs to be of high quality and sometimes even after that, remeshing is required. In this dissertation, a novel, simple and efficient model based on the particle method for isotropic unstructured mesh generation is proposed. The proposed method uses simple point particles with varying cutoff radii and densities. The cutoff radii dictate which surrounding particles exert force on the particle. Several force models are also proposed that affect the convergence speed and accuracy of the system. The mesh generation process follows these steps. First, a target number density of the particles is defined based on the desired refinement in the system. Next, a total number of particles is calculated which is a function of density contour and the system volume. The system is then filled with particles randomly and is set to reach equilibrium. Furthermore, several numerical schemes such as implicit and semi-implicit, and their effect on convergence speed and accuracy are also described. A Voronoi-Tessellation-Delaunay-Triangulation(VT-DT) is developed for two dimensions and three dimensions. VT-DT is used to create mesh from the particle position. The obtained mesh consists of triangular (two dimensions) and tetrahedral (three dimensions) elements. This proposed mesh generation method achieves higher mesh qualities than do its counterparts in prior works. For the mesh, several 2D and 3D benchmark cases are used to demonstrate the capability of the proposed method. Some non-engineering mesh examples are also demonstrated to showcase the robustness of the method. The proposed mesh generation model is simple, efficient, and can be used to generate mesh for any geometry and any level of refinement. Furthermore, the speed of the mesh generation enables this method to be used on applications where remeshing is needed.

Contents

List of Figures

List of Tables

Nomenclature

A	Attractive force amplitude
a	Acceleration
B	Repulsive force amplitude
c	Cell
d	Density
e	exponential function
f	Sample function
h	Smoothing length
i	Particle index i
j	Particle index j
k_B	Boltzman Constant
L_x	Domain length in x direction
m	Mass
P	Force
r	Displacement
t	Time
T	Temperature
V	Volume
v	Velocity
x	Position/independent variable
y	Sample function
$\mathbf{e}$	Unit vector
$\mathbf{F}$	Force vector
$\mathbf{v}$	Velocity vector
$\mathbf{W}$	Kernel function
$\mathbf{X}$	Position vector
$\frac{D<>}{D<>}$	Total derivative
$\frac{\partial}{\partial}$	Partial derivative
Δt	Timestep

Subscripts

0	initial
i	for particle i
j	for particle j
n	time step
q	Non dimensional length

Superscripts

C	Conservative
D	Dissipative
R	Random
B	Body
$*$	In SI units
α	For particle α
β	For particle β

Greek Variables

ω	Weighting function
γ	Dissipative coefficeint
κ	Spring constant
ζ	Random force amplitude
σ	Surface tension
ν	Kinematic viscosity
ρ	Number density
δ	Dirac delta function

Chapter 1

Introduction

Computational Fluid Dynamics(CFD) is a branch of science that solves fluid flow problems using numerical tools. There are three steps involved in any CFD study, pre-processing, simulation, and post-processing. Pre-processing is where the domain is defined and discretized. In this step, any additional or special boundary or initial conditions are also defined. In the simulation step, governing partial differential equations are converted into linear algebraic equations are solved using iteration. The third and final step is to post-process the simulation and analyze the obtained data. CFD has certain advantages and disadvantages compared to experimental studies.

In CFD and structural mechanics simulations, there are two broad numerical techniques to represent the flow/dynamics and simulate it, the Eulerian approach and Lagrangian Approach. Both these approaches are based on how the flow field is represented in them. These will be explained below.

1.1 Computational Approaches of Modeling

1.1.1 Eulerian Approach

In the Eulerian approach, the flow field is specified by a function of position $\mathbf{x}$ and time t. For example, the velocity can be represented by

$$\mathbf{v} = f(\mathbf{x}, t) \tag{1.1}$$

In other words, a control volume is chosen initially and usually remains stationary in the Eulerian approach. In the next step, the control volume is discretized in small elements (mesh), and the governing equations are solved for those mesh elements.

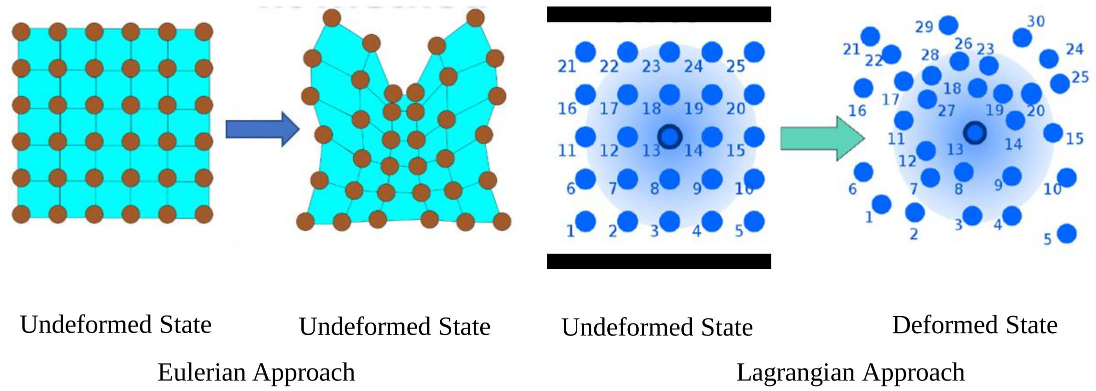


Figure 1.1: The schematic of the Eulerian and Lagrangian approaches and their state in high deformation dynamics

1.1.2 Lagrangian Approach

In the Lagrangian approach, fluid/solid are assumed to be made of small particles. Each fluid/solid particle is tracked through time. As a result, the control volume in the Lagrangian approach is always changing. If the particle position is $\mathbf{x}_0$ at any time t_0, then the dynamics can be determined by $\mathbf{X}(\mathbf{x}_0, t)$.

Unlike the Eulerian approach, the Lagrangian approach is a meshless method, and the governing equations are solved for particles. In the Lagrangian approach, there is no convective term as each grid node follows the path of material at the grid point. In the Lagrangian approach, mesh (the particles) deforms as the mass/material deforms, unlike Eulerian approach where mesh remains the same while the mass/material deformation occurs (see Fig. 1.1). Thus, methods based on Lagrangian approach are also called Mesh-free Particle Methods (MPM).

1.2 Mesh-free Particle Methods

In this section, Mesh-free Particle Methods(MPM) will briefly be introduced. Meshfree Particle Methods are based on the Lagrangian description and have no mesh in their calculation. The governing equations are solved for each particle. MPM can be divided into several classes according to their length scale such as atomistic/microscopic, mesoscopic, and macroscopic. Examples of atomistic, mesoscale and macroscale methods are molecular dynamics method [6]), dissipative particle dynamics (DPD) [7], and Smoothed particle hydrodynamics (SPH) [8], respectively.

MPMs have significant advantages over mesh-based methods [9].

- One advantage of MPM over mesh based method is that there is no need to prescribe any connectivity between the particle. Usually, mesh connectivity need to be defined for mesh-based method.
- Control volume does not need to be described for MPM, the only requirement for MPM is a distribution of the particle and definition of the geometry of the problem.
- Another advantage of using Meshfree Particle Methods is handling particles in large deformation simulation. There is no need for remeshing in particle methods unlike mesh based methods.
- The MPMs can incorporate an enrichment of fine-scale solutions of features.
- Particles can be added easily to the area where refinement is needed.
- MPMs are very powerful for simulations where material failure is involved because the damage of the components, such as fracture can easily be handled.

In flow simulation using MPM, the phenomena are formulated by the interacting particles that have the physical properties of the flow. In this section, a brief overview of some of the particle methods and the idea behind them is described. The particles' motion is governed by the solution of following Ordinary Differential Equations(ODEs)[10]

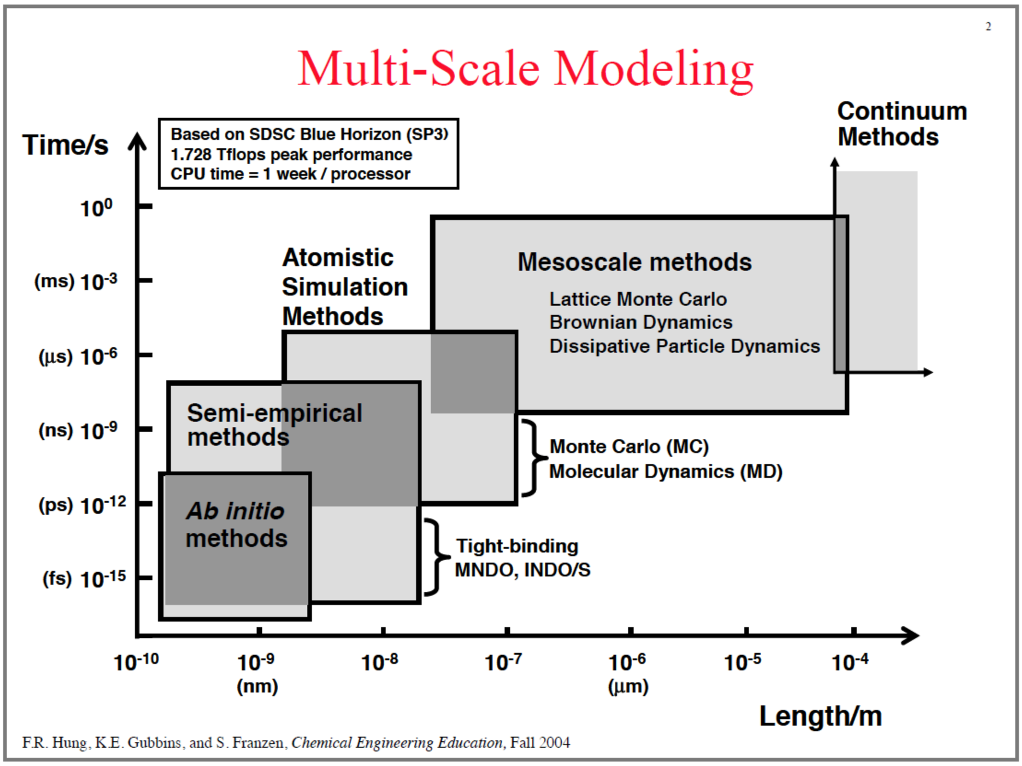
$$\frac{d\mathbf{r}}{dt} = \mathbf{v} \tag{1.2}$$

$$\frac{d\omega_p}{dt} = \Sigma\mathbf{F}(\mathbf{r}_p), \mathbf{r}_q, \omega_p, \omega_q) \tag{1.3}$$

where $\mathbf{x}_p, \mathbf{v}_p$ denote the location and the velocity of p^{th} particle. Here $\mathbf{F}$ represents the dynamics of the simulated physical system and ω_p is some property of the particle (e.g, density, velocity etc.).

In this work, the flow simulations that are considered belong to the mesoscopic scale. Mesoscopic scale (Mesoscale) in physics is a length and time scale which is somewhere in the middle of continuum scale and atomic scale. The length and time scale is also shown in figure 1.2.

The smallest length and time scale shown in the figure is nanometers $(10^{-9}m)$ and picoseconds $(10^{-12}s)$ scale and largest scales are higher than centimeters and seconds. In smaller scales (Atomistic Simulation and smaller), the physics is modeled using either the atoms or semi-empirical methods. In continuum methods, the fluid(and solid) is considered to be a continuum. In continuum scales, the fluid is usually modeled using Navier-Stokes equations. In most of cases, the domain is discretized into small elements and Navier-Stokes equations are solved for the small elements. As mesoscale methods are in the middle of the continuum and atomistic method, their formulation can be derived in two ways. A bottom-up approach from the atomistic method, which usually means a coarse-grained treatment to the atomistic method. For example, dissipative particle dynamics can be understood as a coarse-grained treatment of molecular dynamics. Alternatively, the mesoscale formulation can also be derived using a top-down approach. For example, smoothed particle hydrodynamics' formulation is derived from taking a particle approximation of Navier-Stokes equations.



Multi-Scale Modeling

Based on SDSC Blue Horizon (SP3)
1.728 Tflops peak performance
CPU time = 1 week / processor

F.R. Hung, K.E. Gubbins, and S. Franzen, *Chemical Engineering Education*, Fall 2004

Figure 1.2: The length and time scales for multiscale modeling

1.3 Motivation and Structure of the Thesis

This dissertation is divided into two major sections that fit in an overall particle method umbrella. Particle-based methods are a group of mesh-free methods that involve discreet particles to represent and simulate the system. In the mesh-based methods, the system is divided into small grids/mesh and the physical properties are defined on the center and faces of those meshes, however in particle-based method the particles are arbitrarily distributed and physical properties are assigned to those particles. Due to the particles having their own properties, the complex structure and deformations can be simulated with higher accuracy. Computational costs are usually higher for particle-based methods than mesh-based methods due to large number of particles.

Dissipative Particle Dynamics (DPD) is a mesoscopic stochastic particle-based technique[11]. DPD is mainly used for simulating dynamic and rheological behavior of the simple and complex fluids. Multi-body Dissipative Particle Dynamics(MDPD) is a modified version of conventional dissipative particle dynamics that is capable of simulating fluid-fluid and fluid-solid interactions and is capable of capturing the fluid-fluid interface. In MDPD, modeling fluid-structure interaction is often computationally expensive due to the large number of particles. In chapter 3, a mathematical framework to model solid wall boundary in MDPD for fluid-structure interaction problems is proposed. This wall boundary approach is capable to save significant computation time when doing fluid-structure interaction problem.

In the chapter 4, an application of the boundary wall model is demonstrated. A solid sphere within a liquid droplet is transported due to the wettability gradient on the surface. The wall in this case is modeled using the wall boundary model derived in chapter 3. Several parametric studies are done for this case such as the effect of the wettability gradient and size of the solid sphere. The wall boundary model can save more than 70% of computational in the simulations of solid sphere transport.

One problem often encountered in the continuum method is being able to generate the mesh. It is usually computationally expensive to generate an unstructured mesh. There have been some works on addressing the mesh generation problem in the literature but most of them are either very complex or computationally expensive. A general particle model[12] has also been used recently to generate the mesh but it involves placing particles layer by layer which can be computationally expensive for a large domain. In chapter 2, A particle-based model is proposed that is capable of generating dynamic unstructured mesh. Chapter 5 shows some results of the generated mesh using this model. A triangulation scheme based on Voronoi Tessellation and Delaunay Triangulation is also presented, that is used to convert from particles positions to mesh. Several two-dimensional and three-dimensional cases are used to demonstrate the model's capability.

Chapter 2

Particle Dynamics based Computational Models

In this chapter, the formulation of particle methods will be described. The particle methods are meshless methods where the property of the system is carried via the point particles or particles with definite radii. Depending on the type of particle method, the particles may have mass, viscosity, strain, etc. The particle methods are fundamentally different from the traditional mesh-based methods, such as finite element methods and finite volume methods.

There are two fundamental approaches to describing any physical system using governing equations: the Eulerian and the Lagrangian description. The big difference between these two approaches is that while Eulerian description is a spatial description, Lagrangian is a material description approach. In both approaches, the equations follow similar steps to reach finalized equations that are solved numerically. For example, if we take fluid mechanics then there will be mass momentum and energy conservation equations in both approaches. Another way to understand the difference between these two approaches is from the definition of total time derivative. Total time derivative can be expressed as the sum of local derivative and convective derivative.

$$\frac{D}{Dt} = \frac{\partial}{\partial t} + v^{\alpha}\frac{\partial}{\partial v^{\alpha}} \tag{2.1}$$

where $\frac{D}{Dt}$ is a total time derivative and $\frac{\partial}{\partial t}$ is a partial derivative and the last term is a convective derivative.

In this chapter, the governing equations of two particle methods will be described and then it will be followed by numerical schemes. First, a version of the general particle model is proposed that will be used to generate the unstructured mesh in chapter 5. Next, multi-body dissipative particle dynamics(MDPD) is described. Using this mode, a mathematical framework is developed to model the solid wall boundary condition in chapter 3. The wall boundary model will also be used in chapter 4 to demonstrate an application.

For both particle methods, a cell-linked list is used to avoid redundancy in computational cost and is explained in section 2.2. Cell-Linked list is an efficient way of handling all the neighbors of any particle in the code. Instead of searching through all the particles and calculating force, a cell-linked list allows reducing this number significantly. Numerical schemes for all the models are given in the final section.

2.1 Governing Equations for Each Model

2.1.1 General Particle Model

The motion of each particle is governed by Newton's second law of motion as

$$m_i \frac{d^2\mathbf{r}_i}{dt^2} = m_i \frac{d\mathbf{v}_i}{dt} = \mathbf{F}_i = \sum_j F_{ij} \frac{\mathbf{r}_{ij}}{r_{ij}} \tag{2.2}$$

where $\mathbf{F}_i$ is net force on particle i due to interaction with all the neighboring particles j, while $\mathbf{r}_i$ and $\mathbf{v}_i$ are the position and velocity vectors of the particle i, respectively. F_{ij} is the force on i^{th} particle due to j^{th} particle and m_i is the mass of particle i. In the next section, a detailed description of this force is given.

Force Field

In this section, the force field for the proposed model is presented.

Moaghan[13] used the cut-off distance h_{ij} in smoothed particle hydrodynamics where particles exert force on each other if they lie within a cut-off distance h_{ij}. In this work, the concept of cut-off distance is adopted in the general particle model. In this proposed model, the particles i and j are experience force if they are within a cutoff distance h_{ij} which may vary with particles i and j. h_{ij} is a function of the cut-off distance of particle i and j, density and the mass of the particle i. h_{ij} is defined as

$$h_{ij} = \frac{h_i + h_j}{2}, h_i = \left(\frac{m_i}{\rho}\right)^{(1/d)} \tag{2.3}$$

Here d represents the dimension of the system (e.g., $d = 3$ for three dimensional system) and ρ is the "density field" adopted from smoothed particle hydrodynamics[13] calculated as

$$\rho = \sum_j m_j W_{ij} \tag{2.4}$$

neighborhood definition where W_{ij} is any Kernel function[13] centred at i. One example of Kernel function is a Gaussian function. The dimensions of Kernel function are $1/L^d$. Kernel function will be explained in section 2.1.2.

The particles exerts only repulsive force and the force profile is shown in Figure 2.1b.

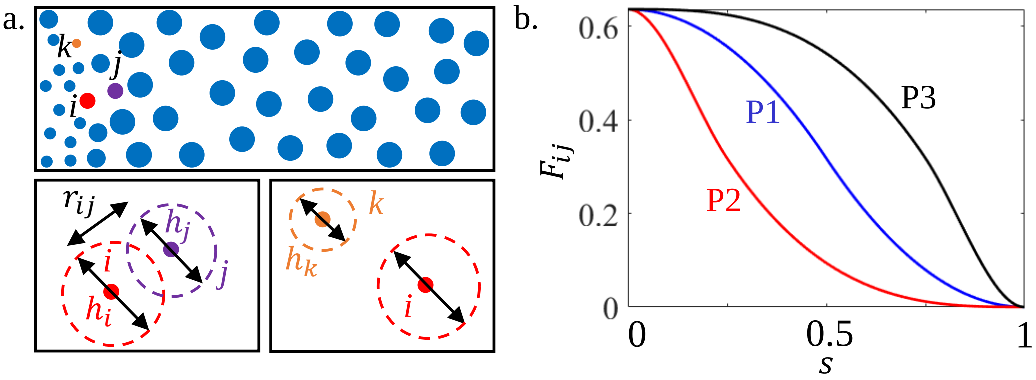


Figure 2.1: The schematic configuration of particles is shown on top in (a). For particle i (red particle), the neighbor j and k are shown below. The particle j and i exerts forces on each other but k and i do not exert any force. The normalized force field distribution is shown in (b)

The force profile is not unique, three sample force profiles $P1$, $P2$ and $P3$ are shown in Fig. 1.1

$$P1 : F_{ij}(s) = \frac{Km_i m_j}{h_{ij}^{d+1}} \begin{cases} -2s^2 + 1 & \text{if } 0 \leq s < 0.5 \\ 2s^2 - 4s + 2 & \text{if } 0.5 \leq s < 1 \\ 0 & \text{if } 1 \leq s \end{cases} \tag{2.5}$$

where $s = \frac{r_{ij}}{h_{ij}}$ and $r_{ij} = |\mathbf{r}_i - \mathbf{r}_j|$ is a displacement vector between particles i and j. Here K is chosen to be 0.6 m^d/s^2Kg for this work. The force profile is not unique and there may be other force profiles as shown in Fig. 2.1b.

2.1.2 Dissipative Particle Dynamics

In dissipative particle dynamics, the system is filled with point particles that interact with each other with three types of forces: conservative force, a dissipative force and a stochastic(random) force. The dissipative and stochastic force can also be called as a ‘pair-wise Brownian dashpot’(Figure 2.2). This Brownian dashpot is momentum conserving (unlike Langevin or Brownian dynamics) that makes the system hydrodynamic at a large scale.

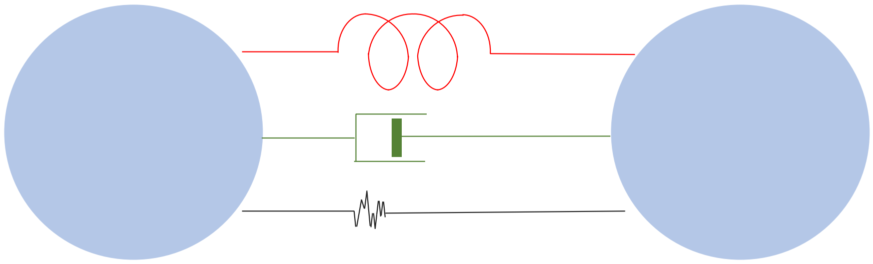


Figure 2.2: Schematic of interaction between two particles in Dissipative Particle Dynamics showing conservative, dissipative, and stochastic forces

Multibody Dissipative Particle Dynamics (MDPD) is a modified version of DPD which is capable of modeling fluid/fluid and fluid/solid interfaces [7]. Several interesting fluid-solid phenomena such as liquid drainage [14], droplet detachment from an AFM tip [15], and several other complex fluid-structure interaction phenomena [16][17][18] have been investigated using DPD and MDPD.

In MDPD, the fluid and solid are made of particles and are governed by Newton’s second law of motion. In the current study, the solid wall is assumed to be frozen for simplicity ([19, 20, 21, 22]) thus force calculation is done only for fluid particles. The equations of motion for each fluid particle are defined as

$$m_i \frac{d^2\mathbf{r}_i}{dt^2} = m_i \frac{d\mathbf{v}_i}{dt} = \mathbf{F}_i = \sum_{(j\neq i)} \mathbf{F}_{ij}^C + \mathbf{F}_{ij'}^D + \mathbf{F}_{ij}^R + \mathbf{F}_B \tag{2.6}$$

where $\mathbf{r}_i$, $\mathbf{v}_i$ and $\mathbf{F}_i$ denote the position, velocity, and net force exerted on particle i by other particles $j \neq i$, respectively. The net force $\mathbf{F}_i$ in Eq. 2.6 includes body force $\mathbf{F}_B$, conservative force $\mathbf{F}_{ij}^C$, dissipative force $\mathbf{F}_{ij}^D$ and random force $\mathbf{F}_{ij}^R$ which are given by [23]:

$$\mathbf{F}_{ij}^C = A\, \omega_C(r_{ij}, R_c)\ \mathbf{e}_{ij} + B\,(\overline{\rho}_i + \overline{\rho}_j)\, \omega_D(r_{ij}, R_d)\ \mathbf{e}_{ij} \tag{2.7}$$

$$\mathbf{F}^{D}_{ij} = \gamma\, \omega_D(r_{ij}, R_c)\ \ (\mathbf{e}_{ij}.\mathbf{v}_{ij})\, \mathbf{e}_{ij}, \tag{2.8}$$

$$\mathbf{F}^{R}_{ij} = \phi\, \omega_R(r_{ij}, R_c)\, \theta_{ij}\, \left(\frac{1}{\sqrt{\Delta t}}\right)\, \mathbf{e}_{ij} \tag{2.9}$$

Conservative forces accounts for material type, dissipative force is accountable for the viscosity in the fluid and the Random force accounts for the thermal fluctuations in the system. In Eqs. 2.7-2.9, $\vec{r}_{ij} = \vec{r}_i - \vec{r}_j$ is the position vector from j^{th} particle to the i^{th} particle with $r_{ij} = |\vec{r}_{ij}|$ and $\vec{e}_{ij} = \vec{r}_{ij}/r_{ij}$. Furthermore, A_{ij} is the attractive force amplitude and B the repulsive force amplitudes along with weighting functions of ω_C and ω_D with a cut-off of radii R_c and $R_d = 0.75R_c$, respectively. In the random force expression, the term $\frac{1}{\sqrt{dt}}$ comes from integration of stochastic differential equation and interpreting the random force as a Wiener process[24]. The weighting functions are given by

$$\omega_C(r_{ij}, R_c) = max(1 - r_{ij}/R_c, 0) \tag{2.10}$$

$$\omega_D(r_{ij}, R_d) = (max(1 - r_{ij}/R_D, 0))^2 \tag{2.11}$$

In the rightmost expression of Eq. 2.7 local density function $\bar{\rho}_i$ is present which denotes how crowded the particles in the local vicinity. Warren [25] proposed the density-dependent repulsion in the conservative force formula empirically and the local density-function was given as [24]

$$\bar{\rho}_i = \frac{15}{2\pi(R_d)^3} \sum_{(j\neq i)} (1 - r_{ij}/R_d) \tag{2.12}$$

In Eqs. 2.8 - 2.9, $\mathbf{v}_{ij} = \mathbf{v}_i - \mathbf{v}_j$ is the velocity of the i^{th} particle relative to the j^{th} particle. The scalars γ and ϕ are the amplitudes of dissipative and random forces, respectively. The system satisfies the fluctuation-dissipation theorem [26] and thus the amplitude of dissipative and random forces are related by the expression

$$\omega_R(r_{ij}, R_c) = \sqrt{(\omega_D(r_{ij}, R_c))} = max(1 - r_{ij}, 0) \tag{2.13}$$

and $\phi^2 = 2\gamma k_B T$ where k_B is Boltzmann constant and T is the absolute temperature of the system[24]. The parameter θ_{ij} is Gaussian white noise with unit bandwidth, and Δt is the time step. Lastly, $\mathbf{F}_B$ accounts for body forces such as gravitational force.

The particle and solid wall particles have soft potential interaction, thus particle penetration is observed for hydrophilic and super hydrophilic systems. To prevent penetration inside the wall, different reflection treatments have been implemented [27][28]. Three reflection modes are popular to achieve impenetrability:

Bounce-back, Specular reflection, and Maxwellian reflections (see Figure 2.3). In bounce-back reflection, the particle is reflected along the same ray that it follows when it impacts the solid wall. The normal and tangential components of the velocity and force are reversed. In Maxwellian reflection, the particle penetrating the solid wall returns with velocity in a Maxwellian distribution. The center of the Maxwellian distribution lies at the wall velocity. In the Specular reflection, only the normal component of the velocity and the forces are reversed. Revenga [27] proved that there is a slip velocity in the case of both Specular and Maxwellian reflection. The bounce-back reflection guarantees the no-slip boundary condition. On similar note, Visser et al.[29], proposed another reflection mechanism for boundary treatment called bounce-forward. In their method, the displacement tangential to the wall during a reflection is preserved, while the position perpendicular to the wall and the velocity change is the same as the original bounce-back reflection method. In this work, bounce-back reflection has been used.

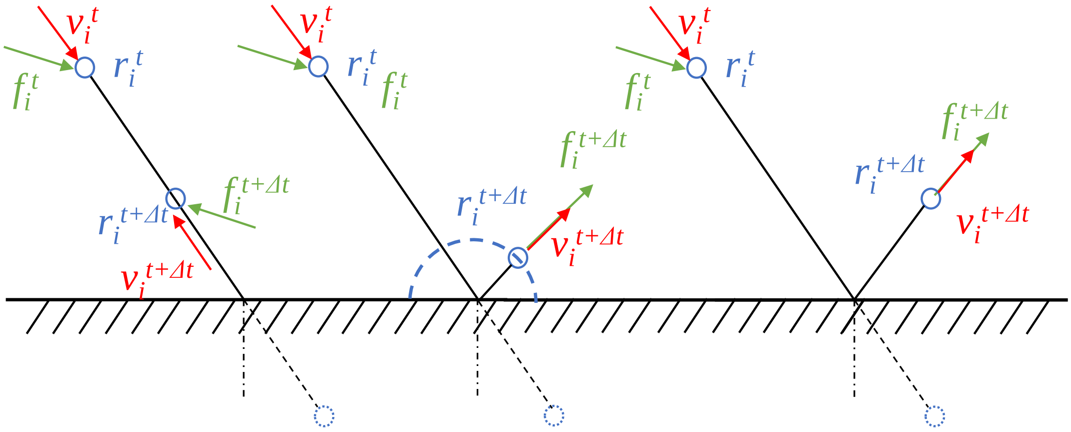


Figure 2.3: Boundary condition in MDPD simulations (bounce-back, Specular, and Maxwellian reflections)

Scaling the Domain

In MDPD, reduced units are commonly used for all quantities. The particle mass, cut-off radius R_c, and system energy are set to be one in MDPD units. The parameters in Eqs. 2.7 - 2.9 for the current modeling approach are listed in Table 2.1 in MDPD units. One of the many advantages of the MDPD (or DPD) method is that it is a scale-free simulation technique which means one can choose any set of reference parameters, and the results would still hold unchanged. In the

Parameters	Symbol	MDPD unit
Time Step	δt	0.01
Particle mass	m	1.0
Cutoff distance	R_c	1.0
Repulsive interaction range	R_d	0.75
Amplitude of $\mathbf{F}_D$	γ	5.61
Amplitude of $\mathbf{F}_{C,A}$	A_{ij}	-40
Amplitude of $\mathbf{F}_{C,B}$	B	25
System Energy	$k_B T$	1.0

Table 2.1: Sample Parametric values for MDPD simulations.

present study, the fluid has no viscous effect because it is quasi-static (low capillary number). The reference length is chosen as LDPD=1 μm. By assuming the fluid density and surface tension in the MDPD units to be d and σ, respectively, the conversion of mass and time from DPD units to physical ones are

$$M_{DPD} = \frac{d^*}{d} L_{DPD}^3 \tag{2.14}$$

$$T_{DPD} = \sqrt{M_{DPD} \frac{\sigma}{\sigma^*}} \tag{2.15}$$

where d^* and σ^* are the physical density and surface tension of the fluid in the real units (e.g., SI unit)[26]. The parameters in Eqs. 2.7-2.9 in the current modeling approach are listed in Table 2.1 in the MDPD units. From the parametric values provided in Table 2.1, the fluid properties can be calculated. The fluid density, kinematic viscosity and surface tension are calculated as $d = 6.0, \nu = 7.45$ and $\sigma = 7.31$ in the DPD units. For water as the fluid, the reference scales are found to be $T_{DPD} = 1.29 \times 10^{-7}$ s, and $M_{DPD} = 1.667 \times 10^{-16}$ kg. These scales are conversion factors from the DPD unit system to the physical SI unit system for water as the fluid.

Using the above-described MDPD model, a wall boundary model is proposed(chapter 3) that is more computationally efficient that the current particle approaches.

Kernel Function

The field function approximation of any function $f(x)$ starts with

$$f(x) = \int_{\Omega} f(\mathbf{x'})\delta(\mathbf{x} - \mathbf{x'})dx' \tag{2.16}$$

where f is a function of position vector $\mathbf{x}$ and $\delta(x - x')$ is a Dirac delta function that is given by

$$\delta(x - x') = \begin{cases} 1 & \text{if } x = x' \\ 0 & \text{if } x \neq x' \end{cases} \tag{2.17}$$

Also, Ω is the volume of the integral that contains $\mathbf{x}$. In Eq. 2.17, Dirac delta function can be replaced by a smoothing function. The equation represents that a function is represented in integral form.

$$f(\mathbf{x}) = \int_{\Omega} f(\mathbf{x'})\,\boldsymbol{W}(\mathbf{x} - \mathbf{x'}, h)dx' \tag{2.18}$$

where $\boldsymbol{W}$ is named smoothing or kernel function. A Kernel function is usually chosen to be an even function that satisfies certain conditions. These conditions are shown below. The first condition is normalization or unity condition

$$\int_{\Omega} \boldsymbol{W}(\mathbf{x} - \mathbf{x'}, h)d\mathbf{x'} = 1 \tag{2.19}$$

The second condition is called delta function properties, it is observed when smoothing length h approaches zero

$$\lim_{h \to 0} \boldsymbol{W}(\mathbf{x} - \mathbf{x'}, h) = \delta(\mathbf{x} - \mathbf{x'}) \tag{2.20}$$

The third condition is compact condition

$$W(x - x', h) = 0 \ \ when \ \ |\mathbf{x} - \mathbf{x'}| > \kappa h \tag{2.21}$$

κ is a constant related to smoothing function for point at $\boldsymbol{x}$.

The integral approximation for the spatial derivative can be obtained

$$< \Delta.f(\mathbf{x}) >= \int_{\Omega} \nabla.f(\mathbf{x'})\,\boldsymbol{W}(\mathbf{x} - \mathbf{x'}, h)d\mathbf{x'} \tag{2.22}$$

which can further be simplified using Divergence theorem

$$< \nabla.f(\mathbf{x}) >= \int_{\Omega} f(\mathbf{x'})\nabla\,\boldsymbol{W}(\mathbf{x} - \mathbf{x'}, h)d\mathbf{x'} \tag{2.23}$$

The continuous integral representation can be written in form of discretized particle approximation

$$f(\mathbf{x}) = \sum_{j=1}^{N} \frac{m_j}{\rho_j} f(\mathbf{x}_j)\, \boldsymbol{W}(\mathbf{x} - \mathbf{x}_j, h) \tag{2.24}$$

where N is the number of particle in support domain of the particle i. The particle approximation for any particle i can be written[8]

$$f(\mathbf{x}_i) = \Sigma \frac{m_j}{\rho_j} f(\mathbf{x}_j)\, \boldsymbol{W}_{ij} \tag{2.25}$$

Similarly,

$$< \nabla . f(\mathbf{x}_i) >= - \sum_{j=1}^{N} \frac{m_j}{\rho_j} f(\mathbf{x}_j) \nabla \boldsymbol{W}_{ij} \tag{2.26}$$

where

$$\boldsymbol{W}_{ij} = \boldsymbol{W}_{ij}(\mathbf{x}_i - \mathbf{x}_j, h) \tag{2.27}$$

$$\Delta \boldsymbol{W}_{ij} = \frac{\mathbf{x}_i - \mathbf{x}_j}{\mathbf{r}_{ij}} \frac{\partial \boldsymbol{W}_{ij}}{\partial \mathbf{r}_{ij}} = \frac{\mathbf{x}_{ij}}{\mathbf{r}_{ij}} \frac{\partial \boldsymbol{W}_{ij}}{\partial \mathbf{r}_{ij}} \tag{2.28}$$

To summarize, the above particle approximation converts continuous integral representations and its derivatives to the discretized summation based on set of particles. If we put density function $\rho(\mathbf{x})$ then approximation becomes

$$\rho_i = \sum_{j=1}^{N} \frac{m_j}{\rho_j} \rho_j W_{ij} \tag{2.29}$$

Furthermore, the support function and neighbor radius assumed is a circle, but it does not necessarily need to be a circle. If particles are needed to be more refined in one axis than the other, then an elliptical support function may also be chosen(as discussed briefly in chapter 6).

2.2 Cell-Linked List Strategy to Avoid Computational Redundancy

In all the particle methods, the force calculation steps involve going over all the particles for each particle. The double loop over N particles is in order of $\mathcal{O}(N^2)$. In an actual simulation with a cutoff radius of r_c, the particle only interacts with only a small number of elements that can be approximated as $4\pi/3r_C^2\rho_d$ where ρ_d

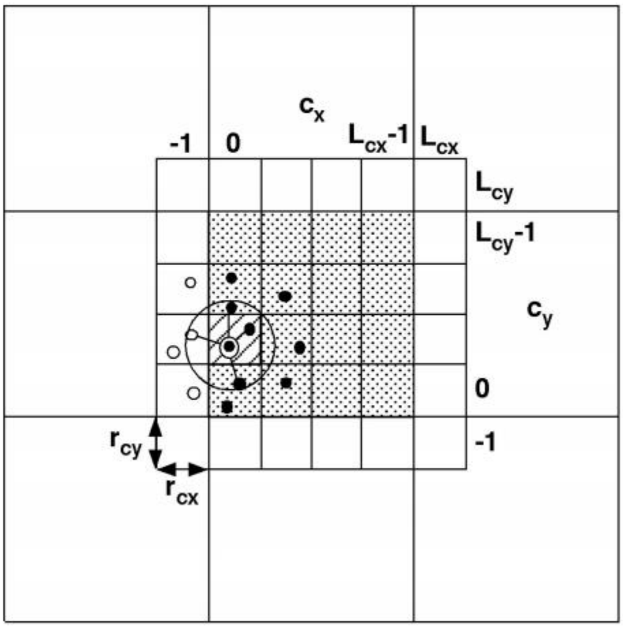


Figure 2.4: Schematic of cell and particle distribution in the domain

is the number density. The cell linked list algorithm reduces this complexity from $\mathcal{O}(N^2)$ to $\mathcal{O}(N)$.

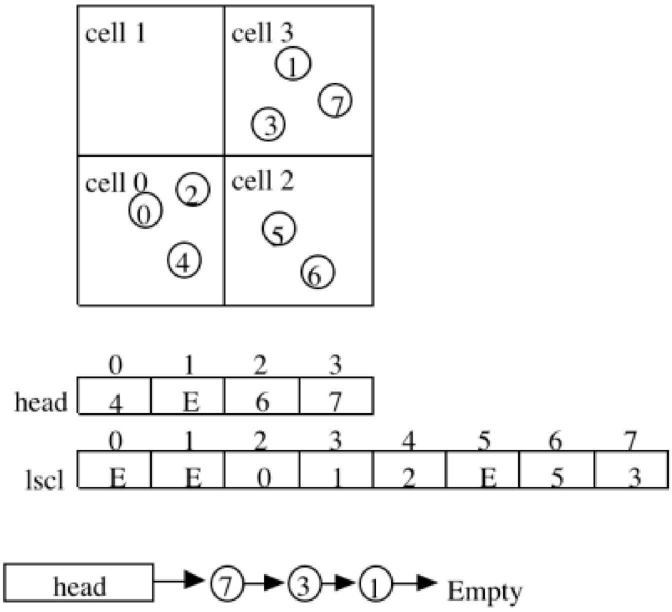
First of all, the system domain is divided into small cells (see Fig. 2.4), where the edge of these small cells must be at least r_C. A good value can be $r_{ci} = L_i/L_{ci}$ where $L_{ci} = [L_i/r_C]$. $[-]$ is a floor function, that is the largest integer less than or equal to that number. If a particle is located in any cell, then all its interaction particles must be in the 26 cells (in the case of three-dimensional system) that surrounds the cell. If there is a cell with index $\overrightarrow{c} = (c_x, c_y, c_z)$ with $0 \leq c_x \leq L_{cx} - 1$, $0 \leq c_y \leq L_{cy} - 1$, $0 \leq c_z \leq L_{cz} - 1$.

$$c = c_x L_{cy} L_{cz} + c_y L_{cz} + c_z \tag{2.30}$$

To find the cell of any particle with position $\overrightarrow{r}$,

$$[htbp] c_i = \lfloor r_i / r_{ci} \rfloor \tag{2.31}$$

These particles in one cell are arranged in a linked list. The flow chart in Fig. 2.5 describes the implementation of the cell-linked list. This is one of many possible implementations of the cell list. In the diagram, lscl is an array implementation of the linked lists and lscl[i] holds the particle index to which the



	0	1	2	3
head	4	E	6	7

	0	1	2	3	4	5	6	7
lscl	E	E	0	1	2	E	5	3

Figure 2.5: A sample data structure showing cell-linked list algorithm

i^{th} particles points. head holds the index of the first particle in the c^th cell, or $head[c] = EMPTY$ if there are no particles in the cell.

2.3 Numerical Scheme

Particles follow Newton's second law of motion to evolve. For the general particle model, the main goal is to optimize the computational time and focus on the speed of the numerical method. Similarly, speed is the main focus for MDPD. Verlet integration is used for these two models as it is a stable and fast method.

2.3.1 Euler Method

The Euler method is the simplest way to integrate or solve ordinary differential equations(ODE). Euler methods also introduce Runge Kutta methods that are also popular for solving ODE.In simple terms, Euler methods can be understood as simply Taylor expansion

$$f(x) = f(x_1) + \frac{1}{1!}\frac{df(x_1)}{dx_1}(x - x_1) + \frac{1}{2!}\frac{d^2 f(x_1)}{dx_1^2}(x - x_1)^2 ... \tag{2.32}$$

Now, if this equation is compared with well know Newton's equation

$$x(t) = x_0 + vt + \frac{1}{2}at^2 \tag{2.33}$$

With above equations, Euler method can be expressed as

$$\frac{dy(t)}{dt} = f(t, y(t)) \tag{2.34}$$

$$y_{n+1} = y_n + f(t_n, y_n)dt \tag{2.35}$$

The euler method approximates the solution of an ODE by using only first derivative. This approximation is bad, and errors associated with this approximation can add up in some cases (Fig. 2.6). In the Euler method, the only way to counter this issue is to decrease the time step, or use a similar more stable method such as backward Euler method. In other words, the Euler method is not always stable and often causes instability if the time step is not small enough. In the case of simple kinetics, another method called Verlet integration may be better suited than the forward Euler method.

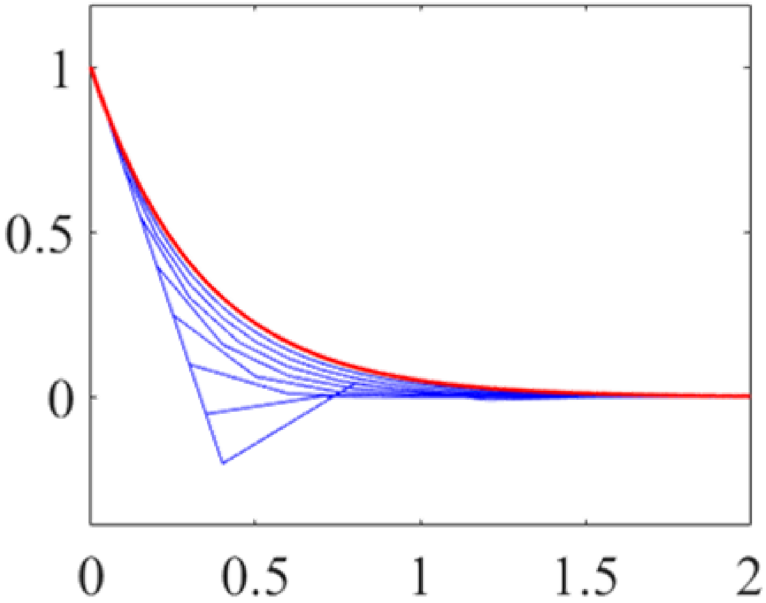


Figure 2.6: Ordinary differential equation $f(x) = e^{-x}$ solution using different discretization. The solutions show negative values for a higher dx that are impossible. Red curve is the exact solution.

2.3.2 Numerical Scheme for general particle model and MDPD

Verlet integration is a popular method to integrate Newton's equations of motion.

$$\mathbf{x}(t+\Delta t) = \mathbf{x}(t) + \mathbf{v}(t)\Delta t + \frac{1}{2}\mathbf{a}\Delta t^2 \tag{2.36}$$

$$\mathbf{v}(t+\Delta t) = \mathbf{v}(t) + \frac{\mathbf{a}(t) + \mathbf{a}(t+\Delta t)}{2}\Delta t \tag{2.37}$$

The above-mentioned equations can further be simplified to

$$\mathbf{v}_{n+\frac{1}{2}} = \mathbf{v}_i + a_n\frac{\Delta t}{2} \tag{2.38a}$$

$$\mathbf{x}_{n+1} = \mathbf{x}_n + \mathbf{v}_{n+\frac{1}{2}}\frac{\Delta t}{2} \tag{2.38b}$$

$$\mathbf{v}_{n+1} = \mathbf{v}_{n+\frac{1}{2}} + a_{n+1}\frac{\Delta t}{2} \tag{2.38c}$$

In the above equation set, the first and the third equation can be combined

$$\mathbf{v}_{n+1} = \mathbf{v}_n + a_n\frac{\Delta t}{2} + a_{n+1}\frac{\Delta t}{2} \tag{2.39}$$

Chapter 3

Analytical Numerical Boundary for Multi-body Dissipative Particle Dynamics

The complex fluids, soft matter, and their behavior involve a large range of length and time scales, and thus simulating them are challenging problems. Dissipative particle dynamics is a group of methods that were developed to address this issue in the mesoscale regime. Mesoscale belongs to the scale between the atomistic scale and continuum scales. If quantum effects are neglected then the atomistic scales are where Hamilton's equations are solved using Molecular Dynamics(MD). The MD simulations are very fast and can be finished in a fraction of second[30]. On the other hand, Navier-Stokes-Fourier hydrodynamics and other continuum theories assume fluid as a continuum. The continuum length scales range from microns to above. The mesoscale is roughly characterized by $10 - 10^4 nm$ and $1 - 10^6 ns$. Due to this scale, the MD simulations require a large number of atoms which is not feasible. Similarly, the continuum simulations start failing due to the small scale and fluid not behaving like a continuum.

Dissipative particle dynamics are a group of methods based on point particle assumption that tackles the complex fluid simulation at the mesoscale. The method became popular due to its simplicity and capability of capturing the phenomena of complex fluids at the mesoscale. The DPD and MDPD model is described in chapter2.

In this chapter, a mathematical framework has been developed to analytically model a complex wall with closed-form expressions in the context of multi-body dissipative particle dynamics (MDPD). There have been several attempts to analytically model the influence of solid walls and non-periodic boundary conditions in the conventional DPD approach[31][18]. However, there is a limited number of studies for these boundary conditions associated with MDPD that capture static and dynamic fluid-structure interactions through direct modeling of fluid-solid particle interactions. In section 3.2.2, the analytical wall approach is presented followed by validation case studies in section 3.2.3. Computational results for one complex geometry with validation of the Wenzel law are discussed in section 3.2.4. A brief derivation of the forces using the integral approach is given in section 3.4.

3.1 State of the Art in MDPD and Wall Boundary

Wetting is an important phenomenon in various applications such as surfactants, spray cooling, and nanolithography[32][33]. A proper description of wetting is essential to many important problems in the modeling of fluid behavior. Droplet contact angle on a surface is a quantitative metric of surface wettability. There are numerous experimental studies measuring liquid contact angles on surfaces of varying chemistry and geometrical complexity [34][35][36][37][38][39]. Robust computational approaches to deal with fluid-structure interaction for various droplet sizes and morphological characteristics of the solid surfaces are vital [40][41].

Molecular dynamics and Monte Carlo simulations have been used to study the behavior of droplets on solid surfaces [42][43][44][45]. While atomistic methods are very accurate when good interaction potentials are available, these methods are only feasible up to small length scales and time scales[45][46][47]. A course-grained method is required to reduce the computational cost at larger scales. There are many different coarse-grained methods that have been used to simulate mesoscale phenomena[48]. Dissipative particle dynamics (DPD) is a particle-based coarse-grained method proposed by Hoogerbrugge et al. [11]. In this approach, each DPD particle is a cluster of several molecules of a specific fluid/solid/vapor phase (analogous to course-grained modeling in molecular dynamics but with additional particle interactions beyond the conservative terms generally applied in MD) [25]. Later, DPD was modified to simulate the hydrodynamics behavior of fluid-solid interaction[16]. In all of the previous studies, dealing with boundary conditions has been a challenging issue [27]. In addition to the fundamental challenges of formulating the wall interactions, common solutions, such as incorporating forces of frozen wall particles, add a drastic computational cost to the modeling. Since the wall particles are frozen, an equivalent interaction formulation from the wall can be found without using the explicit wall particles and significant computational expense can then be saved. The goal of the present work is twofold: (a) to obtain a consistent set of fluid-solid interaction parameters for a wall consisting of explicit particles in MDPD, i.e., normalization, and (b) to develop a semi-analytical framework for MDPD to model the wall without the need for explicit wall particles.

In this work, a modified model for fluid-solid interaction is first discussed which can be used to model the three-phase contact line without local density contribution [49]. This work, for the first time, employs an analytical model (integral approach) for the solid wall boundary condition in MDPD that brings substantial gain in computational efficiency and thus expands the scope of its applicability to curved or complex walls. The model is first normalized to address the discrepancies in the wetting that exist in the present literature and is then validated through several benchmark studies such as a Wenzel model. Moreover, comparisons between both the fully numerical and the semi-analytical (integral force model) approaches are drawn. Time efficiency, accuracy, and limitations of the proposed model are thoroughly discussed.

In prior studies using MDPD, the repulsive component of conservative force (Eq. 2.9) was defined as a function of local density[15]. Local density is measured based on the number of particles, either of the same type (single-type or liquid) or multi-type particles (i.e., liquid and solid), within the cut-off radius R_d of a given particle. In single-type local density calculations, when the water particle approaches the wall, there is a gradual decrease in the local density for both fluid and wall particles due to the reduction in the number of same type particles within each particle's cut-off radius. In this study, multi-type particle-particle interactions have been used. Recently, the repulsive term in the conservative force has been redefined independent of local density, i.e., a new form of conservative force is used[49]:

$$\mathbf{F}_{ij}^{C} = A\,\omega_C(r_{\alpha j}, R_c)\ \mathbf{e}_{\alpha j} + B\,\omega_C(r_{\alpha j}, R_d)\ \mathbf{e}_{\alpha j} \tag{3.1}$$

Using the above expression simplifies the conservative force because there is no local density of the wall in the expression as in conventional MDPD models.

3.2 Results for Wall Boundary

3.2.1 Normalization

In previous studies [7][49][18], the results from the simulation of a droplet on a wall depend on the density of wall particles, which is unphysical when representing a solid wall. To have the results independent of wall density, a normalization of the local density is needed. In prior work, many choices for the number density of the wall have been tried resulting in inconsistent A_{sl}. In several studies, the wall density is taken to be equal to the fluid density [16] while in some other studies the wall number density is significantly higher than the fluid density [49].

ρ_s	A_{sl} range	Method	Local Density Appraoch
6.0	-	MDPD	Multi-type[25]
8.0	15-40	MDPD	Multi-type [50]
4.0-8.0	8-23	DPD	- [18]
6.0	8-22	MDPD	Single -type[31]
6.88	5-40	MDPD	Multi-type[51]
25	6-12	MDPD	Single -type[49]
5.12	25-50	MDPD	Multi-type [26]

Table 3.1: Wall density number ρ_s used in prior work and the range of corresponding A_{sl}.

In Table 3.1, wall density from some studies is shown. As can be seen from Table 3.1, because of different number densities and/or local density approaches, different A_{sl} values or ranges have been obtained in these studies. The large A_{sl} range is a result of variation in liquid and wall density at the interfaces. In other words, when the number density of the wall changes, the magnitude of the force on fluid-particle varies, and thus, the interaction between the wall and fluid-particle differs. Hence, there has not been any consistent scale for A_{sl} for different wettability of a surface. Figure 3.1 shows the effect of A_{sl} on the contact angle of a water droplet on a flat wall for different number densities of the solid wall particles. To calculate the contact angle, the boundary of the droplet is found by calculating the local density of the particles throughout the droplet. As can be seen, the change in the A_{sl} range due to the solid wall density number is clear and it is in agreement with prior work in Table 3.1. This can be explained from Eq. 2.7 where increasing solid wall particle density increases the repulsion force. Hence, a smaller attraction force (in absolute value) is needed to obtain the same contact angle. It is concluded that lower values of A_{sl} result in the same contact angle for a wall with a higher density.

In this study, the fluid particle number is assigned to be around 13,000 and kept within 1% of that number throughout the study. To obtain consistent results independent of the wall density, the solid wall particles' contribution in both local density and forces calculations are normalized. We, therefore, split the summation in the local density of the liquid particle into liquid neighbors' contribution (l) and solid neighbors' contribution (s) as shown in Eq. 3.2(also see Fig. 3.2)

$$\overline{\rho}_i = \frac{15}{2\pi R_d^3}\left[\sum_{j\neq i}^{l}\left(1-\frac{r_{ij}}{R_d}\right)^2 + \frac{\rho_l}{\rho_s}\sum_{j\neq i}^{s}\left(1-\frac{r_{ij}}{R_d}\right)^2\right] \tag{3.2}$$

Note that for simplicity the subscript "i" for the i^{th} particle is dropped in the rest of the work, e.g., ρ F expression due to the wall. In Eq. 3.2, the second term on the right-hand side (summation over solid neighbors) is scaled with the density ratio of liquid and solid. In other words, the contribution of the solid wall particles in either the local density or the force calculations will be normalized as given in Eq. 3.2 for any given value of ρ_s.

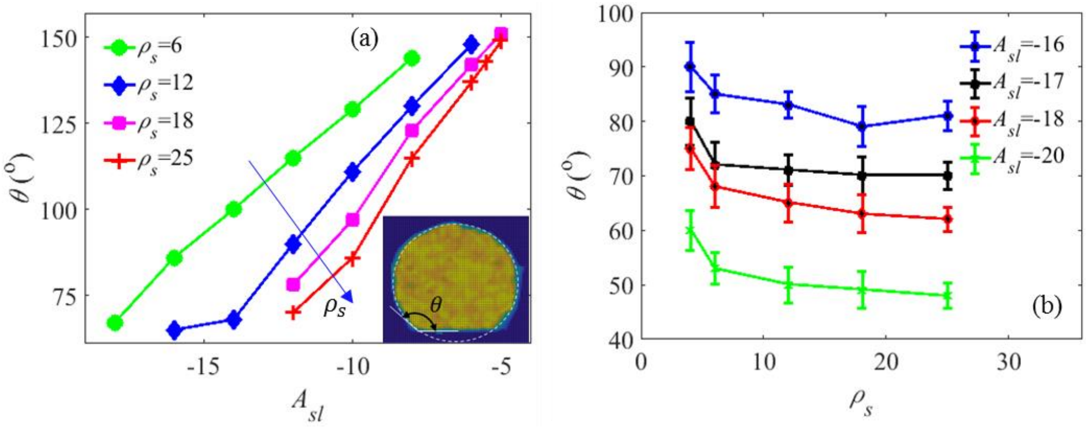


Figure 3.1: Effect of A_{sl} in the droplet contact angle on a flat surface with different solid wall density number ρ_s in (a). After normalization, the effect of solid wall number ρ_s on the droplet contact angle θ for different A_{sl} in (b). The inset figure shows the local density distribution contour used to predict the contact angle using the spherical cap approach in (a).

Figure 3.1 depicts the change in the contact angle of a water droplet of a flat wall after normalization. As expected, the contact angle for each A_{sl} number asymptotically approaches a constant value at a higher solid wall density number. Although a similar approach has been already proposed in DPD modeling [20][18], we apply it to MDPD simulations for the first time in the present investigation. Analytical expressions derived for solid wall interaction forces on adjoining fluid particles are discussed in the following sections.

3.2.2 Wall Boundary Approach

While the wall particles are frozen, these particles still take part in the calculation of the fluid particles as neighbors. More specifically, the fluid particles that are closer to the wall have wall particles within their cut-off radius, and there is a component of force due to these wall particles. This study focuses on imposing a force due to the wall via applying an analytical formulation, rather than considering each pairwise interaction with the frozen wall particles. A set of algebraic expressions (i.e., forces) for the boundary is used to represent the solid wall. This approach significantly reduces the computational cost of simulations, especially for a highly dense wall. The resultant force on an MDPD particle due to the wall is derived by integrating the force due to each individual wall particle. Note that we again split the summation terms into two sets for liquid and for solid neighbors, and an integral approach is then used to find a closed-form expression for the wall particle contribution (see for instance Eq. 3.2) [18]. Generally, these formulae or expressions are given by the following equations:

$$< Y >= \int_{z=h}^{R_d} \int_{r=0}^{\sqrt{R_d^2-z^2}} \rho_l g(r) Y \, r \, dr \, dz \tag{3.3}$$

where $g(r)$ is the radial distribution function of the DPD particle, and $< Y >$ is the integral value of Y within the cut-off radius R_d (or R_c) on the domain Ω_s [20][18] as shown in Fig. 3.2 (particle-particle interaction) and Fig. 3.2 (integral approach). Note that the factor 2π represents the axisymmetry of the domain in the 3D configuration. The value Y can be either a local density function or force component as will be discussed below. Also, because of the high wall density, the radial distribution function $g(r)$ is assumed to be 1.0 in this work [18].

For the local density due to the solid wall, the Y-term is replaced by Eq. 3.2, then the summation is replaced by the integration. The local density due to the wall at particle i, ρ_s), can be expressed as

$$\rho_{i,s}(\delta_d) = \rho_l \left[0.5 - \frac{5}{4}\delta_d + \frac{5}{2}\delta_d^3 - \frac{5}{2}\delta_d^4 + \frac{3}{4}\delta_d^5 \right] \tag{3.4}$$

where $\delta_d = h/R_d$(see Fig. 3.1). This term will be added to the local density for particle i which represents the wall effect in multi-type particle density calculations. In other words, the local density for particle i is now ρ equation where j is the list of the DPD liquid neighbors to the particle i within the cut-off radius R_d.

Similarly, conservative, dissipative and random forces reintegrated in the domain Ω_s. The derivation and final force expressions are given in section 3.4.

Final set of equations can be used to simplify the numerical modeling by removing the solid wall from the calculations. The equations are validated by comparison to the explicit particle approach and forces are compared in Fig 3.1. The model is derived for a flat wall; however, it can also be used for curved walls with a large radius of curvature as will be shown in section 3.2.4. If the radius of curvature of the wall R_{wall} is within a certain limit ($R_{wall} > 3R_c$), the model is valid and can be used with the error being less than 5%.

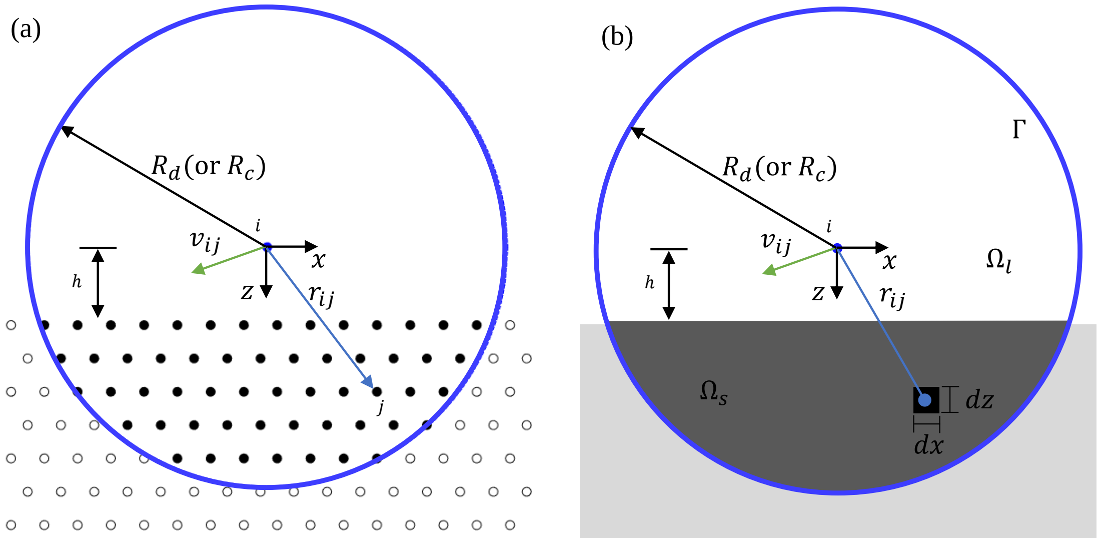


Figure 3.2: Schematic for the portion of wall particles within a cutoff radius R_d (or R_c) that contributes to the force on liquid particle i for particle-particle approach (black shaded particle) in (a) and integral approach (dark-gray area) in (b)

3.2.3 Benchmark Cases

To verify the mode, all the forces(conservative and dissipative) are validated with particle approach. In figure 3.3, one such verification is shown. As it is evident from the figure that the force using the wall boundary model is identical to the particle wall. To further validate our model, we simulate a droplet on a flat wall, and we generate a similar relationship between contact angle and A_{sl} (see Fig. 3.4). For this simulation, the total number of fluid particles is 13,062 in both integral and particle approaches, in particle approach total number of wall particles is $85,312$ where the wall density is chosen to be $\rho_s = 25.0$. We compared this relationship with available literature [49], and an excellent

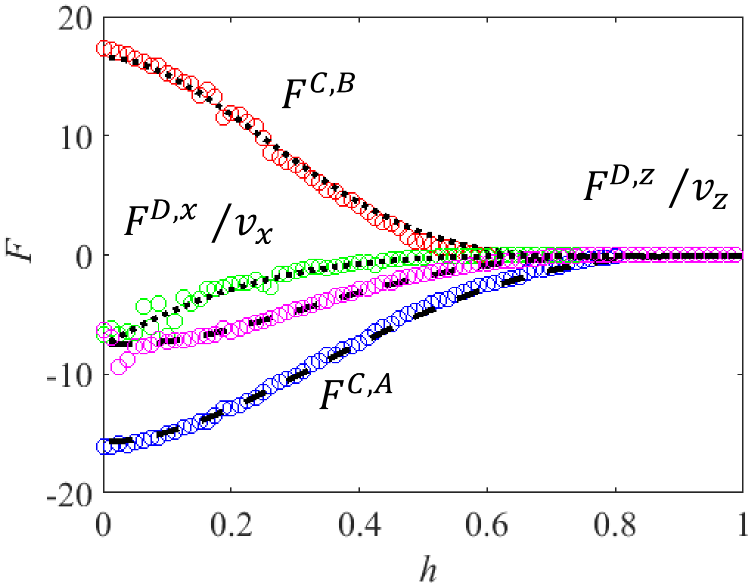


Figure 3.3: Comparison of the conservative, dissipative and random forces between particle-particle and integral approaches for different distance from the wall h.

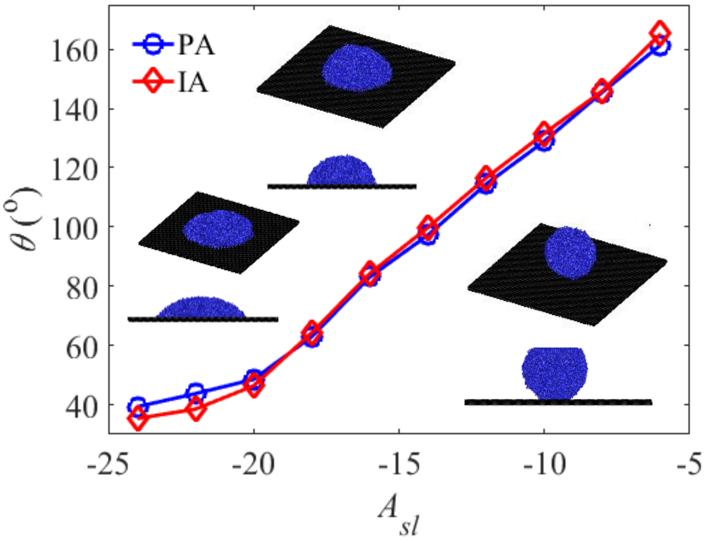


Figure 3.4: Comparison between the particle approach (PA) and integral approach (IA) for contact angle on a flat surface.

agreement is obtained, especially at small A_{sl}. In our analytical approach, the equation is integrated using the value of $\rho_l = 6.0$. With the analytical approach, we save more than 50% of computational cost for this case. More details of computational savings will be discussed in section 3.2.4.

To evaluate the capabilities and stability of the present model, the density fluctuation in the vicinity of the solid wall and thermal energy (k_B T) are calculated for a droplet on a solid flat wall. The same droplet configuration used in Fig. 3.5 is considered here with an A_{sl}=-16 or $\theta \sim 100^o$. For density fluctuation, rectangular blocks (bins) with dimensions of 10100.1 are considered in the core of the droplet as shown in the inset in Fig. 3.1. The average density number (temporal and spatial average) is calculated in these bins in the same fashion of molecular dynamic simulations [52][53][54][55]. As expected, increasing the wall density number damps the density perturbation near the solid wall as shown in Figure 3.5. Moreover, the results from the IA show the reduced fluctuations around the average density (about 6.1). k_BT can be calculated from the kinetic energy for the water droplet, i.e., $3/2k_BT = m/2N$ where N is the number of DPD particles in the water droplet. This specific energy, k_BT, is shown in Figure 3.5 for different wall density numbers for PA and IA. The results show an excellent agreement with the iso-thermal approximation of the MDPD modeling of this system.

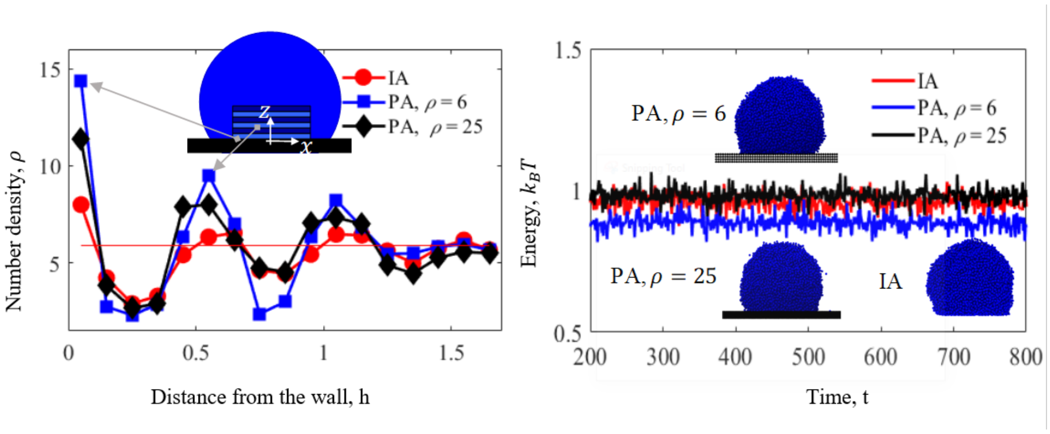


Figure 3.5: Effect of increasing solid wall density on the density fluctuation(left) The density fluctuations observed using the wall boundary model(right).

Further, we use the integral approach (IA) to simulate a droplet on a curved surface and compare it with the particle approach (PA). Contact angle on a droplet for a curved surface can no longer be defined using Young's equation [56], i.e., the contact angle, in this case, is not the same as the one on a flat plate. For

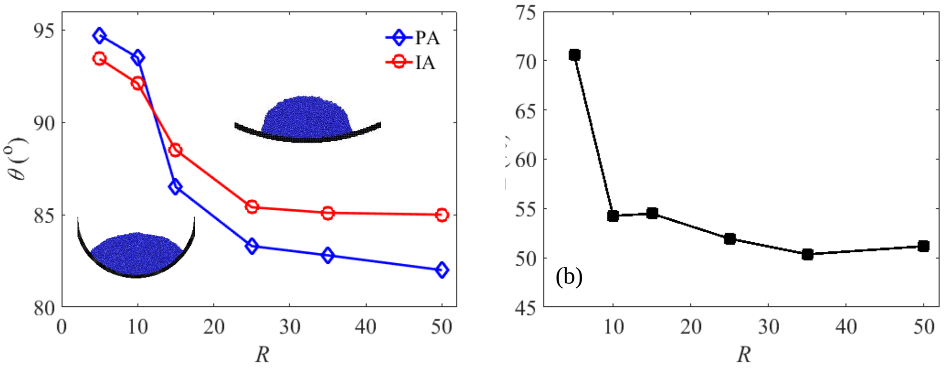


Figure 3.6: Effect of surface curvature on contact angle for the particle approach (PA), and integral approach (IA).

a hydrophilic droplet, the contact angle on a curved surface is higher than the contact angle for a droplet on a flat plate. By varying the radius of the curved surface to infinity, it ultimately corresponds to a flat plate and the contact angle of the droplet converges to its contact angle of the flat plate. In Fig. 3.6, the contact angle of a droplet for one hydrophilic case is shown for various radii of curvature and the contact angle is different for different radii. The contact angle is found to reach a constant value after reaching a certain radius of curvature asymptotically converging to its value on the flat surface.

3.2.4 Computational Efficiency

The analytical model was validated and compared with the explicit all particle simulations in section 3.2.3. For the comparison, a wall number density of 25 has been taken in particle simulations. In this section, the advantage and limitations of the integral approach are given.

The CPU time reduction is expressed using the percentage E(

$$E(\%) = 100\frac{t_{PA} - t_{IA}}{t_{PA}} \tag{3.5}$$

where t is the CPU time used in the simulation, and the subscripts “PA” and “IA” stand for the particle-particle and integral approaches, respectively.

The time consumed to obtain the results shown in Fig. 3.3 is used to calculate E with different solid wall density number as shown in Fig. 3.7. As the wall density number increases, the number of wall particles participating in the computational algorithm also increases. On the other hand, in the integral approach, the

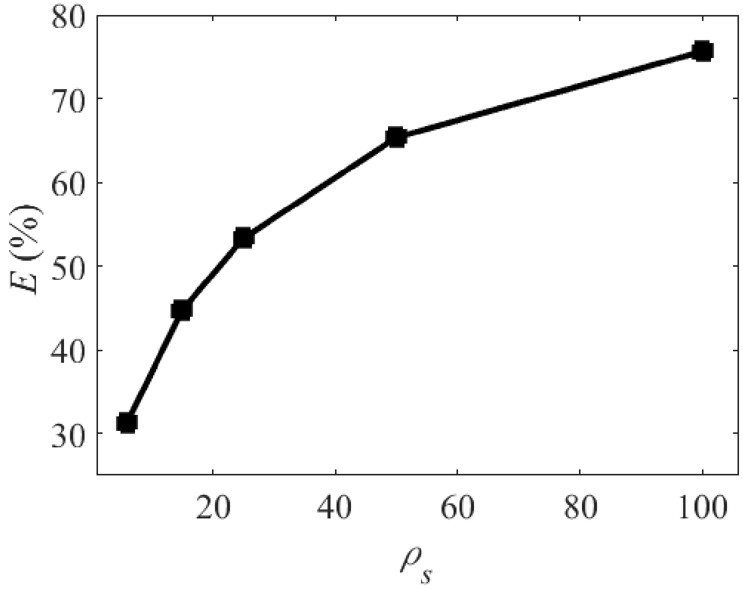


Figure 3.7: The percentage reduction in the CPU time E using the integral approach against its counterpart particle approach.

total number of particles remains the same (liquid particle), thus computational cost remains almost unchanged. Moreover, with increasing complexity in geometry, wall density, and domain size, the particle-particle approach tends to have more computational cost when compared to the integral model. In the integral approach, the computational cost primarily depends on liquid particle numbers, thus when the wall complexity is increased, the integral approach is the more efficient of the two approaches.

To describe the homogeneous wetting regime, the Wenzel model has been previously used [57]. According to the Wenzel model, the contact angle on a rough surface can be estimated by

$$\cos\theta_W = r\cos\theta \tag{3.6}$$

where θ_W is the Wenzel angle, r is the surface roughness ratio and θ is the contact angle on a flat surface. The roughness ratio is defined as

$$r = \frac{A_0}{A_{s0}} \tag{3.7}$$

where A is area of the surface, and A_0 is the projected area of the surface. A 2D geometry is taken into consideration to validate the analytical approach for a complex wall geometry (see Figure 3.8).

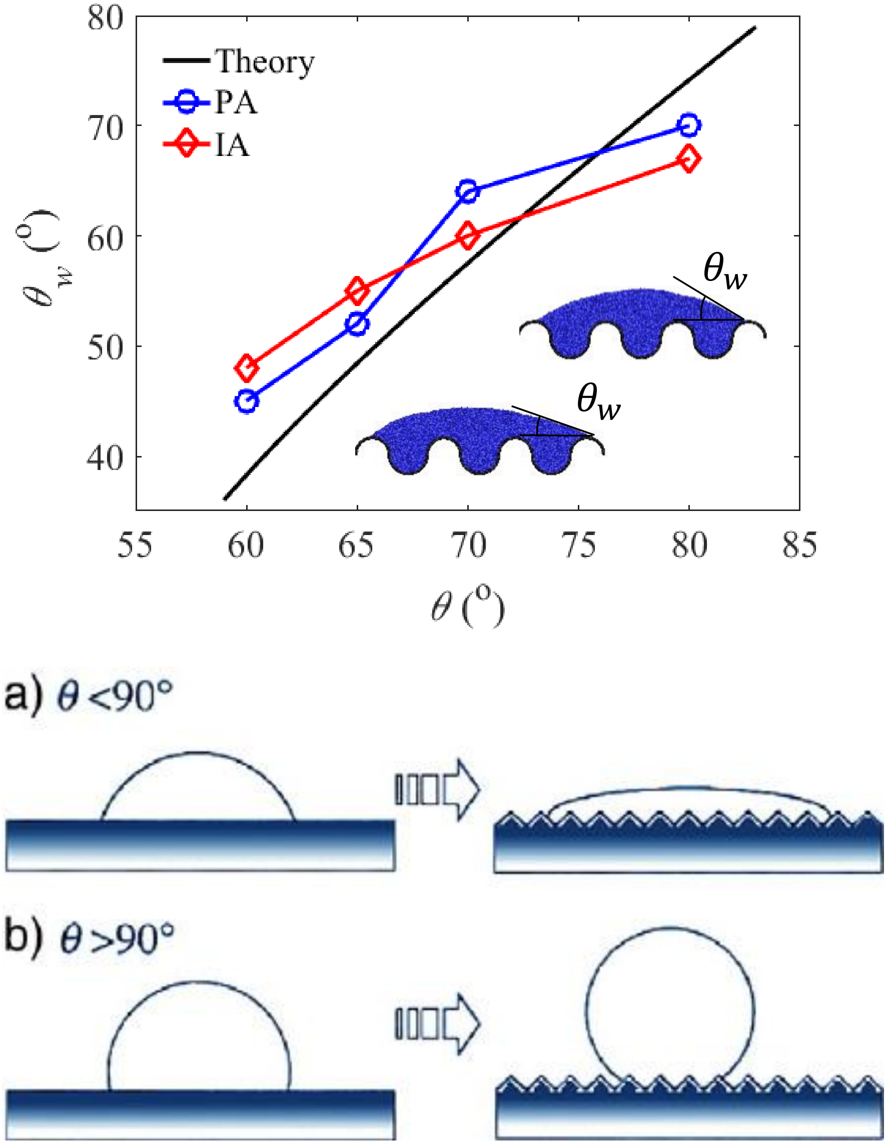


Figure 3.8: The comparison of analytical results with the theoretical equation for Wenzel law.

In Fig. 3.8, results from Wenzel equation validations are shown. The contact angle is depicted against Wenzel contact angle for theoretical, numerical, and semi-analytical counterparts. The analytical approach shows good agreement with both the particle approach and theoretical approaches. This is also an extra validation of the model used in the present work. Finally, a substantial amount of computational time has been saved in such geometries (about 45%) with $\rho_s = 25$. This is because of the highly curved wall, which adds a substantial number of extra particles compared to the flat wall.

3.3 Discussion

In previous MDPD works, different wall densities had been used which resulted in different ranges of A_{sl} values (solid-liquid interaction parameter) for wetting characterization. To solve this discrepancy in the literature, normalization is carried out in this work. A consistent A_{sl} range is obtained for different surface wettability. To do so, a high wall number density number (e.g. 25 or more) is required.

The particle approach used in previous works for the wall is computationally expensive for large domains, high wall density, and complex geometry. In this work, a wall model is developed using a combination of analytical and numerical integration, which substantially reduces the computational cost. As the wall complexity and domain size are increased, it is shown that this model has an advantage over traditional particle approaches. The conventional particle approach for the wall takes about $100 - 300\%$ more time when compared to the presented integral model for the same geometrical configuration. The results for the test cases are compared with those obtained from the particle approach and they match well. Thus, the analytical wall model provided in this investigation is an accurate and simple tool to simulate fluid-structure interaction in MDPD while saving computational time. The presented integral (derived for a flat wall) also gives accurate results for a wall with sufficiently large curvature. The minimum radius of curvature to which it can be applied accurately is $3R_c$.

3.4 Derivation of the Wall Boundary Method

In this work, the forces due to the solid wall are derived using a combination of numerical and analytical expressions. These equivalent forces have been used for all the simulation procedures. In the particle-particle approach, the wall is made of particles (see Figure 3.9) which means for fluid particle i, individual pair-wise forces resulting from each wall particle are calculated. The resultant force on the fluid particle due to the wall is the sum of the forces by each wall particle within the cut-off radius. As the number density increases, the number of particles in the zone within the cut-off radius for particle i increases. Force on the fluid particle due to one wall particle is governed by[49]

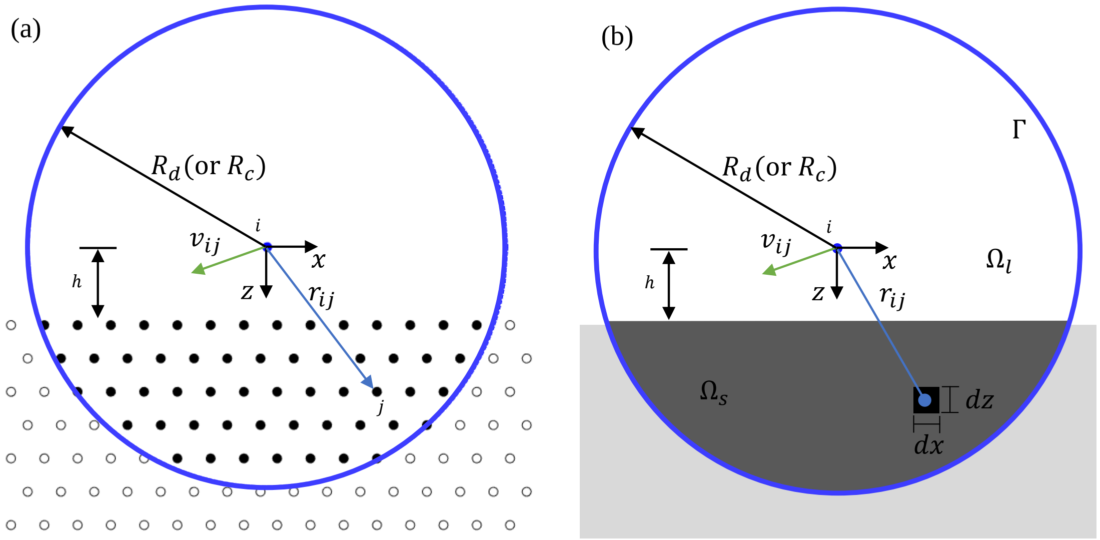


Figure 3.9: Schematic for the portion of wall particles within a cutoff radius R_d (or R_c) that contributes to the force on liquid particle i for particle-particle approach (black shaded particle) in (a) and integral approach (dark-gray area) in (b)

$$\mathbf{F}_{ij}^{C} = A\,\omega_C(r_{ij}, R_c)\ \mathbf{e}_{ij} + B\,\omega_C(r_{ij}, R_d)\ \mathbf{e}_{ij} \tag{3.8}$$

Total conservative force on particle i due to the full wall is governed by

$$\mathbf{F}^{C,A} = \sum_{j\neq i} \mathbf{F}_{ij}^{C} = \sum_{j\neq i} A\,\omega_C(r_{ij}, R_c)\ \mathbf{e}_{ij} \tag{3.9}$$

Ideally, the actual number density of 25.0 or more is enough to make the wall essentially continuous. This wall density is normalized for the fluid number density (i.e., $\rho_l = 6.0$) as explained in chapter 2. For a continuous wall, the summation changes to integration. For instance, if we express any term due to wall in the conservative attractive force term in the integration form then it will become

$$\mathbf{F}^{C,A} = \int\limits_{z=h}^{R_c} \int\limits_{x=0}^{\sqrt{(R_c^2-z^2)}} \int\limits_{\theta=0}^{2\pi} \rho_l A\, \omega_C(r_{ij}, R_c)\ \mathbf{e}_{ij} x dx d\theta dz \tag{3.10}$$

Where $\mathbf{F}^{C,A}$ is attractive component of the conservative force. It can further be simplified and expressed as

$$\mathbf{F}^{C,A} = 2\pi\rho_l A_{ij} \int\limits_{z=h}^{R_c} \int\limits_{x=0}^{\sqrt{(R_c^2-z^2)}} \left(1 - \frac{r}{R_c}\right)\ x dx dz \tag{3.11}$$

where $r^2 = x^2 + z^2$. Upon integration, it is simplified as:

$$\mathbf{F}^{C,A} = 2\pi R_c^3 A_{sl}\rho_l \left[\frac{1}{24} - \frac{1}{4}\delta^2 + \frac{1}{3}\delta^3 - \frac{1}{8}\delta^4\right] \tag{3.12}$$

where $\delta = h/R_c$. As discussed before, the conservative attraction force has identical form but with cut-off radius R_d and amplitude B. Similarly, integration are performed for the dissipative and random components of force, and the following expressions are derived

$$\mathbf{F}^{D,x} = -2\pi\gamma v_x \rho_l \left[\frac{1}{45} - \frac{1}{2}\delta + \delta^3\left(\frac{2}{9} + \frac{1}{3}\log(\delta)\right) + \frac{1}{3}\delta^4 - \frac{1}{20}\delta^5\right] \tag{3.13}$$

$$\mathbf{F}^{D,z} = -\pi\gamma v_z \rho_l \left[\frac{1}{90} + \delta^3\left(\frac{7}{18} + \frac{1}{4}\log(\delta)\right) - \frac{1}{2}\delta^4 + \frac{1}{10}\delta^5\right] \tag{3.14}$$

$$\mathbf{F}^{R} = \zeta\left[\sqrt{2k_BT\mathbf{F}^{D,x}\mathbf{e}_x} + \sqrt{2k_BT\mathbf{F}^{D,z}\mathbf{e}_z}\right] \tag{3.15}$$

where v_x and v_z are the particle's velocity in the x and z directions.

3.5 Maple Code for the Wall Boundary(MDPD)

```
    restart:
int(int((1-sqrt(x^2+z^2)/1)*z/(x^2+z^2)*x,
x =0..sqrt(1^2-z^2)), z = 0..h, AllSolutions);
int(int((1-sqrt(x^2+z^2)/1)*z/(x^2+z^2)*x,
x = 0..sqrt(1^2-z^2)), z = 0..h) assuming h>0;

int(int((1-sqrt(x^2+z^2)/1)*z/(x^2+z^2)*x,
x = 0..sqrt(1^2-z^2)), z = 0..h) assuming h>0,
(-z^2)^(1/2)> 0, (-z^2)^(1/2)<(r_c^2-z^2)^(1/2) ;

int(int((1-sqrt(x^2+z^2)/1)*x^3/(x^2+z^2)*x,
x = 0..sqrt(1^2-z^2)), z = 0..h) assuming h>0,
(-z^2)^(1/2)> 0, (-z^2)^(1/2)<(r_c^2-z^2)^(1/2);

int(int((1-sqrt(x^2+z^2)/1)*x^3/(x^2+z^2)*x,
x = 0..sqrt(1^2-z^2)), z = 0..h, AllSolutions);


int(int((1-sqrt(x^2+z^2)/1)*x*z^2/(x^2+z^2)*x,
x = 0..sqrt(1^2-z^2)), z = 0..h) assuming h>0,
(-z^2)^(1/2)> 0, (-z^2)^(1/2)<(r_c^2-z^2)^(1/2);

int(int((1-sqrt(x^2+z^2)/1)*x*z^2/(x^2+z^2)*x,
x = 0..sqrt(1^2-z^2)), z = 0..h,AllSolutions);
```

Chapter 4

Application of Wall Boundary Method

4.1 Transport of Solid Sphere

In this chapter, the wall boundary method derived in the previous chapter has been used and an application is demonstrated. The specific application chosen is flow measurement. Recently, flow measurement using multi-body dissipative particle dynamics(MDPD) has been used to overcome the challenges in micro and nanoscale. MDPD overcomes both the computational cost and limited computational domain. In this work, MDPD has been used to simulate the dynamics of a solid sphere in mesoscale fluidic systems. A droplet is kept on a surface with a wettability gradient. Due to the wettability gradient, the droplet feels a force field that results in the droplet moving from lower to higher wettability end. A solid sphere is put on the droplet's trajectory and the droplet's motion causes the solid particle to move along with droplet. The interaction and dynamics of the droplet with surface and solid particle have been simulated in this work. For the first time, a velocity field inside a moving droplet around the solid particle is also investigated. The numerical results greatly extend the understanding of the flow field in the micro and nanoscale and can be used for various micro-fluid applications. Although the solid sphere and the wall are modeled using the wall boundary method, the wall and the solid particle are still shown in all the figures to make it simpler to visualize the system.

4.1.1 State of the Art in Particle Transport in Fluid

The physics of fluid flow has been drawing the interests of scientists from old times[58][59]. Fluid dynamics is a fundamental branch of science that deals with fluid in motion and their interaction with the surrounding environment. Fluid-structure interaction has various applications from macro scale such as flow past airfoil and airplane wings[60] to smaller scale such as interaction of the tethered DNA with surrounding fluid[61]. The flow physics of the droplet in microfluidics has been studied using both experiments and numerical simulations[62][63][64]. While liquid droplet dynamics have been extensively studied, the interaction of liquid droplet with solid particle has only limited work in literature. Moumen et al.[62] performed experiments on the droplet's motion on a horizontal surface with a wettability gradient. In the work done by Moumen et al, it was found that the velocity of a droplet was a strong function of position along the wettability gradient. Kinoshita et al.[63] and Lu et al. [64] used microparticles image velocimetry(µPIV) for the internal flow of a droplet and 3D velocity field was reconstructed from the 2D µPIV experimental data. Ma et al.[65] reported an experimental investigation of a flow field inside droplets, and showed that capillary number and droplet geometry had almost no effect on the flow dynamics inside the droplet. In micro and nanoscale, digital microfluidics is a relatively new technology that enables precise control of small liquid droplets[9]. Lan et al.[17] studied the digital microfluidic manipulation of solid microparticles where they designed an optimization tool for the manipulation of the microparticle in microfluidic systems. Experimental methods such as PIV and particle tracking velocimetry(PTV) have been a great tool to study fluid flow[66][67]. There have been several experimental studies with particles in a fluidic environment where the velocity field inside the droplet is observed using PIV and PTV [68][69]. However, there is no computational work where the velocity field inside a moving droplet around a solid particle has been investigated. At bigger scales, continuum methods such as the finite volume method (FVM) and finite elements have previously been used to solve such problems [70][71]. These methods are based on solving Navier stokes equations numerically[72]. On a smaller scale, however, the continuum methods such as FVM are not valid anymore. This happens because there are thermal fluctuations present in the system at this scale and fluid can no longer be modeled as a continuum[73]. Particle methods such as smoothed particle hydrodynamics[74][75] do not consider thermal fluctuations in the system, so there is a need for a computational model which considers random motions of the particle at nanoscales. Atomistic methods such as Molecular dynamics and Monte Carlo simulations have previously been used to study the interaction between fluid and solid surfaces [42][43][44][76][77]. Atomistic methods are very accurate and precise because of their good empirical potential correlation. Even with modern computational capabilities, often these methods are limited to a small length and time scale. [46] MD simulations usually are in the length scale of 10^{-10} meters and time scale of 10^{-12} seconds[47] To solve the issues with limited system size and time limit, MDPD method is being used in this case. The contact angle of a sessile droplet has been a problem for a long time and MDPD had been used to simulate such systems [49][78][79]. Like DPD, the length and time scale of MDPD simulations are larger than atomistic simulation methods. MDPD has been used to examine several phenomena at the mesoscale such as water oil displacement in capillaries[14] coalescence of sessile microdroplet[80] and reproducing interfacial properties of the water-wall systems [81]. However, MDPD has not been used till now to examine the velocity field inside a droplet. In this manuscript, the flow physics of a solid particle inside a moving droplet has been examined, and the effect on its dynamics due to various parameters such as wettability gradient, droplet radius has been studied.

The droplet rests on the wall initially. The wall is assumed to be frozen and the wall boundary model is used to model the wall. The droplet particles, howeve,r experience force from the wall and other droplet particles. These forces are conservative, dissipative, and random. The conservative force accounts for the chemical property of the particles and is different for the wall-droplet and droplet-droplet interaction. The dissipative and random forces are identical for both pairs. A detailed description of these forces are given in chapter 2. The parameter values are given in Table 4.1

The solid sphere is made of solid MDPD particles which are arranged in a high-density manner. The density of the sphere is higher on the outer surface, so it has a shell-like structure. Solid particle's interaction with solid wall has been modelled based on the solid-solid contact model reported by[82]. It is assumed that the substrate material is an elastic material and the solid particle is rigid. The depth of indentation d_m is given by the following equation.

$$d_{in} = \left(\frac{9F_{normal}^2}{16R_sE^2} \right)^3 \tag{4.1}$$

where F_{normal} is normal force, E is the elastic modulus of the substrate and R_s is the radius of the solid sphere.

4.1.2 Configuration and system

The initial setup is shown in the figure. The solid particle is kept on the wall at $x = -10$, and due to the adhesion force, it is attached to the wall. The normal

Parameters	Symbol	MDPD value
Particle mass	m	1.0
System energy	k_BT	1.0
Cut-off radius of attractive force	R_c	1.0
Cut-off radius of repulsive force	R_d	0.75
Attraction parameter (liquid-liquid)	A_{ll}	-40
Repulsion parameter	B_{ll}	25
Amplitude of random force	ϕ	6.0
Time step	ΔT	0.01
Liquid particle density	ρ	6.1
Liquid-Vapor surface tension	σ	7.30
Liquid dynamic viscosity	ν	7.45

Table 4.1: Parametric values for MDPD simulations.

force from the wall balances adhesion force and the net force on the solid particle is zero. The initial position of the liquid droplet is $x = x_i$, the radius of the liquid droplet is R_l. The wettability gradient is applied on the wall (explained in the next section) from x_i to x_f. The contact angle and x_i is 146^o degrees and at x_f, it is 36^o. The simulation is performed in the computational domain of size 120X80×80. Other parameters are tabulated in Table 4.2. The wettability gradient has been modeled by the varying contact angle, which varies according to the following equation.

$$\theta = 146 - \frac{110}{(x_f - x_i)}(x - x_i) \tag{4.2}$$

4.1.3 Dimensional Analysis

In the MDPD, all parameters are used in reduced units and they are given in Table 1. To convert a parameter from MDPD units to physical units, many different approaches have been described in previous works[83][84] . If density, viscosity, and surface tension of fluid in SI units are assumed to be d^*, ν^* and σ^* respectively. The following formulations are used to find the reference length L_{DPD} mass M_{DPD} and time T_{DPD}[26].

$$L_{DPD} = \frac{d^*}{d}(\frac{\nu^*}{nu})^2\frac{\sigma}{\sigma^*} \tag{4.3}$$

$$T_{DPD} = \frac{\nu}{nu^*}L_{DPD}^2 \tag{4.4}$$

Parameters	Symbol	MDPD value
Young's modulus of the wall	E	$1e+6$
Initial position of the liquid droplet	x_0	-40
Attraction parameter with the substrate wall	A_{ss}	-1.0
Repulsion parameter with the liquid	B_{ls}	3
Radius of the solid sphere	R_s	1
Contact angle at the hydrophobic end	θ_0	36
Contact angle at thehydrophilic end	θ_f	146
Initial distance between the droplet and the solid sphere	d	10

Table 4.2: Parametric values used for modeling the solid particle and wettability gradient

$$M_{DPD} = \frac{d^*}{d} L_{DPD}^3 \tag{4.5}$$

If the working fluid is water, then DPD unit length is calculated to be equal to about $10nm$. Using eqs 4.4 and 4.5, the unit time and mass is calculated and found to be $1.3e-10$ s and $1.63e-22$ kg.

4.1.4 Defining the wettability gradient

In this work, the wettability of the surface is an important factor, the droplet is driven by the capillary force which is generated from the wettability gradient. The wettability of the surface can be changed by modifying the conservative force. The repulsion coefficient is kept constant as described in Table 4.1 and the attraction coefficient is changed to modify the wettability of the surface. To formulate the wettability (and static contact angle) as a function of A_{sl}. Figure 4.1 shows the effect of A_{sl} on static contact angle. Water on hydrophilic and hydrophobic is shown in the inset. The data from the current work is compared with the reported data by Chang et al [39]. The contact angle of the droplet matches very well with reported data in the literature [39]. The data from $A_{sl} - \theta$ curve is used to fit a third-order polynomial and A_{sl} as a function of θ is obtained.

$$A_{sl} = 8.825e - 6\theta^3 - 1.888e - 3\theta^2 + 0.3139\theta - 48 \tag{4.6}$$

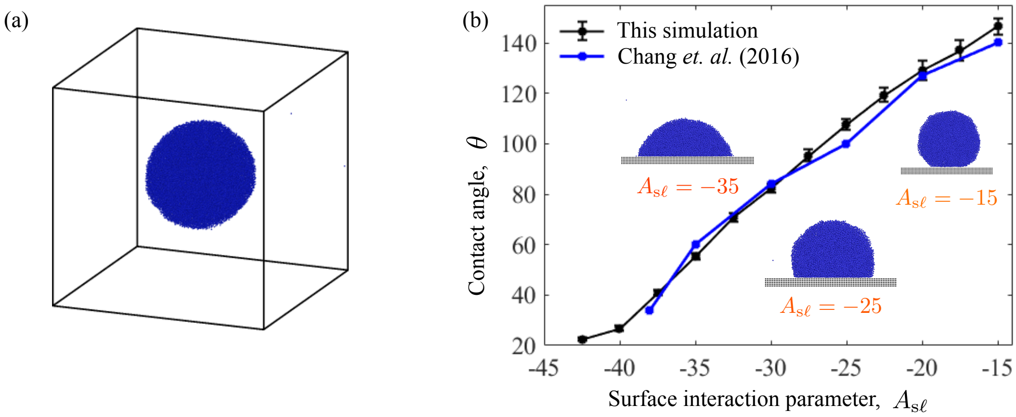


Figure 4.1: The surface tension of the liquid droplet is validated by simulating a droplet(left), the $A_{sl} - \theta$ relationship is validated with literature

4.1.5 Results

Model Validation

In simulations, we calculate the interfacial tension by using Irwing-Kirkwood equation[85]

$$\Gamma_{12} = \int [p_{zz} - \frac{1}{2}(p_{xx} + p_{yy})]dx \tag{4.7}$$

where Γ_{12} is the interfacial tension between fluid 1 and fluid 2. where p_{xx}, p_{yy} and p_{zz} are the three diagonal components of the pressure tensor. The interface in this calculation is $x - y$ plane. In terms of the interparticle forces, this can be expressed as

$$\Gamma_{12} = \frac{1}{A_i}(\sum_{i,j} F^z_{\alpha\alpha'} r_{ijz} - \frac{1}{2}(F^x_{\alpha\alpha'} r^x_{\alpha\alpha'} + F^y_{\alpha\alpha'} r^y_{\alpha\alpha'}); i > j \tag{4.8}$$

where $F^x_{\alpha\alpha'}$, $F^z_{\alpha\alpha'}$, $F^z_{\alpha\alpha'}$ are x,y, and z components of the force between particle i j. A_i is interface area. Similar formula can be applied for solid-liquid interface[86]. The setup for calculating surface tension is shown in Figure 4.1.

In MDPD simulations, viscosity is usually obtained by simulating periodic Poiseuille flow[87][88].We divide our simulation cell into two halves and apply a specific body force in those half in opposite direction. After the steady state is reached, we calculated the velocity profile and compared it with the theoretical results(Figure 4.2).

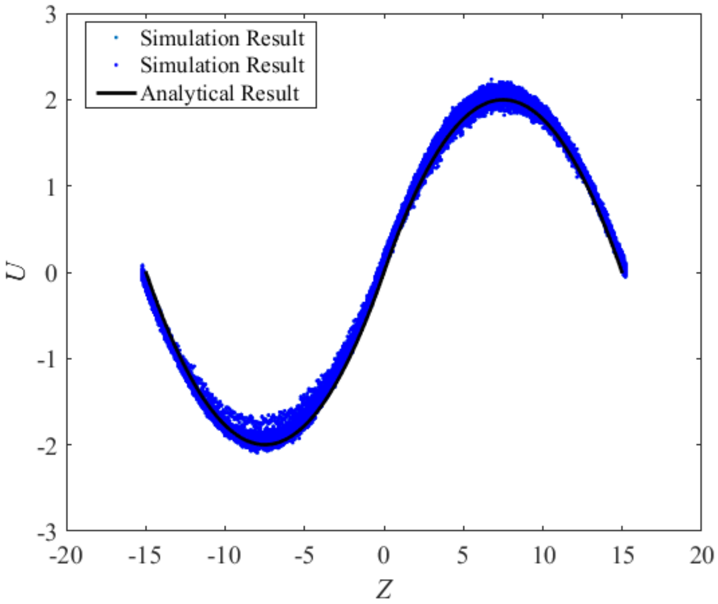


Figure 4.2: The validation of the numerical model and the viscosity using Poiseuille Flow

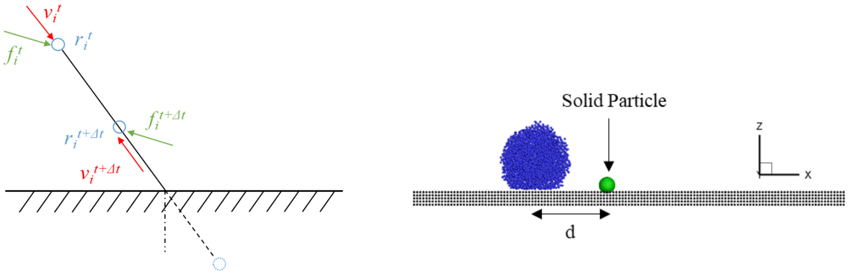


Figure 4.3: Schematic of the bounce-back scheme(left), The initial condition of the system. The surface has a lower contact angle towards the right thus moving the droplet and transporting the solid sphere with it

Transport of Solid Sphere

Initially, the physics flow of liquid droplet without any solid particle on its path is simulated. The wettability gradient on the wall causes the droplet to move from x_i towards x_f. Further, a solid particle is put in the trajectory of the droplet as shown in Figure 4.3. In this case, the droplet still moves because of the wettability gradient and it has similar dynamics before it reaches the solid particle. The travel history of the particle is shown in Figure 4.4. The figure shows the z position of the particle with time. Initially, the z position is zero when the particle is on the wall and the adhesion force is balanced with the normal force. When the solid particle is picked up the z position is higher and then the solid particle comes to an equilibrium height. Various stages and the velocity field of

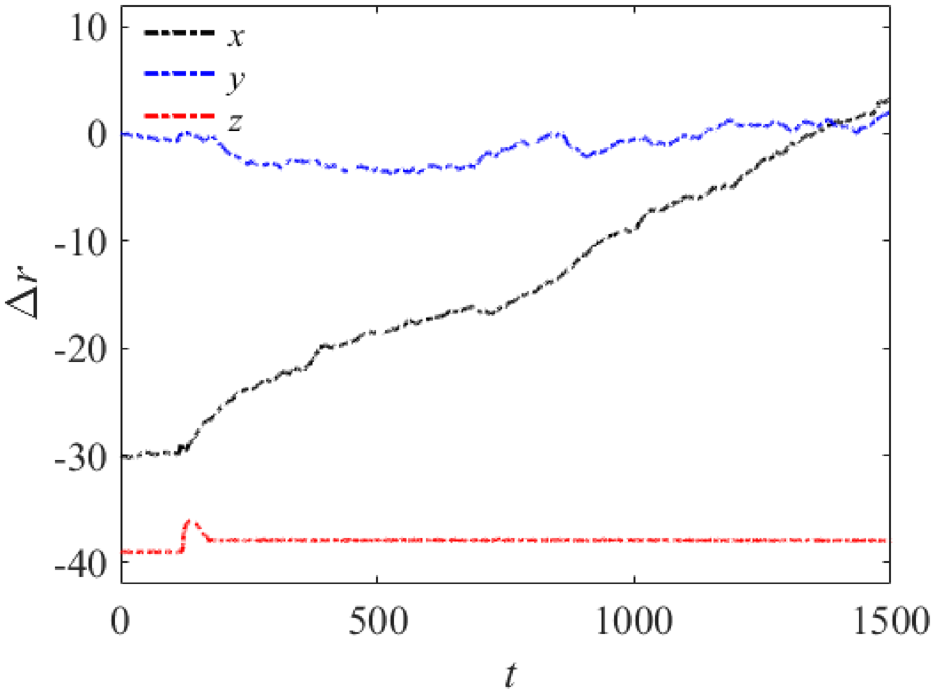


Figure 4.4: The X, Y, and Z coordinate of Center of Mass of the solid sphere. One thing to highlight is that the solid sphere gets picked up by the droplet as reflected in Z-coordinate of the sphere

the droplet and solid particle's dynamics are shown in Figure 4.5. As the droplet reaches the solid particle, initially, the liquid particles surround the solid. In the next stage, the solid is picked up and is lifted high, and then lowers down to a certain height. Further, the droplet keeps moving and the solid particle moves inside the droplet. The solid particle's velocity is less than the velocity of the liquid droplet. Thus, there is a relative velocity between liquid and solid particle and it moves from right to left of liquid. In the final stage, the solid particle stays at the tail end of the droplet. The velocity field in all stages is shown in Figure 4.5. The total time to travel the same distance with and without the particle on the droplet's trajectory is calculated. The droplet takes more time when it carries the particle.

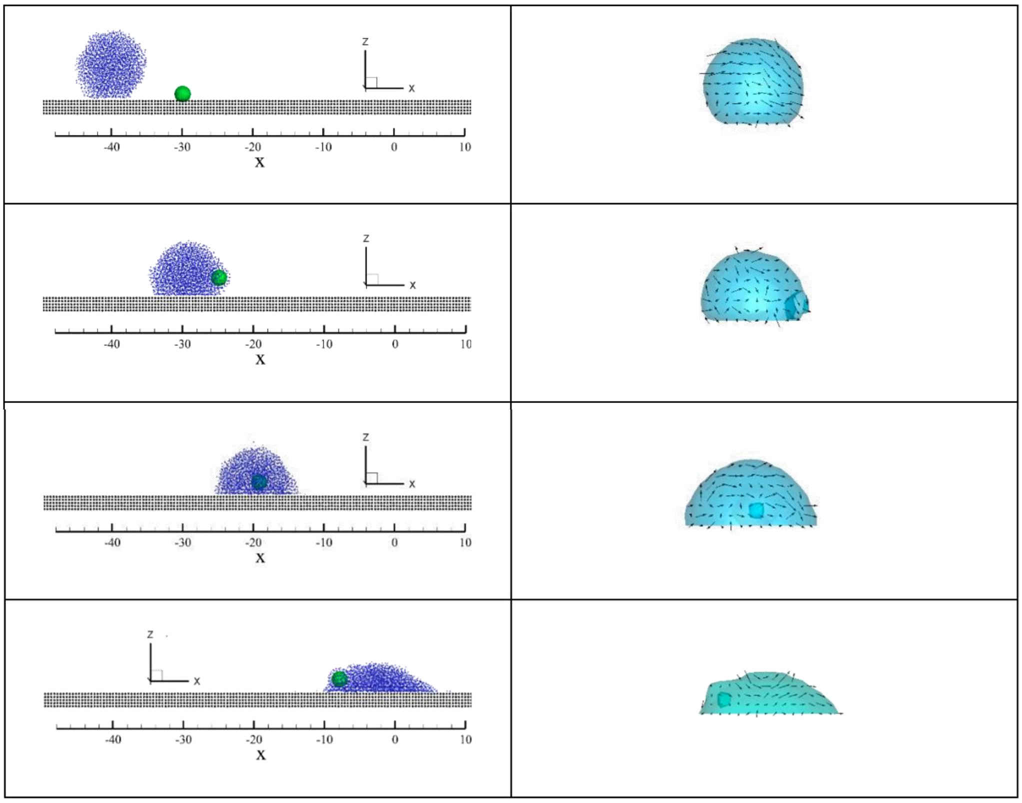


Figure 4.5: Various stages of dynamics of the droplet and solid particle due to wettability gradient

Effect of Droplet Size

The size of the droplet has a great impact on the transport process of the solid particle. For $x_f = 40$, the radius of the droplet is varied, and the impact on the dynamics of the droplet and solid particle is studied. It is observed that as the droplet size is increased, the droplet moves faster. In Figure 4.6, solid particle's position at different times is displayed. When the droplet radius is increased from 5 to 6, the particle velocity is slower but as it is increased further, it becomes higher and higher.

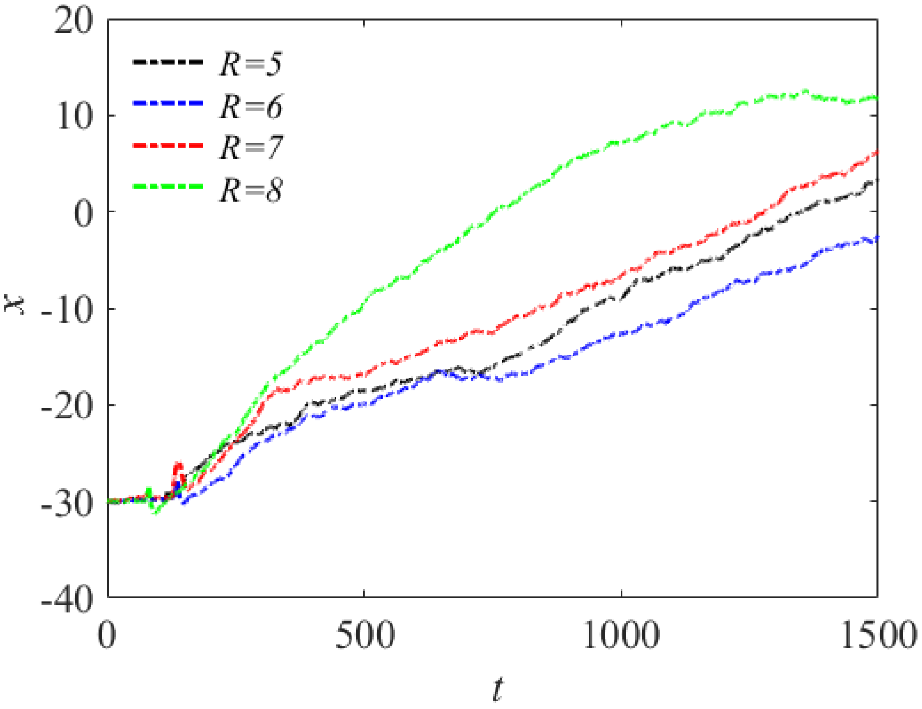


Figure 4.6: The effect of different solid sphere radius, increase in radius causes the faster transport

Effect of Wettability

The solid wall's wettability gradient is one of the factors which is crucial factor for flow physics and the transport of solid particle. The parametric study with different values of x_f has been performed. The values of x_i are kept constant throughout at -40, the different values of x_f chosen are $0, 20$, and 40. Static contact angle at x_f is kept the same at 36^o and the wettability gradient is kept in such a way that the contact varies linearly in the wall. Thus, the wettability gradient is higher when $x_f = 0$ and lower when $x_f = 40$. From Figure 4.7, the droplet and the solid particle move faster when the wettability gradient is higher, and it moves slower for a lower wettability gradient. The droplet reaches the solid particle sooner when the wettability gradient is higher and to travel the same distance, it takes a lower time for a low wettability gradient. In the figure, for the particle to reach from $x = -30$ to $x = -10$, the droplet with $x_f = 0, 10, 20, 40$ take $700, 750, 800$ and 900 (in MDPD unit) respectively. Here it is noticeable that the velocity of the droplet significantly reduces as the static contact angle reduces. In other words, the droplet for $x_f = 0$ travels faster than any other cases

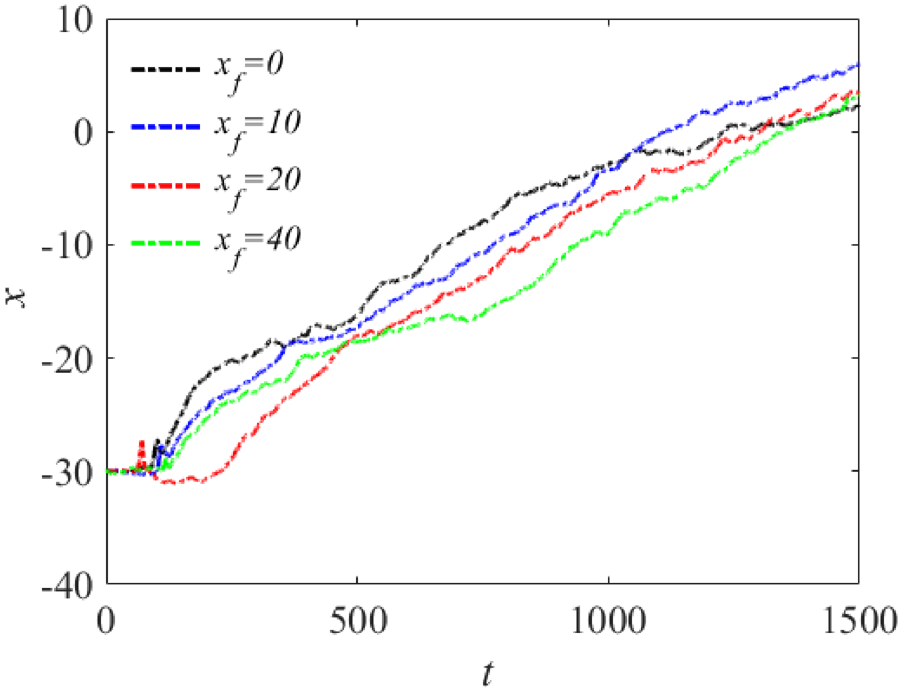


Figure 4.7: The effect of final position x_f, there is no significant difference among them

for initially but as the static contact angle reduces, the velocity slows down, and eventually it reaches to $x = 0$ in the end. For other x_f values, the trend still follows the wettability gradient order. Near $x = 0$, it is evident from the figure that the droplet with $x_f = 20$ and $x_f = 40$ traveled almost the same distance.

4.1.6 Discussion

In the current work, the physics of solid particle within a liquid droplet under a wettability gradient in a mesoscopic fluidic system has been modeled. MDPD has been used to develop the numerical model. Wettability gradient results in different static contact angles for the droplet at its end, which causes the droplet to move. The droplet moves and when it reaches the solid particle, the liquid particle surrounds the solid and lifts it. The solid particle moves inside the droplet from one end to another and then remains in the tail end throughout the process. Various parametric studies have been done in the work, such as the effect of droplet radius. The droplet radius has a mixed effect on the solid particle's velocity. It first decreases but as the radius keeps increasing it increases. Higher wettability gradient results in a higher velocity of the solid particle thus faster transportation. The velocity field inside the droplet is investigated for the first time and the obtained results help understand the fundamentals behind the flow field inside the droplet around a solid in the mesoscale regime.

Chapter 5

Mesh Generation using General Particle Model

In this chapter, the general particle model derived in chapter 2 has been used to generate an isotropic unstructured mesh. Mesh generation is an important stage in computational fluid dynamics (CFD) and structural mechanics. Generating mesh is computationally expensive, and becomes more challenging for a domain with complex geometry and in regions with high field variable gradients. The proposed method is based on the particles in the system interacting with each other. The equilibrium length and mass of the particles depend on the desired refinement in that region. The force potential acts as a driving potential for the particles and cause the particles to move, and rearrange to the final desired locations. The particles achieve a final convergent position that has a uniform density throughout the domain and no resultant force on them. A Voronoi Tessellation-Delaunay Triangulation(VT-DT) method is used to generate mesh from final particle position.

5.1 State of the Art in Mesh Generation

Mesh-based methods such as FEM and FVM are used to solve or simulate a wide range of physical phenomena such as fluid flow, heat transfer, mechanical deformation, and chemical reactor flow [89][90][91]. The principle behind these methods is to divide the physical system into small meshes and convert the partial differential equations (PDEs) into equations with finite form, i.e., algebraic equations. The mesh generation is an important step and often a difficult task for complex domains [92], and the need for a good optimized and efficient mesh generation method becomes important [93][94].

The mesh generation has been studied and researched using many techniques over the last few decades [95][96][97][98][99][100]. The mesh generation methods can generally be divided into four categories(see Figure 5.1): the advancing front methods, Delaunay methods, Octree methods, and particle based methods. In the advancing front methods, mesh generation starts from the boundary elements, and then inside elements are constructed in layers [101][102]. The main challenge in this method is often to decide which face to advance and whether to choose a new or a preexisting vertex [103]. The answer to these two questions critically determines the quality of the resulted mesh [104][93]. Delaunay method [105] uses the triangulation of nodes in the domain. Triangulation is formed and then refined further to add, modify or delete nodes. In the entire process, the

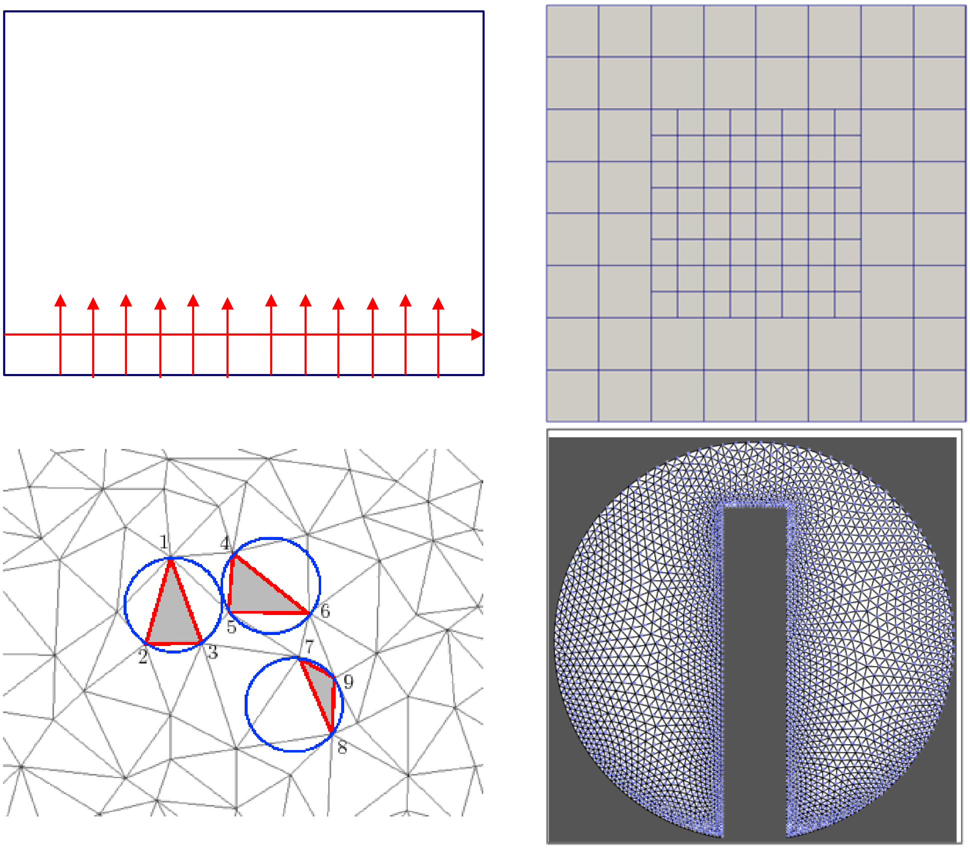


Figure 5.1: Mesh generation methods Advancing front method(top-left), Octree method[1] (top-right), Delaunay Triangulation[2] (bottom-left) and Particle Method[3](bottom-right)

empty circumcircle criterion of the mesh is kept preserved [106][107]. The empty circumcircle criteria dictate that for each of these triangles, the circumcircle of the triangle (from the nodes) contains no internal nodes. The Delaunay method is efficient to produce a good quality mesh in two dimensions but is not guaranteed to have a good quality mesh near the boundary and in three dimensions. There have been modifications to Delaunay methods to counter its drawbacks but they may suffer from other problems such as speed and accuracy of generating new mesh [108][109]. Octree methods [110] use mesh refinement by adding children nodes recursively [111].

In addition to these methods for mesh generation, particle-based methods have also been getting popular recently [112][3][113][114]. One of the first particle-based approaches was proposed by Witkin and Heckbert [115] where the particles are constrained to the surface with a binary potential. They used this method to distribute particles homogeneously on a surface. This method had some disadvantages such as too many free parameters, and the low speed of the convergence. Recently, Meyer et al. [116] improved this method by proposing a new energy function that is scale invariant. They then showed that the resulting method can provide a fast and efficient mechanism to distribute particles on a sample surface. Later, Brosnon et al. [117] extended this idea to generate the mesh using the particles. In these methods, adding and deleting particles continuously is a time taking step, and this slows down the method. Fu et al. used meshless-particle method smoothed particle hydrodynamics (SPH) [3] approach to propose a method using particle relaxation technique to generate a mesh. Using bead-spring model have been used to generate the mesh [118] [119] but most of the methods either do not show the robustness or require a mesh for an initial condition [120]. It is also computationally expensive to generate an initial condition for bead-spring model as it requires adjacency. The method proposed in this work does not need a mesh for the initial condition, does not require adjacency, and is based on a simple force field model making it an efficient and robust method.

Method	Description	Remarks
Advancing Front Technique I [105][121]	Used two steps, first created vertices and then triangulated them to create elements	Computationally expensive
Advancing Front Technique II [122] [123]	vertex creation and element creation are combined. Boundary to inner progression for element creation,	Not optimized, Empirical constant unknown better mesh near the boundary.
Delaunay Method [124] [125]	Particles along with their Delaunay triangulation and then points are added/deleted as necessary.	Good Two dimensional, Problems in three three Dimension
Octree Method[126] [127][128][129]	scientific computing to generate many children nodes recursively	Not optimized for curved geometry
Smoothed Particle Hydrodynamics [112][3][114] [113]	A particle relaxation technique based on SPH formulations Particles are assumed to be fluid, equations of state for pressure need to be solved	computationally expensive
Spring- Bead Based Method [118][119] [120]	Particles connected with extensional and/or torsional spring, mesh is updated dynamically when the system reaches equilibrium,	Good for dynamic system Good for dynamic system

Table 5.1: Summary of some of the previous work on mesh generation

5.2 Defining the System

5.2.1 Initial Condition

In the proposed model, number density is defined as the quantitative measure of how dense the particles are distributed. If the overall density (ρ) is assumed to be uniform in the domain. Therefore, this number density, n_i, depends on the mass of the particle as

$$n_i \propto \frac{1}{m_i} \propto \frac{1}{r_i^d} \tag{5.1}$$

where r_i is the radius of the particle.

To create a mesh with controllable refinement, a target number density and target mass profile is defined for particles in the domain. The mass of particles varies in the domain and depends on the desired refinement. The ratio between largest to smallest mass in the domain is refinement ratio γ. Knowing the number density and mass profile, the total number of particles for the domain can be calculated as.

$$N = \int_{\Omega} n d\mathbf{V} \tag{5.2}$$

Once the total number of particles is found, the initial condition is generated by locating particles according to their mass and number density. Technically, any initial condition should be able to get final convergent results. Szabo[130] proposed a great method to fill the particles in a domain to generate a suitable initial particle location with a given number density. This approach is used in the present work.

5.2.2 Boundary Condition

The boundary treatment in a mesh generation process is important because the geometry of the system can be arbitrary and complex. The proposed force field model can handle fixed boundary and symmetric boundary conditions. In the fixed boundary, the particles initially are distributed along the boundary surface (in three dimensions) or boundary line (in two dimensions) according to their number density and mass. These boundary particles are not allowed to move in the direction normal to the surface. Ghost particles are used to support the boundary particles and particles near the boundary. In the symmetry boundary condition, the mirror image of particles is generated across the symmetry line.

Algorithm 1: Unsteady Force Field

```
compute Total number of particles Eq. 5.2;
distribute particles for initial condition;
for all particle $i$ do
    assign mass; find neighbors;
while convergence criteria do
    for all $i$ do
        compute $\mathbf{F}_i$ Eqs. 2.5 and 5.2;
        compute $\mathbf{v}_i(t + \Delta t)$;
        compute $\mathbf{x}_i(t + \Delta t)$;
        update neighbors;
        update mass ;
```

Algorithm 2: Steady Force Field

```
compute Total number of particles Eq. 5.2;
distribute particles for initial condition;
for all particle $i$ do
    assign mass; find neighbors;
while convergence criteria do
    for all $i$ do
        equate Force for each particle $i$ to be zero (Eq. 5.3);
    generate matrix $A$ to solve for $A\mathbf{x} = B$;
    solve for $x = A\backslash B$;
```

5.2.3 Numerical Details

Although, there is no direct relationship between convergence speed and accuracy of any numerical simulation. Typically, there is a general trade-off between the two. To achieve higher accuracy in simulation, one might have to give in on the convergence speed (computational cost). In mesh generation, the interest is to achieve high convergence speed. The accuracy of simulations matters but it is not higher priority as there is no physics involved. Eventually, the final particle positions is the goal. From the described numerical method and force profile in previous chapter, there is a modification made here. The force profile is taken to be linear

Algorithm 3: Hybrid Method

```
compute Total number of particles Eq. 5.2;
distribute particles for initial condition;
for all particle $i$ do
    assign mass; find neighbors;
while convergence criteria* do
    for all $i$ do
        compute $\mathbf{F}_i$ Eqs. 2.5 and 5.2;
        compute $\mathbf{v}_i(t + \Delta t)$;
        compute $\mathbf{x}_i(t + \Delta t)$;
        update neighbor;
        update mass;
while convergence criteria do
    for all $i$ do
        equate Force for each particle $i$ to be zero (Eq. 5.3);
    generate matrix $A$ to solve for $A\mathbf{x} = B$;
    solve for $x = A\backslash B$;
*Until the particles have reached to almost steady state
```

After calculating the net force on a particle, Verlet integration(section 2.3.2) is used to integrate equations to determine the particle velocity. The algorithm is summarized in the pseudo-code in Algorithm 1.

Steady State Solver

Since the resultant force should eventually be zero on each particle, it is possible to solve the equations directly. The position of the particles can be directly calculated using iteration approach. The speed of this solver strongly depends on the initial condition of the system.

$$\sum_{j} \mathbf{F}_{ij} = 0 \tag{5.3}$$

Hybrid Solver

While the unsteady method is robust and solved implicitly, it is still not as fast as a steady state solver. For the steady state solver to converge, the particles should not be a lot further from their convergent solution. In the hybrid method, the equations are solved implicitly using an unsteady solver until a good transition position. The steady approach is used after a good position is obtained. The hybrid method is both as robust as unsteady and faster than the unsteady method.

5.2.4 Voronoi Tessellation-Delaunay Triangulation Duality

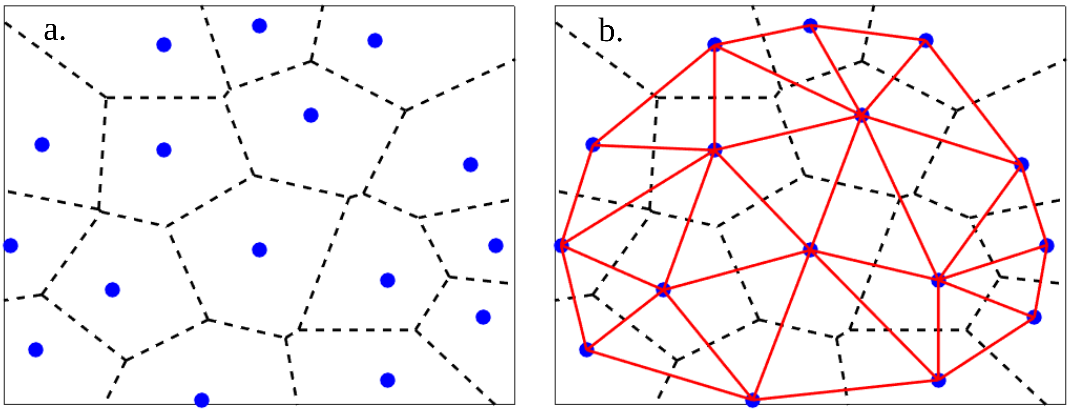


Figure 5.2: A description of transformation from Voronoi Tessellation to Delaunay Triangulation. The blue symbols are nodes and the cell center of Voronoi Tessellation. These nodes become vertices of the triangles in Delaunay triangulation

Voronoi tessellation is a method to divide a domain with several into small cells. For each node, all points in its cell are closer to that node than to any other node. The boundary segments between two cells are equidistant to the two nodes of those corresponding cells. There is a dual relationship between Voronoi Tessellation and Delaunay triangulation, where one can be transformed into another. The circumcentre of Delaunay triangulation, when connected form the Voronoi cells with all the triangulated points on the circle and no point inside the circle. Similarly, the center of Voronoi cells are vertices of Delaunay triangulation. In our proposed mesh generation method, we use this duality to transform the particles into triangulated mesh. We first generate Voronoi tessellation for the convergent particles and then use the duality to find the Delaunay triangulation. The obtained Delaunay triangulation is the desired mesh. One example of converting Voronoi tessellation to Delaunay triangulation is shown in Figure 5.2. In the figure, the blue nodes are the particles and the centre of Voronoi cells. After the triangulation, the same nodes become the vertices of the triangles. The triangulation is shown in red color with Voronoi cells in black dashed lines.

In this work, we use particle position to generate Voronoi tessellation, and then using the duality, we transform it to Delaunay triangulation. The obtained triangulation gives a simple triangular mesh of high quality.

5.3 Results for Mesh Generation

In this section, a criterion for mesh quality is first described. Then, several benchmark cases(in 2D and 3D) are simulated using the proposed method.

5.3.1 Criteria for Mesh Quality

A mesh with smaller elements is usually of higher quality than a coarser element. The quality of the mesh is checked by parameters that will be defined below.

The mesh quality is usually described by several qualitative parameters such as skewness and angle distribution among many others. The isotropic triangular mesh are generated from our proposed method. Various quantitative parameters are used to evaluate the quality of isotropic triangular mesh obtained by the proposed approaches[131][132]. In this work, the goal is not to compare different methods of quantifying the quality of the mesh, but to use a few of them to check the mesh quality. There are multiple definitions and ways to calculate skewness, equiangular definition of skewness is used in the present work,

$$\text{Equiangle Skewnss} = \max\left[\frac{\theta_{max} - \theta_e}{\pi - \theta_e}, \frac{\theta_e - \theta_{min}}{\theta_e}\right] \tag{5.4}$$

where θ_e is the angle of the equiangular face of the cell. Since we are generating triangular mesh, the value is 60^0. θ_{max} and θ_{min} are the maximum and minimum values of the angle for all cells. We also measure the angle of the triangle θ[133].

The value of skewness should not exceed 0.85 and θ for a triangular mesh should be as close to 60^0 as possible for a better mesh. Another property that we calculate is the mesh quality q [134] and it is defined as

$$q = \frac{3\theta_{min}}{\pi} \tag{5.5}$$

The value of q should be close to unity for most of the mesh elements. The mean value of q should exceed 0.5.

5.3.2 Benchmark Cases

A disc and a square

Two cases are considered here in the disc and square case. In the First case, a circular disc with refined mesh on the boundary with larger coarser mesh near the center is simulated. The boundary particles of the disc are fixed and are initially distributed according to the target mass and target number density profile. A rectangular domain with a space of circular shape inside it is the second case. In this case, the desired refinement of the mesh is on the surface at the center. This case represents 2D complex geometry. For these cases, the final convergent position along with triangular mesh is shown in the Figs. 5.3. In Fig. 5.4, the distribution of all the angles for the triangular mesh is plotted in a histogram and a major fraction of all the angles are in the range $45 - 75$ degrees. This signifies that triangular elements are closer to equilateral triangles and mesh elements are of high quality. Further, a mesh quality parameter q defined by Eq. 5.5 is also calculated for each triangular element of the mesh and shown in Figure 5.5. The value of q is found to be very high which is an acceptable value for mesh. The mesh skewness for the circular and square discs are calculated to be 0.11 and 0.13, respectively, and they are well under the threshold value of 0.85.

Five Rings in a square

In this second benchmark case, the capability of handling the curved boundary is demonstrated. A rectangular domain is chosen and five rings are placed inside. The desired mesh is refined closer to the ring boundary and coarser mesh further from the boundary. The final particles position and generated mesh are shown in Figure 5.6. The particle size in this figure is kept constant for visibility. The

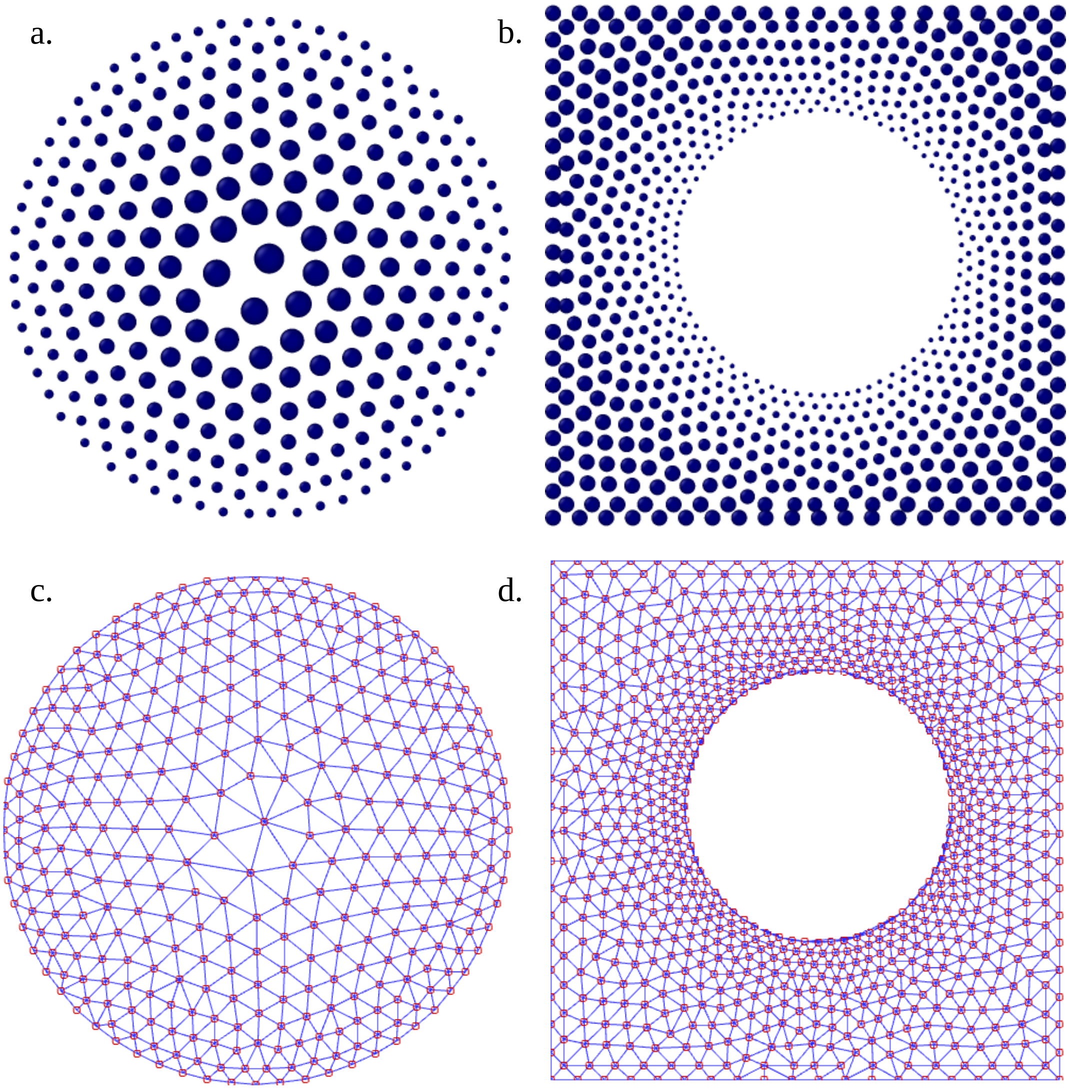


Figure 5.3: A disc and a rectangular domain with disc-shaped space are simulated separately. The final particle position is displayed in figure (top), the radius(proportional to mass) profile is shown by the use of colored contour, The final mesh for both the cases are presented are also shown(bottom)

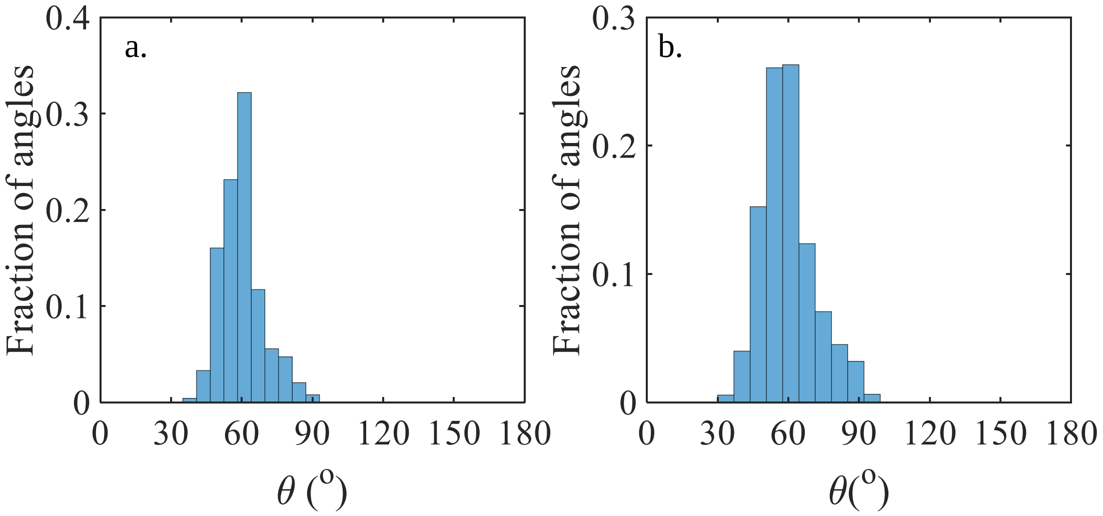


Figure 5.4: The histograms of angles in triangular mesh for the cases in Figure 5.3

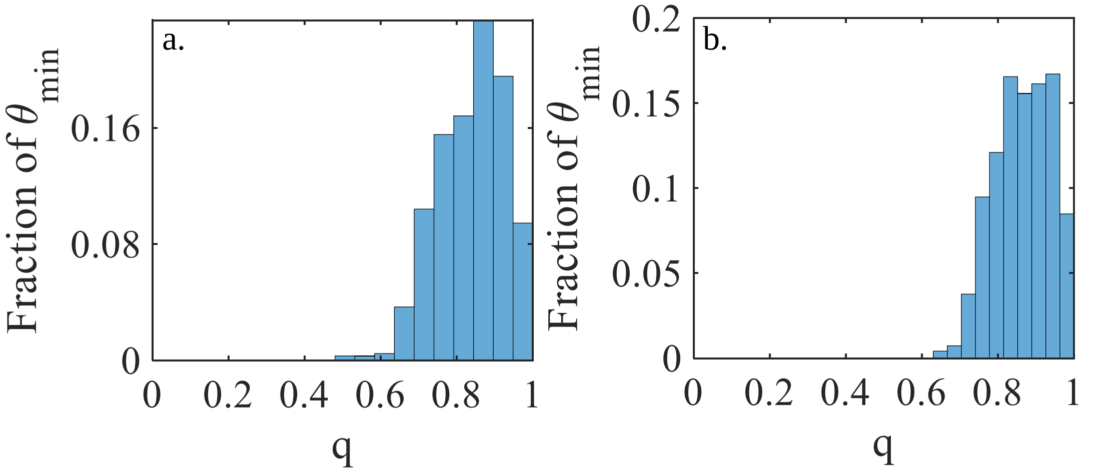


Figure 5.5: The histograms of mesh quality in triangular mesh for the cases in Figure 5.3

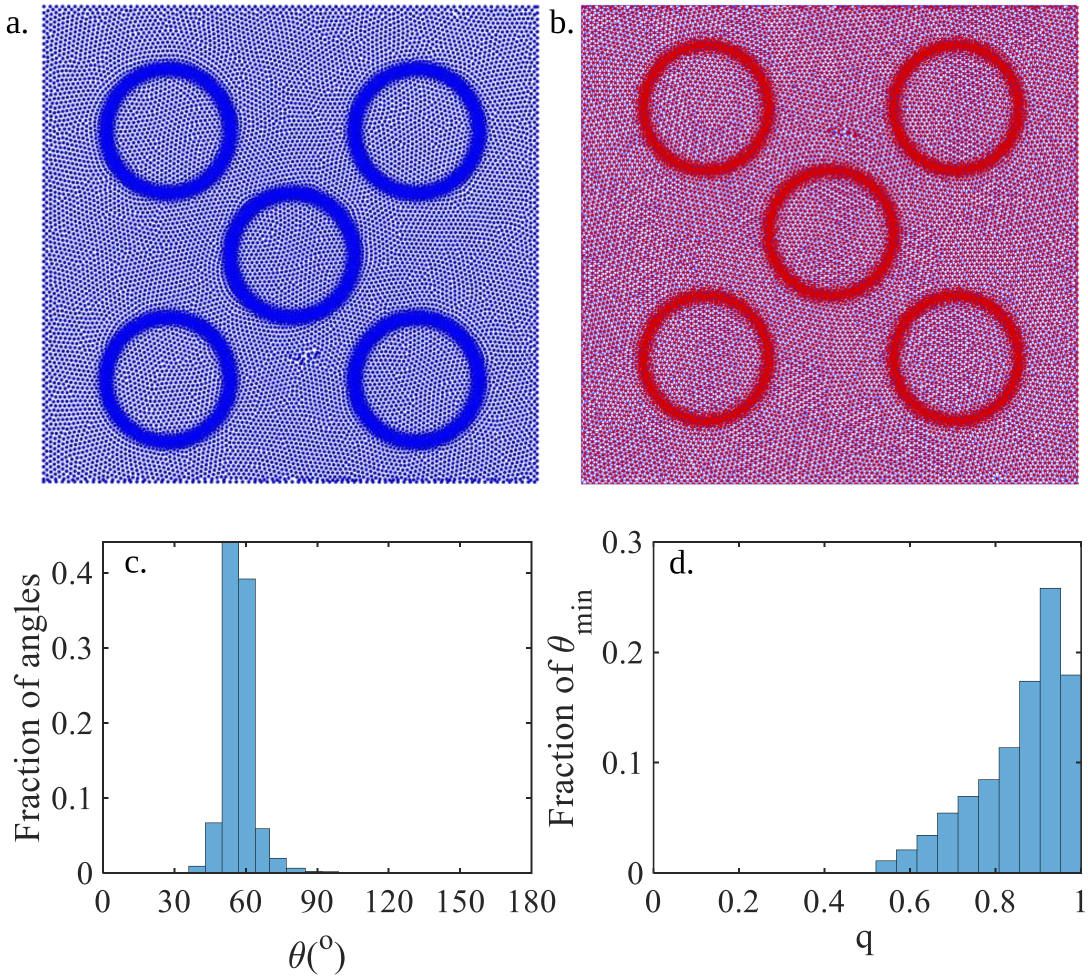


Figure 5.6: a. The particle distribution b. The final mesh for five ring case is shown. c. The distribution of angle in the mesh element and d. Mesh quality q

angle distribution is plotted in a histogram and shown in Figure 5.6c. A majority of angles are in the range $45 - 75$, some mesh elements have some skewness and the angle are not close to 60 degrees but overall the skewness is calculated to be 0.12. The average mesh quality for this case q is calculated to be 0.86 which is well above the cut-off for a good quality mesh.

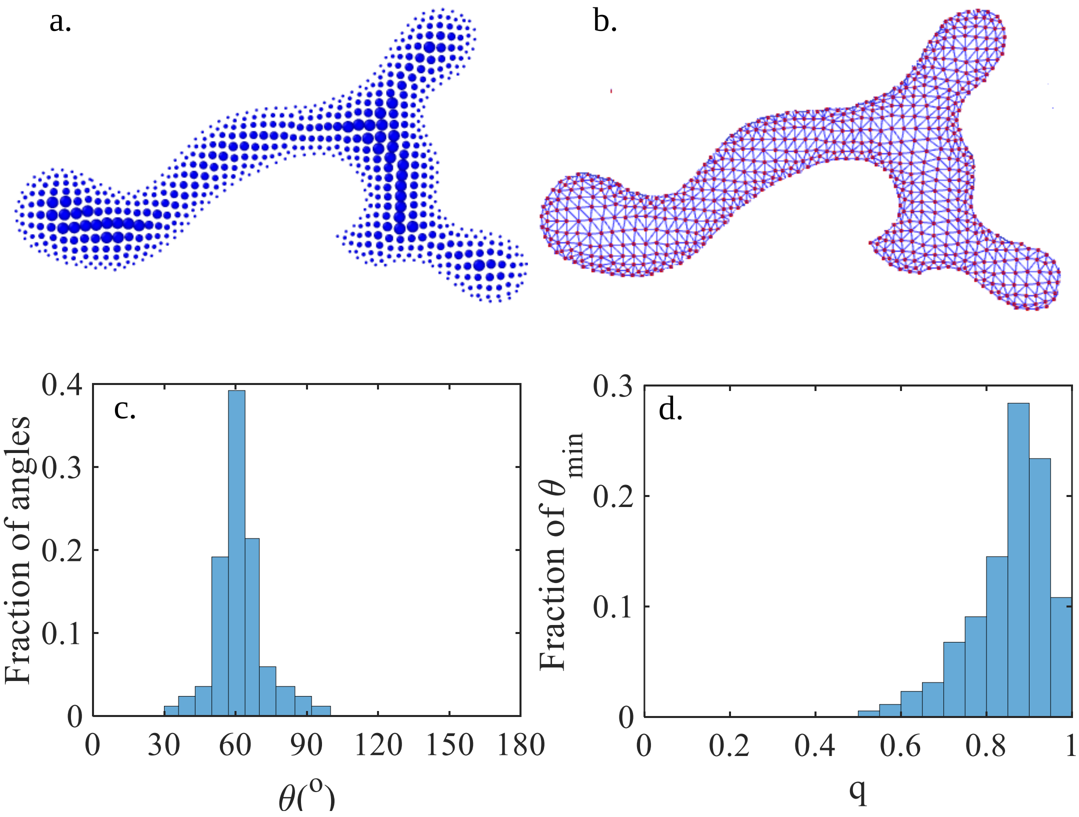


Figure 5.7: a. The particle distribution b. The final mesh for Jordan curve case c. The distribution of angle in the mesh element and d. Mesh quality q

Jordan Curve

A Jordan curve is a planer non-intersecting loop that divides the plane into two sub-regions, inside and outside. The final convergent position of the particles and the mesh generated by using VT-DT duality is shown in Figure 5.7. The method can generate a mesh of good quality. The quality is shown in the histograms of mesh element angle and the mesh quality(q). The particles in this method can be of different size and thus any small or random curvature in the domain can be meshed with the same efficiency and accuracy.

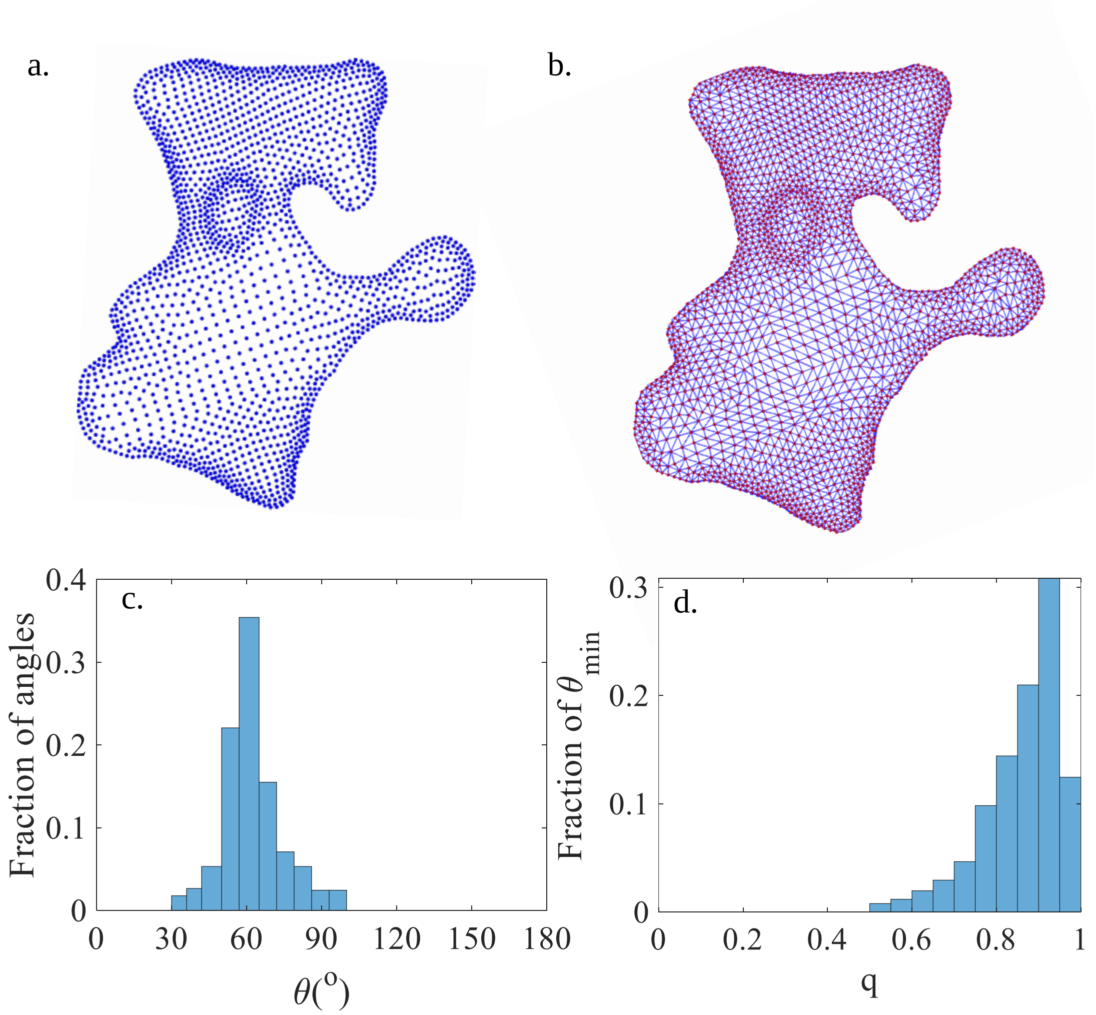


Figure 5.8: a. The particle distribution b. The final mesh for an island in a lake case is shown. c. The distribution of angle in the mesh element and d. Mesh quality q

A lake with an island

A lake with an island is chosen as the next benchmark to generate a mesh. Two random Jordan curves are chosen for the geometry of the lake and island making sure the island is smaller than the lake. The inside sub-region of the lake including the island is the desired area for mesh generation. The boundary of the lake and the interface between the lake and the island is chosen to be a region with refined mesh. The particles are transformed into the triangular mesh and are shown in Figure 5.8. The average Mesh skewness and mesh quality q are found to be 0.26 and 0.76 respectively which is well over the criteria for a good mesh. The angle distribution for triangular mesh element and mesh quality distribution is shown in Figure 5.8.

Porous Media

Porous media often have no strictly defined boundary so it becomes more challenging to generate mesh for them. Using the proposed method, each media can be considered as a Jordan curve and the mesh for the whole domain can be generated. If there is any island-type structure with a different phase, that part can be considered similar to the previous benchmark case of A lake with an island. In this benchmark, a two-phase porous media domain is taken as a benchmark case for mesh generation. The final particle position is obtained and shown in Figure 5.9. The colors of different particles denote different phase. The particles' position is then transformed into triangular mesh and is shown in the Figure 5.9. The triangular elements are as expected and the angle is mostly near 60^0. Furthermore, the mesh quality q is plotted in a histogram and is found to have a mean of 0.89.

Case	**Equiangle Skewness**	**Mesh quality(q)**
One Ring in a square	0.11,0.13	0.86,0.84
Five Rings in a square	0.12	0.86
Jordan Curve	0.2	0.8
Island in a Lake	0.26	0.75
Porous Media	0.17	0.89

Table 5.2: Values of various mesh quality parameters for all the benchmark cases

For all the benchmark cases shown, the obtained mesh is of good quality. The refinement zone is chosen in all the benchmark cases. If the refinement is not needed then the refinement ratio γ can simply be assumed as one. The mass of each particle becomes equal in that case. The mesh quality is quantified with several parameters such as angle distribution, Mesh skewness, and q. The mean values for the last two parameters are also shown in Table 5.1. The angle of all the mesh element is also shown on the histograms for all the cases, and the values of and distribution of the angles reflect the quality of mesh of obtained.

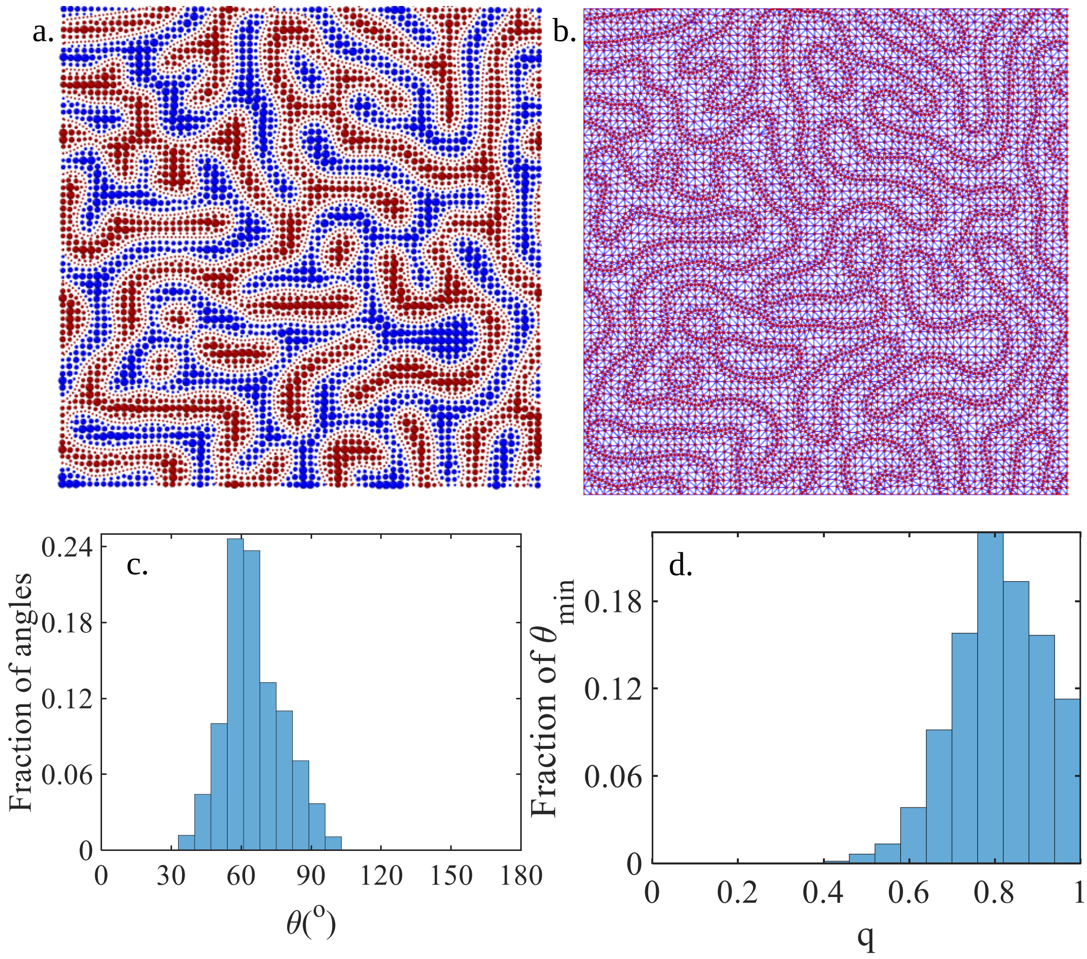


Figure 5.9: a. The particle distribution b. The final mesh for Porous media case is shown. Different colors denote two separate phase c. The distribution of angle in the mesh element and d. Mesh quality q

Three dimensional Cases

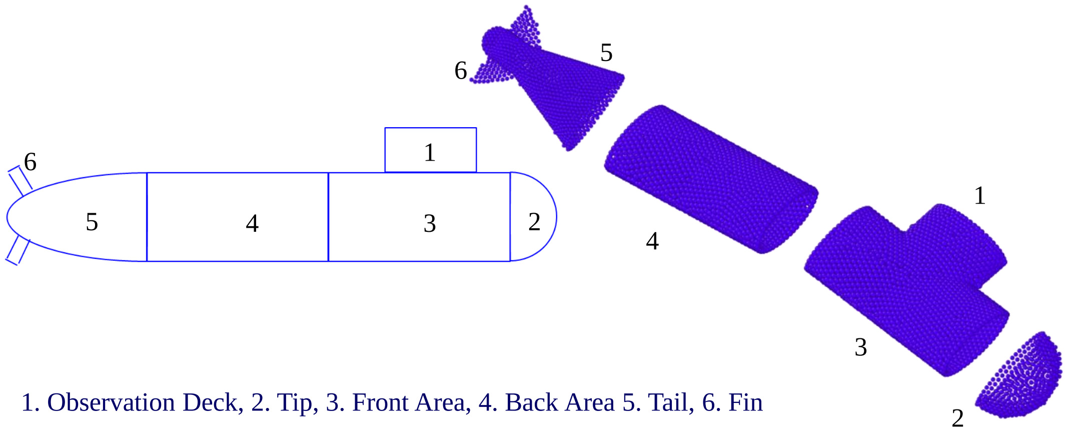

6
5
4
3
2
1
6
5
4
1
3
2
1. Observation Deck, 2. Tip, 3. Front Area, 4. Back Area 5. Tail, 6. Fin

Figure 5.10: Schematic of the blimp geometry(on the left) and particle distribution on the surface

For the three-dimensional cases, there are two things to understand. The surface mesh is in three-dimensional domain but is a two-dimensional mesh. When generating this surface mesh for three-dimensional geometry, one needs to make sure the particles move only on the surface geometry and do not move out of the surface. For two-dimensional cases, it was easier as there was no third axis(eg. Z-axis), however in the three-dimensional cases, a local surface spline needs to be defined. This step is obtained by either using the equation of the surface(if already known) or dividing the surface into small sections and doing a polynomial fit. Once the surface spline is obtained, the simulation is allowed to run and the particles are constrained to the surface. This is achieved by dividing the force on any particle into two components, one parallel to the surface and the second normal to the surface. The particle then is allowed to move only along the surface. This allows for the particle to remain on the surface and reach equilibrium. One example of such motion is shown in 5.10. The geometry of a blimp is chosen and its schematic is shown. To generate the surface mesh for the blimp, a series of three-dimensional splines are generated and the particles are allowed to move only on the surface. The final equilibrium position of the particles on the blimp surface is also shown.

Once the surface particles are obtained, then Voronoi Delaunay triangulation can be used to generate the surface mesh. The next example chosen for surface mesh generation is a teapot. The surface mesh is generated and shown in Figure

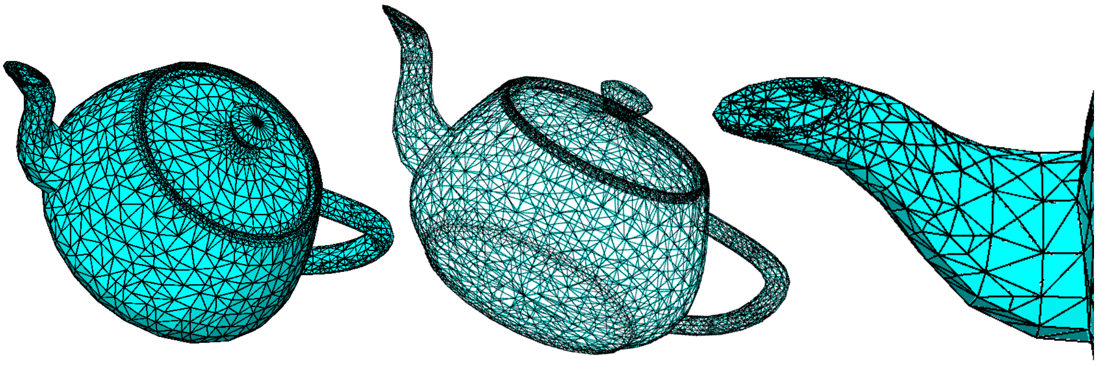


Figure 5.11: A teapot with tetrahedral mesh with two cut sections to show the inside mesh

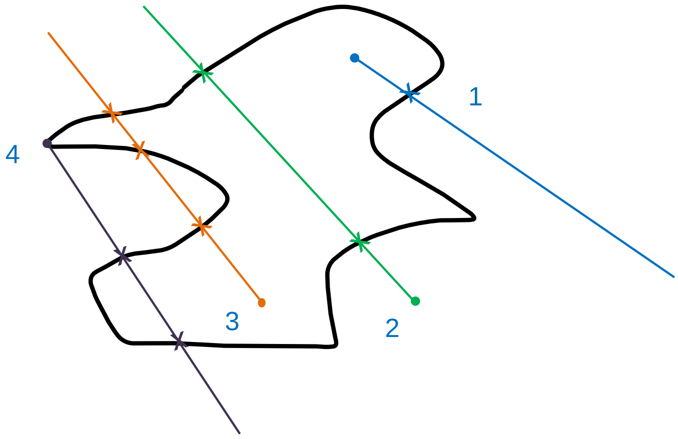


Figure 5.12: Schematic of identifying whether a point is inside or outside the domain

5.11. The teapot geometry has several challenging parts to mesh. On the top, the top knob needs to be meshed finer than other parts, and a more refined mesh is chosen there. Similarly, the spout and the lid-pot interface are also chosen to be more refined. The final mesh with different viewing is shown in Figure 5.11. The mesh quality is calculated and none of the elements are extremely skewed.

To generate a three-dimensional mesh, first, the domain needs to be defined. Once the surface mesh is obtained, the three-dimensional tetrahedral mesh can either be generated inside or outside. To identify whether a point is outside or inside a domain, a line is drawn toward one of the boundaries (Figure 5.12, if the line intersects with the domain boundary odd number of times then the point is inside the domain. Similarly, if the line intersects with the domain boundary an even number of times, then the point lies outside the domain boundary. Using this logic, the inside and outside of the domain are defined. Once this has been determined, the total number of particles are calculated and filled in the system. The system is set to reach equilibrium and final positions of the particles are obtained.

For the rest of the cases, only final mesh is shown as that is the desired result. It is important to note that the final mesh is obtained from the particle positions, similar to shown in the above examples.

Once the particles' positions are obtained, VT-DT duality is used to generate tetrahedral mesh. In the next few paragraphs, several three-dimensional cases are chosen to demonstrate the capability of the method. Next, three-dimensional case of a sphere within a cube is chosen to generate the mesh. This case is can be considered as an extension of benchmark 1 to three dimensions. The convergent particles' positions are shown in Figure 5.13. The Voronoi tessellation obtained in a three-dimensional case is a convex tetrahedral(as opposed to a triangle in the case of two-dimensions). The VT-DT duality still holds in three dimensions and the triangulation results in tetrahedral mesh(triangles in the case of two dimensions). The particles' positions are transformed into the tetrahedral mesh and shown in Figure 5.13. A cross-section of the cube with the particle position(with the sphere in a different color) and mesh section is also shown. Overall the mesh obtained in three-dimensional benchmark is of good quality and can be used for simulations in porous media and representative volume elements methods. Next, a cylinder in a cuboid is chosen. The total number of tetrahedral element chosen for this case are 5285, the mesh generated are adaptive with larger tetrahedral inside. The mesh is refined near the interface. In Figure 5.14, an isotropic and front view of the surface mesh is shown. The surface mesh obtained is of high quality and none of the elements are highly skewed. In Figure 5.15, few slices are

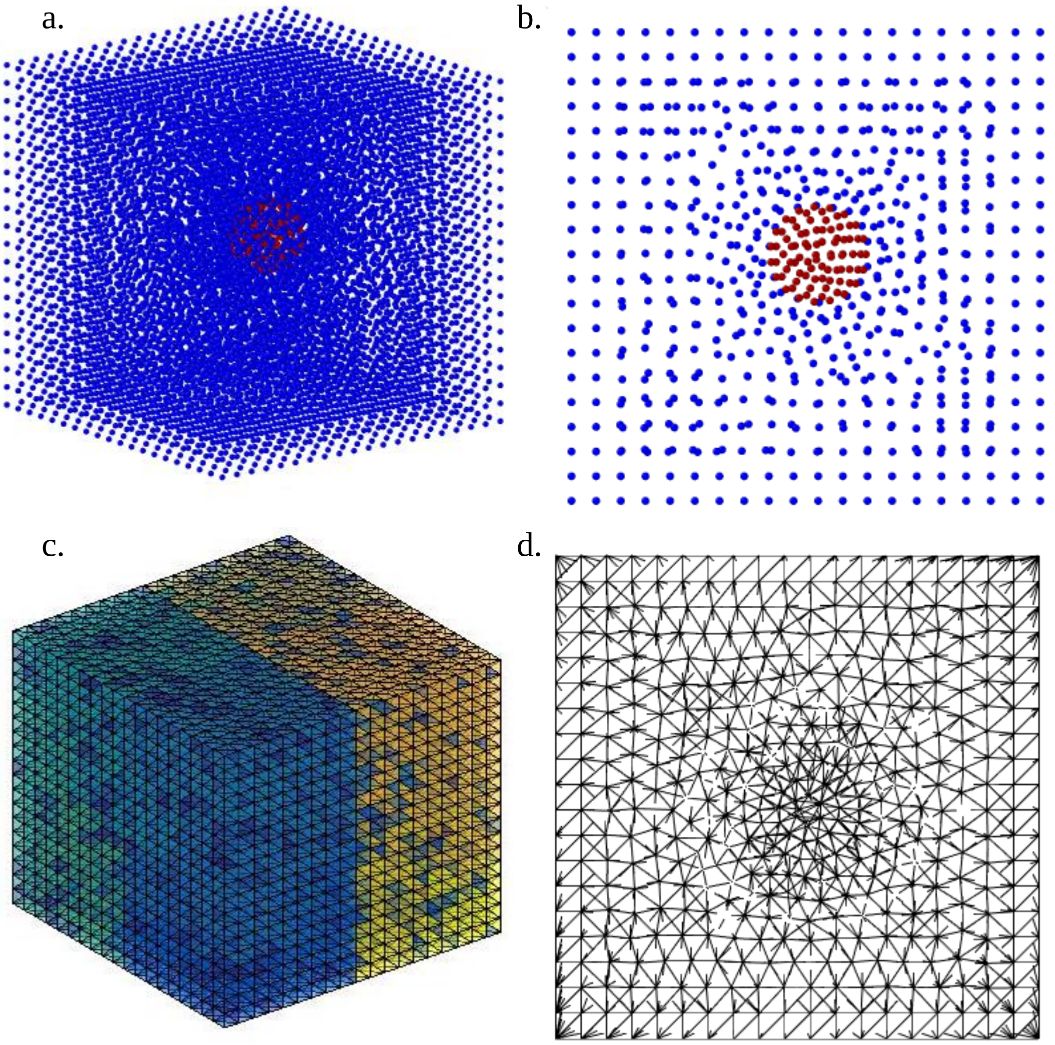


Figure 5.13: a. The particle distribution b. The final mesh for three-dimensional case is shown. c. The distribution of angle in the mesh element and d. Mesh quality q

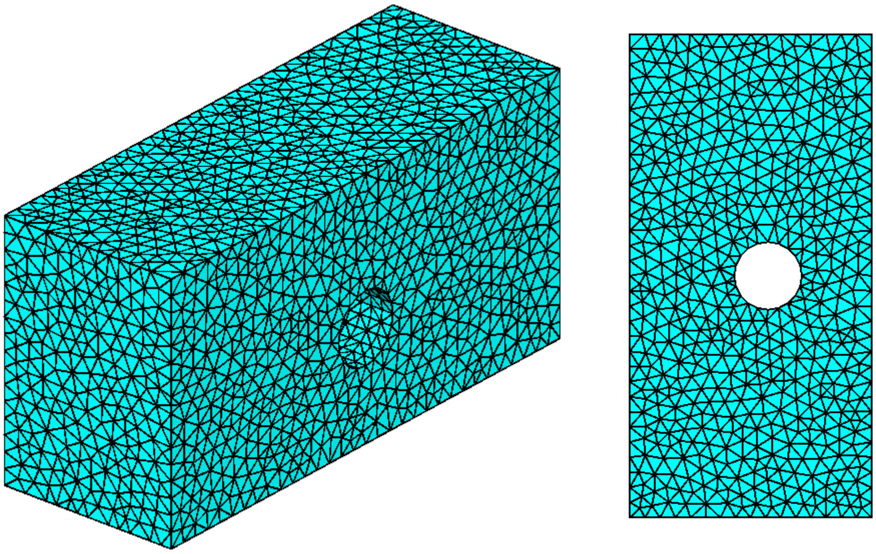


Figure 5.14: A cylinder in a cube and its tetrahedral mesh is shown

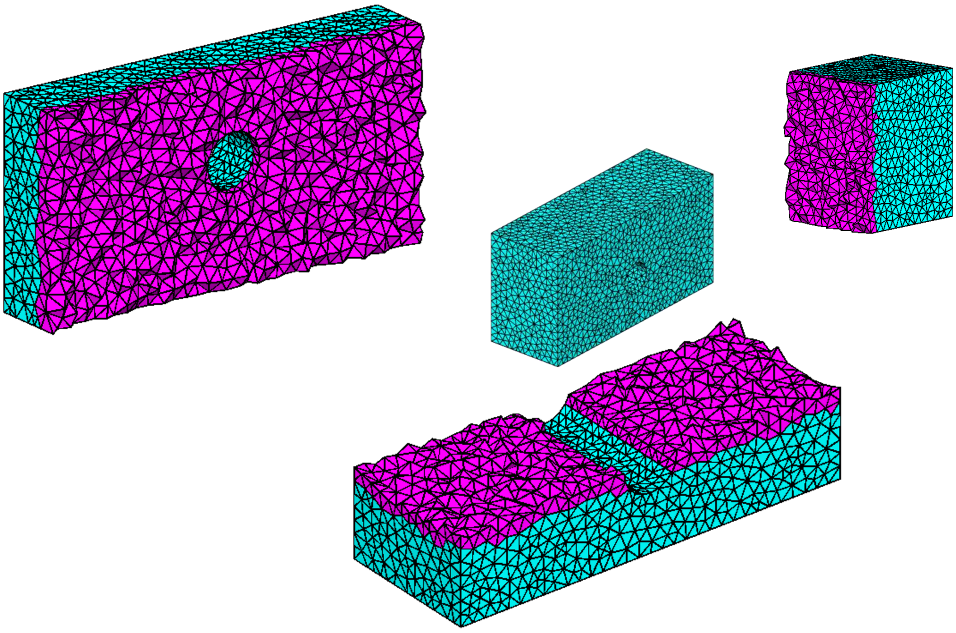


Figure 5.15: For the case 'A sphere within a cube' Different cut sections inside tetrahedral mesh

cut from the mesh to show the tetrahedral mesh. The tetrahedral mesh obtained has no highly skewed element.

Next, several examples from Mechanical Engineering are chosen. First, a hinge is chosen to generate tetrahedral mesh. In Figure 5.16 and 5.17, one part of the hinge and its mesh is shown. The total number of tetrahedral element chosen for the first part of the hinge is 16445. Different slices of the mesh are taken and the inside mesh is shown in 5.17. In the Figure 5.18, the other complementary part of the hinge and its mesh are shown. For this part of the hinge, the total number of tetrahedral elements is 15442.

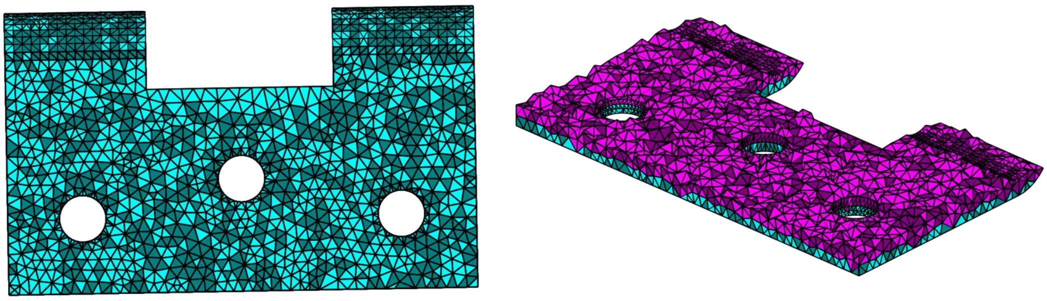


Figure 5.16: a. Three-dimensional tetrahedral mesh for a hinge, A section has also been taken and shown in the right

The next case chosen is a 'nut' with threads. The mesh for this case is shown in 5.19. The overall mesh in this case is of good quality, but near the threads, the elements need to be very small. Because of this, the total number of elements, in this case, is higher than usual(46638).

The next case chosen is a gear geometry. The schematic of the geometry is shown in 5.20. The gear geometry is inspired by the sprocket of the bicycle. The chosen gear has 32 teeth/grooves. These grooves are isosceles trapezoids in shape. The gear also has a center cylinder that has a circular hole in it. The center cylinder is connected to the outer teeth by four hands. Each of these hands has an oval shape hole in it. These types of geometries have holes to reduce the weight. The thickness of the center cylinder is more than the rest of the gear. For this case, the desired mesh chosen is uniform everywhere with a high number of particles. The generated mesh is shown in Figure 5.21. The mesh at different parts with zoomed in and slice (to demonstrate the inner tetrahedral mesh) is shown in 5.22. The total number of elements in this gear is 374692. The last case chosen for this set is a socket. In Figure 5.23, the surface mesh is shown at different angles. In the same figure, a cross-section is also taken to show the inner tetrahedral mesh. The total number of elements in this case is 834125.

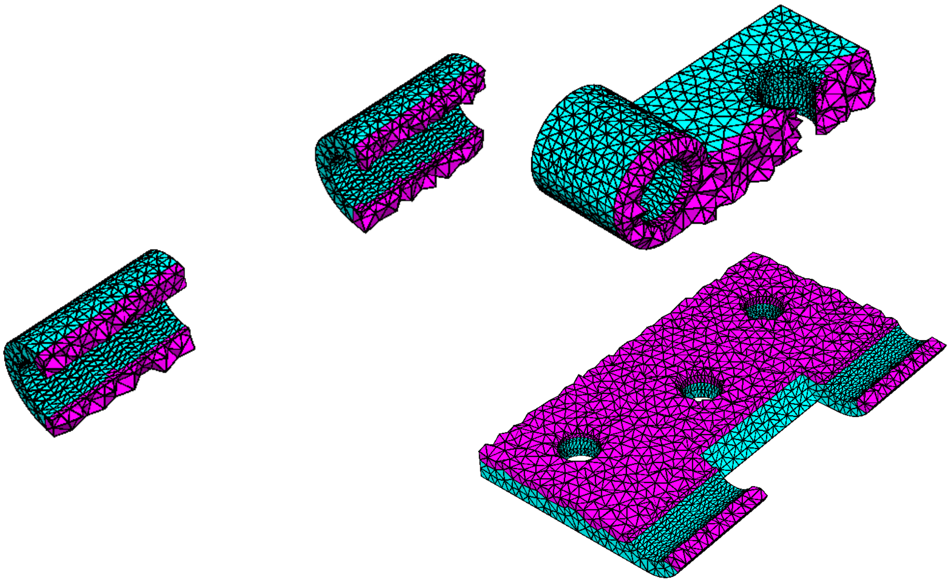


Figure 5.17: a. Some more sections from other axes to show the adaptive meshing mechanism

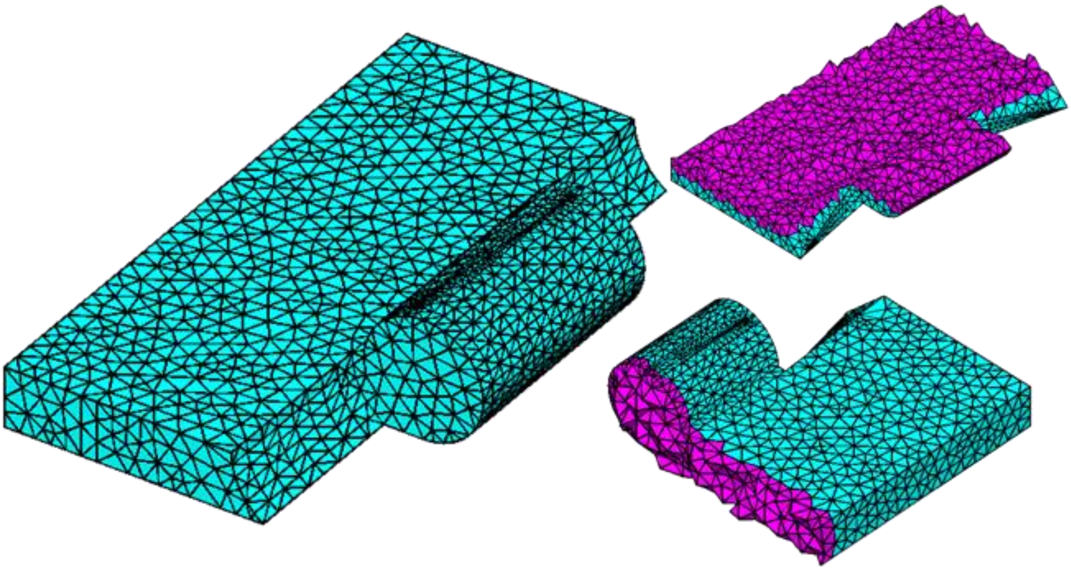


Figure 5.18: For the hinge in 5.16, the mesh for second part of the hinge, the inside tetrahedral mesh are also shown in the right

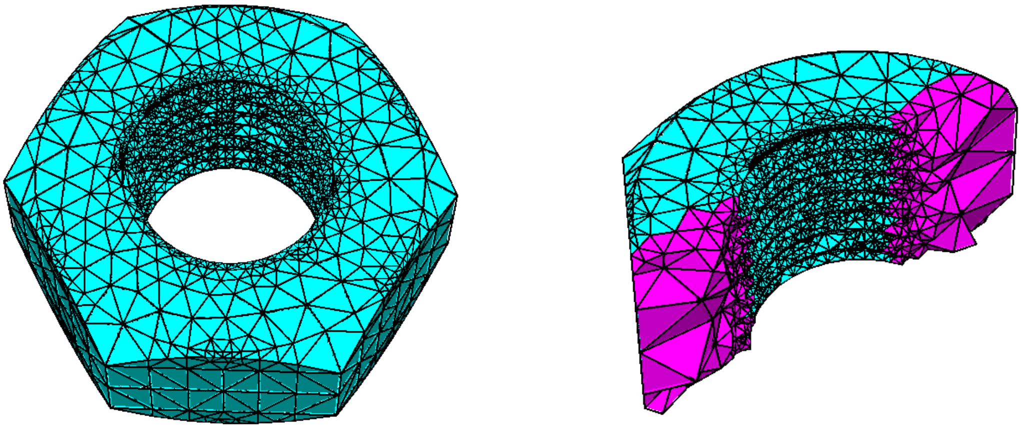


Figure 5.19: The case ’nut’ and its tetrahedral mesh with cut section to show the inside mesh

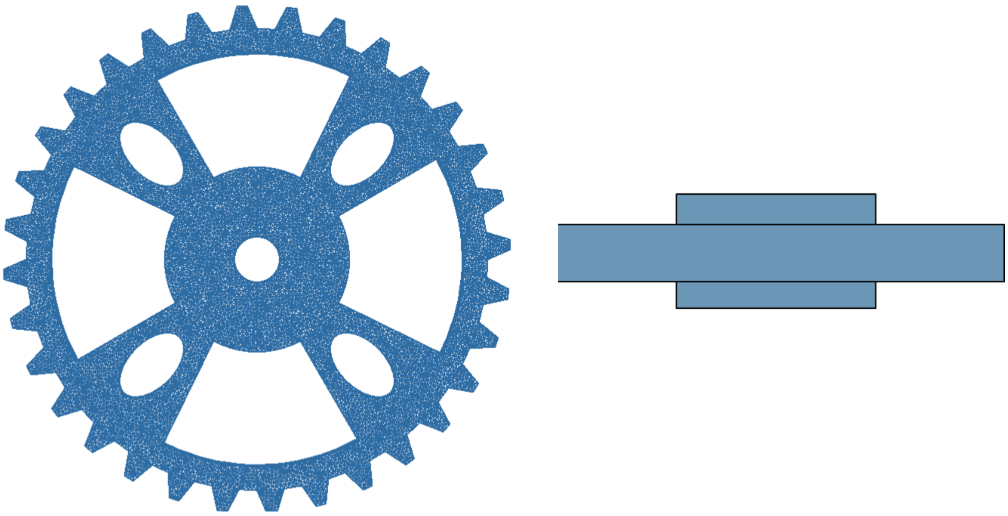


Figure 5.20: Schematic of the gear geometry top view(on the left) and side view(on the right)

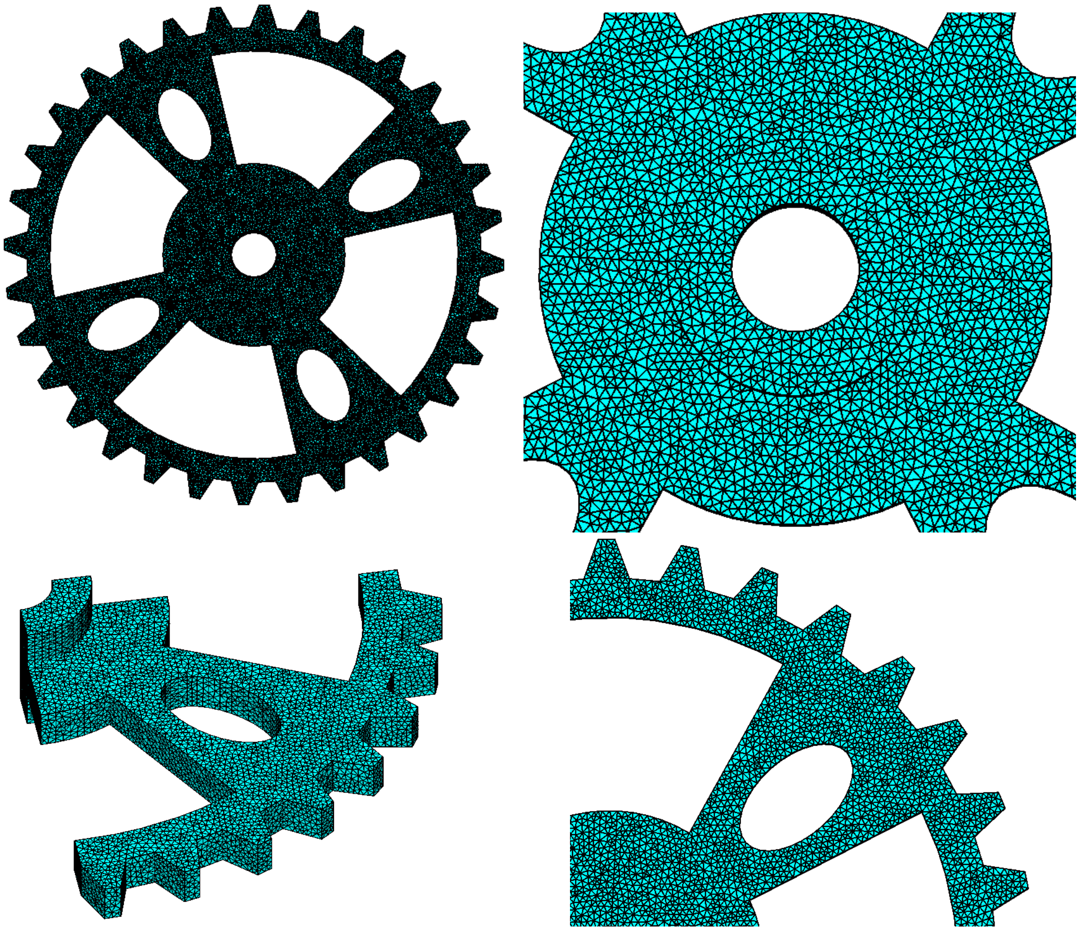


Figure 5.21: The gear full tetrahedral mesh and different sections(zoomed in) to show the detailed mesh

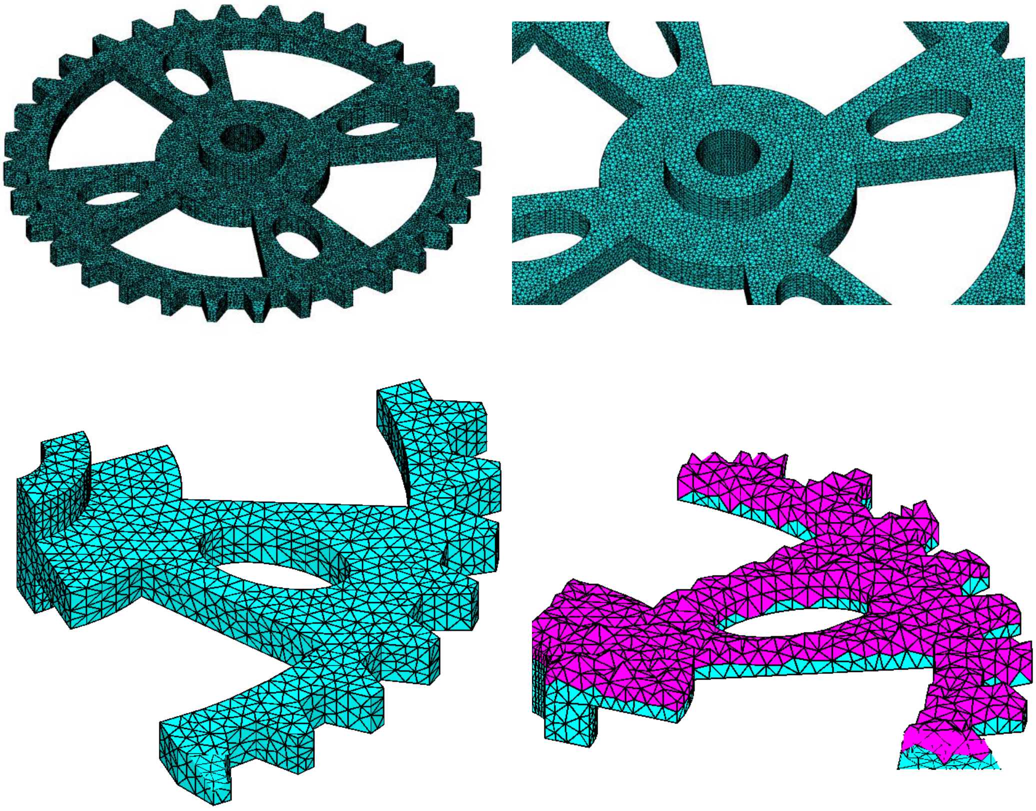


Figure 5.22: Different viewing angle for the gear case, A slice through the X plane to show the inside tetrahedral mesh

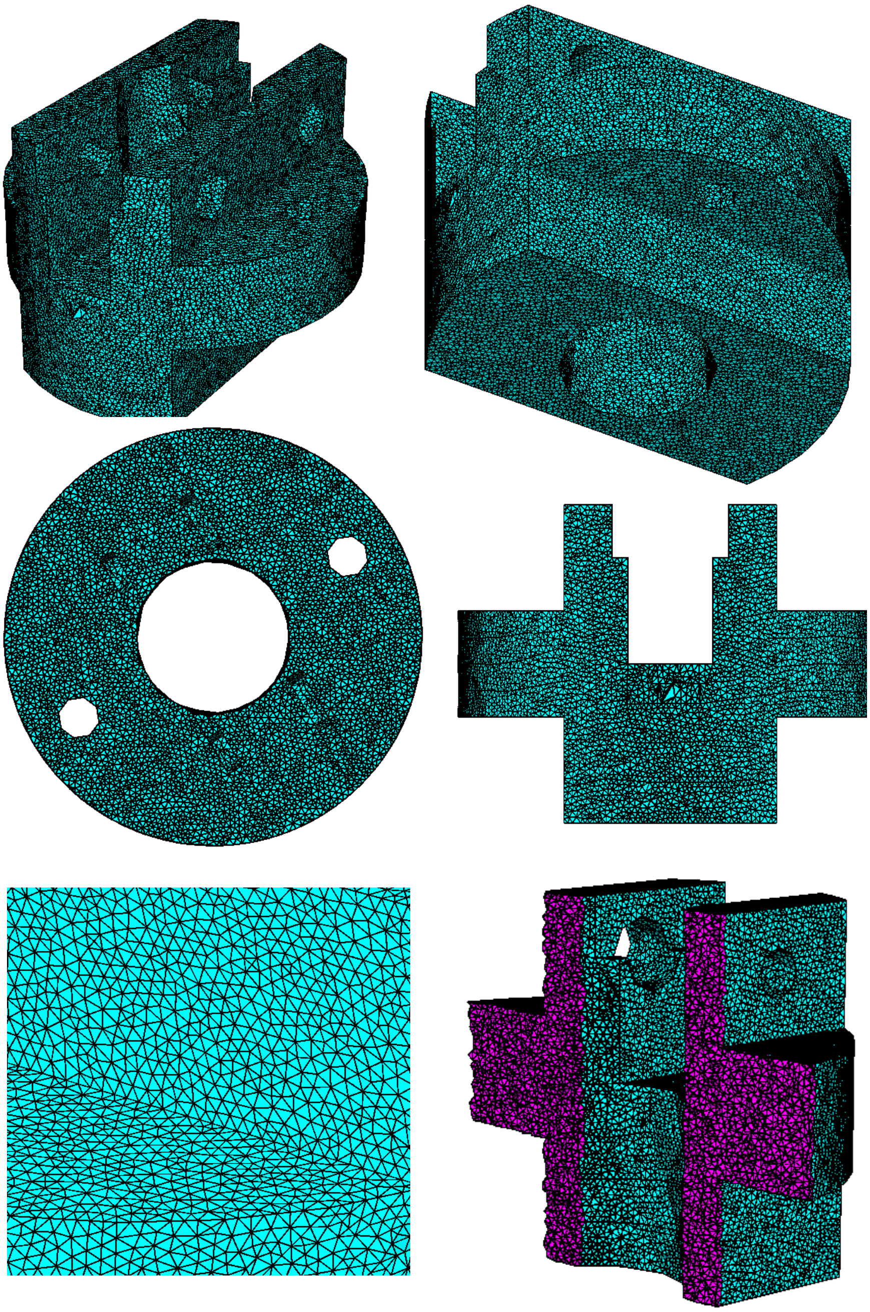


Figure 5.23: A socket and its mesh with detailed mesh with different viewing angles and cross-sectional cut

Apart from the examples from the engineering world, the proposed method is capable of generating a mesh for other examples as well. Unlike engineering world, these geometries are not defined by equations or any shape. For these examples, the geometries are imported using an STL file. After the geometry is imported, surface splines are generated and then the surface mesh is generated. Once the surface mesh has been generated, a similar process is followed to generate the final three-dimensional mesh.

Three cases are chosen for this set of examples. In Figure 5.24, 5.25, and 5.27, a parasaurolophus is chosen to created the mesh. In Figure 5.24, different views of the surface mesh are shown. Inner tetrahedral mesh is shown in Figure 5.25. The total number of elements, in this case, is 20457.

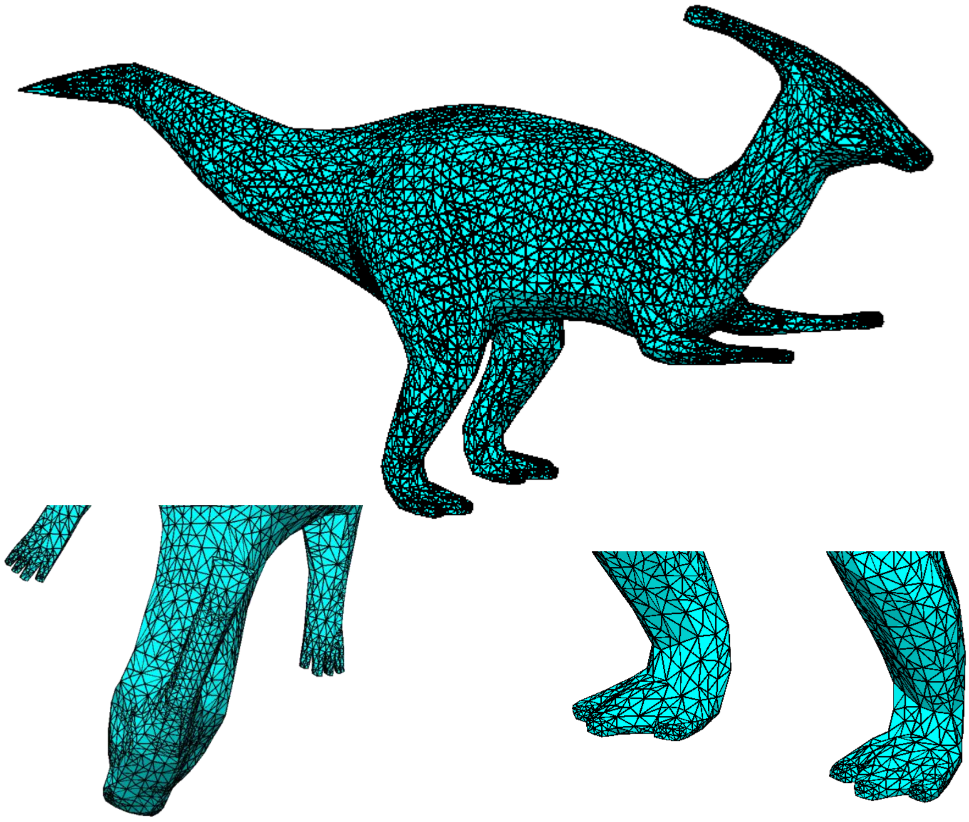


Figure 5.24: Tetrahedral and surface mesh shown for the 'parasaurolophus'

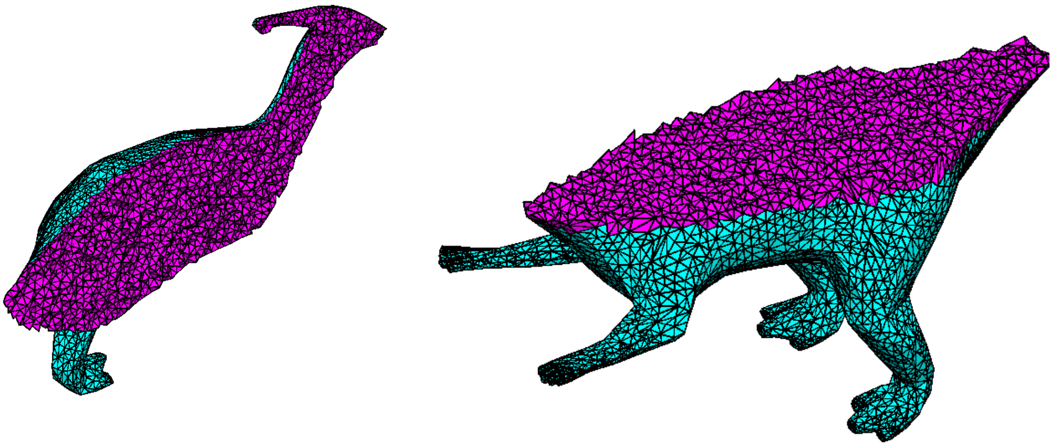


Figure 5.25: Sliced sections for the 'parasaurolophus' to show the tetrahedral mesh

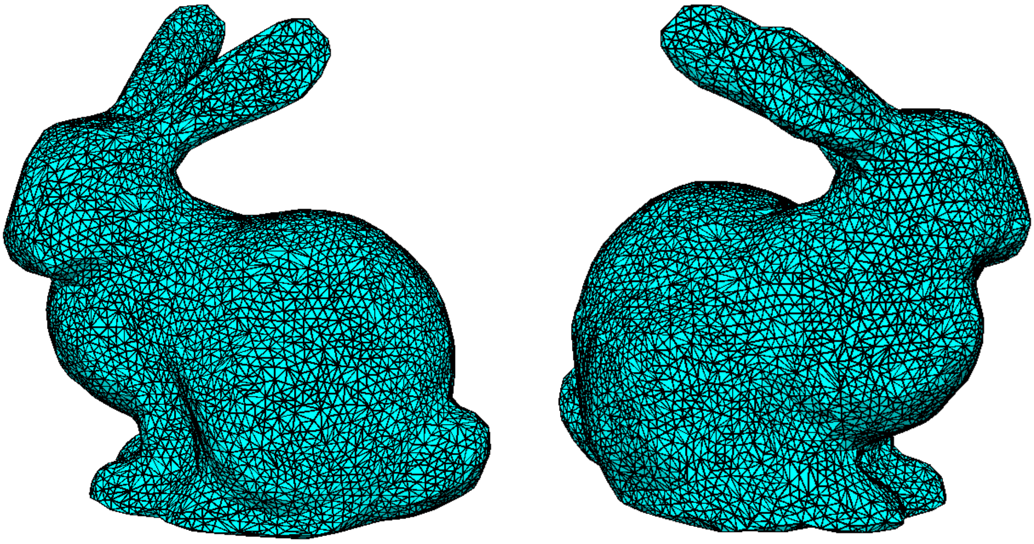


Figure 5.26: Tetrahedral mesh generated for 'Stanford Bunny' and two angles are shown

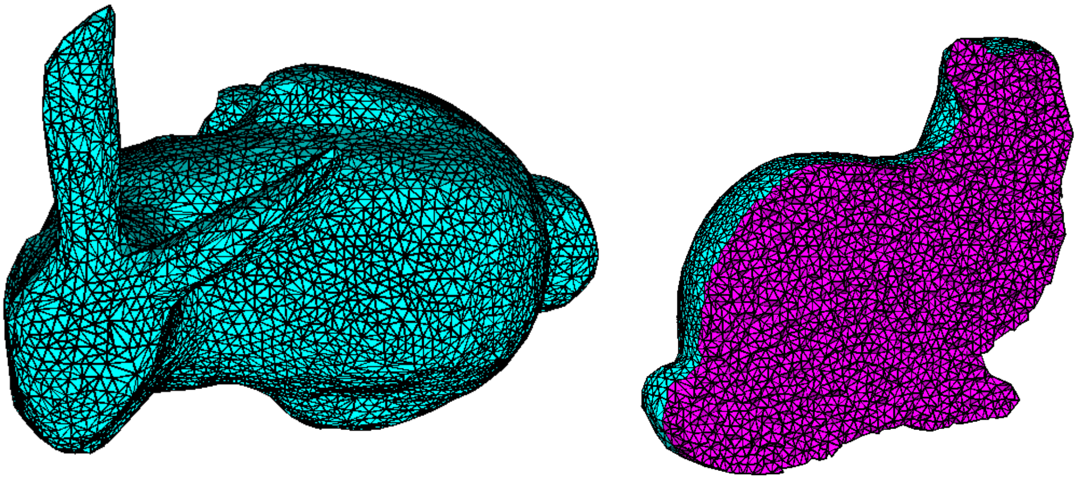


Figure 5.27: The surface mesh of 'parasaurolophus' and Stanford bunny

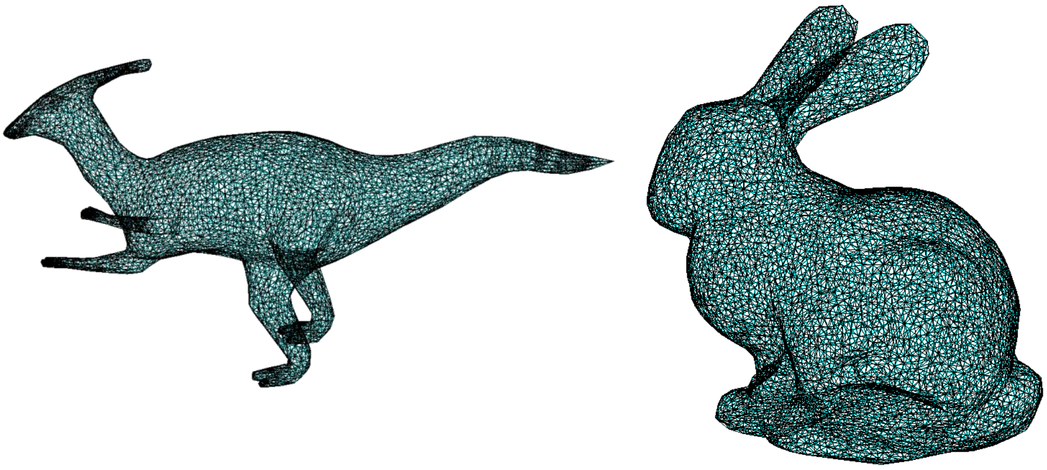


Figure 5.28: The surface mesh of 'parasaurolophus' and Stanford bunny

The next case is the famous Stanford bunny. The surface mesh of the Stanford bunny is shown in Figure 5.26 and 5.27. The geometry of the Stanford bunny is imported from opensource resource as STL file. An STL file defines the surface by way of triangles. The STL file imported has thousands of triangles with the coordinate of the vertices. Once the STL file is imported, the outer surface is re-constructed and three-dimensional splines is generated. Once three dimensional spline is generated, the surface area is calculated and the total number of particles(N) is calculated accordingly. Then, N number of particles are inserted on the surface and set to reach equilibrium. Once the particles reach equilibrium, they are frozen and used as a boundary condition for three-dimensional case. For three-dimensional case, a similar process is followed. First, the total number of particles is calculated and then inserted inside the system. Then the particles are set to reach equilibrium. For the boundary, a bounce back/ghost particle scheme is followed. Final obtained particles go through VT-DT triangulation and a mesh is obtained. The final mesh is shown in Figures 5.26 and 5.27.

5.4 Discussion

5.4.1 Comments on Numerical Schemes

In addition, different numerical schemes have their advantages and disadvantages. In this work, a linear force profile has been used to reduce the computational cost. Furthermore, instead of using an explicit numerical scheme, implicit numerical schemes is also computationally efficient. However, as the number of particles increases, the computational cost saved is lesser in implicit than explicit numerical schemes. Since the goal of the present work is not accuracy but speed, there are several ways it can be achieved and a balance needs to be obtained. The overall interest is in reaching the final particle position, which can also be obtained by simply using an inverse of the matrix algorithms (such as LU decomposition) and finding the final particle positions. It is important to note here that matrix inversion is not easy and can be even more expensive than explicit numerical schemes for the higher number of particles. Thus depending on the number of particles, different schemes can be more optimum than others.

Summary

In this chapter, a novel, computationally efficient method to generate isotropic unstructured mesh using particles has been proposed. First, the boundary of the domain is defined and particles at the boundary are fixed. Next, the target mass and number density profile for the particles in the domain is calculated. The total number of particles is calculated using the target mass profile, dimension, and density of the domain. The target mass profile behaves as a controllable variable that enables control of the mesh including refinement. The three methods are described to solve for the position of the particle. The unsteady method is robust, fast, and is solved implicitly. A second steady method is described that uses the advantages of a good initial condition and is more efficient than the first implicit approach. A third hybrid method is explained that can switch between the unsteady and steady methods dynamically.

A Voronoi tessellation is generated using the final position of the particles. The VT-DT duality is used to generate the triangular mesh. Several benchmark cases are presented in the paper to show the capability of the method. The final mesh quality that is obtained from Delaunay Triangulation is of very good quality and with very little to no skewness.

Chapter 6

Conclusions and Recommendations

In this chapter, a conclusion of the dissertation will be presented in the first section. In the second section, a recommendation for future work will also be described.

6.1 Conclusion

The particle methods are very powerful and robust. For the flow simulation, particle methods such as multi-body Dissipative Particle Dynamics(MDPD) are capable of simulating complex fluid and liquid-vapor interface in mesoscale. However, simulating fluid in mesoscale is computationally expensive as the solid wall particles are taken into consideration even when their dynamics are not the focus. Furthermore, the local density of the fluid near the solid-liquid interface experience oscillation. Local density fluctuations are not physical as the fluid is considered to be incompressible. The reason behind the density fluctuation is also the solid wall and the solid particle density. When trying to solve the density fluctuations, the solid particle density needs to increase, which makes the first problem even worse i.e. computational time is even higher. The local density change in particle also needs to be smooth and not abrupt. So, the increasing particle density is not enough and it needs to be normalized to liquid/fluid particle density. In this work, an analytical-numerical formulation is derived to model the wall boundary in the context of MDPD. The derived wall model is capable of reproducing the same effect as a traditional solid wall made of particles. The advantages of using the proposed wall model are three folds and are as follows

- The wall model is computationally cheaper than the traditional particle wall. Several static and dynamic cases are used to demonstrate the efficiency of the method. In some cases, the method can save as much as 90% of the computational time.
- The density fluctuation near the solid-liquid interface is reduced to less than 0.1% of the traditional particle wall. The reduction of the density fluctuation is done without increasing the particle density(which would further increase the computational cost in a traditional particle wall).
- The wall model is normalized to the fluid density. This eliminates the discrepancies of *correct solid wall particle density* in the literature. Without the normalization, the A_{sl} and contact angle relationship is different in different literatures which physically should not be possible. With normalization, there is only one value of contact angle for a certain A_{sl} Furthermore, the model is also validated for high curvature surface and example of a solid sphere within a droplet is demonstrated.

In this work, several static cases of solid-fluid interaction are chosen to demonstrate the capability of the method. In chapter 4, a dynamic case is chosen to demonstrate the wall model robustness. In this case, a solid sphere is transported inside the droplet on a surface due to a wettability gradient. The solid sphere and solid wall are modeled using the wall boundary model. This example also demonstrates the wall boundary model's capability in handling high curvature surfaces. Furthermore, the effect of solid sphere radius, wettability gradient, and droplet radius is also studied.

Even when the numerical simulations are performed in continuum methods, the particle methods can be used to make them more efficient. In all structural dynamics(and fluid dynamics) simulations, mesh generation is a critical step in terms of the quality of the mesh and mesh generation time. In this dissertation, a novel particle method for isotropic unstructured mesh generation is developed. In structural dynamics, the speed of mesh generation is very important as there might be a need to remesh the system if the initially generated mesh is failing(mesh distortion) with high-speed dynamics(deformation). The mesh generation becomes even more challenging when the domain is very complex(e.g. curved interface). Thus both the quality and the speed of mesh generation are important. In this work, both are solved efficiently. In the current literature, most of the mesh generation methods are based on either Delaunay triangulation, Octree method, or Advancing front methods and all of those have some drawbacks. The proposed particle method is capable of generating two-dimensional and three dimensional isotropic unstructured mesh with high quality.

Particles in the domain have a varying cutoff radii and densities and interact with each other based on a given potential. The cutoff radius and density depend on the desired refined profile. The desired refinement profile is a surface function(for two dimensions) or a volume function(for three dimensions). The total number of particles in the system depends on the desired refinement Next, the particles are filled in the system, and the system is left to reach equilibrium. The time to reach equilibrium depends on the system volume, initial condition, force potential, and numerical scheme. A Voronoi-Tessellation-Delaunay-Triangulation is also developed for two dimensions and three dimensions. VT-DT is used to create a mesh from the particle position. The obtained mesh consists of triangular elements (two-dimensional) and tetrahedral elements (three-dimensions). The method has the following contributions.

- The particle method is capable of generating an isotropic unstructured mesh for any shape or geometry.
- The mesh has full independence on the desired refinement. It is possible to have very refined mesh in some sub-region of the geometry and bigger mesh in others.
- Depending on the complexity of the domain, different numerical schemes are proposed to prioritize speed or stability.

For the mesh, several two-dimensional and three-dimensional benchmark cases are used to demonstrate the capability of the proposed method. Several examples from the non-engineering fields are also demonstrated. The speed of the mesh generation method is fast and thus can be used in places where remeshing is required.

For two-dimensional cases, the final mesh obtains is triangular while tetrahedral shape grids are generated for three-dimensional cases. The mesh quality is also examined and demonstrated by checking the overall skewness of the grids and the skewness of the most skewed grid. Although in the current work, the particle method also has a few limitations such as only isotropic unstructured mesh can be generated. Oftentimes, it might be required to generate structured mesh or anisotropic unstructured mesh. The work, however, is a good beginning for mesh generation using the particle approach and future recommendations will be described in the next section.

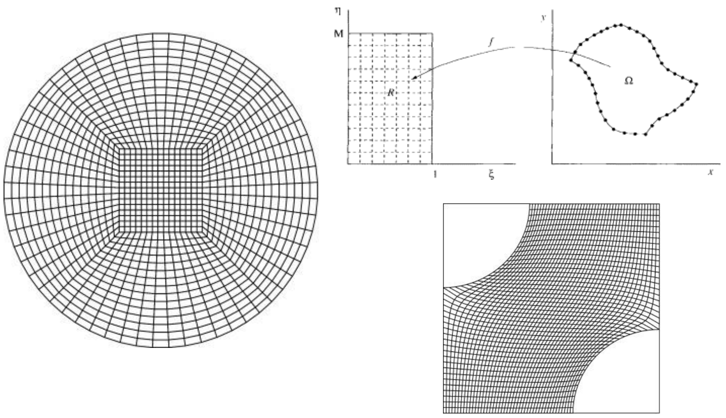


Figure 6.1: An example of generating a structured mesh for a non-rectangular geometry [4][5]

6.2 Future Recommendation

The wall boundary method for MDPD derived in this work eliminates the density fluctuation near the liquid-solid interface and also reduces the computational cost with traditional particle wall. The model is accurate for the wall with a radius of curvature more than three times the cutoff radius of the particle. The wall boundary model for the surface with a lower radius of curvature can be done using a similar analytical-numerical approach. In the present work, the wall boundary model has been used for solid sphere transport using a liquid droplet on a surface with a wettability gradient. The model can also be used for other mesoscale complex fluid cases such as coalescence-induced droplet umping, droplet bouncing from a hydrophobic surface and surface energy-induced droplet jumping. The wall boundary model can also be extended to other versions of DPD such as Energy Dissipative Particle Dynamics and Smoothed Dissipative Particle Dynamics. Furthermore, the wall boundary model is not specific to DPD, it can also be applied to other particle methods such as SPH. That will enable other particle methods to reduce computational time significantly and get rid of any density fluctuations near the wall if any. The mesh generation model presented in this work is capable of generating isotropic unstructured mesh. The current model can further be improved by adding the capability of generating anisotropic unstructured mesh. Anisotropic unstructured mesh can decrease the total number of grids by a large amount. If the gradient of physical properties is high in only one direction, then the desired number of grids is high only in that direction. This modification can be achieved by adding a weighting factor to the axes. In addition, a structured mesh can also be generated using this model. Structured meshes are simple and more efficient. A structured mesh also requires significantly less memory on the computer, which results in the lower computational cost for the simulation. Since structured mesh has all neighbor defined well, accessing the property of the neighboring cell is much easier(e.g. can be stored in an array and neighboring cell can be accessed by $i + 1, i - 1$). Although generating structured mesh is very challenging but normalizing or mapping the domain to a simpler domain and then generating the mesh(see Figure 6.1). Although mapping the domain to a simpler (e.g. rectangular domain) is a simple process but it does not work for all geometries. An efficient method of generating structured mesh that can be useful for several applications and make simulations more efficient.

Bibliography

[1] M.A. Yerry, M.S. Shephard, International Journal for Numerical Methods in Engineering **20**(11), 1965 (1984). DOI 10.1002/nme.1620201103

[2] S. Rebay, Journal of computational physics **106**(1), 125 (1993)

[3] L. Fu, L. Han, X.Y. Hu, N.A. Adams, Computer Methods in Applied Mechanics and Engineering **350**(15), 396 (2019). DOI 10.1016/j.cma.2018.10.052

[4] J.R. Sack, J. Urrutia, *Handbook of computational geometry* (Elsevier, 1999)

[5] J. Tu, G.H. Yeoh, C. Liu, Computational fluid dynamics. Oxford, UK: Elsevier pp. 125–54 (2018)

[6] G. Kresse, J. Hafner, Physical Review B **47**(1), 558 (1993)

[7] P. Warren, Physical Review E **68**(6), 066702 (2003)

[8] G.R. Liu, M.B. Liu, *Smoothed particle hydrodynamics: a meshfree particle method* (World scientific, 2003)

[9] S. Li, W.K. Liu, Appl. Mech. Rev. **55**(1), 1 (2002)

[10] P. Koumoutsakos, Annu. Rev. Fluid Mech. **37**, 457 (2005)

[11] P. Hoogerbrugge, J. Koelman, EPL (Europhysics Letters) **19**(3), 155 (1992)

[12] T. Liu, Automatic higher order mesh generation and movement utilizing spring-field and vector-adding. Ph.D. thesis, University of Tennessee at Chattanooga (2017)

[13] J.J. Monaghan, Annual review of astronomy and astrophysics **30**(1), 543 (1992). DOI 10.1146/annurev.aa.30.090192.002551

[14] C. Chen, L. Zhuang, X. Li, J. Dong, J. Lu, Langmuir **28**(2), 1330 (2012)

[15] M. Ahmadlouydarab, A.A. Hemeda, Y. Ma, Langmuir **34**(1), 198 (2018)

[16] R.D. Groot, P.B. Warren, The Journal of chemical physics **107**(11), 4423 (1997)

[17] C. Lan, S. Pal, Z. Li, Y. Ma, Langmuir **31**(35), 9636 (2015)

[18] Z. Li, X. Bian, Y.H. Tang, G.E. Karniadakis, Journal of Computational Physics **355**, 534 (2018)

[19] C. Chen, C. Gao, L. Zhuang, X. Li, P. Wu, J. Dong, J. Lu, Langmuir **26**(12), 9533 (2010)

[20] Z. Li, Y.H. Tang, H. Lei, B. Caswell, G.E. Karniadakis, Journal of Computational Physics **265**, 113 (2014)

[21] A. Mishra, A. Hemeda, M. Torabi, J. Palko, S. Goyal, D. Li, Y. Ma, Journal of Computational Physics **396**, 416 (2019)

[22] A. Hemeda, S. Pal, A. Mishra, M. Torabi, M. Ahmadlouydarab, Z. Li, J. Palko, Y. Ma, Langmuir (2019)

[23] A. Boromand, S. Jamali, J.M. Maia, Computer Physics Communications **196**, 149 (2015)

[24] P. Espanol, P. Warren, EPL (Europhysics Letters) **30**(4), 191 (1995)

[25] P.B. Warren, Physical Review Letters **87**(22), 225702 (2001)

[26] M. Arienti, W. Pan, X. Li, G. Karniadakis, The Journal of chemical physics **134**(20), 204114 (2011)

[27] M. Revenga, I. Zuniga, P. Espanol, Computer physics communications **121**, 309 (1999)

[28] S.K. Ranjith, B. Patnaik, S. Vedantam, Journal of Computational Physics **232**(1), 174 (2013)

[29] D. Visser, H. Hoefsloot, P. Iedema, Journal of computational Physics **205**(2), 626 (2005)

[30] A. Heinecke, W. Eckhardt, M. Horsch, H.J. Bungartz, *Supercomputing for Molecular Dynamics Simulations: Handling Multi-Trillion Particles in Nanofluidics* (Springer, 2015)

[31] Z. Li, G.H. Hu, Z.L. Wang, Y.B. Ma, Z.W. Zhou, Physics of Fluids **25**(7), 072103 (2013)

[32] D. Bonn, J. Eggers, J. Indekeu, J. Meunier, E. Rolley, Reviews of modern physics **81**(2), 739 (2009)

[33] A. Marmur, Journal of Imaging Science and Technology **44**(5), 406 (2000)

[34] R.N. Wenzel, The Journal of Physical Chemistry **53**(9), 1466 (1949)

[35] H. Kusumaatmaja, J. Yeomans, Langmuir **23**(11), 6019 (2007)

[36] M. Amrei, M. Davoudi, G. Chase, H.V. Tafreshi, Separation and Purification Technology **180**, 107 (2017)

[37] D. Venkateshan, H.V. Tafreshi, Colloids and Surfaces A: Physicochemical and Engineering Aspects **538**, 310 (2018)

[38] M. Abolghasemibizaki, R.L. McMasters, R. Mohammadi, Journal of colloid and interface science **521**, 17 (2018)

[39] M. Abolghasemibizaki, C.J. Robertson, C.P. Fergusson, R.L. McMasters, R. Mohammadi, Physics of Fluids **30**(2), 023105 (2018)

[40] G. Hou, J. Wang, A. Layton, Communications in Computational Physics **12**(2), 337 (2012)

[41] P. Meakin, Z. Xu, Progress in Computational Fluid Dynamics, an International Journal **9**(6-7), 399 (2009)

[42] A.V. Lukyanov, A.E. Likhtman, ACS nano **10**(6), 6045 (2016)

[43] S. Do Hong, M.Y. Ha, S. Balachandar, Journal of colloid and interface science **339**(1), 187 (2009)

[44] D. Sergi, G. Scocchi, A. Ortona, Fluid phase equilibria **332**, 173 (2012)

[45] N. Ojaghlou, H.V. Tafreshi, D. Bratko, A. Luzar, Soft matter **14**(44), 8924 (2018)

[46] S. Plimpton, Journal of computational physics **117**(1), 1 (1995)

[47] T. Arias, M. Payne, J. Joannopoulos, Physical Review B **45**(4), 1538 (1992)

[48] T. Kinjo, S. Hyodo, Molecular Simulation **33**(4-5), 417 (2007)

[49] N. Ghorbani, A. Pishevar, Computational Particle Mechanics **5**(1), 113 (2018)

[50] C.C. Chang, Y.J. Sheng, H.K. Tsao, Physical Review E **94**(4), 042807 (2016)

[51] J. Zhao, S. Chen, N. Phan-Thien, Molecular Simulation **44**(3), 213 (2018)

[52] P.B. Paramonov, S.F. Lyuksyutov, The Journal of chemical physics **123**(8), 084705 (2005)

[53] A. Hemeda, R. Esteves, J. McLeskey Jr, M. Gad-el Hak, M. Khraisheh, H.V. Tafreshi, International Communications in Heat and Mass Transfer **98**, 304 (2018)

[54] I. Pagonabarraga, D. Frenkel, The Journal of Chemical Physics **115**(11), 5015 (2001)

[55] I.V. Pivkin, G.E. Karniadakis, Physical review letters **96**(20), 206001 (2006)

[56] H. Dobbs, International Journal of Modern Physics B **13**(27), 3255 (1999)

[57] A. Marmur, Langmuir **19**(20), 8343 (2003)

[58] Y. Yeh, H. Cummins, Applied Physics Letters **4**(10), 176 (1964)

[59] H.S. Hele-Shaw, Nature **58**(1509), 520 (1898)

[60] A. Jameson, in *10th Computational fluid dynamics conference* (1991), p. 1596

[61] S. Litvinov, X. Hu, N. Adams, Journal of Physics: Condensed Matter **23**(18), 184118 (2011)

[62] N. Moumen, R.S. Subramanian, J.B. McLaughlin, Langmuir **22**(6), 2682 (2006)

[63] H. Kinoshita, S. Kaneda, T. Fujii, M. Oshima, Lab on a Chip **7**(3), 338 (2007)

[64] H.W. Lu, F. Bottausci, J.D. Fowler, A.L. Bertozzi, C. Meinhart, et al., Lab on a Chip **8**(3), 456 (2008)

[65] S. Ma, J.M. Sherwood, W.T. Huck, S. Balabani, Lab on a Chip **14**(18), 3611 (2014)

[66] R.J. Adrian, et al., Annual review of fluid mechanics **23**(1), 261 (1991)

[67] S. Fu, P.H. Biwole, C. Mathis, Building and Environment **87**, 34 (2015)

[68] C.D. Meinhart, S.T. Wereley, J.G. Santiago, Experiments in fluids **27**(5), 414 (1999)

[69] H. Maas, A. Gruen, D. Papantoniou, Experiments in fluids **15**(2), 133 (1993)

[70] H.K. Versteeg, W. Malalasekera, *An introduction to computational fluid dynamics: the finite volume method* (Pearson education, 2007)

[71] P. Jenny, S. Lee, H.A. Tchelepi, Journal of computational physics **187**(1), 47 (2003)

[72] J. Anderson, *Computational Fluid Dynamics*. Computational Fluid Dynamics: The Basics with Applications (McGraw-Hill Education, 1995)

[73] J.M.O. De Zarate, J.V. Sengers, *Hydrodynamic fluctuations in fluids and fluid mixtures* (Elsevier, 2006)

[74] R.A. Gingold, J.J. Monaghan, Monthly notices of the royal astronomical society **181**(3), 375 (1977)

[75] L.B. Lucy, The astronomical journal **82**, 1013 (1977)

[76] A. Mishra, Y. Ma, A. Gopinath, in *APS Division of Fluid Dynamics Meeting Abstracts* (2019), pp. L21–001

[77] T. Liu, A. Mishra, M. Torabi, A.A. Hemeda, J. Palko, Y. Ma, in *ASTFE Digital Library* (Begel House Inc., 2019)

[78] M. Torabi, A.A. Hemeda, A. Mishra, T. Liu, Y. Ma, in *ASTFE Digital Library* (Begel House Inc., 2019)

[79] M. Torabi, A.A. Hemeda, J.W. Palko, Y. Feng, Y. Cao, Y. Ma, Physical Review E **100**(3), 033102 (2019)

[80] M. Ahmadlouydarab, C. Lan, A.K. Das, Y. Ma, Physical Review E **94**(3), 033112 (2016)

[81] A. Ghoufi, P. Malfreyt, Physical Review E **83**(5), 051601 (2011)

[82] K.L. Johnson, K. Kendall, a. Roberts, Proceedings of the royal society of London. A. mathematical and physical sciences **324**(1558), 301 (1971)

[83] R.M. Füchslin, H. Fellermann, A. Eriksson, H.J. Ziock, The Journal of chemical physics **130**(21), 214102 (2009)

[84] A. Ghoufi, P. Malfreyt, Journal of Chemical Theory and Computation **8**(3), 787 (2012)

[85] J. Irving, J.G. Kirkwood, The Journal of chemical physics **18**(6), 817 (1950)

[86] J. Jones, J. áNoel Ruddock, N. Spenley, et al., Faraday Discussions **112**, 129 (1999)

[87] J. Backer, C. Lowe, H. Hoefsloot, P. Iedema, The Journal of chemical physics **122**(15), 154503 (2005)

[88] D. Visser, H.C. Hoefsloot, P.D. Iedema, Journal of computational Physics **214**(2), 491 (2006)

[89] J.C. Chai, H.S. Lee, S.V. Patankar, Journal of Thermophysics and Heat Transfer **8**(3), 419 (1994). DOI 10.2514/3.559

[90] J. Kim, D. Kim, H. Choi, Journal of Computational Physics **171**(1), 132 (2001). DOI 10.1006/jcph.2001.6778

[91] F. Moukalled, L. Mangani, M. Darwish, et al., *The finite volume method in computational fluid dynamics* (Springer, 2016)

[92] J.F. Thompson, Computing Systems in Engineering **3**(1-4), 393 (1992). DOI 10.1016/0956-0521(92)90124-2

[93] P.J. Frey, H. Borouchaki, P.L. George, Computer methods in applied mechanics and engineering **157**(1-2), 115 (1998). DOI 10.1016/S0045-7825(97) 00222-3

[94] T.J. Baker, Finite Elements in Analysis and Design **25**(3-4), 243 (1997). DOI 10.1016/S0168-874X(96)00032-7

[95] D. Bommes, B. Levy, N. Pietroni, E. Puppo, C. Silva, M. Tarini, D. Zorin, Computer Graphics Forum **32**(6), 51 (2013). DOI 10.1111/cgf.12014

[96] M. Fortunato, P.O. Persson, Journal of Computational Physics **307**, 1 (2016). DOI 10.1016/j.jcp.2015.11.020

[97] C.B. Marsh, R.J. Spiteri, J.W. Pomeroy, H.S. Wheater, Computers & Geosciences **119**, 49 (2018). DOI 10.1016/j.cageo.2018.06.009

[98] J. Guo, F. Ding, X. Jia, D.M. Yan, Computer-Aided Design **109**, 49 (2019). DOI 10.1016/j.cad.2018.12.005

[99] B. Roget, J. Sitaraman, V. Lakshminarayan, A. Wissink, Computers & Fluids **200**, 104429 (2020). DOI 10.1016/j.compfluid.2020.104429

[100] Qianqian Fang, D.A. Boas, in *2009 IEEE International Symposium on Biomedical Imaging: From Nano to Macro* (2009), pp. 1142–1145. DOI 10.1109/ISBI.2009.5193259

[101] E.A. Sadek, International Journal for Numerical Methods in Engineering **15**(12), 1813 (1980). DOI 10.1002/nme.1620151206

[102] T.D. Blacker, M.B. Stephenson, International journal for numerical methods in engineering **32**(4), 811 (1991). DOI 10.1002/nme.1620320410

[103] D.L. Marcum, N.P. Weatherill, AIAA Journal **33**(9), 1619 (1995). DOI 10.2514/3.12701

[104] R. Löhner, Engineering with computers **12**(3-4), 186 (1996). DOI 10.1007/BF01198734

[105] C. Frederick, Y. Wong, F. Edge, International Journal for Numerical Methods in Engineering **2**(1), 133 (1970). DOI 10.1002/nme.1620020112

[106] K.J. Roberts, W.J. Pringle, J.J. Westerink, Geoscientific Model Development **12**(5), 1837 (2019). DOI 10.5194/gmd-12-1847-2019

[107] C. Yoon, Journal of the Earthquake Engineering Society of Korea **23**(1), 83 (2019). DOI 10.9798/KOSHAM.2012.12.6.151

[108] D.T. Lee, A.K. Lin, Discrete & Computational Geometry **1**(3), 201 (1986). DOI 10.1007/BF02187695

[109] Q. Du, V. Faber, M. Gunzburger, SIAM review **41**(4), 637 (1999). DOI 10.1137/S0036144599352836

[110] R. Schneiders, International Journal of Computational Geometry & Applications **10**(04), 383 (2000). DOI 10.1142/S021819590000022X

[111] M.S. Shephard, M.K. Georges, International Journal for Numerical methods in engineering **32**(4), 709 (1991). DOI 10.1002/nme.1620320406

[112] L. Fu, S. Litvinov, X.Y. Hu, N.A. Adams, Journal of Computational Physics **341**, 447 (2017). DOI 10.1016/j.jcp.2016.11.016

[113] L. Fu, X.Y. Hu, N.A. Adams, Communications in Computational Physics, doi **10** (2020). DOI 10.4208/cicp.OA-2019-0049

[114] Z. Ji, L. Fu, X. Hu, N. Adams, Computer Methods in Applied Mechanics and Engineering **363**, 112881 (2020). DOI 10.1016/j.cma.2020.112881

[115] A.P. Witkin, P.S. Heckbert, in *Proceedings of the 21st annual conference on Computer graphics and interactive techniques* (1994), pp. 269–277. DOI 10.1145/1198555.1198656

[116] M.D. Meyer, P. Georgel, R.T. Whitaker, in *International Conference on Shape Modeling and Applications 2005 (SMI'05)* (IEEE, 2005), pp. 124–133. DOI 10.1109/SMI.2005.41

[117] J.R. Bronson, J.A. Levine, R.T. Whitaker, in *Proceedings of the 19th International Meshing Roundtable* (Springer Berlin Heidelberg, Berlin, Heidelberg, 2010), pp. 279–296. DOI 10.1007/978-3-642-15414-0-17

[118] F.J. Blom, International Journal for Numerical Methods in Fluids **32**(6), 647 (2000). DOI 10.1002/(SICI)1097-0363(20000330)32:6⟨647::AID-FLD979⟩3.0.CO;2-K

[119] D. Zeng, C.R. Ethier, Finite Elements in Analysis and Design **41**(11-12), 1118 (2005). DOI 10.1016/j.finel.2005.01.003

[120] T. Liu, L. Wang, S.L. Karman, B. Hilbert, in *54th AIAA Aerospace Sciences Meeting* (2016). DOI 10.2514/6.2016-1673

[121] S. Lo, International journal for numerical methods in engineering **21**(8), 1403 (1985). DOI 10.1016/0045-7949(94)90434-0

[122] J. Peraire, M. Vahdati, K. Morgan, O.C. Zienkiewicz, Journal of computational physics **72**(2), 449 (1987). DOI 10.1016/0021-9991(87)90093-3

[123] Y. Ito, A.M. Shih, A.K. Erukala, B.K. Soni, A. Chernikov, N.P. Chrisochoides, K. Nakahashi, Mathematics and Computers in Simulation **75**(5-6), 200 (2007). DOI 10.1016/j.matcom.2006.12.008

[124] N.P. Weatherill, Computers & Mathematics with Applications **24**(5-6), 129 (1992). DOI 10.1016/0898-1221(92)90045-J

[125] H. Si, International Journal for Numerical Methods in Engineering **75**(7), 856 (2008). DOI 10.1002/nme.2318

[126] M. Bern, D. Eppstein, J. Gilbert, Journal of computer and system sciences **48**(3), 384 (1994). DOI 10.1016/S0022-0000(05)80059-5

[127] Y. Ito, A.M. Shih, B.K. Soni, International Journal for Numerical Methods in Engineering **77**(13), 1809 (2009). DOI 10.1002/nme.2470

[128] L. Marechal, in *Proceedings of the 18th international meshing roundtable* (Springer, 2009), pp. 65–84. DOI 10.1007/978-3-642-04319-2-5

[129] T. Tu, D.R. O'hallaron, J.C. Lopez, Engineering with Computers **20**(2), 117 (2004). DOI 10.1007/s00366-004-0283-5

[130] P.G. Szabo, in *Proceedings of the International Conference on Computer Systems and Technologies and Workshop for PhD Students in Computing* (2009), pp. 1–6. DOI 10.1145/1731740.1731804

[131] P.J. Frey, H. Borouchaki, International journal for numerical methods in engineering **45**(1), 101 (1999). DOI 10.1002/(SICI)1097-0207(19990510)45:1⟨101::AID-NME582⟩3.0.CO;2-4

[132] P. Pebay, T. Baker, Mathematics of computation **72**(244), 1817 (2003). DOI 10.1090/S0025-5718-03-01485-6

[133] H. Nguyen, J. Burkardt, M. Gunzburger, L. Ju, Y. Saka, Computational geometry **42**(1), 1 (2009). DOI 10.1016/j.comgeo.2008.04.002

[134] J. Sarrate, J. Palau, A. Huerta, Communications in numerical methods in engineering **19**(7), 551 (2003). DOI 10.1002/cnm.585