University of Alberta

Direct Numerical Simulation of Particulate Flows

By

Veeramani Chidambaranathan ©

A thesis submitted to the Faculty of Graduate Studies and Research
in partial fulfillment of the requirements for the degree of
Doctor of Philosophy

in

Chemical Engineering

Department of Chemical and Materials Engineering

Edmonton, Alberta
Fall 2007

Library and
Archives Canada

Published Heritage
Branch

395 Wellington Street
Ottawa ON K1A 0N4
Canada

Bibliothèque et
Archives Canada

Direction du
Patrimoine de l'édition

395, rue Wellington
Ottawa ON K1A 0N4
Canada

Your file Votre référence
ISBN: 978-0-494-32939-9
Our file Notre référence
ISBN: 978-0-494-32939-9

Abstract

Over the last decade, the fictitious domain approach used along with the Lagrange multiplier technique has become very popular for direct numerical simulation of particulate flow problems. In the present work, a new formulation based on the fictitious domain approach but without the need to use Lagrange multipliers is proposed. The traditionally used fractional step method is followed to split Navier-Stokes equation on the basis of operators and explicit equations relating the interaction force between solid (particle) and fluid phases is derived. From these equations implicit expressions for particle and fluid velocities are derived as function of current and previous time step velocities. The most straightforward way to resolve them is via an iterative method but a direct extrapolation scheme is also possible. The latter approach makes the overall numerical scheme much cheaper. Validation of the new formulation is performed for single particle motion of fundamental importance like settling of a particle in a closed box and migration of a neutrally buoyant particle in Poiseuille flow in a circular channel. For multiparticle interaction, a new collision model based on the Stereomechanical impact theory and lubrication theory is developed and empirical relationship between coefficient of restitution e and Stokes number St established by experimental data available in literature is used. Validation of collision for bouncing particle and application to other physical situations like interaction with slant wall, drafting, kissing and tumbling interaction of two settling particles, interaction of settling and neutrally buoyant particles and sedimentation of multiple particles is explored. Parallel implementation of the code with moderate speed-up achieved on both shared memory and distributed memory architecture is presented. Finally, the application of fictitious domain approach to progressing cavity pumps to simulate the fluid flow in a double helical shaped stator due to a single helical rotor is presented.

Acknowledgement

I wish to thank my supervisors Dr. K. Nandakumar and Dr. P. Minev for their ideas, advice and encouragement. I am grateful to them for the freedom they gave me to explore different ideas. I wish to thank Dr. Fraser and Shirley Russell for the teaching fellowship which helped me develop my teaching capabilities and thanks to Dr. K. Nandakumar for being my teaching mentor.

I wish to thank all my colleagues in the lab for all the helpful discussions and comments. Special thanks to Dr. Yarub Al-Jahmany, Srinath and the workshop technicians in our department who helped with experiments. I am also grateful to Dr. Carolina Diaz-Goano and Dr. Tong Chen, for sharing their knowledge regarding various aspects of the algorithm that they had worked on. I would like to acknowledge the support of our department system administrator Bob Barton for his help with department computing facilities and patiently answering all my queries. Help from Martin Siegert, SFU and PETSc team, ANL is also much appreciated with regards to cluster computing. I thank my friends at home away from home for their advice and support.

Finally, I wish to thank my parents, sister and brother-in-law, my grandparents and all my family members for their patience and love. Special thanks to my little niece Sreenitha for bringing cheers and happiness during the uncertain final years of my studies.

Contents

List of Tables

List of Figures

List of Symbols

a, b, c	general co-ordinates	m
A_c	Area of cross-section	m^2
C	relative compression velocity	$m.s^{-1}$
$\mathbf{D}$	Fluid rate of strain tensor	s^{-1}
dh	Gap step size	m
d_p	particle diameter	m
D_r	Rotor diameter	m
dt	Time step size	s
e	coefficient of restitution	
ϵ	Eccentricity of the rotor	m
$\mathbf{e}_f$	unit vector in the direction of flow	
$\mathbf{e}_g$	unit vector in the direction of gravity	
$\mathbf{F}$	Force	$kg.m.s^{-2}$
$\mathbf{f}$	Right hand side vector	
f	coefficient of friction	
$\mathbf{F}_i^c$	Force due to collision	$kg.m.s^{-2}$
$\mathfrak{F}$	Interaction force	$kg.m.s^{-2}$
$\mathbf{F}_L$	Sub-grid lubrication force	$kg.m.s^{-2}$

LIST OF SYMBOLS

$\mathbf{F}_P$	Sub-grid particle lubrication force	$kg.m.s^{-2}$
Fr	Froude number	
$\mathbf{F}_W$	Sub-grid wall lubrication force	$kg.m.s^{-2}$
$\mathbf{g}$	Acceleration due to gravity	$m.s^{-2}$
G	center of gravity	m
$\mathbb{H}$	Hilbert space	
h	Gap between particle surface and boundary	m
h_0	Sub-grid length usually equal to grid spacing	m
I	Moment of Inertia	$kg.m^2$
i	Counter	
k	Number of substeps	
K_1	Shape correction factor	
$\mathbb{L}$	Gradient matrix	
$\mathbb{L}^T$	Transpose of gradient matrix	
$\mathbb{M}$	Mass matrix	
M	mass of the particle	kg
m^*	reduced mass $m^* = \left(\frac{1}{m_1} + \frac{1}{m_2}\right)^{-1}$	kg
$\mathbf{n}$	unit normal	
N	Rotational speed of the rotor	rpm
N_p	Total number of pressure nodes in the whole domain	
n_p	Number of pressure nodes	
N_v	Total number of velocity nodes in the whole domain	
n_v	Number of velocity nodes	
P	Dynamic pressure	$kg.m^{-1}.s^{-1}$

LIST OF SYMBOLS

p	Total pressure	$kg.m^{-1}.s^{-2}$
P_1	Space of polynomials of first order	
$\mathbb{P}_2$	Space of polynomials of second order	
P_N	Number of particles	
P	Particle domain	
P_r	Rotor pitch length	m
P_s	Stator pitch length	m
$\hat{P}$	Trial pressure field	$kg.m^{-1}.s^{-1}$
Q	Volumetric flow rate	$m^3.s^{-1}$
Q	image point	
q	weighting function for pressure	
Q_t	Theoretical volumetric flow rate	$m^3.s^{-1}$
$\mathbf{R}$	Space of real functions	
$\mathbf{r}$	Particle's position vector	m
R	Particle Radius	m
Re	Particle Reynolds number	
R^*	reduced radius $R^* = \left(\frac{1}{R_1} + \frac{1}{R_2}\right)^{-1}$	m
R_r	Rotor radius	m
$\mathbb{S}$	Stiffness matrix	
S	Surface area	m^2
S	relative sliding velocity	$m.s^{-1}$
s	substep time	s
$\mathbb{S}_p$	Pressure stiffness matrix	
St	Particle Stokes number	

LIST OF SYMBOLS

$\mathbf{T}$	Torque	$kg.m^2.s^{-2}$
t	Time	s
T_i	Contact time	s
$\mathbf{u}$	Fluid velocity field	$m.s^{-1}$
$\mathbf{U}$	Particle velocity	$m.s^{-1}$
$\mathbf{u}_2$	Fluid velocity in particle domain	$m.s^{-1}$
$\mathbf{u}_\Gamma$	Fluid velocity at the boundary	$m.s^{-1}$
u^*	Intermediate velocity	$m.s^{-1}$
U_∞	Free stokes settling velocity	$m.s^{-1}$
u^{**}	Divergence free intermediate velocity	$m.s^{-1}$
$U_\perp$	Velocity perpendicular to the wall	$m.s^{-1}$
$\hat{\mathbf{u}}$	Trial velocity field	$m.s^{-1}$
$\tilde{\mathbf{u}}$	Convected velocity field	$m.s^{-1}$
V	Volume of the particle	m^3
v	weighting function for velocity	
$\mathbf{V}$	Weighting function for particle translational velocity	$m.s^{-1}$
$\mathbf{v}$	Weighting function for fluid velocity	$m.s^{-1}$
v_c	Compression velocity	$m.s^{-1}$
$\mathbb{W}$	Space of weighting functions	
$\mathbf{w}$	weighting functions	
$w_{i,j,k}$	Gaussian weights in 3D tetrahedral geometry	
W_a	Particle velocity after collision	$m.s^{-1}$
W_b	Particle velocity before collision	$m.s^{-1}$
$\mathbf{X}$	Center of mass of the particle	m

LIST OF SYMBOLS

x	Position of any general point	m
$\mathbf{X}_w$	position of closest point on wall	m

Greek symbols

α	friction angle	rad
$\boldsymbol{\delta}$	Kronecker tensor	
Δ	finite changes	
δ	prefix for correction terms	
∂	Boundary of the domain	
ϕ	Basis functions for velocity	
Γ	Boundary of fluid domain	
γ	critical angle in collision model	rad
$\boldsymbol{\Lambda}$	space of Lagrange multipliers	
$\boldsymbol{\lambda}$	Lagrange multipliers	
μ	Dynamic viscosity	$kg.m^{-1}.s^{-1}$
$\boldsymbol{\mu}$	function belonging to Lagrange multiplier space $\boldsymbol{\Lambda}$	
$\boldsymbol{\nabla}$	Gradient Operator	m^{-1}
Ω	Computational domain	
ψ	Basis functions for pressure	
ρ	Density	$kg.m^{-3}$
$\boldsymbol{\sigma}$	Fluid stress tensor	$kg.m^{-1}.s^{-2}$
τ	time constants	s^{-1}
θ_s	sliding direction	rad
$\boldsymbol{\omega}$	Particle angular velocity	$rad.s^{-1}$
ξ, η, ζ	Local co-ordinates	

LIST OF SYMBOLS

ξ	weighting function for particle angular velocity	$rad.s^{-1}$

Subscripts

0	Initial conditions
1	Fluid domain
1	first particle
2	Solid domain
2	second particle
a	approach phase
c	characteristic scales
i	$i-$th particle domain
in	inlet
out	outlet
r	restitution phase

Superscripts

n	n-th time level
$'$	Dimensionless quantities

Chapter 1

Introduction

1.1 Fluid-particle systems

Multiphase fluid dynamics is a fast developing and already a vastly diversified area of research. Any chemical engineering unit operation involves multiphase dynamics between two or more phases. In our research, we focus on dynamics between solid and liquid phases and hence processes involving solid-liquid interactions are of special interest to us. Some examples of such processes are sedimentation, fluidization and particulate conveying. The principles involved in each of these processes are exploited in various unit operations e.g. sedimentation principle would apply to clarifiers, thickeners, rake settlers, hydro-cyclones, etc. Differences in these processes arise due to presence of different forces acting on the two phases which have physically different properties. In sedimentation, the force is gravity acting on phases of different densities. In fluidization, the force of gravity is offset by motion of fluid resulting in dispersion of particles in fluid rather than separation by settling. Particulate conveying is an example in which not only particles get convected with flow along the axial direction but they also migrate in radial direction under the action of flow field as observed by Segré and Silberberg [48]. This opens a new perspective that motion of particles is not only due to external forces but also due to hydrodynamic forces. This leads us to wonder whether such secondary motion of particles exist in other processes like sedimentation or fluidization too? Questions like this have justified re-investigation of simple processes like sedimentation which were once thought to be well understood. Our research will focus around sedimentation and issues related to sedimentation.

Before we go any further we need to clarify definition of a particle as it applies to our research, thereby define the domain of our modeling effort. Depending on the concentration and size distribution of the particulate phase we might infringe upon colloidal suspensions at one extreme and free settling of particles at the other. While a colloidal suspension would require us to account for surface forces on particle leading to Brownian motion of particles and electro-kinetic effects, free settling particles can be handled accurately by considering only hydrodynamic forces between particles and fluid. In order to avoid complications due to colloidal behavior, we will restrict our particle size to be greater than 10 μm [46] and upper limit is arbitrarily set to 10 cm.

It is useful to briefly define the physical parameters characterizing the fluid-particle system. There are a variety of dimensionless parameters that can be used. In this thesis, we use the Reynolds Number ($\text{Re} = \frac{U_c L_c}{\nu}$), where the characteristic velocity U_c and characteristic length L_c are problem dependent and ν is the kinematic viscosity of the fluid medium. In the majority of the thesis, we are concerned with the spherical particle motion in the fluid and hence take the diameter of the particle as the characteristic length scale and terminal settling velocity of the particle as the characteristic velocity scale. In the final chapter on progressing cavity pumps the diameter of the rotor is taken as the characteristic length scale and the inverse of speed of the rotation (in rad/s) defines the characteristic time scale. While Reynolds number is used to relate the inertial forces in fluid to viscous forces, we use Froude number ($\text{Fr} = \frac{U_c^2}{gL_c}$) to characterize the ratio of inertial forces to gravity. The density ratio of the fluid to particle is the third important parameter.

If two particles get very close to each other or the particle is near the wall of a container, a new spatial scale appears which is orders of magnitude different from the particle scale. This scale cannot be explicitly resolved in numerical discretization and requires additional modelling efforts. We use lubrication theory and stereomechanical impact theory for this purpose. We introduce the Stokes number that characterizes the ratio of the particle inertia to the fluid viscous forces ($\text{St} = \frac{1}{9}\frac{\rho_2}{\rho_1}\text{Re}$).

1.2 Literature Review

Although the free settling particle has been exhaustively researched, it has very limited practical applications. Nonetheless, it is an ideal starting point for our modeling approach due to the extensive literature available. On the other hand, investigation of multiple particles in fluids has been mostly experimental. Only recent growth of computing power coupled with ingenious numerical methods has spawned numerous tools to analyze these problems on computers. In our study we are developing one such tool to understand the physics of multi-particle systems.

1.2.1 Single particle scale

The emphasis at this scale is to capture in fine detail the phenomena that take place at the same length scale as the particle diameter or radius. Studies at the particle scale can be divided into two categories viz. particle-fluid interaction and particle-particle-fluid interaction. Various configurations have been studied like flow around a fixed particle or freely moving particle in fluid which itself may be at rest or moving. Similarly for particle-particle interaction, one particle may be fixed or both particles may be free to move in a fluid at rest or in motion. Of course, the motion of the particle is intimately coupled to the motion of fluid; they both influence each other. We will briefly cover studies done on each of these scenarios, within the limits of our assumptions.

Fixed particle: Flow around a fixed sphere has been studied at various flow regimes defined by different Reynolds number (Re) of flow. The stages of flow separation ($\mathrm{Re} \sim 20$), axis-symmetric wake formation and growth of the vortex rings in these wakes (up to $\mathrm{Re} \sim 210$), breaking of symmetry and double thread wake formation (up to $\mathrm{Re} \sim 270$), shedding of vortices in the form of interlocking hairpins (after $\mathrm{Re} \sim 300$) has been extensively studied experimentally and numerically [29, 58]. Ploumhans et al. [44] using vorticity-stream function formulation of Navier-Stokes equations have demonstrated the capability of numerical methods to capture iso-surfaces of hairpin vortices at $\mathrm{Re} \sim 1000$. Presently, one of the goals of direct numerical simulations is to extend this limit further by accurate description of vortex structures and be able to locate them precisely.

Settling particle: Various experimental methods have been employed to determine the trajectory of a settling particle in an otherwise quiescent fluid. Mordant and Pinton [37] employed an acoustic technique using ultrasonic waves to locate the particle and Doppler shift in frequency of the reflected waves due to motion of the particle to determine the settling velocity. Their main focus was the initial acceleration phase of settling particle starting from rest. ten Cate et al. [51] used particle imaging velocimetry (PIV) measurements to capture the settling velocity of the particle as well as the velocity field of the surrounding fluid. They have reported comprehensive results in all three stages of particle settling in a closed box viz. acceleration, steady settling and deceleration near bottom wall. We have used their results to validate our numerical code for predicting settling velocity of particles at various Reynolds numbers.

Simulation of single settling particle is used for code validation by many researchers. Usually, the particle is started from rest at the central axis of the container which may have a square, rectangular or circular cross-section. An interesting twist to this usual method is to start the particle off-centered as done by Feng et al. [15] for 2D circular particle and by Yu et al. [58] for 3D spherical particle. At low-Re, a circular particle approaches the central axis with an overshoot while a spherical particle monotonically approaches the axis. At high-Re, a circular particle oscillates about the axis while a spherical particle goes into a helical motion close to the wall. Broadly speaking they behave in the same way except that a spherical particle has three dimensional freedom of motion.

Two particle interactions: The drafting-kissing-tumbling (DKT) phenomenon of wake interaction between two settling particles has been experimentally demonstrated by Fortes et al. [16]. Two dimensional dynamic simulations of two particles settling in a channel have shown formation of stable staggered configuration with damped oscillations of the circular particles at low-Re [15] and DKT phenomenon at higher Re for circular particles [25, 15] and spherical particles [13]. Aidun and Ding [3] have investigated settling of cylinders and shown the existence of Hopf bifurcation, period doubling and chaotic dynamics at progressively higher Re and dimensionless external force (basically gravity).

The particle surface plays an important role in interaction between particles and particle and wall. Two models viz. stick/rotate model and roll/slip model have been proposed by Davis [9] based on the lubrication theory of mobility functions [28] which describe the motion of two particles approaching or receding from each other. In stick/rotate model, the particles touch each other at one contact point and then rotate about this point while they also translate. In roll/slip model, one particle rolls on the surface of the other and may slip if loading force exceeds friction between the surfaces. Zeng et al. [59] have found that roll/slip model describe experimentally observed interactions better. Ekiel-Jezewska et al. [14] have analyzed roll/slip or without slip model using four parameters viz. static friction coefficient; kinetic friction coefficient; wall effect parameter and gap between interacting particles and arrived at good fits for their experimental results. The particle-wall interaction has been addressed by many researchers [6, 4, 34, 20, 21]. Stokes number (St) which captures relative importance of particle inertia to fluid viscous forces, has been found to be the deciding parameter in interaction of a particle with walls. Joseph et al. [34], using a pendulum method and Gondret et al. [21], using a bouncing sphere method agreed on a critical St $\sim$ 10 in order for the particle to rebound from the wall.

Numerically, particle-particle and particle-wall interactions are simulated using a parametric repulsive force model [17, 18, 55]. Johnson and Tezduyar [30] implemented collision between particles using the coefficient of restitution, as typically done for solid-body collisions in air (fluid lubrication neglected). Diaz-Goano et al. [13] have implemented a collision strategy based on displacement of particles along the line joining their centers whenever particles come within a predetermined distance. All these methods maintain a small safety zone between the surfaces of particles to avoid ambiguity due to overlapping domains. The collision method suggested by Singh et al. [49] allows for particles to overlap before they are separated. They track the node at which overlap occurs to remove it from list of unknowns for one of the particles involved in the collision. A common philosophy in studies involving large scale simulations is that a method geared towards simulation of macro-scale events need not spend too much computational effort on accurately capturing micro-scale events like collision between two particles as long as collective behavior of the multi-particle system is captured (accurately!!).

1.2.2 Multi-particle scale

Ample indications to the need for revision of above philosophy are provided by experiments on hydrodynamic dispersion of a settling particle in a suspension. A theoretical model for hydrodynamic dispersion was given by Davis and Hill [10] based on pair-wise interaction of particles, hence applicable only to dilute suspensions. In practice a settling particle may encounter an ensemble of particles or may settle through interstitial fluid undisturbed. To determine the vertical and horizontal dispersivities of velocity, experiments with different settling particles in a suspension of neutrally buoyant particles were carried out by Abbott et al. [1] and in a sedimenting suspension by Nicolai et al. [39, 40]. Their main conclusions are that horizontal dispersion on any time scale and vertical dispersion averaged over a long period of time is best modeled as a Fickian process i.e. they both increase linearly with time. However, vertical dispersion in short time steps is deterministic.

Abbott et al. [1] have derived empirical relationships between dispersivities and the size ratio of settling particle to suspended particle and solid volume fraction of the suspension. It would be interesting to simulate these experiments with a direct numerical simulation method to verify the empirical results.

Experimental investigation of multiple particles can only give us qualitative information; clearer insight into these processes can be gained through numerical methods. Glowinski et al. [17, 18] showed settling of 504 and 6400 circular particles, respectively, in a 2D cell. These simulations have successfully captured large scale swirl formations in gravity driven flows which are reminiscent of Rayleigh-Taylor instability. Pan et al. [43] have also simulated fluidization of 1024 spheres in 2D domain. Wan and Turek [55] using a fictitious boundary method have simulated settling of 790 circular particles of various sizes and 3600 circular particles of uniform size in a 2D cell. Johnson and Tezduyar [31, 32] have been able to simulate 100 and 1000 spherical particles, respectively using their space-time finite element method and massively parallel computing architecture.

1.3 Numerical Methods

Early numerical simulations of solid-fluid flows were based on continuum theory which treats solid and fluid as interpenetrating mixture [60]. The conservation equations are derived by ensemble averaging of mass or momentum either for each phase separately or for entire mixture. The interaction terms between the phases are modeled separately for closure. Hence the interaction terms do not necessarily reflect the exact nature of interactions between solid and fluid phases. The next age of numerical methods employed physically based equations like the Navier-Stokes and continuity equations for the fluid and Newton's second law of motion to model rigid particles. But, the interaction between solids and fluid was still achieved using empirical forms of hydrodynamic forces. The most recent methods developed during the last decade are called direct numerical simulation (DNS) which directly calculate hydrodynamic interaction forces from the flow field, and the equations of motion for particle and fluid are fully coupled. Excellent reviews of various DNS methods are available in the literature viz. Lattice-Boltzmann Equation (LBE) based method [7], Volume of Fluid (VoF) method [47], Front tracking method [52], Level set method [41], Arbitrary Eulerian Lagrangian (ALE) method [26] and Distributed Lagrange Multiplier (DLM) method [18]. There are many other methods existing and many new ones are being developed based on the above mentioned methods. Except for the Lattice-Boltzmann method which ultimately derives from kinetic theory, other methods are based on Navier-Stokes equation. The volume of fluid, front tracking and level-set methods has been extensively applied for free surface and interfacial flow simulations. The last two of these DNS methods were developed by researchers working on a grand challenge project to develop what they call *"particle movers"*. As the name suggests, these methods are targeted towards solving particulate flow problems.

Both methods avoid explicit computation of hydrodynamic forces and moments between fluid and particles by using a combined momentum equation of the fluid-particle system. The core concept behind ALE methods is the ALE mesh movement technique. Since the particles are free to move, the fluid domain constantly changes because the particle-fluid boundary changes. To accurately capture the *"Lagrangian"* particle motion, the nodes of the mesh are *"arbitrarily"* distributed in the *"Eulerian"* fluid domain. This implies re-meshing is required every time step and solution from one time step has to be projected to the next time step. The DLM approach obviates re-meshing and projection by providing a uniform fixed mesh throughout the simulation by employing the concept of a *"fictitious domain"*. The particle domain is integrated into fluid domain, so that the fluid now fills the whole *"fictitious"* domain. The combined momentum equation is split to reflect this change. The fictitious domain is solved for fluid motion and then particle motion is imposed on the nodes, actually in particle domain, using Lagrange multipliers. As opposed to distributed Lagrange multipliers for each particle, Diaz-Goano et al. [13] use a single Lagrange multiplier defined for the whole domain. Hence, their method is appropriately called the Global Lagrange Multiplier (GLM) method.

1.4 Thesis Overview

In the next chapter (chapter 2), we present the formulation and the algorithm used to solve the Navier-Stokes equations. Chapter 3 is devoted to the derivation and a discussion of the pros and cons of previous methods and the new method proposed here. We first present the basic methodology involved in formulating the Lagrange multiplier fictitious domain approach as proposed by Glowinski et al. [17]. Then we look at an algorithm proposed by Diaz-Goano et al. [13] that further couples the unsteady effects like added mass and Basset forces into the interaction force between particles and fluid through a global Lagrange multiplier. Going one step further, an efficient discretization algorithm is devised that explicitly eliminates the Lagrange multiplier and therefore significantly reduces the overall operation count.

One of the major short-comings of the fictitious domain method is the loss of resolution in capturing the particle-fluid interface. In chapter 4, we develop a mechanism to avoid loosing accuracy by dynamic meshing of the particle domain. We look at two different approaches for this purpose: sub-division of the surface elements and body fitted surface elements. We will show effectiveness of these techniques by testing them on a sedimenting particle problem at low Reynolds number and analyzing the angular velocity data. It will be shown that body fitted meshing at the particle-fluid interface can better control the spurious oscillations that appear due to the non-symmetric shape of an ill resolved particle interface.

The entire code is then validated for stability and consistency by comparing two fundamental test cases of a single particle motion. In chapter 5, we present details of settling and migrating particle studies and their comparison to experimental and numerical data available in literature. We also compare the performance of the new formulation to the formulation given by Diaz-Goano et al. [13].

Once the single particle motion is validated, stage is set to test the cases of multiple particle systems. Two major problems have to be addressed in this case. The first one is the resolution of the interaction between particles when they are very close to each other. Second, the computational complexity increases dramatically with the number of particles and this in turn requires an efficient parallelization of the code. We address the issue of particle collision modelling in chapter 6. It is based on the lubrication theory which is applicable in the Stokes regime (Re $<<$ 1). It is well known that fluid inertia can be neglected in this regime. We also use the Stokes number (St) which relates the particle inertia to fluid viscosity. It has been confirmed in a number of experiments [34, 33, 35, 20, 21] that whenever St $<$ 10, the particle inertia can be neglected with respect to fluid viscosity which is quite analogous to the way fluid inertia is neglected with respect to fluid viscosity when Re $<<$ 1. However at the other extreme when particle inertia is important (i.e. at high Stokes number $\sim$ 100 or higher), a stereomechanical impact based collision model works best. The challenge is to bridge these theories with a numerical model for collision in the intermediate regime where the dynamics is a rich interplay of viscous forces and inertial forces. A new collision model based on stereomechanical impact and lubrication theories in concert with experimental data is proposed. A number of two particle interaction scenarios is presented and a validation of the bouncing motion of spherical particle is attempted. Finally, a multiple particle sedimentation case is considered.

In chapter 7, we present a novel application of the fictitious domain method to Progressing Cavity Pumps (PCP). This application illustrates that the algorithm can be extended to non-spherical particles. In this case the motion of the rotor is prescribed unlike the dynamic motion of spherical particles in previous cases. The geometry of the solid boundary that moves inside the fluid is essential for the algorithm to work as it is presented here. However, the present implementation of rotor motion in a stator geometry is very useful because a wealth of information about the torque requirement and hence power requirement to operate these pumps built from various combination of geometric parameters can be estimated.

Finally, we will conclude with major contributions embodied in this thesis and suggestions for future work.

Chapter 2

The Governing Equations and the Numerical Methods

2.1 Governing Equations

2.1.1 Computational Domain

The fluid domain is denoted by Ω_1, where subscript '1' is used for fluid phase and the entire particle domain is denoted by Ω_2, where subscript '2' is for solid phase. To specifically refer to the domain of i-th particle subscript 'i' is added. For example, $\Omega_{2,1}$ refers to domain of first particle. The entire domain of particle is $\Omega_2 = \bigcup_{i=1}^{P_N}$, where there are P_N particles. An all encompassing domain for fluid and particle is defined as $\Omega = \Omega_1 \bigcup \Omega_2$. The boundary of the fictitious domain is Γ

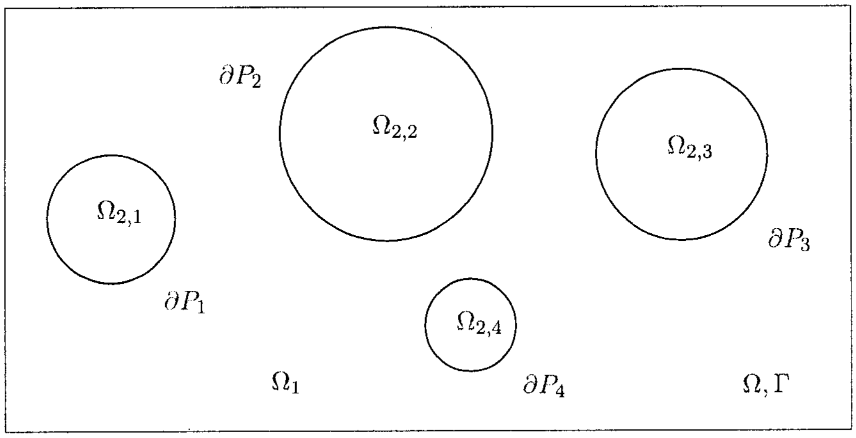


Figure 2.1: Illustration of computational domain

and the boundary of $i-$th particle is ∂P_i, therefore the effective total boundary for the fluid is $\Gamma \bigcup_{i=1}^{P_N} \partial P_i$. To simplify the notation, subscripts for fluid and particle is used only if both phases are present. Further, we will avoid the subscript '2' for particle when it is clear, for example we always refer to $i-$th particle only. No special distinction is made between dimensional and dimensionless quantities as these should be obvious from the context. An example domain is shown in figure 2.1.

2.1.2 Equations for fluid domain

The Navier-Stokes equation (2.1) is used to model incompressible fluid along with the incompressibility constraint obtained from equation of continuity (2.2).

$$\rho_1 \frac{\mathrm{D}\mathbf{u}_1}{\mathrm{D}t} = \rho_1 \mathbf{g} + \boldsymbol{\nabla} \cdot \boldsymbol{\sigma}_1 \tag{2.1}$$

$$\boldsymbol{\nabla} \cdot \mathbf{u}_1 = 0 \quad \text{in}\, \Omega_1 \tag{2.2}$$

where, $\mathbf{u}_1$ is the velocity field vector in fluid domain, ρ_1 is the density of the fluid, $\mathbf{g}$ is the acceleration due to gravity and $\boldsymbol{\sigma}_1$ is the stress tensor. To model the Newtonian behavior of the fluid the stress tensor is related to strain tensor through constant viscosity.

$$\boldsymbol{\sigma}_1 = -p_1 \boldsymbol{\delta} + 2\mu \mathbf{D}[\mathbf{u}_1] \tag{2.3}$$

$$\mathbf{D}[\mathbf{u}_1] = \frac{1}{2}[\boldsymbol{\nabla}\,\mathbf{u}_1 + (\boldsymbol{\nabla}\,\mathbf{u}_1)^T] \tag{2.4}$$

where, p_1 is the scalar pressure field in fluid domain, $\boldsymbol{\delta}$ is the Kronecker tensor, μ is the viscosity of fluid and $\mathbf{D}[\mathbf{u}_1]$ is the rate of strain tensor. To complete the problem specification the initial condition and the boundary conditions for fluid domain must be specified. The initial conditions must be specified as follows:

$$\mathbf{u}_1(t=0) = \mathbf{u}_0 \quad \text{in}\, \Omega_1; \qquad \boldsymbol{\nabla} \cdot \mathbf{u}_0 = 0 \tag{2.5}$$

where, $\mathbf{u}_0$ is the initial fluid velocity field which should be divergence free. The particles in the fluid present internal no slip boundary that are set as follows:

$$\mathbf{u}_1 = \mathbf{U}_i + \boldsymbol{\omega}_i \times \mathbf{r}_i \quad \text{on}\, \partial P_i(t), \quad i = 1, \cdots, P_N \tag{2.6}$$

where, $\mathbf{U}_i$, $\boldsymbol{\omega}_i$ and $\mathbf{r}_i = x - \mathbf{X}_i$ are the linear velocity, angular velocity and position of the $i-$th particle's surface element x with respect to its center $\mathbf{X}_i$ respectively. The boundary conditions on the external boundaries depends on the problem under consideration. In general the following boundary conditions are applicable is most situations. The terminology introduced here will be used throughout the thesis.

Inlet Boundary Condition

The inlet boundary condition is used in case of a channel flow problem. We will encounter a horizontal channel flow when considering particle migration problems and a vertical channel when considering two settling particles. For a circular channel of radius R, the Hagen-Poiseuille flow profile given by the expression:

$$\mathbf{u}_1 = \mathbf{u}_{max} \left[1 - \left(\frac{\mathbf{r}}{R}\right)^2\right] \mathbf{e}_f \tag{2.7}$$

is used. Here, $\mathbf{r}$ is the distance to any node on the inlet from the center of the inlet, $\mathbf{u}_{max}$ is the maximum channel velocity at the center, and $\mathbf{e}_f$ is the flow direction.

Outlet Boundary Condition

The outlet boundary condition is used for channel flow problem and open container problems where the surface is open to the atmosphere. This physical situation is usually modelled through the so-called stress-free boundary condition which specifies zero stress on the outlet (Γ). When projection schemes are used to discretize the Navier-Stokes equation (which is the case of this thesis), this condition is often approximated by:

$$p_1 = 0 \quad \text{on}\,\Gamma \tag{2.8}$$

$$\frac{\partial u_{1,1}}{\partial n} = \frac{\partial u_{1,2}}{\partial n} = \frac{\partial u_{1,3}}{\partial n} = 0 \quad \text{on}\,\Gamma \tag{2.9}$$

where, $u_{1,1}$, $u_{1,2}$ and $u_{1,3}$ are components of fluid velocity and n is the outward normal to the outlet.

Wall boundary condition

The wall boundary condition is of Dirichlet type and for solid impermeable wall used in closed or open containers, it is given by:

$$\mathbf{u}_1 = 0 \tag{2.10}$$

2.1.3 Equations for particle domain

The rigid body equations describing motion of particles are: Newton's second law of motion:

$$M_i \frac{\mathrm{D}\mathbf{U}_i}{\mathrm{D}t} = M_i \mathbf{g} + \mathbf{F}_i \tag{2.11}$$

Euler's rotation equation:

$$I_i \frac{\mathrm{D}\boldsymbol{\omega}_i}{\mathrm{D}t} + \boldsymbol{\omega}_i \times I_i \boldsymbol{\omega}_i = \mathbf{T}_i \tag{2.12}$$

where, M_i, I_i are the mass and moment of inertia; $\mathbf{U}_i$, $\boldsymbol{\omega}_i$ are the velocity and angular velocity of the $i-$th particle; $\mathbf{F}_i$, $\mathbf{T}_i$ are the hydrodynamic force and torque on the particle. The kinematic equations for linear and angular velocity of the $i-$th particle are:

$$\frac{\mathrm{D}\mathbf{X}_i}{\mathrm{D}t} = \mathbf{U}_i \tag{2.13}$$

$$\frac{\mathrm{D}\boldsymbol{\theta}_i}{\mathrm{D}t} = \boldsymbol{\omega}_i \tag{2.14}$$

where, $\mathbf{X}_i$ and $\boldsymbol{\theta}_i$ are the position and angular displacement of the particle. The hydrodynamic force and torque due to fluid on the particle can be obtained from fluid stress tensor as follows:

$$\mathbf{F}_i = -\int_{\partial P_i} \boldsymbol{\sigma}_1 \cdot \mathbf{n}_1 \mathrm{d}S \tag{2.15}$$

$$\mathbf{T}_i = -\int_{\partial P_i} \mathbf{r}_i \times (\boldsymbol{\sigma}_1 \cdot \mathbf{n}_1) \mathrm{d}S \tag{2.16}$$

Here, $\mathbf{n}_1$ is the normal to fluid that is at the fluid-particle interface it will be into the particle and since particle's normal will be into the fluid $\mathbf{n}_i = -\mathbf{n}_1$. Finally to complete the problem specification for particle equations, the initial conditions for the particle are as follows:

$$\mathbf{U}_i(t=0) = \mathbf{U}_{i,0} \tag{2.17}$$

$$\boldsymbol{\omega}_i(t=0) = \boldsymbol{\omega}_{i,0} \tag{2.18}$$

$$\mathbf{X}_i(t=0) = \mathbf{X}_{i,0} \tag{2.19}$$

$$\boldsymbol{\theta}_i(t=0) = \boldsymbol{\theta}_{i,0} \tag{2.20}$$

The ideal case of spherical rigid particles allows significant simplification of the particle kinematics equations. Due to spherical symmetry, the moment of inertia is isotropic i.e. it does not depend on the orientation, therefore the rotation equation (2.12) simplifies to:

$$I_i \frac{\mathrm{D}\boldsymbol{\omega}_i}{\mathrm{D}t} = \mathbf{T}_i \tag{2.21}$$

Since, we don't need angular displacement the equations (2.14) and (2.20) can be ignored. The angular velocity of the particle can be extracted from the curl of the fluid velocity in the particle domain as follows [see 12, chapter 4]:

$$\boldsymbol{\omega}_i = \frac{0.5}{V_i} \int_{\Omega_i(t)} (\boldsymbol{\nabla} \times \mathbf{u})\, \mathrm{d}\Omega_i \tag{2.22}$$

Note that here $\mathbf{u}$ is the global velocity field after fluid domain is extended into the particle domain and V_i is the volume of the i–th particle.

2.2 Numerical Methods

Central to the idea of the fictitious domain approach is the solution of the fluid equations without regard for any internal boundaries (in present case particle boundaries), on the entire computational domain Ω. The basic Navier-Stokes solver is therefore built first and validated. In this section, the solution method for Navier-Stokes solver is described as it applies to incompressible Newtonian fluid alone. The dynamics of particles will be included in subsequent chapters.

The Navier-Stokes equation (2.1) is a second order non-linear equation. The non-linearity due to convection term and second order due to diffusion term. In addition, the problem is complicated further due to pressure-velocity coupling which in the case of incompressible fluids requires that pressure changes propagate instantly through the entire domain maintaining divergence free velocity field. In real world of course there is nothing that is incompressible and pressure takes a finite time (no matter how small) to be felt at a distant location.

It is convenient while discretizing the equations in time to split the advection and diffusion terms. The operator splitting technique leads to the advection equation stated as an initial value problem and second order linear Stokes problem. The method of characteristics is used to solve the advection problem. The integration is performed pointwise along the characteristic line using the Euler explicit scheme. The Stokes problem is temporally resolved using second order backward difference scheme and spatially resolved using Galerkin Finite Element Method. The pressure-velocity coupling is addressed using projection method implementing pressure correction scheme. These methods are briefly recounted in this section for the sake of completeness. For details on selection of these methods see thesis work of Diaz-Goano [12], chapter 2. The Navier-Stokes equation in dimensionless form is written as:

$$\left(\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \boldsymbol{\nabla} \mathbf{u}\right) = -\boldsymbol{\nabla} P + \frac{1}{Re} \boldsymbol{\nabla}^2 \mathbf{u} \tag{2.23}$$

Note that the dynamic pressure P has been introduced by combining total pressure and gravity terms.

2.2.1 Method of Characteristics

The advection equation resulting from operator splitting technique applied on equation (2.23) can be written as:

$$\begin{aligned} \frac{\partial \tilde{\mathbf{u}}^{n-i}(s)}{\partial s} &= -(\tilde{\mathbf{u}}^{n-i}(s) \cdot \boldsymbol{\nabla})\tilde{\mathbf{u}}^{n-i}(s); \quad 0 \le s \le (i+1)\Delta t, \quad i = 0, \cdots, k \\ \tilde{\mathbf{u}}^{n-i}(0) &= \mathbf{u}^{n-i} \end{aligned} \tag{2.24}$$

Here, s is the sub-timestep within the method of characteristics. $k = 0$, for a first order scheme and $k = 1$, for a second order scheme. The objective is to get advected velocities $\tilde{\mathbf{u}}^{n-i}$ at time level $n+1$. The following steps are followed:

1. Estimate the advecting velocity by second order extrapolation of the velocity field.

$$\mathbf{u}^{n+1} = 2\mathbf{u}^{n} - \mathbf{u}^{n-1} \tag{2.25}$$

Of course, this is only possible in and after second time step. For first order advection equation, the initial condition specified for velocity field is used.

2. Determine the foot of the characteristic by solving the initial value problem.

$$\frac{\mathrm{d}\mathbf{X}_x^{n+1}}{\mathrm{d}t} = \mathbf{u}^{n+1}(\mathbf{X}_x^{n+1}(t), t) \quad \forall t \in [t^{n-i}, t^{n+1}] \tag{2.26}$$

where, $\mathbf{X}_x^{n+1}(t^{n+1}) = x$ means x is the position on the curve at time t^{n+1} (initial condition). By going back in time, we find the foot of the same characteristic curve at t^{n-i} as $\mathbf{X}_x^{n+1}(t^{n-i})$. This is achieved using Euler explicit scheme:

$$\mathbf{X}_x^{n+1}(t^{n-i}) = \mathbf{X}_x^{n+1}(t^{n+1}) - i\Delta t \mathbf{u}^{n+1} \tag{2.27}$$

3. Search the finite elements containing the foot $(\mathbf{X}_x^{n+1}(t^{n-i}))$. If not found sub-timestepping is used until the element is located.

4. Interpolate the velocity at the foot of the characteristic using finite element interpolation.

$$\tilde{\mathbf{u}}^{n-i}(x) = \mathbf{u}^{n-i}(\mathbf{X}_x^{n+1}(t^{n-i}), t^{n-i}) \tag{2.28}$$

Implementation of the search algorithm is discussed by Minev and Ethier [36].

2.2.2 Projection method

The advected velocities are employed in the second order backward difference and pressure at the previous time level P^n is used to resolve the Stokes part of the equation.

$$\left(\frac{3\mathbf{u}^* - 4\tilde{\mathbf{u}}^n + \tilde{\mathbf{u}}^{n-1}}{2\Delta t}\right) = -\boldsymbol{\nabla} P^n + \frac{1}{Re}\boldsymbol{\nabla}^2\mathbf{u}^* \tag{2.29}$$

This velocity field is then projected on a divergence free subspace via a pressure correction scheme [22] to obtain divergence free velocity field $\mathbf{u}^{**}$ and the corresponding pressure field P^{n+1}.

$$\begin{aligned}
\left(\frac{3\mathbf{u}^{**} - 4\tilde{\mathbf{u}}^n + \tilde{\mathbf{u}}^{n-1}}{2\Delta t}\right) &= -\boldsymbol{\nabla} P^{n+1} + \frac{1}{Re}\boldsymbol{\nabla}^2\mathbf{u}^{**} && (2.30)\\
\boldsymbol{\nabla}\cdot\mathbf{u}^{**} &= 0 \quad \text{in}\,\Omega && (2.31)\\
\mathbf{u}^{**}\cdot\mathbf{n}_1 &= 0 \quad \text{on}\,\Gamma && (2.32)
\end{aligned}$$

By subtracting the equations (2.29) and (2.30) and taking divergence we obtain the pressure correction (2.33) required to make $\mathbf{u}^*$ field divergence free.

$$\boldsymbol{\nabla}^2(\delta P) \approx \frac{3\boldsymbol{\nabla}\cdot\mathbf{u}^*}{2\Delta t} \tag{2.33}$$

The higher order derivatives have been neglected. The final divergence free velocity field $\mathbf{u}^{**}$ and pressure P^{n+1} are obtained as:

$$\begin{aligned}
\mathbf{u}^{**} &\approx \mathbf{u}^* - \frac{2\Delta t}{3}\boldsymbol{\nabla}(\delta P) && (2.34)\\
P^{n+1} &= P^n + \delta P && (2.35)
\end{aligned}$$

2.2.3 Galerkin Finite Element Method

The Galerkin Finite Element Method (GFEM) is a special application of the method of weighted residuals. GFEM uses the basis functions (also called shape functions or interpolation functions) for weighting functions (also called test functions) as well. A variety of basis functions and weighting functions have been employed to get variety of finite element methods. Further, one can choose to discretize the computational domain using a variety of finite elements. In the present case, the second order Lagrange polynomials are employed as basis functions and since it is a GFEM, also as weighting functions. Second order Taylor-Hood elements are employed to discretize the domain. A standard Taylor-Hood element in its local co-ordinate system is shown in the figure 2.2. In order to satisfy stability criterion specified by Ladyzhenskaya - Babuška - Brezzi(LBB) condition, the pressure must be approximated at least one order lower than the velocity. Therefore, the velocity field is approximated at all the ten nodes of the Taylor-Hood element

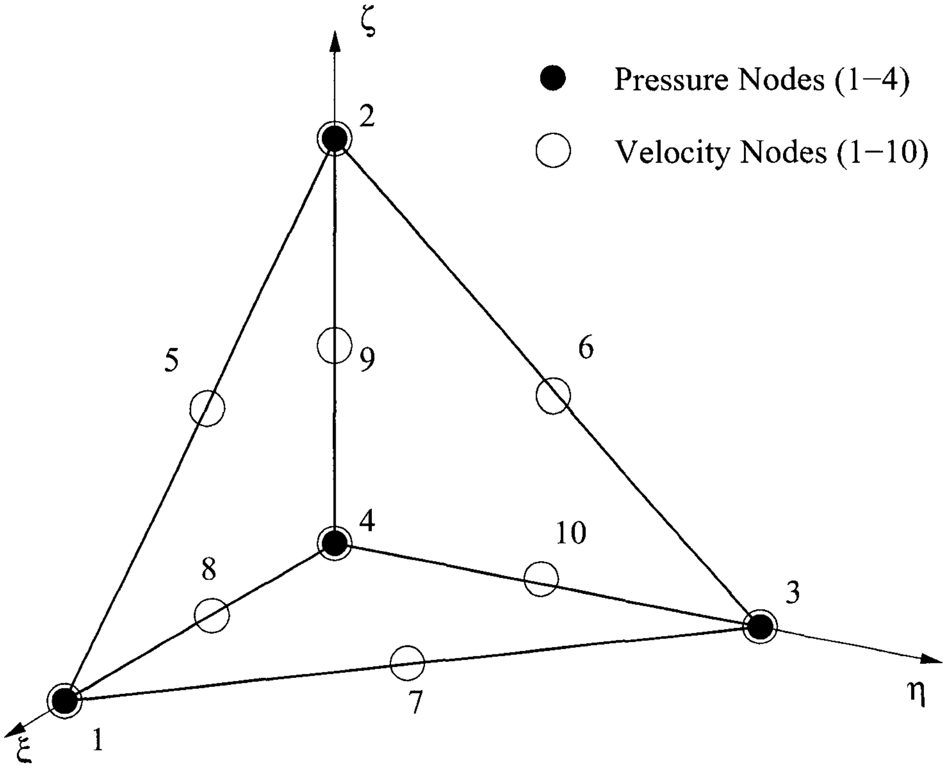


Figure 2.2: The standard Taylor-Hood element

and pressure is approximated at only four corner nodes. The velocity interpolation function is defined as:

$$\mathbf{u} \approx \hat{\mathbf{u}} = \sum_{i=1}^{n_v} \hat{\mathbf{u}}_i \phi_i \tag{2.36}$$

The hat indicates that velocity determined by numerical methods is only an approximation, different from exact solution to the partial differential equations. $n_v = 10$ is the number of velocity nodes per element and $\hat{\mathbf{u}}_i$ is the value of velocity at the $i-$th node. The basis functions for velocity at the n_v nodes are given by:

$$\begin{aligned}
\phi_1 &= \xi(2\xi - 1) \\
\phi_2 &= \zeta(2\zeta - 1) \\
\phi_3 &= \eta(2\eta - 1) \\
\phi_4 &= (1 - 2(\xi + \zeta + \eta))(1 - (\xi + \zeta + \eta)) \\
\phi_5 &= 4\xi\zeta \\
\phi_6 &= 4\zeta\eta \\
\phi_7 &= 4\xi\eta \\
\phi_8 &= 4\xi(1 - (\xi + \zeta + \eta)) \\
\phi_9 &= 4\zeta(1 - (\xi + \zeta + \eta)) \\
\phi_{10} &= 4\eta(1 - (\xi + \zeta + \eta))
\end{aligned} \tag{2.37}$$

The interpolation function for pressure is similarly defined as:

$$\hat{P} = \sum_{i=1}^{n_p} \hat{P}_i \psi_i \tag{2.38}$$

The basis functions for pressure at the $n_p = 4$ corner nodes of the standard element are given by:

$$\begin{aligned}
\psi_1 &= \xi \\
\psi_2 &= \zeta \\
\psi_3 &= \eta \\
\psi_4 &= (1 - (\xi + \zeta + \eta))
\end{aligned} \tag{2.39}$$

Note that the basis functions for velocity are second order polynomials ($\mathbb{P}_2$) and basis function for pressure are first order polynomials (P_1) and hence the finite element is also refered to as $\mathbb{P}_2 - P_1$ element. Both these polynomials are defined locally within an standard element and their derivatives are not continuous across elements. Therefore, they are said to belong to space of $\mathbb{C}^0$ functions. The application of GFEM to complete Navier-Stokes equation is illustrated in the thesis work of Diaz-Goano [12]. In the present case, the equations that need to be discretized in space are equation (2.29) for $\mathbf{u}^*$, equation (2.33) for δP and equation (2.34) for $\mathbf{u}^{**}$. The time co-efficients are defined as $\tau_0 = \frac{3}{2\Delta t}$, $\tau_1 = -\frac{2}{\Delta t}$ and $\tau_2 = \frac{1}{2\Delta t}$. In vectorial notation the equations look as follows:

$$\left(\tau_0 - \frac{1}{Re}\boldsymbol{\nabla}^2\right)\mathbf{u}^* = -\left(\tau_1\tilde{\mathbf{u}}^n + \tau_2\tilde{\mathbf{u}}^{n-1}\right) - \boldsymbol{\nabla}P^n \qquad \text{by (2.29)} \qquad (2.40)$$

$$\boldsymbol{\nabla}^2(\delta P) = \tau_0\boldsymbol{\nabla}\cdot\mathbf{u}^* \qquad \text{by (2.33)} \qquad (2.41)$$

$$\tau_0\left(\mathbf{u}^{**} - \mathbf{u}^*\right) = -\boldsymbol{\nabla}(\delta P) \qquad \text{by (2.34)} \qquad (2.42)$$

The first and third equations are in fact three equations each when written componentwise (in 3D problem), while second equation is just one. Therefore, there are in total seven equations to be solved for fluid alone. The procedure for converting these partial differential equations to algebraic equation is detailed below in terms of global co-ordinates. The integrations that are involved are performed in local co-ordinates using Gaussian quadratures. The steps involved in GFEM are illustrated for one component (x-component), but other components can be derived in similar manner.

1. The equation (2.40) for x-component can be written as:

$$\left(\tau_0 - \frac{1}{Re}\boldsymbol{\nabla}^2\right)u = -\left(\tau_1\tilde{u}^n + \tau_2\tilde{u}^{n-1}\right) - \frac{\partial P^n}{\partial x} \qquad (2.43)$$

2. Obtain the weak formulation using the method of weighted residuals:

$$\int_\Omega\left(\tau_0 u\cdot v - \frac{1}{Re}\boldsymbol{\nabla}^2 u\cdot v\right)\mathrm{d}\Omega = -\int_\Omega\left(\tau_1\tilde{u}^n + \tau_2\tilde{u}^{n-1}\right)\cdot v\mathrm{d}\Omega - \int_\Omega\frac{\partial P^n}{\partial x}\cdot v\mathrm{d}\Omega \qquad (2.44)$$

In general, the velocity test function $\mathbf{v} \in \mathbb{W}$ where the space of test functions is defined as $\mathbb{W}_0 = \{\mathbf{w}|\mathbf{w} \in \mathbb{H}^1(\Omega), \mathbf{w} = 0 \,\mathrm{on}\, \Gamma\}$. Applying Green's identity and using the boundary condition on weighting function we get:

$$\int_\Omega\left(\tau_0\left(u\cdot v\right) + \frac{1}{Re}\left(\boldsymbol{\nabla}u\cdot\boldsymbol{\nabla}v\right)\right)\mathrm{d}\Omega = -\int_\Omega\left(\tau_1\tilde{u}^n + \tau_2\tilde{u}^{n-1}\right)\cdot v\mathrm{d}\Omega - \int_\Omega\frac{\partial P^n}{\partial x}\cdot v\mathrm{d}\Omega \qquad (2.45)$$

Since, the requirement of smoothness from velocity is reduced by one order, the formulation is called weak formulation.

3. Substitute the trial function for velocity and take weighting functions the same as the basis functions. In doing this the method ensures that the residual, which is approximated from basis functions, is made orthogonal to every member of the functional space formed by the basis functions themselves. Since the only vector that can be orthogonal to the complete space defined by the basis functions is the null vector, the residual is forced to be zero.

$$\begin{aligned}&\int_\Omega \tau_0 \left(\sum_{j=1}^{N_v} \hat{u}_j \phi_j \cdot \phi_i\right) + \frac{1}{Re} \left(\boldsymbol{\nabla} \left(\sum_{j=1}^{N_v} \hat{u}_j \phi_j\right) \cdot \boldsymbol{\nabla} \phi_i\right) \mathrm{d}\Omega \\ &= - \int_\Omega \left(\tau_1 \widetilde{u}^n + \tau_2 \widetilde{u}^{n-1}\right) \cdot \phi_i \mathrm{d}\Omega - \int_\Omega \sum_{k=1}^{N_p} \hat{P}_k^n \frac{\partial \psi_k}{\partial x} \cdot \phi_i \mathrm{d}\Omega \\ &\forall \quad i = 1, \cdots, N_v\end{aligned} \tag{2.46}$$

$$\begin{aligned}&\sum_{j=1}^{N_v} \hat{u}_j \int_\Omega \tau_0 \left(\phi_j \cdot \phi_i\right) + \frac{1}{Re} \left(\boldsymbol{\nabla} \phi_j \cdot \boldsymbol{\nabla} \phi_i\right) \mathrm{d}\Omega \\ &= - \int_\Omega \left(\tau_1 \widetilde{u}^n + \tau_2 \widetilde{u}^{n-1}\right) \cdot \phi_i \mathrm{d}\Omega - \sum_{k=1}^{N_p} \hat{P}_k^n \int_\Omega \phi_i \cdot \frac{\partial \psi_k}{\partial x} \mathrm{d}\Omega \\ &\forall \quad i = 1, \cdots, N_v\end{aligned} \tag{2.47}$$

Here, N_v is the total number of velocity nodes in entire domain. In matrix notation,

$$\left(\tau_0 \mathbb{M}^x + \frac{1}{Re} \mathbb{S}^x\right) \{\hat{u}\} = \{f^x\} - \mathbb{L}^x \{P^n\} \tag{2.48}$$

$$\begin{aligned}M_{i,j}^x &= \int_\Omega \left(\phi_j \cdot \phi_i\right) \mathrm{d}\Omega \\ S_{i,j}^x &= \int_\Omega \left(\boldsymbol{\nabla} \phi_j \cdot \boldsymbol{\nabla} \phi_i\right) \mathrm{d}\Omega \\ L_{i,k}^x &= \int_\Omega \phi_i \cdot \frac{\partial \psi_k}{\partial x} \mathrm{d}\Omega \\ f_i^x &= - \int_\Omega \left(\tau_1 \widetilde{u}^n + \tau_2 \widetilde{u}^{n-1}\right) \cdot \phi_i \mathrm{d}\Omega\end{aligned}$$

Here, $\mathbb{M}$ is the mass matrix, $\mathbb{S}$ is the stiffness matrix and $\mathbb{L}$ is the gradient matrix. The known values, boundary conditions, source and sink terms are included in vector **f**. In the same manner algebraic equations can be obtained for other components.

The GFEM applied to equation (2.41) is as follows:

1. The method of weighted residuals gives:

$$\int_{\Omega} \nabla^2(\delta P)\cdot q \mathrm{d}\Omega = \tau_0 \int_{\Omega} (\nabla\cdot\mathbf{u}^*)\cdot q \mathrm{d}\Omega \tag{2.49}$$

   The pressure test function $q \in \mathbb{H}^1$.

2. Obtain the weak formulation by applying Green's identity:

$$\int_{\Omega} \nabla(\delta P)\cdot\nabla q \mathrm{d}\Omega = \tau_0 \int_{\Omega} \nabla q\cdot\mathbf{u}^* \mathrm{d}\Omega \tag{2.50}$$

3. Substitute the trial function for pressure and take the weighting functions the same as the basis functions.

$$\int_{\Omega} \nabla\left(\sum_{j=1}^{N_p}(\delta\hat{P}_j)\psi_j\right)\cdot\nabla\psi_k \mathrm{d}\Omega = \tau_0 \int_{\Omega} \nabla\psi_k\cdot\left(\sum_{i=1}^{N_v}\hat{\mathbf{u}}_i^*\phi_i\right)\mathrm{d}\Omega \qquad \forall \quad k = 1,\cdots,N_p \tag{2.51}$$

   Performing the dot product:

$$\sum_{j=1}^{N_p}(\delta\hat{P}_j)\int_{\Omega}\nabla\psi_j\cdot\nabla\psi_k \mathrm{d}\Omega = \tau_0\int_{\Omega}\sum_{d=1}^{3}\left(\frac{\partial\psi_k}{\partial x_d}\right)\left(\sum_{i=1}^{N_v}\hat{u}_{d,i}^*\phi_i\right)\mathrm{d}\Omega \qquad \forall \quad k = 1,\cdots,N_p \tag{2.52}$$

   Changing the order of summation and integration:

$$\sum_{j=1}^{N_p}(\delta\hat{P}_j)\int_{\Omega}\nabla\psi_j\cdot\nabla\psi_k \mathrm{d}\Omega = \tau_0\sum_{d=1}^{3}\left(\sum_{i=1}^{N_v}\left(\int_{\Omega}\frac{\partial\psi_k}{\partial x_d}\phi_i \mathrm{d}\Omega\right)\cdot\left(\hat{u}_{d,i}^*\right)\right) \qquad \forall \quad k = 1,\cdots,N_p \tag{2.53}$$

   In matrix notation,

$$\begin{aligned}\mathbb{S}_p\{\delta\hat{P}\} &= \tau_0\left((\mathbb{L}^x)^T\hat{u}_1^* + (\mathbb{L}^y)^T\hat{u}_2^* + (\mathbb{L}^z)^T\hat{u}_3^*\right)\\ &= \tau_0\mathbb{L}^T\mathbf{u}^* \\ (S_p)_{k,j} &= \int_{\Omega}\nabla\psi_j\cdot\nabla\psi_k \mathrm{d}\Omega\end{aligned} \tag{2.54}$$

   Here, $\mathbb{S}_p$ is the pressure stiffness matrix and transpose of the gradient matrix ($\mathbb{L}^T$) is the divergence matrix. $\hat{u}_1^*, \hat{u}_2^*$ and $\hat{u}_3^*$ are components of the $\hat{\mathbf{u}}^*$

The final equation to find correct velocity equation (2.42) after GFEM gives the algebraic systems (all three components condensed):

$$\tau_0\mathbb{M}\{\hat{\mathbf{u}}^{**} - \hat{\mathbf{u}}^*\} = -\mathbb{L}\{\delta\hat{P}\} \tag{2.55}$$

2.3 Summary

The final algorithm used to solve for the fluid domain is as follows:

1. Determine the convected velocities at time levels n and $n-1$ as $\tilde{\mathbf{u}}^n$ and $\tilde{\mathbf{u}}^{n-1}$.

2. Solve the Stokes part:

$$\left(\tau_0\mathbb{M} + \frac{1}{Re}\mathbb{S}\right)\hat{\mathbf{u}}^* = \mathbf{f} - \mathbb{L}\{\hat{P}^n\} \tag{2.56}$$

3. Solve for pressure correction:

$$\mathbb{S}_p\{\delta\hat{P}\} = \tau_0\mathbb{L}^T\hat{\mathbf{u}}^* \tag{2.57}$$

4. Solve for velocity correction:

$$\hat{\mathbf{u}}^{**} = -\frac{1}{\tau_0}\mathbb{M}^{-1}[\mathbb{L}\{\delta\hat{P}\}] + \hat{\mathbf{u}}^* \tag{2.58}$$

This is the final fluid velocity for time level $n+1$ and computations for next time level is continued with this velocity. The numerical integration of all the integrals are carried out using Gaussian Quadrature. The linear algebraic equations obtained are solved using conjugate gradient method. Two standard benchmark tests viz. backward facing step and shear driving cavity flow were carried out by Diaz-Goano [12] to validate the solver package. The implementation of particle domain is the topic of next chapter.

Chapter 3

Lagrange Multipliers: to use or not to use?

In this chapter, first the idea of using a fictitious domain and distributed Lagrange multipliers (DLM) approach to constrain the fluid to behave like a rigid body as proposed by Glowinski et al. [17] is briefly discussed. Then the global Lagrange multiplier (GLM) approach in a similar framework proposed by Diaz-Goano et al. [13] and a minor correction to the same algorithm [54] is discussed. Finally, a modification of the latter algorithm is proposed which elliminates the Lagrange multiplier for imposition of the rigid body motion.

3.1 Distributed Lagrange Multiplier

The formulation of Glowinski et al. [17] proceeds with the following basic three steps to solve fluid-particle system.

§**3.1.1** Derive the combined equation of motion for fluid and particle.

§**3.1.2** Extend the fluid domain to the particle domain.

§**3.1.3** Remove rigid body constraint in particle domain and impose it as side constraint using Lagrange multipliers.

3.1.1 Combined Equation of Motion

The purpose of this step is to eliminate the hydrodynamic force and torque between solid and liquid phases. By using the variational principle, the Navier-Stokes equation (2.1), the equation for translation (2.11) and rotation (2.21) for particle are combined as follows:

$$\begin{aligned}&\int_{\Omega_1}\left(\rho_1\frac{\mathrm{D}\mathbf{u}_1}{\mathrm{D}t}-\rho_1\mathbf{g}-\boldsymbol{\nabla}\cdot\boldsymbol{\sigma}_1\right)\cdot\mathbf{v}\mathrm{d}\Omega\\&+\left(M_i\left(\frac{\mathrm{D}\mathbf{U}_i}{\mathrm{D}t}-\mathbf{g}\right)-\mathbf{F}_i-\mathbf{F}_i^c\right)\cdot\mathbf{V}_i\ +\left(I_i\frac{\mathrm{D}\boldsymbol{\omega}_i}{\mathrm{D}t}-\mathbf{T}_i\right)\cdot\boldsymbol{\xi}_i=0\end{aligned}\tag{3.1}$$

Note that there is an additional external force term $\mathbf{F}_i^c$ added to equation (2.11) to account for the collision interaction force with other particles and boundaries. This term is further expanded in the chapter dedicated to collision modelling (chapter 6). The combined functional space of fluid and particle velocities is defined as:

$$\begin{aligned}\mathbb{V}_1(t)=\{&(\mathbf{u},\mathbf{U}_i,\boldsymbol{\omega}_i)|\mathbf{u}\in\mathbb{H}^1(\Omega_1)^3,\mathbf{U}_i\in\mathbf{R}^3,\boldsymbol{\omega}_i\in\mathbf{R}^3,\\&\mathbf{u}=\mathbf{U}_i+\boldsymbol{\omega}_i\times\mathbf{r}_i\,\text{on}\,\partial P_i(t),\,\text{and}\,\mathbf{u}=\mathbf{u}_\Gamma(t)\,\text{on}\,\Gamma\}\end{aligned}$$

And the space of combined weighting functions is defined as:

$$\begin{aligned}\mathbb{W}_1(t)=\{&(\mathbf{v},\mathbf{V},\boldsymbol{\xi})|\mathbf{v}\in\mathbb{H}^1(\Omega_1)^3,\mathbf{V}\in\mathbf{R}^3,\boldsymbol{\xi}\in\mathbf{R}^3,\\&\mathbf{v}=\mathbf{V}_i+\boldsymbol{\xi}_i\times\mathbf{r}_i\,\text{on}\,\partial P_i(t),\,\text{and}\,\mathbf{v}=0\,\text{on}\,\Gamma\}\end{aligned}$$

After some mathematical manipulation (See A.1) using vector and tensor identities and using the fact that $\mathbf{v}=\mathbf{V}_i+\boldsymbol{\xi}_i\times\mathbf{r}_i$ on $\partial P_i(t)$, the stress divergence term can be expanded to cancel the hydrodynamic force and torque terms. The final combined equation of motion can be written as:

$$\begin{aligned}&\int_{\Omega_1}\rho_1\left(\frac{\mathrm{D}\mathbf{u}_1}{\mathrm{D}t}-\mathbf{g}\right)\cdot\mathbf{v}\mathrm{d}\Omega+\left(M_i\left(\frac{\mathrm{D}\mathbf{U}_i}{\mathrm{D}t}-\mathbf{g}\right)-\mathbf{F}_i^c\right)\cdot\mathbf{V}_i\\&+\left(I_i\frac{\mathrm{D}\boldsymbol{\omega}_i}{\mathrm{D}t}\right)\cdot\boldsymbol{\xi}_i=-\int_{\Omega_1}\boldsymbol{\sigma}_1:\mathbf{D}[\mathbf{v}]\mathrm{d}\Omega\end{aligned}\tag{3.2}$$

Note that this equation is only applicable in the fluid domain (Ω_1).

3.1.2 Define the Fictitious Domain

In order to extend it into the entire computational domain (Ω), the following space of functions is defined.

$$\begin{aligned}\mathbb{V}(t)=\{&(\mathbf{u},\mathbf{U}_i,\boldsymbol{\omega}_i)|\mathbf{u}\in\mathbb{H}^1(\Omega)^3,\mathbf{U}_i\in\mathbf{R}^3,\boldsymbol{\omega}_i\in\mathbf{R}^3,\\&\mathbf{u}=\mathbf{U}_i+\boldsymbol{\omega}_i\times\mathbf{r}_i\,\text{in}\,\Omega_i(t),\,\text{and}\,\mathbf{u}=\mathbf{u}_\Gamma(t)\,\text{on}\,\Gamma\}\end{aligned}$$

And the space of combined weighting functions is defined as:

$$\mathbb{W}(t) = \{(\mathbf{v}, \mathbf{V}, \boldsymbol{\xi}) | \mathbf{v} \in \mathbb{H}^1(\Omega)^3, \mathbf{V} \in \mathbf{R}^3, \boldsymbol{\xi} \in \mathbf{R}^3,$$
$$\mathbf{v} = \mathbf{V}_i + \boldsymbol{\xi}_i \times \mathbf{r}_i \,\text{in}\, \Omega_i(t), \,\text{and}\, \mathbf{v} = 0 \,\text{on}\, \Gamma\}$$

According to the definition of this space, the fluid velocity trial function must not only satisfy no slip boundary condition on the surface of particle $\partial P_i(t)$ but must satisfy rigid body motion throughout particle domain $\Omega_i(t)$ at any time. Now, the additional constraint introduced i.e.

$$\mathbf{u}_2 = \mathbf{U}_i + \boldsymbol{\omega}_i \times \mathbf{r}_i \,\text{in}\, \Omega_i(t) \tag{3.3}$$

and similarly for weighting function, can be used to derive the particle equation in integral form (See A.2). The particle equation of motion is written in integral form as follows:

$$\begin{aligned} &\int_{\Omega_i} \rho_1 \left(\frac{\mathrm{D}\mathbf{u}_2}{\mathrm{D}t} - \mathbf{g}\right) \cdot \mathbf{v} \mathrm{d}\Omega - \frac{\rho_1}{\rho_i} \left(M_i \left(\frac{\mathrm{D}\mathbf{U}_i}{\mathrm{D}t} - \mathbf{g}\right)\right) \cdot \mathbf{V}_i \\ &- \frac{\rho_1}{\rho_i} \left(I_i \frac{\mathrm{D}\boldsymbol{\omega}_i}{\mathrm{D}t}\right) \cdot \boldsymbol{\xi}_i = - \int_{\Omega_i} \boldsymbol{\sigma}_2 : \mathbf{D}[\mathbf{v}] \mathrm{d}\Omega \end{aligned} \tag{3.4}$$

The fluid velocity inside the particle domain is denoted as $\mathbf{u}_2$ to distinguish it from fluid domain velocity $\mathbf{u}_1$. Once fluid is extended into particle domain we have a global fluid velocity $\mathbf{u}$. The similar notation is followed for fluid stresses. The combined weak formulation in the fictitious domain is thus:

$$\begin{aligned} &\int_{\Omega} \rho_1 \left(\frac{\mathrm{D}\mathbf{u}}{\mathrm{D}t} - \mathbf{g}\right) \cdot \mathbf{v} \mathrm{d}\Omega + \sum_{i=1}^{P_N} \Bigg[\left(1 - \frac{\rho_1}{\rho_i}\right) \Bigg(M_i \left(\frac{\mathrm{D}\mathbf{U}_i}{\mathrm{D}t} - \mathbf{g}\right) \cdot \mathbf{V}_i + \\ &I_i \left(\frac{\mathrm{D}\boldsymbol{\omega}_i}{\mathrm{D}t}\right) \cdot \boldsymbol{\xi}_i \Bigg) - \mathbf{F}_i^c \cdot \mathbf{V}_i \Bigg] = - \int_{\Omega} \boldsymbol{\sigma} : \mathbf{D}[\mathbf{v}] \mathrm{d}\Omega \end{aligned} \tag{3.5}$$

3.1.3 Introduce Lagrange Multipliers

The final step relaxes the rigid body constraint from the fictitious domain and imposes it separately as a side constraint using Lagrange multipliers. In order to do this, the following space of functions is defined:

$$\widetilde{\mathbb{V}}(t) = \{\mathbf{u} | \mathbf{u} \in \mathbb{H}^1(\Omega)^3, \mathbf{u} = \mathbf{u}_\Gamma(t) \,\text{on}\, \Gamma\}$$

And corresponding space of combined weighting functions is defined as:

$$\widetilde{\mathbb{W}}(t) = \{\mathbf{v} | \mathbf{v} \in \mathbb{H}^1(\Omega)^3, \mathbf{v} = 0 \,\text{on}\, \Gamma\}$$

It is clear from the functional space that the motive is to get rid of the particles from the entire computational domain so that the fluid can be solved in the entire domain without concern for particles. But of course eventually particle equations must be added and it is added to equation (3.5) as follows:

$$\begin{aligned}&\int_\Omega \rho_1 \left(\frac{\mathrm{D}\mathbf{u}}{\mathrm{D}t} - \mathbf{g}\right) \cdot \mathbf{v} \mathrm{d}\Omega + \sum_{i=1}^{P_N} \left[\left(1 - \frac{\rho_1}{\rho_i}\right) \left(M_i \left(\frac{\mathrm{D}\mathbf{U}_i}{\mathrm{D}t} - \mathbf{g}\right) \cdot \mathbf{V}_i \right.\right. \\ &\left.\left. + I_i \left(\frac{\mathrm{D}\boldsymbol{\omega}_i}{\mathrm{D}t}\right) \cdot \boldsymbol{\xi}_i \right) - \mathbf{F}_i^c \cdot \mathbf{V}_i \right] = - \int_\Omega \boldsymbol{\sigma} : \mathbf{D}[\mathbf{v}] \mathrm{d}\Omega \qquad (3.6) \\ &+ \sum_{i=1}^{P_N} \langle \boldsymbol{\lambda}_i, \mathbf{v} - (\mathbf{V}_i + \boldsymbol{\xi}_i \times \mathbf{r}_i) \rangle_{\Omega_i(t)}\end{aligned}$$

$$\langle \boldsymbol{\mu}_i, \mathbf{u} - (\mathbf{U}_i + \boldsymbol{\omega}_i \times \mathbf{r}_i) \rangle_{\Omega_i(t)} = 0 \quad \forall \boldsymbol{\mu}_i \in \boldsymbol{\Lambda}_i(t),\ i = 1, \cdots, P_N \qquad (3.7)$$

The variational form of the Lagrange multiplier term added to the right hand side suggests that the Lagrange multiplier is like a velocity correction term and this correction to the global fluid velocity must go to zero or in other words the global velocity should match the particle velocity exactly inside the particle domain. Glowinski et al. [17] suggested another interpretation for Lagrange multipliers as an additional body force required to maintain the rigid body motion in $\Omega_i(t)$. It is this interpretation that Diaz-Goano et al. [13] capitalized on. As suggested in [17], if the space of Lagrange multipliers $\boldsymbol{\Lambda}_i(t)$ is taken as $\mathbb{H}^1(\Omega_i)^3$ then the inner product could be written as:

$$\langle \boldsymbol{\mu}_i, \mathbf{v} \rangle_{\Omega_i(t)} = \int_{\Omega_i(t)} \left(\boldsymbol{\mu}_i \cdot \mathbf{v} + \boldsymbol{\nabla} \boldsymbol{\mu}_i \cdot \boldsymbol{\nabla} \mathbf{v} \right) \mathrm{d}\Omega_i \qquad (3.8)$$

Using this definition, it easy to see that the strong form of the fluid equation leading to equation (3.6) should have been:

$$\rho_1 \left(\frac{\mathrm{D}\mathbf{u}}{\mathrm{D}t} - \mathbf{g}\right) = \boldsymbol{\nabla} \cdot \boldsymbol{\sigma} + \left(\boldsymbol{\lambda}_i - \boldsymbol{\nabla}^2 \boldsymbol{\lambda}_i\right) \quad \text{in } \Omega_i(t) \qquad (3.9)$$

along with the natural boundary condition to take care of the surface integral that appears in the weak form:

$$\left(\mathbf{n}_1 \cdot \boldsymbol{\nabla}\right) \boldsymbol{\lambda}_i = \mathbf{n}_1 \cdot (\boldsymbol{\sigma}_2 - \boldsymbol{\sigma}_1) \quad \text{on } \partial P_i(t) \forall i = 1, \cdots, P_N \qquad (3.10)$$

Thus, $\left(\boldsymbol{\lambda}_i - \boldsymbol{\nabla}^2 \boldsymbol{\lambda}_i\right)$ is like an additional body force due to the $i-$th particle.

The equations discussed above are not the complete set to solve the fluid-particle system. One must also take care of the continuity equation for the fluid. Here, we have focused on the particle and fluid interaction. Glowinski et al. [17]

have used operator splitting to decompose the equation into fractional time steps and Galerkin finite element method for spatial discretization of all the terms. One important point to highlight is that they have used separate triangular meshing for particle domain in 2D [17, 18] and collocation method with points on surface of particle in 3D [42] and solved for Lagrange multiplier separately for each particle, thus a Distributed Lagrange Multiplier (DLM) scheme results. In the next section, a Global Lagrange Multiplier (GLM) scheme modelling Lagrange multipliers as the interaction force is discussed.

3.2 Global Lagrange Multiplier

The formulation of Diaz-Goano et al. [13] is derived in a similar framework as above, but combined equation of motion (§3.1.1) is avoided. Instead separate equations of motion for the fluid and the particle phase are derived along with an interaction force term. In a strong form the set of equations for the fictitious domain obtained after step 2 is as follows:

$$\rho_1\left(\frac{\mathrm{D}\mathbf{u}}{\mathrm{D}t} - \mathbf{g}\right) = \boldsymbol{\nabla}\cdot\boldsymbol{\sigma} - \mathbf{F}, \quad \boldsymbol{\nabla}\cdot\mathbf{u} = 0 \qquad \text{in}\,\Omega, \tag{3.11}$$

$$\Delta M_i \frac{\mathrm{D}\mathbf{U}_i}{\mathrm{D}t} = \Delta M_i \mathbf{g} + \int_{\Omega_i} \mathbf{F}\,\mathrm{d}\Omega_i, \qquad i = 1, \cdots, P_N \tag{3.12}$$

$$\boldsymbol{\omega}_i = \frac{0.5}{V_i}\int_{\Omega_i} (\boldsymbol{\nabla}\times\mathbf{u})\,\mathrm{d}\Omega_i, \qquad i = 1, \cdots, P_N \tag{3.13}$$

$$\mathbf{u} = \mathbf{U}_i + \boldsymbol{\omega}_i \times \mathbf{r}_i \quad \text{in}\,\Omega_i, \qquad i = 1, \cdots, P_N \tag{3.14}$$

The quantity $\Delta M_i = \int_{\Omega_i} (\rho_i - \rho_1)\,\mathrm{d}\Omega_i$ is the buoyant mass of the particle. It is apparent by comparing equations (3.9) and (3.11), that the Lagrange multiplier can be defined as a global interaction force ($\mathbf{F}$). Since the Lagrange multiplier also lies in the same space as the global velocity space i.e. $\mathbb{H}^1_0(\Omega)$, a global Lagrange multiplier $\boldsymbol{\lambda}$ is defined as the solution to the boundary value problem:

$$\begin{aligned} -\alpha\boldsymbol{\lambda} + \mu\boldsymbol{\nabla}^2\boldsymbol{\lambda} &= \mathbf{F} \quad \text{in}\,\Omega, \\ \boldsymbol{\lambda} &= 0 \quad \text{on}\,\Gamma \end{aligned} \tag{3.15}$$

where, $\alpha > 0$ is a constant. Note that this Lagrange multiplier is defined over the entire domain Ω, unlike the distributed Lagrange multipliers defined before. The advantage of GLM is that one single system of equations is to be solved every time-step, unlike the DLM which has to be solved for each particle separately each time-step. The equations for GLM can be discretized on the same mesh used for velocity calculation and hence there is no need for separate particle meshes. It was envisaged that such collective imposition of rigid body constraint at all the particle's domains would be computationally efficient when the solid fraction in the multiphase flow increases. But, this approach also causes problems due to inaccurate resolution of fluid-particle boundary. The details of the particle boundary resolution is discussed in chapter 4.

The final set of equations for an incompressible Newtonian fluid and rigid spherical particles can be written as:

$$\rho_1 \frac{\mathrm{D}\mathbf{u}}{\mathrm{D}t} = -\boldsymbol{\nabla} P + \mu \boldsymbol{\nabla}^2 \mathbf{u} + \alpha \boldsymbol{\lambda} - \mu \boldsymbol{\nabla}^2 \boldsymbol{\lambda}, \quad \boldsymbol{\nabla} \cdot \mathbf{u} = 0 \text{ in } \Omega, \tag{3.16}$$

$$\Delta M_i \frac{\mathrm{D}\mathbf{U}_i}{\mathrm{D}t} = \Delta M_i \mathbf{g} - \int_{\Omega_i} \alpha \boldsymbol{\lambda} \mathrm{d}\Omega_i - \mu \int_{\partial P_i} \frac{\partial \boldsymbol{\lambda}}{\partial \mathbf{n}_1} \mathrm{d}S, \tag{3.17}$$

$$\boldsymbol{\omega}_i = \frac{0.5}{V_i} \int_{\Omega_i} (\boldsymbol{\nabla} \times \mathbf{u})\, \mathrm{d}\Omega_i, \tag{3.18}$$

$$\mathbf{u}_2 = \mathbf{U}_i + \boldsymbol{\omega}_i \times \mathbf{r}_i, \tag{3.19}$$

$$\frac{\partial \mathbf{X}_i}{\partial t} = \mathbf{U}_i, \tag{3.20}$$

where, $i = 1, \cdots, P_N$ and dynamic pressure P is introduced. While $\mathbf{u}$ is the global fluid velocity field that needs to be corrected, $\mathbf{u}_2$ is the correct velocity field in particle domain that needs to be imposed. This distinction is made to avoid confusion during rigid body iteration. Before rigid body iterations they are different and after rigid body iteration $\mathbf{u} = \mathbf{u}_2$. The Gauss divergence theorem is applied to the second order of equation (3.15) as follows:

$$\int_{\Omega_i} \mu \boldsymbol{\nabla} \cdot \boldsymbol{\nabla} \boldsymbol{\lambda} \mathrm{d}\Omega_i = \mu \int_{\partial P_i} \mathbf{n}_2 \cdot \boldsymbol{\nabla} \boldsymbol{\lambda} \mathrm{d}S = -\mu \int_{\partial P_i} \mathbf{n}_1 \cdot \boldsymbol{\nabla} \boldsymbol{\lambda} \mathrm{d}S$$

According to the discussion above with regards to interpretation of Lagrange multiplier in equation (3.9), it was noted that normal gradient of Lagrange multiplier across the interface is equal to the difference in stresses inside and outside the particle domain. This term would cancel off if the stress continuity could be established across the interface (as would be the case with bubbles and drops) but in case of rigid particles this term is crucial in diffusion of momentum or vorticity from the particle surface into the fluid. But, according to Diaz-Goano et al. [13], this term is so small in case of spherical particles that it can be neglected.

Diaz-Goano et al. [13] also use fractional time stepping like Glowinski et al. [17]. But the advection part of the equation is handled using method of characteristics rather than using GFEM. The numerical solution of fluid domain equations is same as the procedure outlined in chapter 2, §2.2 for solving Naiver-Stokes equations. In the next section, the imposition of rigid body constraint as reported in Veeramani et al. [54] is presented. This implementation of the numerical algorithm gives much better accuracy as compared to rigid body constraint substep in [13].

3.2.1 Rigid Body Motion

This is the final step in the overall algorithm where finally the fluid "feels" the presence of the particles. The solution of fluid velocity field over the entire computation domain is obtained as $\mathbf{u}^{**}$ following the procedure outlined in chapter 2, §2.2. This velocity field is taken as the initial guess for the iterations performed to correct fluid field for particle motion. The steps involved are presented below. All steps are shown for second order scheme only, but the first time step is first order.

1. Predict the particle position:

$$\mathbf{X}_i^{p,n+1} = \mathbf{X}_i^{n-1} + 2\Delta t \mathbf{U}_i^n \tag{3.21}$$

2. Set the initial guesses:

$$\begin{aligned}
&\boldsymbol{\lambda}^{0,n+1} = 0,\\
&\mathbf{u}^{0,n+1} = \mathbf{u}^{**},\\
&\tau_0 \mathbf{U}_i^{0,n+1} = -\tau_1 \mathbf{U}_i^n - \tau_2 \mathbf{U}_i^{n-1} + \mathbf{g},\\
&\boldsymbol{\omega}_i^{0,n+1} = \frac{0.5}{V_i}\int_{\Omega_i} (\boldsymbol{\nabla}\times \mathbf{u}^{0,n+1})\,\mathrm{d}\Omega_i,\\
&\mathbf{u}_2^{0,n+1} = \mathbf{U}_i^{0,n+1} + \boldsymbol{\omega}_i^{0,n+1} \times \left(\mathbf{x} - \mathbf{X}_i^{p,n+1}\right)
\end{aligned} \tag{3.22}$$

3. Define iterative correction to a quantity Q by δQ i.e. $\delta Q^{k+1} = Q^{k+1} - Q^k$. For all iteration index $k \geq 0$, the Lagrange multiplier correction is given by:

$$\begin{aligned}
&\left(1 + \frac{\rho_1}{\rho_i - \rho_1}\right)(\alpha I - \mu \boldsymbol{\nabla}^2)\delta\boldsymbol{\lambda}^{k+1,n+1}\\
&\quad = -(\rho_1 \tau_0 I - \mu \boldsymbol{\nabla}^2)(\mathbf{u}^{k,n+1} - \mathbf{u}_2^{k,n+1}), \quad \text{in } \Omega_i\\
&\left(1 + \frac{\rho_1}{\rho_i - \rho_1}\right)(\alpha I - \mu \boldsymbol{\nabla}^2)\delta\boldsymbol{\lambda}^{k+1,n+1} = 0, \quad \text{in } \Omega_1\\
&\delta\boldsymbol{\lambda}^{k+1,n+1} = 0 \quad \text{on } \Gamma
\end{aligned} \tag{3.23}$$

The fluid velocity correction is calculated as:

$$\begin{aligned}
&(\rho_1 \tau_0 I - \mu \boldsymbol{\nabla}^2)\delta\mathbf{u}^{k+1,n+1} = (\alpha I - \mu \boldsymbol{\nabla}^2)\delta\boldsymbol{\lambda}^{k+1,n+1} \quad \text{in } \Omega\\
&\delta\mathbf{u}^{k+1,n+1} = 0 \quad \text{on } \Gamma
\end{aligned} \tag{3.24}$$

Finally, particle velocity correction for this substep is:

$$
\begin{aligned}
\Delta M_i \tau_0 \delta \mathbf{U}_i^{k+1,n+1} &= -\left(\frac{\rho_i - \rho_1}{\rho_i}\right) \int_{\Omega_i} \alpha(\mathbf{u}^{k,n+1} - \mathbf{u}_2^{k,n+1}) \mathrm{d}\Omega \\
&+ \mu \left(\frac{\rho_i - \rho_1}{\rho_i}\right) \int_{\partial P_i} \frac{\partial}{\partial \mathbf{n}_1} (\mathbf{u}^{k,n+1} - \mathbf{u}_2^{k,n+1}) \mathrm{d}S \\
V_i \boldsymbol{\omega}_i^{k+1,n+1} &= 0.5 \int_{\Omega_i} \boldsymbol{\nabla} \times \mathbf{u}^{k+1,n+1} \mathrm{d}\Omega_i \\
\mathbf{u}_2^{k+1,n+1} &= \mathbf{U}_i^{k+1,n+1} + \boldsymbol{\omega}_i^{k+1,n+1} \times \left(\mathbf{x} - \mathbf{X}_i^{p,n+1}\right) \quad \text{in } \Omega_i \\
\mathbf{u}_2^{k+1,n+1} &= 0 \quad \text{in } \Omega_1
\end{aligned}
\tag{3.25}
$$

where, $i = 1, \cdots, P_N$ and I is the identity operator. Upon convergence after $k = N$ iterations, set $\mathbf{u}^{n+1} = \mathbf{u}^{N+1,n+1}, \mathbf{U}_i^{n+1} = \mathbf{U}_i^{N+1,n+1}, \boldsymbol{\omega}_i^{n+1} = \boldsymbol{\omega}_i^{N+1,n+1}$ and $\boldsymbol{\lambda}^{n+1} = \boldsymbol{\lambda}^{N+1,n+1}$.

4. Correct the particle position:

$$
\mathbf{X}_i^{n+1} = \mathbf{X}_i^n + \frac{\Delta t}{2} \left(\mathbf{U}_i^{n+1} + \mathbf{U}_i^n\right) \tag{3.26}
$$

While Glowinski et al. [17] had employed conjugate gradient type of iteration, a Richardson iteration scheme is employed in equations (3.23)-(3.25). Note that this iteration scheme is slightly different from the iteration suggested in Diaz-Goano et al. [13] because the update of particle velocity $\delta \mathbf{U}_i^{k+1,n+1}$ is performed here using the difference $(\mathbf{u}^{k,n+1} - \mathbf{u}_2^{k,n+1})$ directly rather than the Lagrange multiplier increment $\delta \boldsymbol{\lambda}^{k+1,n+1}$. It was generally observed that it takes less iterations and is more accurate. The Laplacian operator used to obtain Lagrange multiplier increment tends to diffuse it out of the particle domain which is actually what is required for the fluid boundary layer correction but the integral of the multiplier increment in the particle domain is reduced and the correction to particle velocity is lower than that required. Therefore is takes more iterations to match the particle and fluid velocity.

The equations are further discretized in space by means of $\mathbb{P}_2 - P_1$ tetrahedral elements in the same manner as illustrated in §2.2.3. The computation of the integrals over Ω_i involved in the iteration above is not very straightforward because the finite element grid does not fit the surfaces of the particles in general. In [13], Gauss quadrature with modified weights were used but the inaccuracy in the computation of the integrals leads to relatively large oscillations in the angular velocity and relatively inaccurate velocity of the centroids of the particles. In chapter 4, a scheme for subdivision of elements on the surface of the particle domain is discussed which approximates the spherical surface of particles much more accurately and gives better control of the angular velocity of the particles. The formulation that we have presented here will be refered to as the iterative algorithm for further comparisons.

3.3 Non-Lagrange Multiplier Approach

In order to completely avoid using Lagrange multipliers, the formulation for fluid and particle is derived as separate equations along with an interaction force just like in the case of Diaz-Goano et al. [13]. But instead of defining a global Lagrange multiplier using this interaction force, the interaction force itself is eliminated [53].

3.3.1 Formulation

Again we start with the conservation equation for fluid momentum (2.1) and mass (2.2) and equation of motion for particle (2.11). For an incompressible Newtonian fluid and rigid particles these equations can be written as:

$$\rho_1 \frac{\mathrm{D}\mathbf{u}_1}{\mathrm{D}t} = -\boldsymbol{\nabla} P_1 + \mu \boldsymbol{\nabla}^2 \mathbf{u}_1, \quad \boldsymbol{\nabla} \cdot \mathbf{u}_1 = 0 \text{ in } \Omega_1 \tag{3.27}$$

$$\rho_i V_i \frac{\mathrm{D}\mathbf{U}_i}{\mathrm{D}t} = (\rho_i - \rho_1)\, V_i \mathbf{g} + \mathbf{F}_i, \quad \forall i = 1, \cdots, P_N \tag{3.28}$$

Then the fluid properties are extended into the particle domain and hence we can define a global fluid stress in the particle domain as $\boldsymbol{\sigma} = -P\boldsymbol{\delta} + \mu[\boldsymbol{\nabla}\,\mathbf{u} + (\boldsymbol{\nabla}\,\mathbf{u})^T]$. This fluid stress, which is based on dynamic pressure, can then be used to evaluate the hydrodynamic force (2.15) on the particle (using Gauss divergence theorem) as follows :

$$\mathbf{F}_i = \int_{\Omega_i} \boldsymbol{\nabla} \cdot \boldsymbol{\sigma} \mathrm{d}\Omega_i \tag{3.29}$$

Note that whenever a buoyancy force is applied to a particle, a fluid stress based on dynamic pressure must be used for consistency. The equations can be non-dimensionalized using characteristic length scale L_c and velocity scale U_c. Multiplying all the equations by $\frac{L_c}{\rho_1 U_c^2}$, we get the final equations in non-dimensional form:

$$\frac{\mathrm{D}\mathbf{u}}{\mathrm{D}t} = -\boldsymbol{\nabla} P + \frac{1}{Re} \boldsymbol{\nabla}^2 \mathbf{u}, \quad \boldsymbol{\nabla} \cdot \mathbf{u} = 0 \text{ in } \Omega_1 \tag{3.30}$$

$$\frac{\rho_i}{\rho_1} \frac{\mathrm{D}\mathbf{U}_i}{\mathrm{D}t} = \left(\frac{\rho_i - \rho_1}{\rho_1}\right) \frac{1}{Fr} \mathbf{e}_g + \frac{1}{V_i} \int_{\Omega_i} \boldsymbol{\nabla} \cdot \boldsymbol{\sigma} \mathrm{d}\Omega_i \tag{3.31}$$

where, $i = 1, \cdots, P_N$, $\boldsymbol{\sigma} = -P\boldsymbol{\delta} + \frac{1}{Re}[\boldsymbol{\nabla}\,\mathbf{u} + (\boldsymbol{\nabla}\,\mathbf{u})^T]$ and $\mathbf{e}_g$ is the unit vector in the direction of gravity. We don't make any distinction between dimensional and dimensionless quantity as this should be obvious from the context. The three governing parameters in this set of equations are Reynolds number Re, Froude number Fr and density ratio $\frac{\rho_i}{\rho_1}$. The non-dimensional numbers are defined as:

$$Re = \frac{\rho_1 U_c L_c}{\mu} \text{ and } Fr = \frac{U_c^2}{L_c g}$$

The next step is to define the interaction force in the same fashion as introduced by Diaz-Goano et al. [13].

$$\mathbf{F} = \begin{cases} -\frac{\mathrm{D}\mathbf{u}}{\mathrm{D}t} - \boldsymbol{\nabla} P + \frac{1}{Re}\boldsymbol{\nabla}^2\mathbf{u}, & \text{in } \Omega_i,\ \forall i = 1, \cdots, P_N \\ 0 & \text{in } \Omega_1 \end{cases} \tag{3.32}$$

Using this definition the Navier-Stokes equation is extended to the entire domain Ω and the particle equation modified as:

$$\frac{\mathrm{D}\mathbf{u}}{\mathrm{D}t} = -\boldsymbol{\nabla} P + \frac{1}{Re}\boldsymbol{\nabla}^2\mathbf{u} - \mathbf{F}, \quad \boldsymbol{\nabla}\cdot\mathbf{u} = 0 \text{ in } \Omega \tag{3.33}$$

$$\frac{\rho_i}{\rho_1}\frac{\mathrm{D}\mathbf{U}_i}{\mathrm{D}t} - \frac{1}{V_i}\int_{\Omega_i}\frac{\mathrm{D}\mathbf{u}}{\mathrm{D}t}\mathrm{d}\Omega_i = \left(\frac{\rho_i - \rho_1}{\rho_1}\right)\frac{1}{Fr}\mathbf{e}_g + \frac{1}{V_i}\int_{\Omega_i}\mathbf{F}\mathrm{d}\Omega_i \tag{3.34}$$

Since, the fluid occupying the particle domain must accelerate like a rigid particle, we can impose the condition:

$$\frac{\mathrm{D}\mathbf{U}_i}{\mathrm{D}t} = \frac{1}{V_i}\int_{\Omega_i}\frac{\mathrm{D}\mathbf{u}}{\mathrm{D}t}\mathrm{d}\Omega_i \tag{3.35}$$

and therefore write the particle equation after some simplification as:

$$\frac{\mathrm{D}\mathbf{U}_i}{\mathrm{D}t} = \frac{1}{Fr}\mathbf{e}_g + \frac{1}{V_i}\left(\frac{\rho_1}{\rho_i - \rho_1}\right)\int_{\Omega_i}\mathbf{F}\mathrm{d}\Omega_i \tag{3.36}$$

At this point, it is more convenient to define another modified interaction force as follows:

$$\mathfrak{F} = \begin{cases} \frac{1}{Fr}\mathbf{e}_g + \left(\frac{\rho_1}{\rho_i - \rho_1}\right)\mathbf{F}, & \text{in } \Omega_i,\ \forall i = 1, \cdots, P_N \\ 0 & \text{in } \Omega_1 \end{cases} \tag{3.37}$$

and rewrite the set of equations:

$$\frac{\mathrm{D}\mathbf{u}}{\mathrm{D}t} = -\boldsymbol{\nabla} P + \frac{1}{Re}\boldsymbol{\nabla}^2\mathbf{u} + \left(\frac{\rho_i - \rho_1}{\rho_1}\right)(\mathbf{G} - \mathfrak{F}), \quad \boldsymbol{\nabla}\cdot\mathbf{u} = 0 \text{ in } \Omega \tag{3.38}$$

$$\frac{\mathrm{D}\mathbf{U}_i}{\mathrm{D}t} = \frac{1}{V_i}\int_{\Omega_i}\mathfrak{F}\mathrm{d}\Omega_i \tag{3.39}$$

where,

$$\mathbf{G} = \begin{cases} \frac{1}{Fr}\mathbf{e}_g, & \text{in } \Omega_i, \forall i = 1, \cdots, P_N \\ 0 & \text{in } \Omega_1 \end{cases} \tag{3.40}$$

Note that by defining the interaction force in this way, we have progressively added forces that result in interaction between particle and fluid. In the original formulation of Glowinski et al. [17], only the hydrodynamic force $\mathbf{F}_i$ was identified as the interaction force and subsequently it was eliminated. Then, in the formulation of Diaz-Goano et al. [13], the interaction force $\mathbf{F}$ accounted not only for hydrodynamic force but also the added mass and Basset forces represented in terms of the material derivative. Finally, the interaction force identified here adds the force of gravity to the list. The latest approach also has additional benefits in terms of numerical computations which we will describe in the next section.

In addition to the set of two momentum equations above, we still need to satisfy the rigid body constraint in particle domain given by:

$$\mathbf{u} = \mathbf{U}_i + \boldsymbol{\omega}_i \times (\mathbf{x} - \mathbf{X}_i), \text{ in } \Omega_i, \forall i = 1, \cdots, P_N \tag{3.41}$$

where, the angular velocity $\boldsymbol{\omega}_i$ is given by (2.22), which was derived in Diaz-Goano et al. [13] using Stokes Theorem from the above relationship and position of the particle $\mathbf{X}_i$ is given by (2.13). Since, this condition does not contain the interaction force itself, it is a side constraint for the two momentum equations and therefore it is natural to impose it via Lagrange multipliers. It turns out, however, that an explicit equation for the interaction force can be derived as shown in the next section.

3.3.2 Discretization

As is the norm, the time discretization is carried out using operator splitting procedure. Here, we adhere to the second order pressure correction version of the procedure described in [13]. In case of a single phase of incompressible flow the pressure correction scheme is second order accurate in time as proved by Guermond and Minev [23]. In the present case though, only a formal second order accuracy can be claimed because of the fictitious domain formulation of the problem.

Advection-Diffusion step

The advection part of the Navier-Stokes equation is solved using method of characteristics in the same way as previously described in §2.2.1. Once the advected velocities $\widetilde{\mathbf{u}}^n$ and $\widetilde{\mathbf{u}}^{n-1}$ are obtained at time levels n and $n-1$ respectively, then the discretization of stokes part of the equation (3.38) is written as:

$$\left(\tau_0 - \frac{1}{Re}\boldsymbol{\nabla}^2\right)\mathbf{u}^* = -\left(\tau_1\widetilde{\mathbf{u}}^n + \tau_2\widetilde{\mathbf{u}}^{n-1}\right) - \boldsymbol{\nabla}P^n + \left(\frac{\rho_i - \rho_1}{\rho_1}\right)\mathbf{G}, \text{ in } \Omega \qquad (3.42)$$
$$\mathbf{u}^* = 0 \quad \text{on } \Gamma$$

where, $i = 1, \cdots, P_N$. Note unlike the equation (2.40), here the gravity term is also added in the right hand side forcing terms. This means the fluid already "feels" a heavier or lighter particle in it at this step. Also, since the gravity term is already taken care of in this term, there is no external force term as used by Diaz-Goano et al. [13] for particle velocity prediction in equation (3.22). A better prediction for particle velocity is obtained by discretization of the equation (3.35) as follows:

$$\tau_0\mathbf{U}_i^* + \tau_1\mathbf{U}_i^n + \tau_2\mathbf{U}_i^{n-1} = \frac{1}{V_i}\int_{\Omega_i^{n+1}}\left(\tau_0\mathbf{u}^* + \tau_1\widetilde{\mathbf{u}}^n + \tau_2\widetilde{\mathbf{u}}^{n-1}\right)\mathrm{d}\Omega_i \qquad (3.43)$$

In order to apply the gravity force and to evaluate the integral over the domain Ω_i^{n+1}, we must know the particle position at this time level $(n+1)$. Unfortunately, at this point we can only predict the position of the particle explicitly using the following discretization of equation (2.13) as follows:

$$\mathbf{X}_i^{p,n+1} = \mathbf{X}_i^{n-1} + 2\Delta t\mathbf{U}_i^n \qquad (3.44)$$

Since, this is an explicit scheme there are strict constraints on what the maximum time step size Δt can be, as we also find out in the validation studies.

Projection step

The next step is to impose the incompressibility constraint on the velocity field and also update the pressure by solving the following equations:

$$\tau_0\left(\mathbf{u}^{**} - \mathbf{u}^*\right) = -\boldsymbol{\nabla}\left(P^{n+1} - P^n\right), \qquad \text{in } \Omega \qquad (3.45)$$
$$\boldsymbol{\nabla}\cdot\mathbf{u}^{**} = 0, \qquad \text{in } \Omega \qquad (3.46)$$
$$\mathbf{u}^{**}\cdot\mathbf{n}_1 = 0 \qquad \text{on } \Gamma \qquad (3.47)$$

The details of the projection method to solve this system of equation is discussed in §2.2.2.

Rigid body constraint

This is the final step where rigid body motion is imposed in Ω_i. In [17, 13], this part is implemented in an iterative fashion on the Lagrange multiplier. Here, the explicit derivation of the interaction force $\mathfrak{F}$ is shown, which will allow us to correct the fluid and particle velocity predicted in the first step. Following the splitting idea of Marchuk-Yanenko, the discretization of equations (3.38), (3.39) and (3.41) can be written as:

$$\tau_0\left(\mathbf{u}^{n+1}-\mathbf{u}^{**}\right)=-\left(\frac{\rho_i-\rho_1}{\rho_1}\right)\mathfrak{F},\ \text{in } \Omega \tag{3.48}$$

$$\tau_0\left(\mathbf{U}_i^{n+1}-\mathbf{U}_i^{*}\right)=\frac{1}{V_i}\int_{\Omega_i^{n+1}}\mathfrak{F}\mathrm{d}\Omega_i \tag{3.49}$$

$$\mathbf{u}^{n+1}-\left(\mathbf{U}_i^{n+1}+\boldsymbol{\omega}_i^{n+1}\times\left(\mathbf{x}-\mathbf{X}_i^{p,n+1}\right)\right)=0,\ \text{in } \Omega_i^{n+1} \tag{3.50}$$

where, $i=1,\cdots,P_N$. The integral of equation (3.48) over the particle domain Ω_i^{n+1} and the equation (3.49) in integral form are:

$$\tau_0\int_{\Omega_i^{n+1}}\left(\mathbf{u}^{n+1}-\mathbf{u}^{**}\right)\mathrm{d}\Omega_i=-\left(\frac{\rho_i-\rho_1}{\rho_1}\right)\int_{\Omega_i^{n+1}}\mathfrak{F}\mathrm{d}\Omega_i$$

$$\tau_0\int_{\Omega_i^{n+1}}\left(\mathbf{U}_i^{n+1}-\mathbf{U}_i^{*}\right)\mathrm{d}\Omega_i=\int_{\Omega_i^{n+1}}\mathfrak{F}\mathrm{d}\Omega_i$$

Substracting and rearranging the resulting equation yields:

$$\tau_0\left[\int_{\Omega_i^{n+1}}\left(\mathbf{u}^{n+1}-\mathbf{U}_i^{n+1}\right)\mathrm{d}\Omega_i+\int_{\Omega_i^{n+1}}\left(\mathbf{U}_i^{*}-\mathbf{u}^{**}\right)\mathrm{d}\Omega_i\right]=-\frac{\rho_i}{\rho_1}\int_{\Omega_i^{n+1}}\mathfrak{F}\mathrm{d}\Omega_i$$

The integral of rigid body constraint over the particle domain yields:

$$\int_{\Omega_i^{n+1}}\left(\mathbf{u}^{n+1}-\mathbf{U}_i^{n+1}\right)\mathrm{d}\Omega_i=\int_{\Omega_i^{n+1}}\boldsymbol{\omega}_i^{n+1}\times\left(\mathbf{x}-\mathbf{X}_i^{p,n+1}\right)\mathrm{d}\Omega_i=0$$

Substituting this result in the preceding equation, gives the integral of interaction force in particle domain as:

$$\int_{\Omega_i^{n+1}}\mathfrak{F}\mathrm{d}\Omega_i=\tau_0\frac{\rho_i}{\rho_1}\int_{\Omega_i^{n+1}}\left(\mathbf{u}^{**}-\mathbf{U}_i^{*}\right)\mathrm{d}\Omega_i \tag{3.51}$$

The interaction force over the entire computational domain can be obtained by first subtracting equation (3.49) from equation (3.48) and substituting rigid body constraint:

$$\begin{aligned}-\left(\frac{\rho_i-\rho_1}{\rho_1}\right)\mathfrak{F}-\frac{1}{V_i}\int_{\Omega_i^{n+1}}\mathfrak{F}\mathrm{d}\Omega_i&=\tau_0\left(\mathbf{u}^{n+1}-\mathbf{U}_i^{n+1}+\mathbf{U}_i^{*}-\mathbf{u}^{**}\right)\\&=\tau_0\left(\boldsymbol{\omega}_i^{n+1}\times\left(\mathbf{x}-\mathbf{X}_i^{p,n+1}\right)+\mathbf{U}_i^{*}-\mathbf{u}^{**}\right)\ \text{by (3.50)}\end{aligned}$$

Then substituting equation (3.51) and introducing the Heavyside function defined as:

$$\mathbf{H} = \begin{cases} 1 & \text{in } \Omega_2 \\ 0 & \text{in } \Omega_1 \end{cases}$$

we get:

$$\begin{aligned} -\left(\frac{\rho_i - \rho_1}{\rho_1}\right)\mathfrak{F} = \tau_0 \Bigg[& \frac{\rho_1}{\rho_i}\frac{1}{V_i}\int_{\Omega_i^{n+1}} (\mathbf{u}^{**} - \mathbf{U}_i^*)\, d\Omega_i \\ & + \boldsymbol{\omega}_i^{n+1} \times \left(\mathbf{x} - \mathbf{X}_i^{p,n+1}\right) + \mathbf{U}_i^* - \mathbf{u}^{**}\Bigg]\mathbf{H} \end{aligned} \tag{3.52}$$

Now, the interaction force can be completely eliminated by substituting equation (3.51) and equation (3.52) into equations (3.49) and (3.48) respectively, to get:

$$\begin{aligned} \mathbf{u}^{n+1} = \mathbf{u}^{**} + \Bigg[& \frac{\rho_1}{\rho_i}\frac{1}{V_i}\int_{\Omega_i^{n+1}} (\mathbf{u}^{**} - \mathbf{U}_i^*)\, d\Omega_i \\ & + \left(\boldsymbol{\omega}_i^{n+1} \times \left(\mathbf{x} - \mathbf{X}_i^{p,n+1}\right)\right) + \mathbf{U}_i^* - \mathbf{u}^{**}\Bigg]\mathbf{H} \end{aligned} \tag{3.53}$$

$$\mathbf{U}_i^{n+1} = \left(1 - \frac{\rho_1}{\rho_i}\right)\mathbf{U}_i^* + \frac{\rho_1}{\rho_i}\frac{1}{V_i}\int_{\Omega_i^{n+1}} \mathbf{u}^{**} d\Omega_i \tag{3.54}$$

The fluid velocity equation as formulated above is implicit because of the angular velocity term $\boldsymbol{\omega}_i^{n+1}$, which is calculated from fluid velocity itself by equation (2.22) as:

$$\boldsymbol{\omega}_i^{n+1} = \frac{0.5}{V_i}\int_{\Omega_i(t)} (\boldsymbol{\nabla} \times \mathbf{u}^{n+1})\, d\Omega_i \tag{3.55}$$

One way to avoid implicit formulation is to evaluate angular velocity based on $(\boldsymbol{\nabla} \times \mathbf{u}^{**})$ instead of $(\boldsymbol{\nabla} \times \mathbf{u}^{n+1})$ or otherwise compute the equation iteratively. If the particle is expected to have high angular velocity changes (which is true in the case of migration of a neutrally buoyant particle in a Poiseuille flow) then one must perform iterative computation of angular velocity and fluid velocity. In cases where the particle does not have appreciable angular velocity (for example in case of low Reynolds number settling particles) it is found that the explicit method is sufficient i.e. use $(\boldsymbol{\nabla} \times \mathbf{u}^{**})$. Finally, the position of the particle is corrected as:

$$\mathbf{X}_i^{n+1} = \mathbf{X}_i^n + \frac{\Delta t}{2}\left(\mathbf{U}_i^{n+1} + \mathbf{U}_i^n\right) \tag{3.56}$$

The spatial discretization of the set of equations in the advection-diffusion step, the projection step and of equation (3.53) is performed by means of $\mathbb{P}_2 - P_1$

tetrahedral elements in the same manner as illustrated in §2.2.3. The computation of the integrals over Ω_i involves Gauss quadrature over original elements in the fluid mesh and new sudivided elements created near the fluid-particle interface. This procedure is discussed in detail in the following chapter. The system of linear equations that results is solved using a conjugate gradient solver. The formulation presented in this section will be referred to as the direct algorithm as opposed to the iterative algorithm used before. The direct and iterative algorithms can also be referred to as non-Lagrangian and Lagrangian approach respectively.

Chapter 4

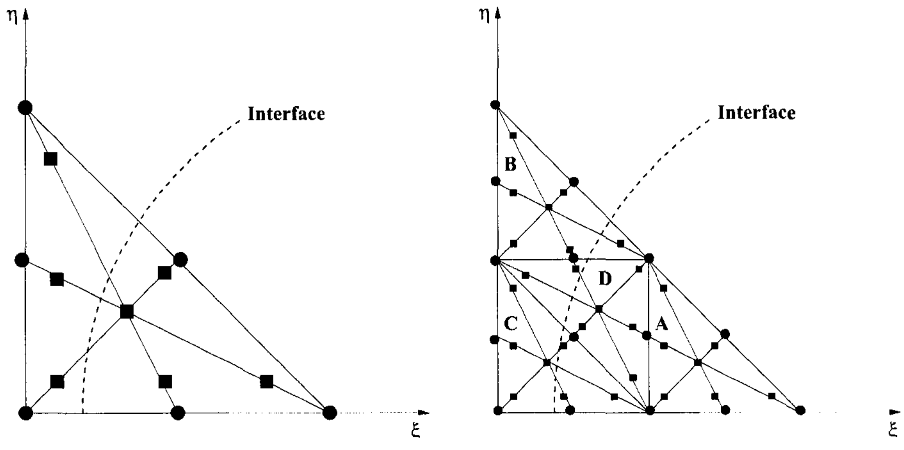
Dynamic Meshing

4.1 Motivation

The fictitious domain method avoids separate meshing of the particle domain. However, when evaluating the integrals resulting from the weak formulation, we need information about volume and surface defining the particle and its boundary. In general, as the particle moves its surface may not align with the fixed base mesh on which fluid motion is solved. In other numerical methods like Arbitrary Lagrangian Eulerian (ALE) method, the base mesh is made to fit the body of the particle at its initial position. Then as the particle moves, the mesh nodes laying on the surface of particle are moved along with it. This causes the elements to get skewed and therefore every few iterations the mesh needs to be remade. Obviously, this is computationally too expensive since it involves assembling the matrix structure according to the new connectivity at least partially if not fully. In the light of this computational overhead, the advantage that the fictitious domain approach provides by using a single fixed base mesh is worth preserving. Of course this freedom from re-meshing comes with the responsibility that we compute the integrals accurately. Specifically, we are interested in the accurate computation of the integrals in the right hand side of the equations (3.53) to (3.55).

4.2 Gaussian Quadrature

Gaussian quadratures are used for performing numerical integrations. The integrands in our case are at most fourth order polynomials such as those arising in the mass matrix calculation. In [13], the Gaussian integrations are performed based on adjusted weights for Gaussian points that are found inside the particle volume when the element is cut by the interface. The weights ($w_{i,j,k}$) are adjusted such that their sum is equal to one. There is no additional meshing needed for the particle in this approach. This method for a two dimensional standard triangle is illustrated in figure 4.1. This simple fix does not seem to give accurate results as discussed in the following section.

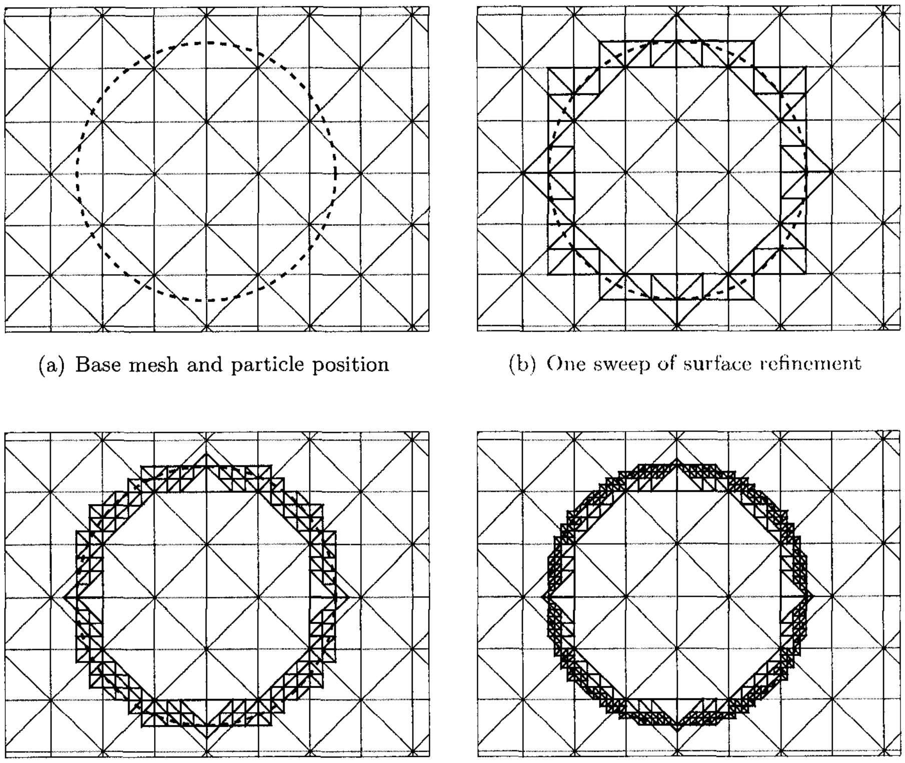


(a) Gauss points in parent element (b) Gauss points in subdivided elements

Figure 4.1: The Gauss points represented with filled squares are shown here for a standard triangular element

4.2.1 Refinement by subdivision

A second method is to refine the mesh on the surface of the particle. An unstructured tetrahedral mesh refinement is described by Bey [5] based on regular sub-division of the element into eight equal volume tetrahedra. Further, the elements that are not regularly refined have to be irregularly refined to preserve consistency. In our case, the refinement is needed only for the accurate evaluation of the integrals and not for the discretization of the equations; therefore we avoid irregular refinement and do only regular refinement of elements on the boundary. The elements inside the particle domain are kept and elements outside are dissolved. The procedure can be repeated any number of times on the resulting elements that are on the boundary. Number of sweeps corresponds to the number of times the procedure is applied. The initial base mesh, particle meshing created after one, two and three sweeps of refinement are shown in figure 4.2. These figures are obtained by extracting a transverse plane through the center of the particle. Two different base meshes with different resolution were built to test the algorithm. The base meshes cover sufficient three dimensional space to accomodate one particle of diameter $d_p = 1.0$. The coarse mesh has node spacing of $0.2d_p$ and the finer mesh has node spacing of $0.1d_p$. Times t reported are the total program time for one step. The objective is to get the particle volume as accurately as possible. The radius of the particle being 0.5, the exact volume of the sphere should be $V = 0.5236$. These test results are presented in table 4.1. Each successive sweep leads to a more accurate definition of the boundary and the volume. The drawback is that there will be an exponential increase of number of nodes on the boundary which increase the total processing time of the new elements. The processing time includes the interpolation of velocities at the new



(a) Base mesh and particle position

(b) One sweep of surface refinement

(c) Two sweeps of surface refinement

(d) Three sweeps of surface refinement

Figure 4.2: Successive refinement of mesh on the particle surface

Table 4.1: Test results of particle surface mesh refinement

Mesh details	Sweeps	Number of full elements	Number of surface elements	Volume	Exec time	% error
Coarse mesh (320 Elements)	0	384	224	0.4867	1.21	7.05
	1	2800	1552	0.5150	7.98	1.64
	2	13020	6092	0.5215	34.96	0.40
Finer mesh (5000 Elements)	0	4688	1800	0.5175	14.61	1.17
	1	19760	8952	0.5221	54.89	0.29
	2	91710	43620	0.5232	242.32	0.08

nodes from nodes in the base mesh and the actual computation of the integrals. If we use the estimate of particle volume as a measure of the accuracy achieved, then it is clear from the table 4.1 that even for a coarse mesh, one sweep of refinement is sufficient to reduce the error below 2.0%. This is a good tolerance to specify for engineering calculations. We can conclude from the error estimates that when a finer mesh is employed, the surface refinement is unnecessary. Often when a coarse mesh is employed to compute on a large domain with large number of particles, surface refinement can come in handy. In this case, one sweep of surface refinement is recommended to improve the accuracy. At the same time, more than one sweep of refinement is an overkill because we already have good accuracy and tremendous increase in computations associated with large number of new elements is unnecesary.

The effect of this approach on distribution of gauss points is illustrated for the two dimensional standard triangle in the figure 4.1. The volume of elements that are cut by the fluid-particle interface is one-eigth of the original element. In the two dimensional illustration shown in figure 4.1, the element 'A' is completely inside the particle domain. Therefore, integration over this new element will be accurate. However, using the adjusted weights for Gaussian quadrature over the remaining elements still leads to some error.

4.2.2 Refinement by body-fitted mesh

A third method is to sub-divide the boundary elements into body fitted elements. Again these elements are used only for integration purposes and consistency of subdivision with neighbouring elements outside the particle domain does not matter. Ruprecht and Müller [45] described an edge-based tetrahedral sub-division method which was adapted by Chen et al. [8] for computation of interface in multiphase flows. We have implemented a similar method to get the body-fitted elements. In general, the elements on the surface can be cut by the spherical surface of the particle into a combination of tetrahedron, pyramid or prism. The resulting geometry depends on the number of edges cut and the number of nodes that fall inside the particle domain. It is also possible that some of the nodes fall on the surface of the particle. Since it is difficult to accurately infer this because of round off errors in position, we use a small buffer zone around the node in question to decide whether it is inside or outside.

The various configurations that are possible are obtained by visualization exercise and are tabulated in table 4.2. The nodes can either be inside the particle domain, on the interface (i.e. they lie in the buffer zone) or outside the particle domain. For a spherical particle, the nodes can be easily classified based on their distance from the center of the particle. For non-spherical particles, the classification would depend on the geometry of the shape and its orientation. The numbers in brackets in the caption of each case denotes the number of nodes on the interface and number of nodes inside the domain. We cannot code for a geometry until at least one node is internal and edge is cut between an internal and an external/interfacial node. In the figures, internal nodes are shown with filled circles (•), nodes on the surface are shown with empty circles (○) and nodes that are new are marked by a cross (×). The intersection of the particle surface with the elements are shown with curved lines.

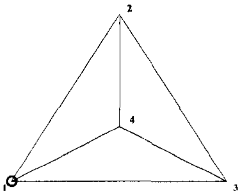
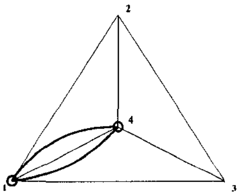
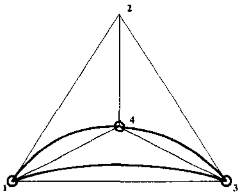
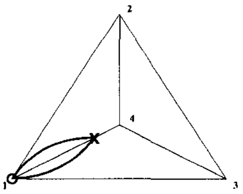
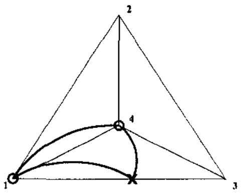
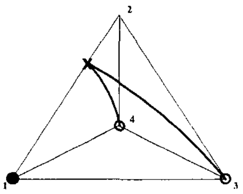
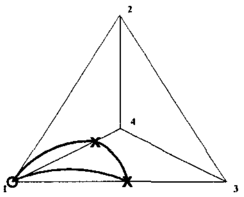
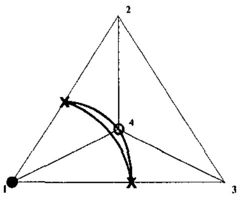
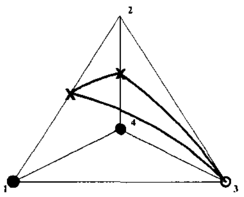
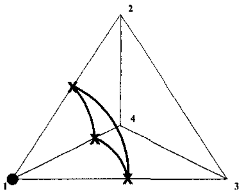
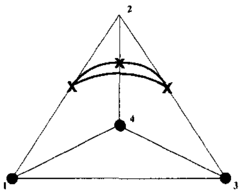
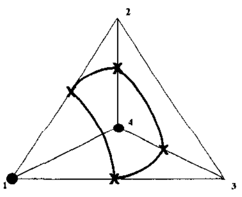
Some special cases where none of the edges of the element are cut but some of the corner nodes fall on the surface can also occur as shown in figure 4.3.

The cases which lead to creation of new elements are labeled and same labels are used in the code. The description of each labeled case is as follows:

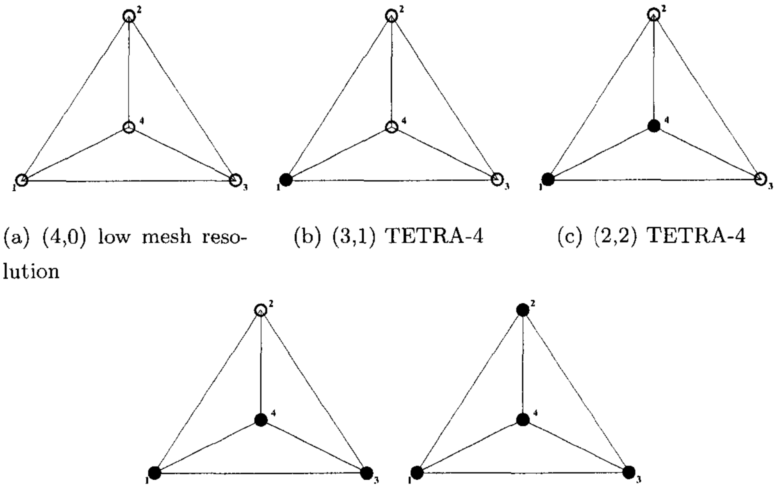
TETRA-1 Three new nodes corresponding to three edges cut are created. These three nodes along with one original node form the corner of the new tetrahedron. Mid-nodes on the all the edges of the tetrahedron must be created to make it a second order Taylor-Hood element. The edges that are formed between two newly defined nodes are curved. Therefore, defining mid-nodes for these edges using linear interpolation of co-ordinates leads to some error.

TETRA-2 This case is different from the last case in one of the edges which is

Table 4.2: Configuration of different geometric elements that is possible with element cut by spherical particle boundary

		Number of corner nodes in/on the particle domain		
		1	2	3
Number of edges cut	0	(a) (1,0)† 	(b) (2,0)† 	(c) (3,0)† 
	1	(d) (1,0)† 	(e) (2,0)† 	(f) (2,1) TETRA-3 
	2	(g) (1,0)† 	(h) (1,1) TETRA-2 	(i) (1,2) PYRAM-3 
	3	(j) (0,1) TETRA-1 	(k) (1,1)† NOT POSSIBLE	(l) (0,3) PRISM-3 
	4	(m) (0,1)† NOT POSSIBLE	(n) (0,2) PRISM-2 	(o) NOT POSSIBLE

† (on,in) nodes, we cannot code for a geometry until at least one node is internal and edge is cut between an in and on/out node.



(a) (4,0) low mesh resolution

(b) (3,1) TETRA-4

(c) (2,2) TETRA-4

(d) (1,3) TETRA-4

(e) (0,4) TETRA-4 (fully in)

Figure 4.3: Configuration of cut elements leading to full tetrahedra

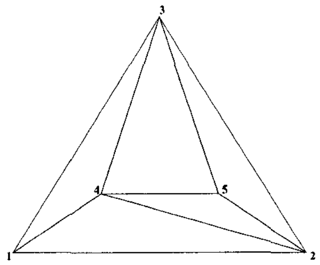
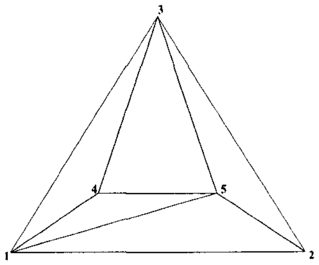
now formed between one internal node and one node on the surface. Therefore, the edge is not taken as cut and is assumed to be fully inside the particle domain. No mid-node is created for this edge, since both the end nodes are original nodes. The mid-node is recycled from the connectivity table of the original element. Mid-nodes must be created for edges cut and new edges formed.

TETRA-3 This case is similar to previous case with two of its edges formed by internal and surface node connection. Same procedure as used for last case is used.

TETRA-4 These are set of special cases where all the corner nodes are either inside or on the surface of particle. Therefore, there is no need to create any new elements.

PYRAM-3 This case results in a pyramid geometry which must be subdivided into two tetrahedra. The division can be performed in two ways as shown in table 4.3. The resulting tetrahedra are shown in the figure. The difference between the two templates is whether one connects nodes 2 and 5 or 1 and 5. The decision to use a particular template is made on the basis of distance between the nodes that get connected. Usually, the nodes pair that gives

Table 4.3: Templates for sub-division of pyramid into two tetrahedra

Template	 (a) Pyramid 1	 (b) Pyramid 2

Connectivity	{ 1, 2, 3, 4}
	{ 2, 3, 4, 5}
		{ 1, 2, 3, 5}
		{ 1, 5, 3, 4}

longest diagonal must be connected to give elements with good aspect ratio. The surface element is created using the face {3,4,5}. The creation of mid-nodes is similar to the procedure decribed for above cases.

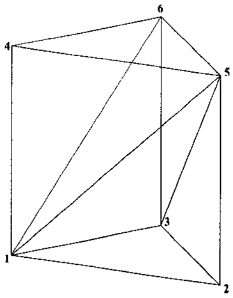
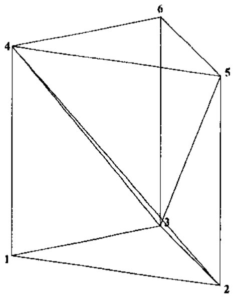
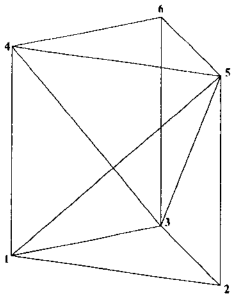
PRISM-2 This case results in a prism geometry which must be divided into three new tetrahedra. The division can be performed in six different ways as shown in table 4.4. The resulting tetrahedra are shown in the figure. Again, the decision to use a particular template is made on the basis of distance between the nodes that get connected. The surface elements for the odd templates are the faces {2,3,5} and {3,6,5}. For even templates, the surface elements come from faces {2,3,6} and {2,6,5}.

PRISM-3 This case again gives rise to prism geometry and using the same template described in the case we can create three tetrahedra.

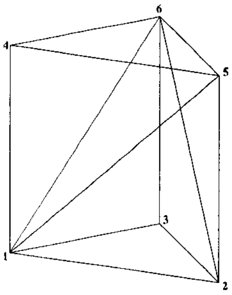
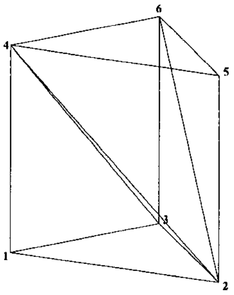
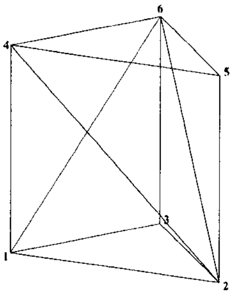
Again, the values of the variable solved at the nodes are interpolated from the parent element nodes. In this method, we get well defined surface elements and volume elements of good aspect ratio. The number of nodes on the surface is also much reduced as compared to the sub-division method. One can also use an hybrid approach i.e. refine the surface elements by one sweep of sub-division refinement and then create body-fitted elements (figure 4.4(a)). We found that this approach does not give any better resolution than just body fitted elements (figure 4.4(b)).

The placement of gauss points for the case of body-fitted mesh is illustrated for the case of two dimensional triangular element in figure 4.5. Since the quadrature

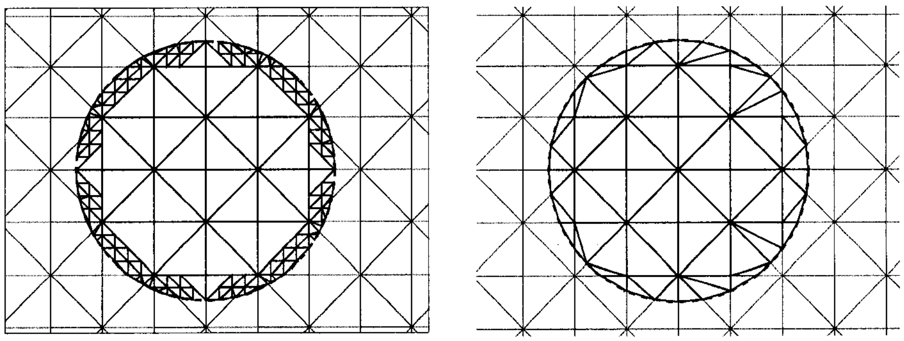
Table 4.4: Templates for sub-division of prism into three tetrahedra

Odd template	(a) Prism 1  	(b) Prism 3  	(c) Prism 5 

Connectivity	{ 2, 3, 1, 5}
	{ 3, 1, 5, 6}
	{ 1, 5, 6, 4}
		{ 1, 2, 3, 4}
		{ 2, 3, 4, 5}
		{ 3, 4, 5, 6}
			{ 1, 2, 3, 5}
			{ 1, 5, 3, 4}
			{ 3, 4, 5, 6}

Even template	(d) Prism 2  	(e) Prism 4  	(f) Prism 6 

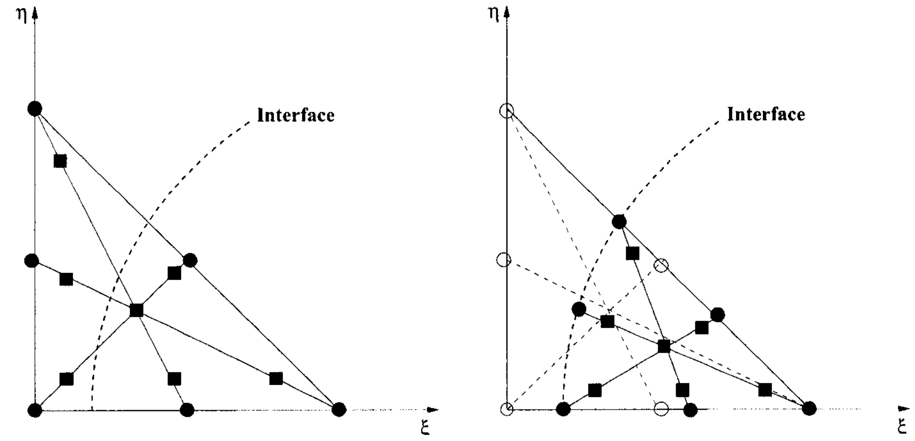
Connectivity	{ 1, 2, 3, 6}
	{ 1, 2, 6, 5}
	{ 1, 5, 6, 4}
		{ 1, 2, 3, 4}
		{ 2, 3, 4, 6}
		{ 2, 6, 4, 5}
			{ 1, 2, 3, 6}
			{ 2, 6, 4, 5}
			{ 2, 6, 1, 4}



(a) Body-fitted elements after two sweeps of subdivision

(b) Body-fitted elements without subdivision

Figure 4.4: Body fitted mesh on the particle surface with and without subdivision



(a) Gauss points in parent element

(b) Gauss points in body-fitted element

Figure 4.5: The Gauss points represented with filled squares are shown here for a body-fitted triangular element

rule now applied on the entire body-fitted element without adjustment of weights, the quadrature is not only consistent but also accurate.

4.3 Control of angular velocity

In this section, the trajectory of a settling particle is analyzed in three dimensions. The Reynolds number is defined $Re = \frac{\rho_1 U_c L_c}{\mu}$ where for a settling particle case U_c is the free settling velocity, L_c is the diameter of the particle. In this particular test case $Re = 1.5$. At this low Reynolds number the wake of the particle is axis-symmetric and therefore the path of the particle should be straight downwards without any rotation or oscillations associated with vortex shedding.

The particle domain is refined using subdivision of surface elements method and with body-fitted elements. The results for the angular velocity of the particle as a function of time are compared to the results obtained without any particle meshing in figure 4.6.

It is obvious that an accurate definition of particle surface is very important in order to reduce the non-physical oscillation originating from accumulating numerical error. If the particle surface is not symmetric (point symmetry), this irregularity on surface leads to non-uniform torque on the particle causing oscillations.

4.4 Conclusion

We can conclude that the resolution lost due to not using a separate particle mesh for Lagrange multiplier calculations (as done by Glowinski et al. [17]) can be regained by using particle meshing. It is recommended to use body-fitted meshes because it not only leads to accurate calculation of Gaussian quadrature but does so with lowest number of additional nodes for the new elements. We find in our later studies on migration of particles in Poiseuille flow that controlling the angular velocity is crucial to get convergence of results. The details of this study are discussed in a later chapter.

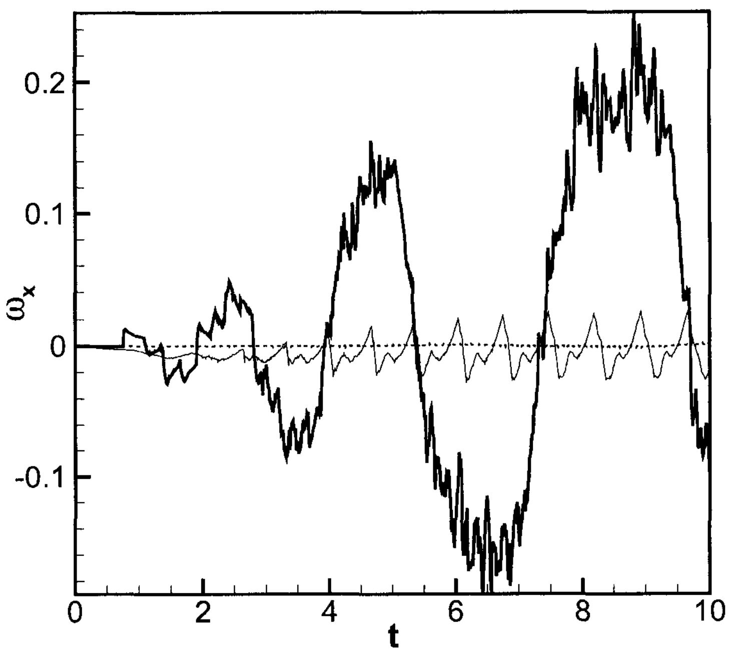


Figure 4.6: Comparison of the angular velocity of a settling particle when calculations are performed using different methods of particle meshing: No mesh (Thick solid), Surface mesh (thin solid) and Body-fitted mesh (dashed).

Chapter 5

Motion of a Single Particle

Before the numerical code can be used to obtain insight into new problems, it must be validated using known behaviour of well studied physical situations. The Navier-Stokes equation solver part had been validated for standard test cases like backward facing step and cavity flow [12]. In this chapter we will focus on the validation of single particle motion in a fluid. First we will validate our new formulation for the case of a single settling spherical particle with experimental data available in literature. We will compare the convergence of solution for the new formulation i.e. non-Lagrange multiplier approach or direct method with the old formulation i.e. global Lagrange multiplier approach or iterative method.

Then in the next section we will look at the convergence of solution for the migration of neutrally buoyant particle in Poiseuille flow in a circular channel. This is a challenging problem to validate, unless the angular velocity calculation is accurate. We will see how the dynamic meshing of particle plays a crucial role in solution convergence for this problem.

5.1 Validation of settling particle problem

5.1.1 Experimental data

The performance of the code was validated using the experimental results published by ten Cate et al. [51]. They used particle image velocimetry (PIV) to calculate the settling of spherical particles at different Reynolds numbers. The fluid medium used was silicon oil suspended with neutrally buoyant hollow glass spheres approximately 10 μm in size. The settling particle used, was a precision Nylon bearing with a diameter of $d_p = 15\ mm$ and density $\rho_2 = 1120\ kg/m^3$.

As the particle settles, it disturbs the fluid around it and the neutrally buoyant glass spheres follow the flow around the settling particle. The fluid velocity was determined by displacement of the glass spheres between two consecutive frames captured by a high speed camera through cross correlation.

The experimental data are available for different particle Reynolds numbers obtained by changing the type of silicon oil used. The different cases used for validation are tabulated in table 5.1.

Table 5.1: Setup of settling experiments.

ρ_1 $[kg/m^3]$	μ_1 $[Pa \cdot s]$	u_∞ $[m/s]$	$Re = \left(\frac{\rho_1 \cdot u_\infty \cdot d_p}{\mu_1}\right)$	$St = \left(\frac{1}{9} Re \frac{\rho_2}{\rho_1}\right)$	$Fr = \left(\frac{g d_p}{U_i^2}\right)$
970	0.373	0.038	1.5	0.19	0.009
965	0.212	0.060	4.1	0.53	0.025
962	0.113	0.091	11.6	1.50	0.056
960	0.058	0.128	31.9	4.13	0.111

The dimensions of the container used are $depth \times width \times height = 100 \times 100 \times 160\ mm$. The dimensions of the mesh used in our simulation are scaled from container dimensions such that the diameter of particle is 1 unit. The Reynolds number (Re) is based on particle diameter and free settling velocity of the particle. The Stokes number (St) is the ratio of particle inertial force to the fluid viscous force. The Froude number (Fr) is the ratio of gravity to inertia of the particle.

5.1.2 Numerical set-up

In all the simulations for a settling particle, with the new formulation the angular velocity at the time level $n+1$ in the equation (3.53) is extrapolated according to $(\nabla \times \mathbf{u}^{n+1}) = (\nabla \times \mathbf{u}^{**})$. The range of Reynolds numbers in the experiments is too low to expect any appreciable angular velocity.

The size of the mesh in non-dimensional units are $7 \times 7 \times 11$. Two different meshes with resolution as detailed in table 5.2 were made. The distribution of nodes are as given in the table, for example $(-3.5 : 0.5 : -1.0)$ which is expanded as $(-3.5, -3.0, -2.5, -2.0, -1.5, -1.0)$ is chosen to give good resolution around particle at all times and also to reduce the total number of nodes.

Combination of value in x, y and z dimension gives the co-ordinates of a corner node. The result is a cubic array of nodes. Further nodes are introduced into each of the cube's edges and five second order tetrahedra are formed. The resulting

Table 5.2: Placement of nodes for single particle settling simulation

Mesh type	Dimension	Placement
coarse	x and z	$-3.5 : 0.5 : -1.0$, $-0.6 : 0.2 : 0.6$, $1.0 : 0.5 : 3.5$
	y	$0.0 : 0.2 : 11.0$
finer	x and z	$-3.5 : 0.5 : -1.0$, $-0.6 : 0.1 : 0.6$, $1.0 : 0.5 : 3.5$
	y	$0.0 : 0.1 : 11.0$

meshes have 134139 and 467261 nodes respectively. Henceforth, these will be referred to as the coarse and fine mesh. A lateral and transverse section of the meshes are plotted in figure 5.1.

The experiments are reported for a single particle settling in a closed box, hence wall boundary conditions on all six faces of the domain are used in the numerical simulations.

5.1.3 Comparison and Convergence

Convergence in time

In figure 5.2(a) for coarse mesh and figure 5.2(b) for finer mesh, we compare the vertical velocity component of the settling particle for the four different Reynolds numbers tabulated in table 5.1.

The direction of gravity is in negative y-direction. Three different time step resolutions of $0.05, 0.01$ and 0.005 are used. They are also compared to the experimental results from ten Cate et al. [51]. There is a clear convergence towards the experimental results with the decrease of time step size for both mesh resolutions. The deviation that occurs at small Reynolds numbers and large times is due to the fact that the numerical speed of the particle is consistently greater than the experimental value at each time step and its cumulative effect is that particle reaches the bottom faster. Hence, the deceleration of the particle occurs earlier than in the case of experiments. At the bottom wall after the particle surface is past the last grid point, we add a sub-grid lubrication force to decelerate the particle. For further discussion on sub-grid lubrication model refer to §6.2. This additional force results in the slight discontinuity in the trajectory seen near the end time steps. However, the convergence of the results with respect to time step is obvious.

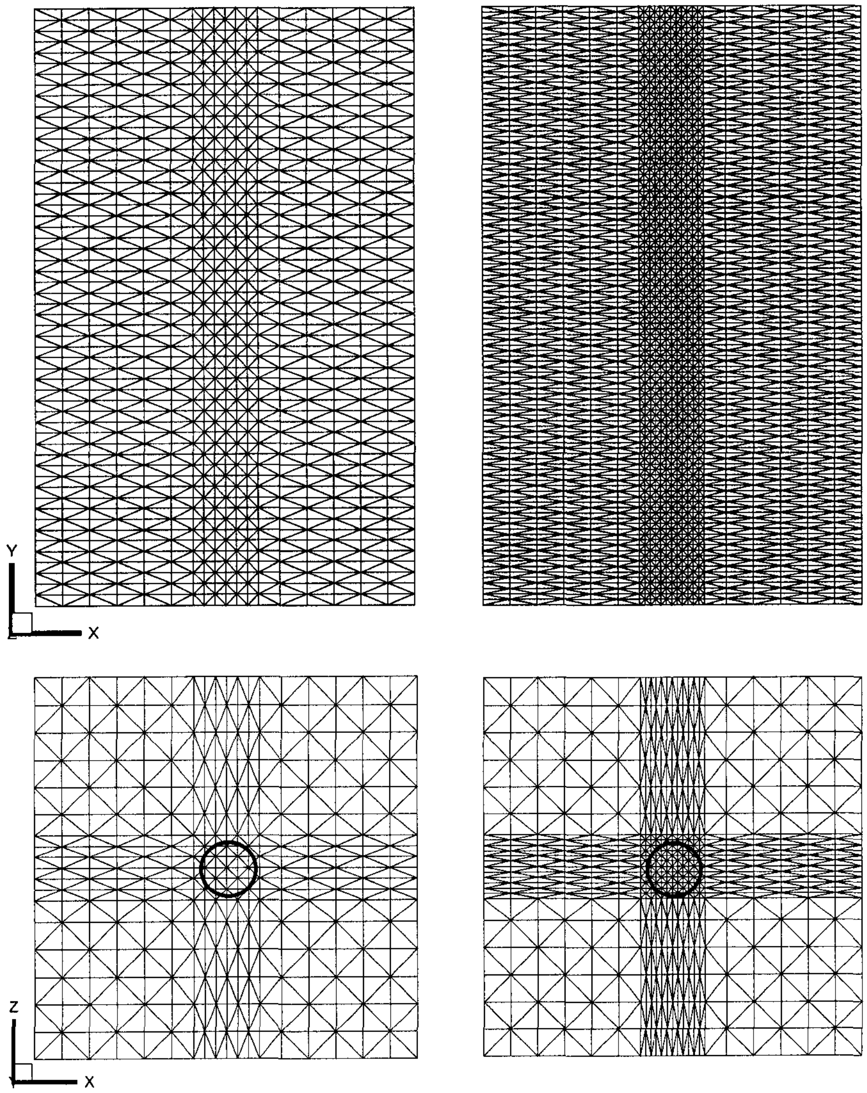
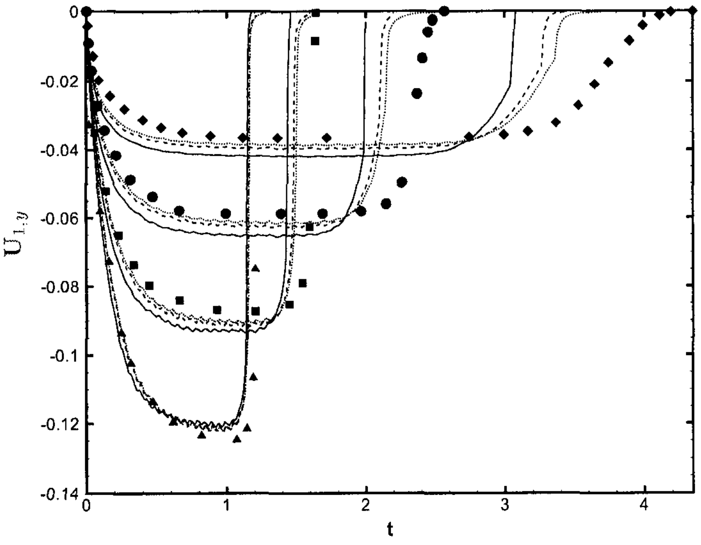
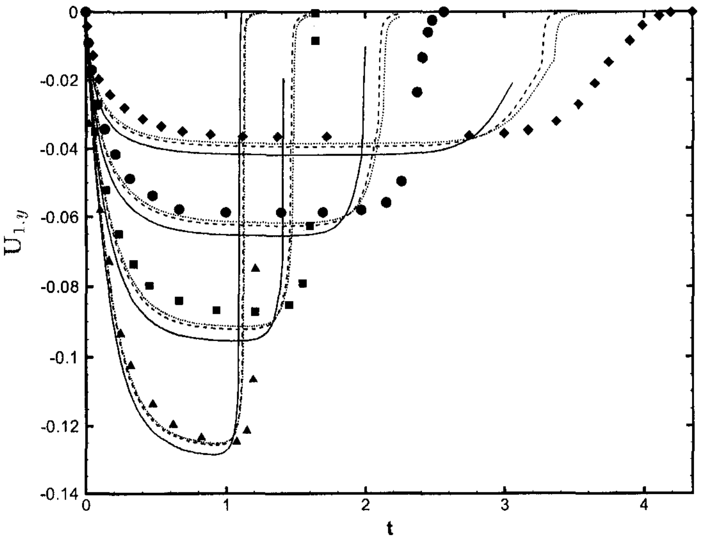


Figure 5.1: Comparison of coarse and fine meshes



(a) coarse mesh



(b) fine mesh

Figure 5.2: Comparison for numerical results at three different time step sizes 0.05 (solid lines), 0.01 (dashed lines) and 0.005 (dotted lines) with experimental results (symbols) for the vertical component of the particle velocity at four different Reynolds numbers: Re = 1.5 (◆), 4.1 (•), 11.6 (■), 31.9 (▲).

Convergence in space

In figure 5.3(a) and 5.3(b) the comparison of coarse and fine mesh simulations at two different time step sizes is done for the four different Reynolds numbers. The grid resolution only improves the results marginally at relatively higher Reynolds numbers and on the basis of these simulations one can not confirm grid independent results.

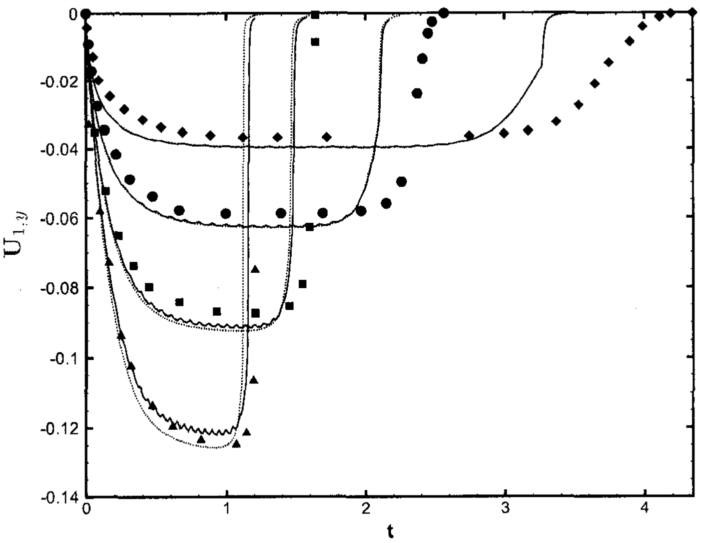
One characteristic feature in these graphs is the presence of variations in the terminal settling velocity, particularly salient at coarse grid resolution. This is a numerical artifact due to the fictitious domain formulation. Since there is no separate particle mesh, the resolution of particle volume is dependent on the grid resolution. In case of the coarser mesh, the particle's volume varies significantly as it moves across the grid. The variation in particle volume is reflected in its settling velocity. The artifact seems to go away with increased mesh resolution and they have little to do with time step size because the variations can be seen on all plots of coarse mesh results with different time steps (figure 5.2(a)). One solution to remedy this problem is to use dynamic particle meshing as discussed in chapter 4.

The tables 5.3 and 5.4 record the absolute value and variation in terminal settling velocities obtained for coarse and fine meshes at different time steps for the four Reynolds numbers. The experimental results reported by ten Cate et al. [51] is also included.

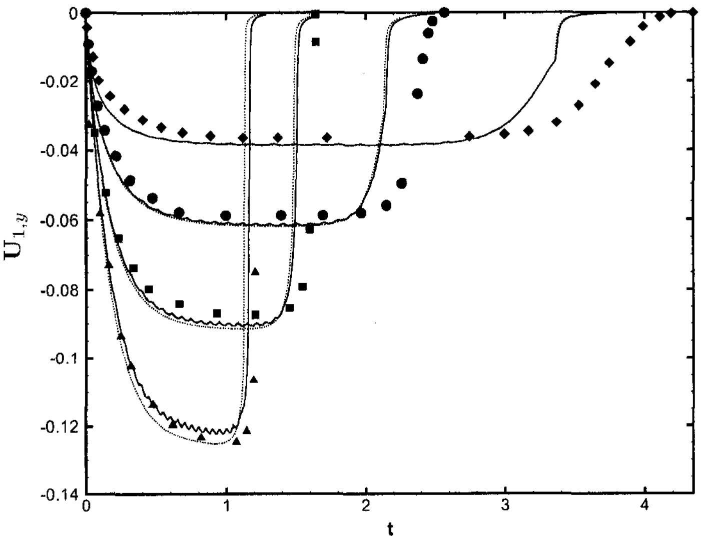
Table 5.3: Terminal settling velocities in cm/s from simulations in coarse mesh along with variations as the particle moves on the grid

$Re \downarrow$	Direct method			Experiment
$dt \rightarrow$	0.05	0.01	0.005	-
1.5	4.19 ± 0.02	3.95 ± 0.02	3.86 ± 0.02	3.60
4.1	6.52 ± 0.03	6.24 ± 0.03	6.14 ± 0.03	5.72
11.6	9.30 ± 0.05	9.11 ± 0.05	9.02 ± 0.05	8.73
31.9	12.02 ± 0.07	12.10 ± 0.08	12.13 ± 0.09	12.35

Clearly, for both coarse and fine mesh simulations the terminal settling velocity data suggests the same trends as shown in the plots. The convergence in space and even grid independence is debatable because when we take the experimental results as the benchmark, it is observed that coarse mesh simulations give marginally



(a) $dt = 0.01$



(b) $dt = 0.005$

Figure 5.3: Comparison for numerical results with coarse (134139 nodes, solid lines) and fine (467261 nodes, dotted lines) meshes with experimental results (symbols) for the vertical component of the particle velocity at four different Reynolds numbers: Re = 1.5 (♦), 4.1 (•), 11.6 (■), 31.9 (▲).

Table 5.4: Terminal settling velocities in cm/s from simulations in finer mesh along with variations as the particle moves on the grid

$Re \downarrow$	Direct method			Experiment
$dt \rightarrow$	0.05	0.01	0.005	-
1.5	4.18 ± 0.004	3.96 ± 0.003	3.87 ± 0.004	3.60
4.1	6.56 ± 0.005	6.26 ± 0.006	6.18 ± 0.007	5.72
11.6	9.54 ± 0.008	9.23 ± 0.01	9.14 ± 0.01	8.73
31.9	12.8 ± 0.02	12.54 ± 0.02	12.50 ± 0.02	12.35

better results than finer mesh simulations at least with the time resolutions that we have employed. Unlike the pure advection case where the Courant-Friedrichs-Lewy (CFL) condition provides some framework to chose appropriate time step size for a given grid resolution, we don't have an explicit rule to follow which would relate the time step size (Δt) and the space resolution (Δx). Moreover the meshes being non-uniform in node spacing moving from inside towards the wall boundaries and velocities ranging from terminal settling velocity near the particle to zero at the walls, further complicate the choice of resolution in both space and time. To avoid complications due to non-uniform mesh, we can restrict ourselves to uniform mesh. However, in case of uniform mesh the number of nodes required to resolve the domain becomes overwhelming especially in 3D case. For example, for the dimensions of the settling particle simulations the uniform coarse mesh with node spacing of 0.2 leads to $464,891$ nodes and a finer mesh with node spacing of 0.1 leads to $3,854,701$. Therefore, the meshes that we have now are the best possible to study convergence.

Comparison with iterative method

In figure 5.4 and table 5.5 the results obtained in the present algorithm which is direct and explicit in the sense of the rigid body constraint, are compared to the results for the same problem obtained with the iterative method proposed in Diaz-Goano et al. [13].

The iterative method performs on an average between 2 and 11 iterations per time step (for an accuracy of 0.01 in a maximum norm) depending on the Reynolds number as shown in table 5.6.

Obviously the number of iterations is higher for larger time steps. For the same problem, the results from the direct method are better which is probably

Table 5.5: Terminal settling velocities in cm/s from simulations in coarse mesh using direct method and iterative method using time step size of $dt = 0.01$

Re ↓	Direct method	Iterative method
1.5	3.95 ± 0.02	4.15 ± 0.05
4.1	6.24 ± 0.03	6.74 ± 0.04
11.6	9.11 ± 0.05	9.59 ± 0.07
31.9	12.10 ± 0.08	12.60 ± 0.09

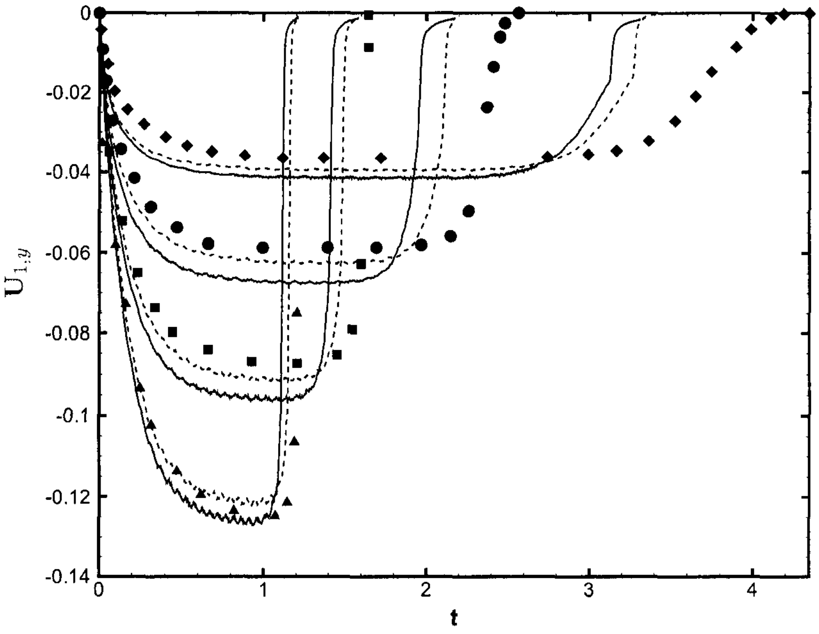


Figure 5.4: Comparison of numerical (lines) and experimental (symbols) results on a coarse mesh (134139 nodes) using old iterative method (solid lines) and new direct method (dashed lines) with time step size dt = 0.01. Experimental results are at Re = 1.5 (♦), 4.1 (•), 11.6 (■), 31.9 (▲).

Table 5.6: Average number of iterations for rigid body constraint in iterative method

Time step size	Reynolds numbers			
dt	1.5	4.1	11.6	31.9
0.05	11	8	6	5
0.01	3	3	2	2

due to the better initial guess for the end-of-step fluid velocity. The reason for the better prediction is that in the present algorithm the gravity which is the dominant driving force is transferred into the Navier-Stokes equations. Thus, the present method is more efficient than the rigid body iteration suggested in Diaz-Goano et al. [13]. In order to lend quantitative basis for this qualitative conclusion, the simulation of settling particle at Re = 11.6 was run using both algorithms on the coarse mesh with 134139 nodes and with time step size of $dt = 0.1$ for 100 time steps. The total execution time for the iterative method was 7762 seconds as opposed to 1054 seconds for the direct method. The table 5.7 gives detailed time profile for the various stages of the algorithms. It can be seen that most of the time in iterative approach is spent in the rigid body iterations and consequently the new algorithm which avoids the rigid body iteration step solves the same problem in significantly shorter time than the iterative algorithm.

Table 5.7: Distribution of time spent on various stages in iterative and direct algorithms

	Iterative algorithm		Direct algorithm	
Stages	Avg. time (s)	% (Total)	Avg. time (s)	% (Total)
Matrix assembly	61.3	0.8	58.6	5.6
First order time stepping	15.5	0.2	11.5	1.1
Convection routines	133.7	1.7	117.6	11.2
Trajectory calculations	15.5	0.2	11.7	1.1
Inverting Stokes operator	226.9	2.9	253.9	24.1
Pressure projection scheme	55.9	0.7	55.1	5.2
Imposing incompressibility criteria	217.9	2.8	229.1	21.7
Imposing rigid-body motion	7031.0	90.6	-	-
Particle predictor	-	-	48.4	4.6
Particle corrector	-	-	245.7	23.3
Saving end-of-step results	3.9	0.0	22.4	2.1

5.2 Validation of migrating particle problem

5.2.1 Experimental data

The well known fact that particles can migrate in a direction completely orthogonal to the primary flow direction in a pipe, will be the basis of our next validation problem. Segré and Silberberg [48] were the first to observe that particles in suspension subjected to Poiseuille flow in a pipe, tend to migrate to a position that is $0.6R$ from the centerline of the pipe, R being the radius of the pipe.

5.2.2 Numerical set-up

We performed the numerical simulations as reported by Pan and Glowinski [42] to validate our code for a single neutrally buoyant particle migration in a circular Poiseuille flow. The flow is along the y-axis. The dimensions used for the particle and channel are shown in figure 5.5. The same parameters as used by Pan and Glowinski [42] are used in the following simulations. The radius of the channel is $R = 2.5$ and its length is $L = 10$, the diameter of the spherical particle is $d_p = 0.75$. The maximum velocity U_m corresponding to the axial velocity maximum of Poiseuille flow profile is 20. The densities of the particle and the fluid are set to 1.0 and the viscosity of the fluid is also set to 1.0. The figure 5.6 shows the contours of the Poiseuille flow profile.

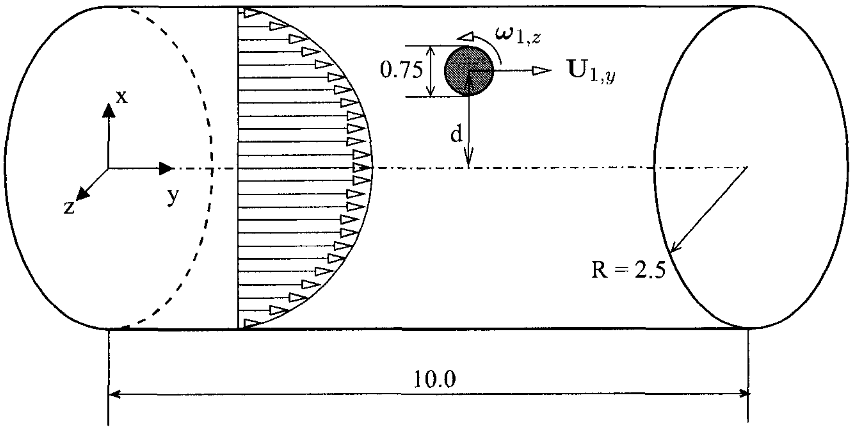
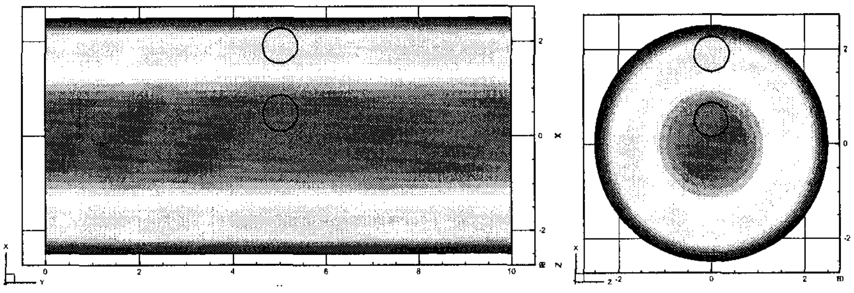


Figure 5.5: Dimensions of the channel and particle

The particle is released from two different radial positions from the centerline along the positive x-axis. Due to the velocity gradient in the flow profile, the particle is expected to develop angular velocity along the z-axis. From the initial positions at $d = 0.2R$ and $d = 0.75R$, the particle is expected to migrate towards the wall and the centerline respectively, until it reaches the final equilibrium position of $d = 0.6R$ from the centerline.



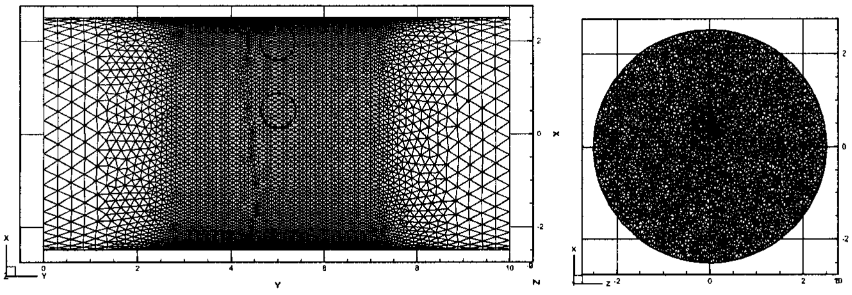
(a) Lateral section

(b) Transverse section

Figure 5.6: Poiseuille flow contours in circular channel. For any single simulation one of the initial positions of the particle indicated by the circles is chosen.

The node spacing between corner nodes of the second order elements used to discretize the grid was 0.1. The unstructured mesh prepared using this grid resolution results in 1.07×10^6 velocity nodes and $136,613$ pressure nodes. In figure 5.7, transverse and lateral section through the channel mesh are shown. A moving reference frame is used at the center of the particle along the flow direction to prevent the particle from getting outside the finely resolved region of the channel. The mesh is denser at the center to resolve the particle motion and is progressively coarsened towards the ends to reduce the total number of nodes.

The transient flow simulation is carried out in two steps. First the steady flow profile around the particle is allowed to develop in the channel by fixing the particle at its initial position. The particle is allowed to rotate freely but it cannot translate in any direction. Then in the second step, the simulation is restarted using the steady state solution as the initial condition and the particle is now allowed to move freely. The motion of the particle in the axial direction (y-axis) is cancelled by the advection of fluid in the opposite direction by using a moving reference frame in that direction. The first step is referred to as constrained motion and the second step is referred to as free motion.



(a) Lateral section

(b) Transverse section

Figure 5.7: Unstructured non-uniform mesh used for particle migration in a channel simulation. For any single simulation one of the initial positions of the particle indicated by the circles is chosen.

5.2.3 Comparison and Convergence

The particle's final velocity and angular velocity at the different radial positions obtained from the constrained motion are compiled in table 5.8 and those from free motion are compiled in table 5.9. The tables compare the results obtained from Distributed Lagrange Multiplier (DLM) simulations done in Pan and Glowinski [42], Arbitrary Lagrangian Eulerian (ALE) simulations of Yang et al. [56] and the present direct method. For the constrained motion simulations, the steady state is achieved after 1000 timesteps at the time step size of 0.005. The values of velocities don't change upto the third decimal place thereafter. The free motion simulations were continued for 10000 time steps to get the behavior of the particle for 50 time units after it is released.

In figures 5.8(a) and 5.8(b), we compare the radial position of the migrating particle as a function of time. The time step size is a crucial factor to get better accuracy for the rate of migration of the particle. The simulation with $dt = 0.005$ gives much more accurate results as compared to $dt = 0.01$, but it still does not capture the correct migration rate when compared to $dt = 0.001$ used by Pan and Glowinski [42]. However, the code is able to capture the final equilibrium position accurately. The total execution time for 5000 time steps with time step size of 0.01 is 294 hrs which means it takes on an average 3.5 minutes per time step on an AMD Opteron^{TM} processor model 252 with cpu speed of 2193.5 MHz. The total execution time for 10000 time steps with time step size of 0.005 is 901.5 hrs which means it takes on an average 5.5 minutes per time step on a an AMD

Table 5.8: Comparison of results for constrained motion simulations

$U_m =$ 20.0	DLM	ALE	Present	DLM	ALE	Present
d/R		U_y			ω_z	
0.10	19.4965	19.493	19.580	0.7751	0.7875	0.7812
0.20	18.8841	18.881	18.965	1.5514	1.5725	1.5734
0.30	17.8656	17.862	17.973	2.3235	2.3539	2.3537
0.40	16.4442	16.439	16.558	3.0872	3.1284	3.1523
0.50	14.6210	14.616	14.756	3.8409	3.8918	3.9060
0.60	12.3957	12.388	12.515	4.5824	4.6424	4.7483
0.70	9.7338	9.705	9.933	5.2798	5.3295	5.3735
0.75	8.1822	8.127	8.387	5.5765	5.5710	5.7214

Table 5.9: Comparison of results for free motion simulations

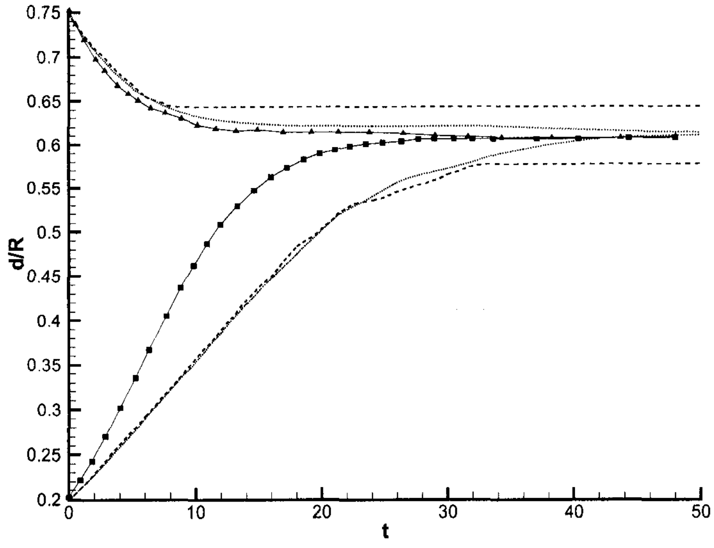
$U_m =$ 20.0	d_e/R		U_y		ω_z	
$d_0/R \rightarrow$	0.20	0.75	0.20	0.75	0.20	0.75
DLM	0.6058	0.6058	12.2353	12.2365	4.6359	4.6286
ALE	0.6011	0.6011	12.364	12.364	4.6513	4.6513
Present	0.6106	0.6139	12.2585	12.1806	4.7574	4.7790

Opteron^{TM} processor model 252 with cpu speed of 997.073 MHz. Ideally, the simulations should have been run on similar architecture if not the same machine but still these simulations indicate more than doubling of total execution time as the time step size is halved. Since the convergence of the algorithm is clear from the two time step sizes used here, we didn't attempt further refined calculations.

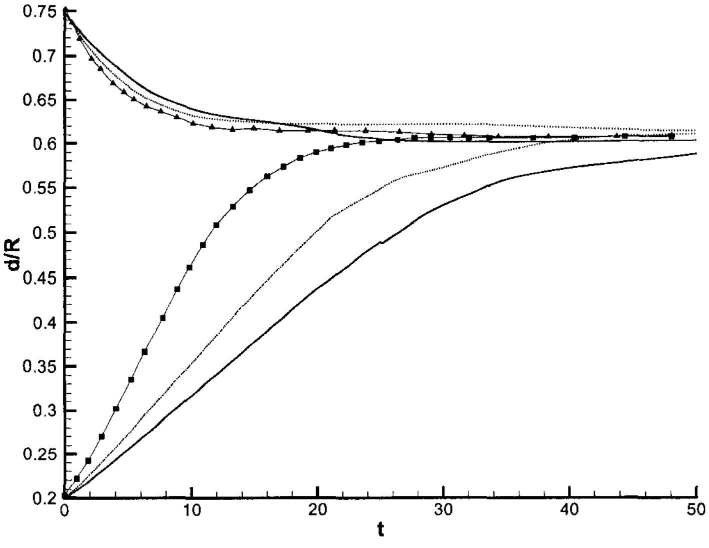
Both the simulations were carried out with particle meshing, which substantially improve the accuracy of the angular velocity and volume of the particle as already shown in chapter 4. While ALE method is accurate in defining the particle domain because of the fact that it uses adaptive mesh refinement by changing the computational grid, fictitious domain methods lack such accuracy due to the fixed mesh. Therefore, other ways of improving the accuracy must be sought if needed. Pan and Glowinski [42] use a grid of collocated points on the surface of particle to achieve this accuracy and we have employed dynamic mesh refinement to achieve the same.

In figure 5.8(a), the difference in trajectory with and without particle meshing is shown. The migration of the particle to the equilibrium position is related to the angular slip velocity of the particle in some sense [56]. The angular slip velocity is defined as the difference of particle and fluid angular velocities. The difference of angular slip velocity at any radial position and the equilibrium position is proposed to be the driving force that causes the migration of the particle. This is found to be true at least for lower Reynolds numbers while for higher Reynolds numbers the analysis is complicated due to the presence of multiple equilibrium positions [56]. The consequence is that if the driving force is not calculated accurately then the particle can stop migrating at an intermediate position before reaching the equilibrium position. We think that this could explain why the particle stops migrating when no particle meshing is used.

In figures 5.9(a)-5.10(b), the translational and angular velocities of the particle are compared. These figures further support the above hypothesis. In general, one can infer from the figures that the migration of particle from the wall to the equilibrium position is much more accurately solved as compared to the migration from the centerline to the equilibrium position. The velocity gradients near the wall is much higher for Poiseuille flow profile as compared to near the center of the channel. Higher gradients means higher angular velocity for the particle (see table 5.8). The relative error (or noise to signal ratio) in near wall driving force is consequently much lower as compared to error in driving force near the center-

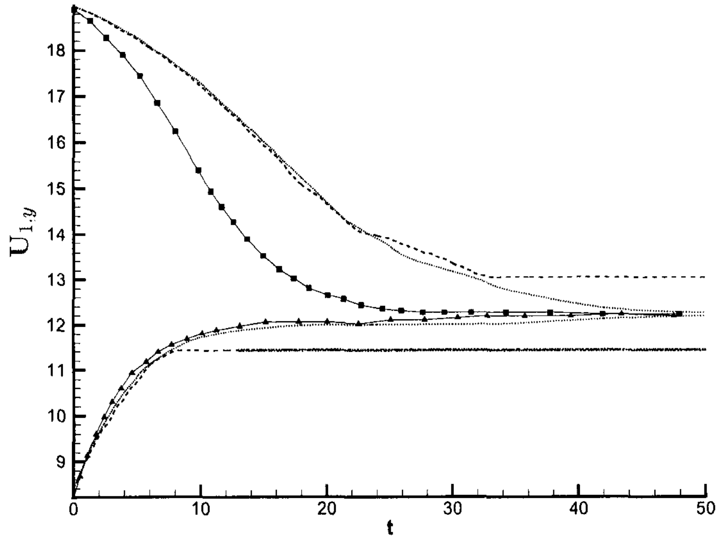


(a) The effect of particle meshing: with (dotted) and without (dashed) particle meshing. $dt = 0.005$

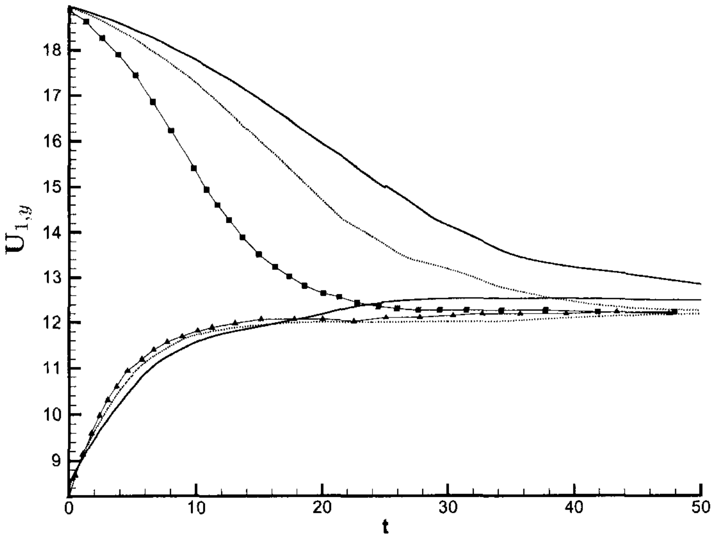


(b) The effect of time step size: $dt = 0.01$ (solid) and $dt = 0.005$ dotted

Figure 5.8: The radial position of the particle as a function of time for starting positions 0.2 and 0.75. The connected symbols are results of [42].

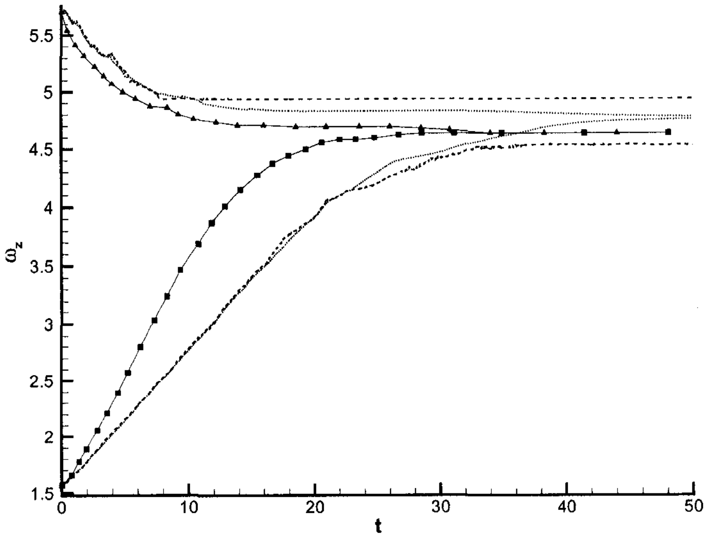


(a) The effect of particle meshing: with (dotted) and without (dashed) particle meshing. $dt = 0.005$

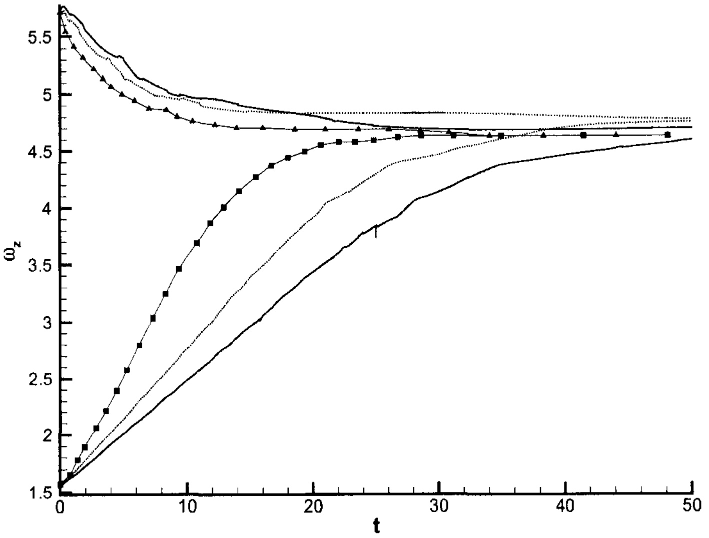


(b) The effect of time step size: $dt = 0.01$ (solid) and $dt = 0.005$ dotted

Figure 5.9: The translational velocity of the particle as a function of time for starting positions 0.2 and 0.75. The connected symbols are results of [42].



(a) The effect of particle meshing: with (dotted) and without (dashed) particle meshing. $dt = 0.005$



(b) The effect of time step size: $dt = 0.01$ (solid) and $dt = 0.005$ dotted

Figure 5.10: The angular velocity of the particle as a function of time for starting positions 0.2 and 0.75. The connected symbols are results of [42].

line. Better driving force resolution therefore translates into better calculation of particle trajectory.

5.3 Conclusion

Validation of the algorithm using two fundamentally different physical experiments of single particle motion inside a fluid is done. The first validation case considered is that of the settling of a single particle in a closed box at four different Reynolds numbers and the second case is that of a migrating neutrally buoyant particle in Hagen-Poiseuille flow in a circular channel. For the first case, the entire trajectory of the particle from initial acceleration, terminal settling and final deceleration near the bottom wall is compared with experimental data available in the literature. The terminal settling velocity predicted by the numerical simulations is within 7% accuracy for the lowest Reynolds number and within 2% accuracy for the highest Reynolds number tested. The convergence in time is shown using three different time step sizes. The convergence in space is also attempted using two different mesh resolutions, however grid independence could not be concluded. For the second case, the entire transient migration of a neutrally buoyant particle towards and away from the wall of a circular channel is tested. Two different time step sizes are used to capture the trajectory as accurately as possible and are compared to the simulations reported in the literature using other methods. With smaller time steps closer agreement to other simulations is observed. The equilibrium position of the particle is well known to be at 60% of the radius of the channel away from the axis. The results from the simulations are accurate to within 3% of this experimental value.

Chapter 6

Collision Modelling

6.1 Introduction

With multiple particles the inevitable event of collision between particles can not be ignored. The accuracy with which the collision is modelled can change the way multiple particles behave in fluid. The many body problem is quite challenging due to the time and space resolution required to resolve the collision process.

Generally, when two particles collide in a sparse inviscid medium like air, the time duration of the entire collision process is only a few microseconds (0.01 ms [21]). The trajectory of the particles before and after the collision can be accurately calculated based on conservation of momentum and energy laws. On the other hand, inside a viscous medium one can encounter a variety of situations depending on the inertia of the particle and the inertia and viscosity of the fluid. The ratio of the fluid inertial force to viscous force is nicely captured by the Reynolds number. Now enter the particle inertia, we need to define a new dimensionless parameter called Stokes number, which is the product of density ratio of the particle to the fluid and the Reynolds number. In short, it captures the ratio of particle inertia to fluid viscous force. The hydrodynamic radius of influence of a single particle can stretch for hundreds of diameters in a sufficiently viscous medium and the hydrodynamic interaction force due to the approaching particles start acting well ahead of actual contact. Therefore, an accurate contact time is difficult to determine. Similarly, the hydrodynamic interaction of settling particle towards a plane wall [6] and along side a vertical wall is well known [24]. If the Stokes number is quite low, the momentum of the particles could be completely dissipated by the viscous force of the intervening fluid. The particles never come in contact.

In this case, a Stokesian dynamics based model best estimates the motion of the fluid. The hydrodynamic force and torque of the fluid on the particles is used to determine their further course [27]. While this kind of modelling works well for smooth particles, rough particles on the other hand may actually come in contact due to their surface perturbations [4, 50]. At sufficiently higher Stokes numbers, the particles can actually come into contact because the fluid is not able to completely dissipate their momentum and collision may unfold in a milder form as it occurs in inviscid medium.

An interesting explanation of the sequence of events can be provided by the characteristic time scales involved in the process. One time scale is the time that the particle takes to travel a characteristic length of one diameter and this is generally refer to as convection time scale. Another time scale is the time that the fluid takes to drain from between the particles which depends on viscosity of the fluid and hence this time scale can be referred to as diffusion time scale. There is no reason why the fluid should drain faster than the approaching particles because the draining of fluid is supposedly due to the approaching particles. So we can discard this possibility. Now, if the fluid drains as fast as the particles are approaching, then eventually there will be contact. On the other hand, if the fluid drains slower than the approaching particle then the pressure in the intervening fluid will increase and the particle will be decelerated (in lubrication limit to zero velocity) or if the particle can deform, it will deform storing part of its kinetic energy as potential energy due to its elasticity. When this energy is released it will rebound. In the present case we don't consider deformation of the particles because of the rigid body assumption made for our numerical model. However it is worth taking a note that the elastohydrodynamic theory proposed by Davis et al. [11] is based on the fact that the deformation energy stored in the particle due to the fluid pressure is responsible for rebound.

In between the extremes of the Stokesian regime where the fluid governs the dynamics and rigid body dynamics where particles have dominant role to play, a range of possibilities exist. One has to choose the appropriate model depending on the range of Stokes and Reynolds number one is working at.

In this chapter, we discuss some of the attempts made at modelling this tough problem, however none of the approaches discussed below is a universal model for collision. In general, we allow the dynamics to be determined by the Navier-Stokes equation unless or until the gap between the particle surfaces or the particle surface and the boundary (wall) is less than the grid spacing. First, we will look at the interaction of single particle with the wall by using a sub-grid lubrication force as suggested by ten Cate et al. [51]. A similar force based on stokesian dynamics can be applied between the particles [38]. Then collision modelling based on rigid body dynamics [19] with correction for presence of fluid [35, 57] is explored.

6.2 Sub-grid lubrication model

6.2.1 For particle wall interaction

As the name suggests this force is required only when the particle surface lies in the gap between the last node above the bottom wall boundary and the bottom wall itself. Since, we don't have any intervening nodes to solve Navier-Stokes equation for the fluid; we need to supplement the lubricating force of fluid in the gap. The form of the lubrication force needed is discussed by Nguyen and Ladd [38] for their lattice-Boltzmann simulation of particle-particle and particle-wall interaction and applied to the low Reynolds number settling particle case by ten Cate et al. [51]. The basic assumption is that when two surfaces (of rigid bodies) approach each other closely, the flow can be treated as low-Reynolds number Stokes flow. Consequently, the sub-grid lubrication force is written as follows:

$$\mathbf{F}_W = -6\pi R_i \mathbf{U}_{\perp,i}\mu\left(\frac{R_i}{h} - \frac{R_i}{h_0}\right) \quad h < h_0 \tag{6.1}$$

where, R_i is the radius of the $i-$th particle, $U_{\perp,i}$ is the velocity of the particle, perpendicular to the bottom wall, μ is viscosity of the fluid, h is the gap between particle surface and the bottom wall and h_0 is the grid spacing of the base mesh. The force is based on Stokes drag and is valid only in the sub-grid gap. The negative sign signifies that the force is always in the direction opposite to that of particle velocity. Irrespective of whether the particle is moving towards or away from the wall the force is going to decelerate the particle. In order to derive the non-dimensional version of the force, we can start with the following linear equation of particle motion:

$$\Delta M_i \frac{\mathrm{d}\mathbf{U}_i}{\mathrm{d}t} = \mathbf{F}_W \tag{6.2}$$

The lubrication force acts on the buoyant mass of the particle because the particle is immersed in fluid. For experiments where free particle strikes a solid wall coated with a thin layer of fluid, the equation will use total mass of the particle.

The equation is discretized and expression for mass and lubrication force (6.1) is substituted to get:

$$\frac{\Delta \mathbf{U}_i}{\Delta t} = -9\frac{\mu}{2R_i(\rho_i - \rho_1)}\mathbf{U}_{\perp,i}\left(\frac{1}{h} - \frac{1}{h_0}\right) \tag{6.3}$$

The particle diameter can be used as the characteristic length scale and free settling velocity is used as the characteristic velocity scale to define the particle Reynolds Re as before. Retaining the same symbols in dimensionless form, the correction for velocity required for a time step size of Δt is given by:

$$\Delta \mathbf{U}_i = -\frac{9}{Re}\frac{1}{\left(\frac{\rho_i}{\rho_1} - 1\right)}\mathbf{U}_{\perp,i}\left(\frac{1}{h} - \frac{1}{h_0}\right)\Delta t \tag{6.4}$$

It is clear from this form of the equation that the artificial lubrication correction is strong at lower Reynolds numbers and weak at higher Reynolds numbers. The dimensionless grid spacing and time step size dependence are reminiscent of heuristic nature of the force. The velocity correction obtained from this expression is used in the predictor and the corrector steps for particle position (equations (3.44) and (3.56) respectively).

6.2.2 For particle-particle interaction

A similar lubrication force model can be derived for particle-particle interaction. The formulation of this force is also discussed in Nguyen and Ladd [38] and is as follows:

$$\mathbf{F}_P = -6\pi\mu\frac{R_1^2R_2^2}{(R_1 + R_2)^2}\left(\frac{1}{h} - \frac{1}{h_0}\right)(\mathbf{U}_1 - \mathbf{U}_2)\cdot\frac{\mathbf{r}_{12}}{|\mathbf{r}_{12}|} \quad h < h_0 \tag{6.5}$$

Here, R_1 and R_2 are radii of the interacting particles, $\mathbf{U}_1$ and $\mathbf{U}_2$ are particle velocities and $\mathbf{r}_{12}$ is the vector from center of particle 1 to center of particle 2. The dot product results in the projection of relative velocity along the line joining the centers. The force as before acts opposite to the relative motion to decelerate it. This equation can be non-dimensionalized in the same fashion as above, which finally leads to velocity correction for particle 1 and 2 as follows:

$$\Delta \mathbf{U}_1 = -\frac{9}{Re}\frac{1}{\left(\frac{\rho_{2,1}}{\rho_1} - 1\right)}\left(\frac{R_2}{R_1 + R_2}\right)^2(\mathbf{U}_1 - \mathbf{U}_2)\left(\frac{1}{h} - \frac{1}{h_0}\right)\Delta t \tag{6.6}$$

$$\Delta \mathbf{U}_2 = -\frac{9}{Re}\frac{1}{\left(\frac{\rho_{2,2}}{\rho_1} - 1\right)}\frac{R_1}{R_2}\left(\frac{R_1}{R_1 + R_2}\right)^2(\mathbf{U}_2 - \mathbf{U}_1)\left(\frac{1}{h} - \frac{1}{h_0}\right)\Delta t \tag{6.7}$$

Here, the domain subscript 2 explicitly indicates particle domain and 1 indicates fluid domain. The gap between the particle surfaces is denoted by h and h_0 is the grid spacing as before. The additional factor of $\frac{R_1}{R_2}$ appears in the second equation when we choose the characteristic length scale to be the diameter of particle 1. The expression for velocity correction for particle 1 reduces to that for velocity correction when it interacts with wall (6.4), if the particle 2 is the plane wall with large radius ($R_2 \rightarrow \infty$).

The lubrication force model being based on Stokesian dynamics, gives importance to resolving fluid motion. It works well when the inertial effects of particle and fluid are negligible i.e. only for low Reynolds number and low Stokes number limit. The model is effective in decelerating the particle to rest when interaction occurs between particles or between a particle and a wall. It has been found experimentally for higher Stokes numbers that particles can rebound from the wall or another particle. To capture this physics, rigid body dynamics based model would be more appropriate.

6.3 Stereomechanical impact based model

A two body interaction model is first developed for collision between two spheres based on the graphical approach of Goldsmith [19] for two body impact in three dimensions. Stereomechanical theory of impact is an abstract model that does not account for any stress waves originating at point of contact or any deformation of the impacting bodies. Since we model rigid bodies, the complication due to deformation is not involved. Further our particles are spherical; we need not bother about vibrations [19, pg.4]. Therefore this theory perfectly suits our purpose for modelling collision between two particles. The entire process of collision is analyzed in two phases: approach/compression phase and restitution phase. The extent of recovery after collision determines the conversion of energy stored in the body during compression phase of collision to kinetic energy during restitution phase of collision. The energy loss is captured by defining the effective coefficient of restitution (e) as follows:

$$e = -\frac{\mathbf{U}_{r1} - \mathbf{U}_{r2}}{\mathbf{U}_{a1} - \mathbf{U}_{a2}} \tag{6.8}$$

where, $\mathbf{U}_{r1}$ and $\mathbf{U}_{r2}$ are velocities after collision and $\mathbf{U}_{a1}$ and $\mathbf{U}_{a2}$ are velocities before collision for particles 1 and 2 respectively. $e = 1$, means complete recovery or elastic collision and $e = 0$, means no recovery or plastic collision i.e. the particles stick after collision. In general, this definition is applied only to the component of the velocities that is normal to contacting surfaces. To resolve tangential components the coefficient of friction (f) is introduced. The values of these parameters depend on the materials that are taking part in the collision and therefore must be determined from experiments. This is the major drawback of this model. The submerged particle collision experiments of Yang and Hunt [57] provide some experimental correlations for normal coefficient of restitution as a function of impact angles in three different ranges of Stokes number and the submerged pendulum experiments of Joseph et al. [34], Joseph and Hunt [33] provide values for frictional coefficients. However, these experimental results do not cover the entire range of impact angles $0 - 90^\circ$ or the entire range of Stokes numbers. There are difficulties in finding coefficients of restitution at lower range of Stokes number stemming from the fact that this range belongs to the dynamics of fluid. Yang and Hunt [57] have found in their experiments that the target sphere starts to move before the collision, making the estimation of coefficient of restitution obtained in the low Stokes number regime unreliable. The elastohydrodynamic theory developed by Davis et al. [11] has been shown to fit the experimental data by Kantak and Davis [35] fairly well. They arrived at a simple relationship betweem coefficient of restitution and Stokes number as follows:

$$e_{wet} = \begin{cases} e_{dry}\left(1 - \frac{St_c}{St}\right) & St > St_c \\ 0 & St < St_c \end{cases} \tag{6.9}$$

The value of critical Stokes number is dependent on material properties. In general, the critical Stokes number is found to be 10 [34] and e_{dry} values for various materials have been compiled by Gondret et al. [21] based on the bouncing height of various balls on a flat plane. The Stokes number is defined as:

$$St_n = \frac{m^*(\mathbf{U}_1 - \mathbf{U}_2)}{6\pi\mu R^{*2}} \tag{6.10}$$

Here, $m^* = \left(\frac{1}{m_1} + \frac{1}{m_2}\right)^{-1}$ is the reduced mass of the two interacting bodies, similarly $R^* = \left(\frac{1}{R_1} + \frac{1}{R_2}\right)^{-1}$ is the reduced radius. In terms of dimensionless quantities the Stokes number can be written as:

$$St_n = \frac{1}{9} Re U_n \frac{\rho_{2,1}}{\rho_1} \frac{(1 + \frac{R_1}{R_2})^2}{(1 + \frac{m_1}{m_2})} \tag{6.11}$$

Here, Re is the particle Reynolds number defined as $\frac{2R_1\rho_1 U_c}{\mu}$ when diameter of particle 1 is taken as characteristic length scale and U_c is some characteristic velocity scale. U_n is the dimensionless relative approach velocity of the particles or a particle towards wall. For collision with a massive plane wall the factors with radius and mass ratio drops out.

6.3.1 Two sphere impact in three dimensions

We follow the graphical procedure outlined in Goldsmith [19] to obtain final velocities of colliding spheres based on their initial velocities. The figure 6.1 shows the velocities involved in three dimensional impact of two spheres. With reference

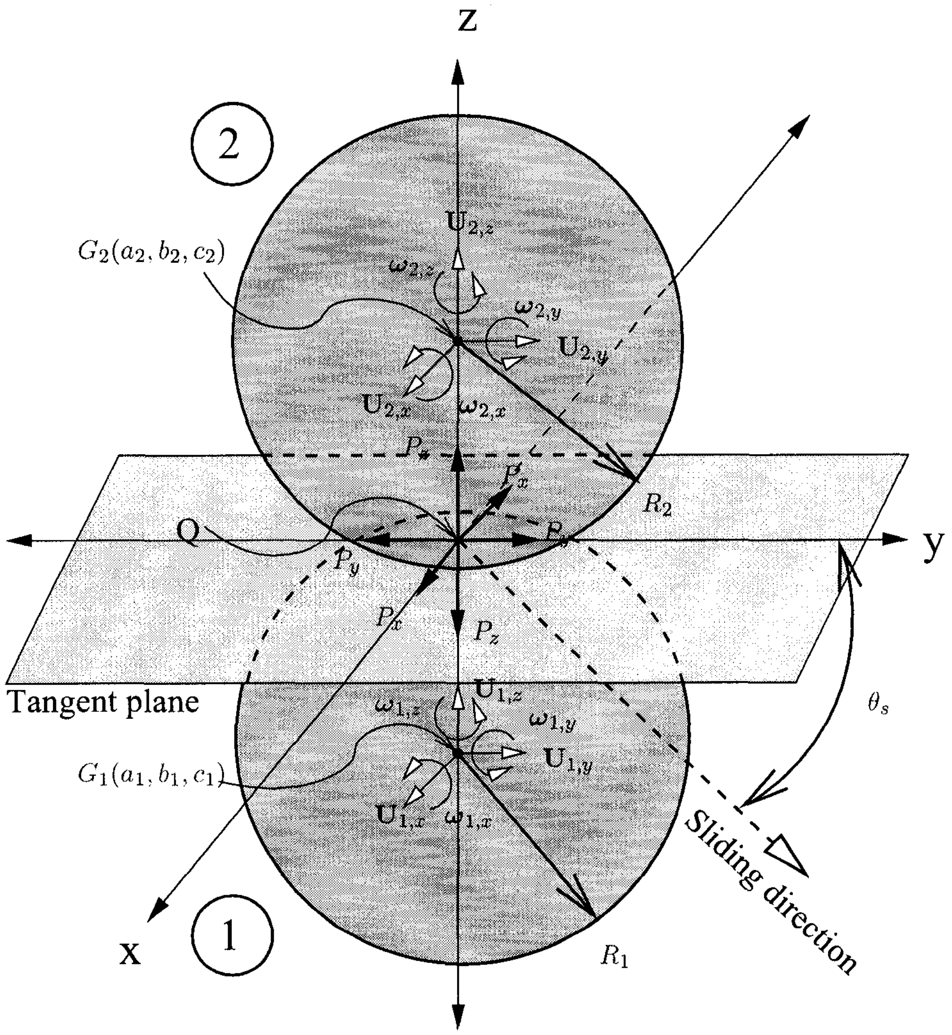


Figure 6.1: Impact of two spheres in three dimensional motion

to the figure, the linear and angular momentum equations can be written as:

$$m_1\left(\mathbf{U}_{1,x}-\mathbf{U}_{1,x0}\right)=-\mathbf{P}_x \qquad m_2\left(\mathbf{U}_{2,x}-\mathbf{U}_{2,x0}\right)=\mathbf{P}_x \tag{6.12}$$

$$m_1\left(\mathbf{U}_{1,y}-\mathbf{U}_{1,y0}\right)=-\mathbf{P}_y \qquad m_2\left(\mathbf{U}_{2,y}-\mathbf{U}_{2,y0}\right)=\mathbf{P}_y \tag{6.13}$$

$$m_1\left(\mathbf{U}_{1,z}-\mathbf{U}_{1,z0}\right)=-\mathbf{P}_z \qquad m_2\left(\mathbf{U}_{2,z}-\mathbf{U}_{2,z0}\right)=\mathbf{P}_z \tag{6.14}$$

$$\frac{2}{5}m_1R_1^2\left(\omega_{1,x}-\omega_{1,x0}\right)=R_1\mathbf{P}_y \qquad \frac{2}{5}m_2R_2^2\left(\omega_{2,x}-\omega_{2,x0}\right)=R_2\mathbf{P}_y \tag{6.15}$$

$$\frac{2}{5}m_1R_1^2\left(\omega_{1,y}-\omega_{1,y0}\right)=-R_1\mathbf{P}_x \qquad \frac{2}{5}m_2R_2^2\left(\omega_{2,y}-\omega_{2,y0}\right)=-R_2\mathbf{P}_x \tag{6.16}$$

$$\frac{2}{5}m_1R_1^2\left(\omega_{1,z}-\omega_{1,z0}\right)=0 \qquad \frac{2}{5}m_2R_2^2\left(\omega_{2,z}-\omega_{2,z0}\right)=0 \tag{6.17}$$

It is assumed that spheres can freely pivot around the contact point, so there is no change is angular velocity in normal direction. One can discard this assumption provided we know how to model the spin coefficient of friction. Otherwise, the relative velocities of body 1 with respect to body 2 are:

$$\begin{bmatrix} S\cos\theta_s \\ S\sin\theta_s \\ C \end{bmatrix} = \begin{bmatrix} \mathbf{U}_{1,x}-\mathbf{U}_{2,x}+R_2\omega_{2,y}+R_1\omega_{1,y} \\ \mathbf{U}_{1,y}-\mathbf{U}_{2,y}-R_2\omega_{2,x}-R_1\omega_{1,x} \\ \mathbf{U}_{1,z}-\mathbf{U}_{2,z} \end{bmatrix} \tag{6.18}$$

The initial sliding velocities S_0, C_0 and sliding direction θ_0 are determined by putting initial velocities in above equations. The sliding velocities in terms of impulses are obtained as follows:

$$S_0\cos\theta_0 - S\cos\theta_s = \frac{7}{2}\left(\frac{m_1+m_2}{m_1m_2}\right)\mathbf{P}_x \tag{6.19}$$

$$S_0\sin\theta_0 - S\sin\theta_s = \frac{7}{2}\left(\frac{m_1+m_2}{m_1m_2}\right)\mathbf{P}_y \tag{6.20}$$

$$C_0 - C = \left(\frac{m_1+m_2}{m_1m_2}\right)\mathbf{P}_z \tag{6.21}$$

The planes for no-sliding and maximum compression are:

$$S=0:\mathbf{P}_x=\frac{2}{7}\left(\frac{m_1m_2}{m_1+m_2}\right)S_0\cos\theta_0 \tag{6.22}$$

$$\mathbf{P}_y=\frac{2}{7}\left(\frac{m_1m_2}{m_1+m_2}\right)S_0\sin\theta_0 \tag{6.23}$$

$$C=0:\mathbf{P}_z=\left(\frac{m_1m_2}{m_1+m_2}\right)C_0 \tag{6.24}$$

In figure 6.2, we show the 3 dimensional version of the image point trajectory in impulse space when the angle of sliding is fixed as θ_0. The planes of no-sliding and maximum compression are also shown. The plane for termination of impact is added parallel to $C = 0$ plane. The normal impulse at termination of impact will be $(1 + e)$ times the normal impulse at the maximum compression plane. This relationship is result of the way coefficient of restitution (e) is defined. The

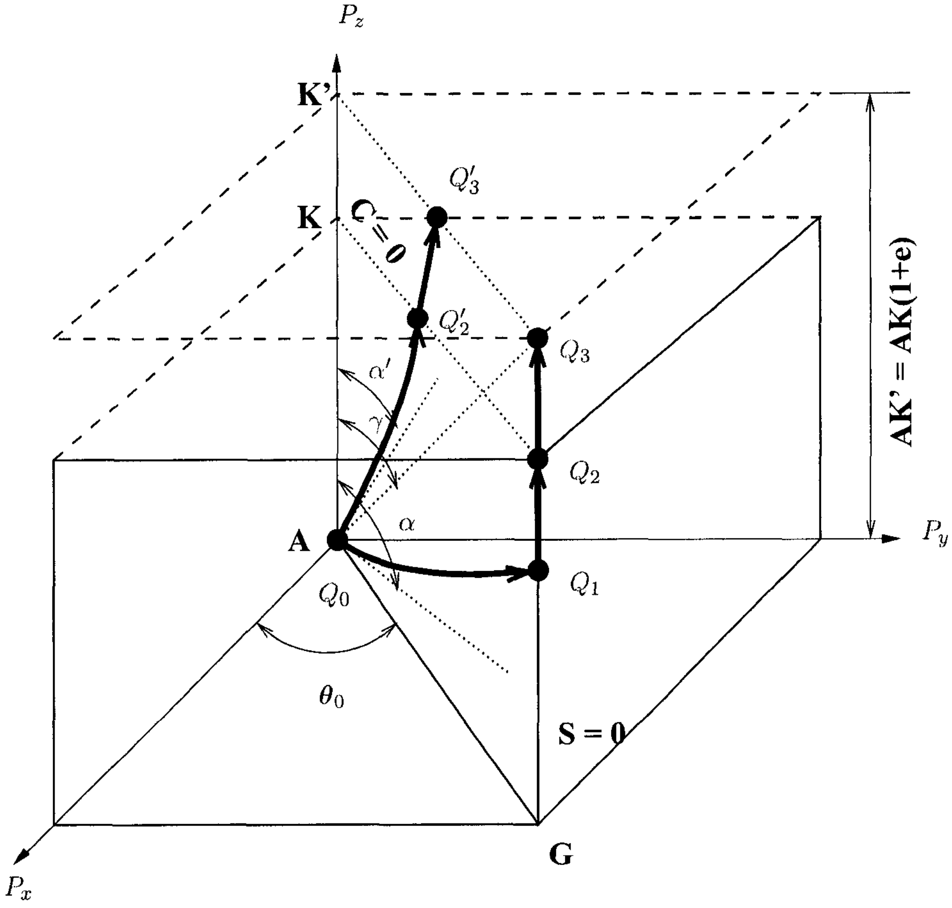


Figure 6.2: Motion of image point in three dimensional impact between two spheres

relevant lengths are:

$$AG = \frac{2}{7}\left(\frac{m_1 m_2}{m_1 + m_2}\right) S_0 \tag{6.25}$$

$$AK = \left(\frac{m_1 m_2}{m_1 + m_2}\right) C_0 \tag{6.26}$$

$$AK' = \left(\frac{m_1 m_2}{m_1 + m_2}\right) C_0 \, (1 + e) \tag{6.27}$$

The critical angle γ is given by:

$$\tan\gamma = \frac{AG}{AK'} = \frac{2S_0}{7C_0\,(1+e)} \tag{6.28}$$

The impulses at the image point Q, follow the curve defined by frictional impulses available. This curve will be a straight line in our case because we are assuming constant coefficient of friction. If friction is assumed to act at limiting value initially, then the path from Q_0 to Q_1 is described by the line:

$$\frac{\mathrm{d}P_x}{\cos\theta_s} = \frac{\mathrm{d}P_y}{\sin\theta_s} = \frac{\mathrm{d}P_z}{1/f} \tag{6.29}$$

Depending on the value of friction angle (α), the following two situations can arise:

a) $f = \tan\alpha > \tan\gamma$: Friction is acting at limiting value right from the start of sliding. The image point follows the curve described by equation (6.29) from Q_0 to point Q_1 which lies on the no-sliding line. Then it reaches maximum compression plane at Q_2 and finally termination of impact at Q_3. The impulses at the termination point are:

$$\mathbf{P}_{x,3} = \frac{2}{7}\left(\frac{m_1 m_2}{m_1+m_2}\right) S_0\cos\theta_0 \tag{6.30}$$

$$\mathbf{P}_{y,3} = \frac{2}{7}\left(\frac{m_1 m_2}{m_1+m_2}\right) S_0\sin\theta_0 \tag{6.31}$$

$$\mathbf{P}_{z,3} = \left(\frac{m_1 m_2}{m_1+m_2}\right) C_0\,(1+e) \tag{6.32}$$

The final velocities at termination can be obtained by substituting the above impulses into impulse-momentum equation. For particle-1 we get:

$$\mathbf{U}_{1,x} = \mathbf{U}_{1,x0} - \frac{2}{7(1+M)}(S_0\cos\theta_0) \tag{6.33}$$

$$\mathbf{U}_{1,y} = \mathbf{U}_{1,y0} - \frac{2}{7(1+M)}(S_0\sin\theta_0) \tag{6.34}$$

$$\mathbf{U}_{1,z} = \mathbf{U}_{1,z0} - \frac{(1+e)}{(1+M)}C_0 \tag{6.35}$$

$$\omega_{1,x} = \omega_{1,x0} + \frac{5}{7R_1(1+M)}(S_0\sin\theta_0) \tag{6.36}$$

$$\omega_{1,y} = \omega_{1,y0} - \frac{5}{7R_1(1+M)}(S_0\cos\theta_0) \tag{6.37}$$

$$\omega_{1,z} = \omega_{1,z0}, \tag{6.38}$$

and for particle-2 as follows:

$$\mathbf{U}_{2,x} = \mathbf{U}_{2,x0} + \frac{2M}{7(1+M)}(S_0\cos\theta_0) \tag{6.39}$$

$$\mathbf{U}_{2,y} = \mathbf{U}_{2,y0} + \frac{2M}{7(1+M)}(S_0\sin\theta_0) \tag{6.40}$$

$$\mathbf{U}_{2,z} = \mathbf{U}_{2,z0} + \frac{M(1+e)}{(1+M)}C_0 \tag{6.41}$$

$$\omega_{2,x} = \omega_{2,x0} + \frac{5M}{7R_2(1+M)}(S_0\sin\theta_0) \tag{6.42}$$

$$\omega_{2,y} = \omega_{2,y0} - \frac{5M}{7R_2(1+M)}(S_0\cos\theta_0) \tag{6.43}$$

$$\omega_{2,z} = \omega_{2,z0} \tag{6.44}$$

Here, $M = m_1/m_2$ is the ratio of the particle masses.

b) $f = \tan\alpha' < \tan\gamma$: The image point follows the curve described by equation (6.29) from Q_0 to Q_2 where the approach phase ends and continues to termination of impact at Q_3. The impulses at termination are:

$$\mathbf{P}_{z,3} = \left(\frac{m_1 m_2}{m_1 + m_2}\right) C_0 (1+e) \tag{6.45}$$

$$\mathbf{P}_{x,3} = f\cos\theta_0 \left(\frac{m_1 m_2}{m_1 + m_2}\right) C_0 (1+e) \tag{6.46}$$

$$\mathbf{P}_{y,3} = f\sin\theta_0 \left(\frac{m_1 m_2}{m_1 + m_2}\right) C_0 (1+e) \tag{6.47}$$

It is assumed that the sliding direction is constant θ_0. The final velocities in this case are given for particle-1 by:

$$\mathbf{U}_{1,x} = \mathbf{U}_{1,x0} - f\cos\theta_0 \frac{1+e}{1+M}C_0 \tag{6.48}$$

$$\mathbf{U}_{1,y} = \mathbf{U}_{1,y0} - f\sin\theta_0 \frac{1+e}{1+M}C_0 \tag{6.49}$$

$$\mathbf{U}_{1,z} = \mathbf{U}_{1,z0} - \frac{1+e}{1+M}C_0 \tag{6.50}$$

$$\omega_{1,x} = \omega_{1,x0} + f\sin\theta_0 \frac{5\,(1+e)}{2R_1\,(1+M)}C_0 \tag{6.51}$$

$$\omega_{1,y} = \omega_{1,y0} - f\cos\theta_0 \frac{5\,(1+e)}{2R_1\,(1+M)}C_0 \tag{6.52}$$

$$\omega_{1,z} = \omega_{1,z0}, \tag{6.53}$$

and for particle-2 by:

$$\mathbf{U}_{2,x} = \mathbf{U}_{2,x0} + f\cos\theta_0 \frac{M\,(1+e)}{1+M} C_0 \tag{6.54}$$

$$\mathbf{U}_{2,y} = \mathbf{U}_{2,y0} + f\sin\theta_0 \frac{M\,(1+e)}{1+M} C_0 \tag{6.55}$$

$$\mathbf{U}_{2,z} = \mathbf{U}_{2,z0} + \frac{M\,(1+e)}{1+M} C_0 \tag{6.56}$$

$$\omega_{2,x} = \omega_{2,x0} + f\sin\theta_0 \frac{5M\,(1+e)}{2R_2\,(1+M)} C_0 \tag{6.57}$$

$$\omega_{2,y} = \omega_{2,y0} - f\cos\theta_0 \frac{5M\,(1+e)}{2R_2\,(1+M)} C_0 \tag{6.58}$$

$$\omega_{2,z} = \omega_{2,z0} \tag{6.59}$$

6.4 Numerical Implementation

In the last two sections we develop collision models based on the two extreme cases: either lubrication theory or the stereomechanical impact theory. In most practical situations, neither of the two theories is completely applicable. Here we consider in detail the case of a settling particle when it is close to the bottom wall. In chapter 5, we had already employed the lubrication based model to good accuracy when the settling particle within the sub-grid distance of the bottom wall (See figure 6.3). While the particle is well inside the fluid domain, the solution of the Navier-Stokes equation with boundary conditions of stationary wall and moving particle surface can capture the deceleration of the particle. The difficulty with finite grids is when particle surface gets in between the penultimate node before wall and the boundary node on the wall. At this point we had employed the approximate lubrication force given by equation (6.1). A major drawback of this method is that it is not able to simulate rebounding particles. In this section we develop a numerical procedure that uses the lubrication force and the stereomechanical impact model in a sequence to predict the correct rebound velocity if rebounce occurs.

6.4.1 The collision mechanism

The collision mechanism has two functions to perform; first is detection of collision and second correcting particle velocities in case of collision. At every time step, the collision detection function checks for collision between any two particles at the position predicted by (3.44) and again after the final particle positions are

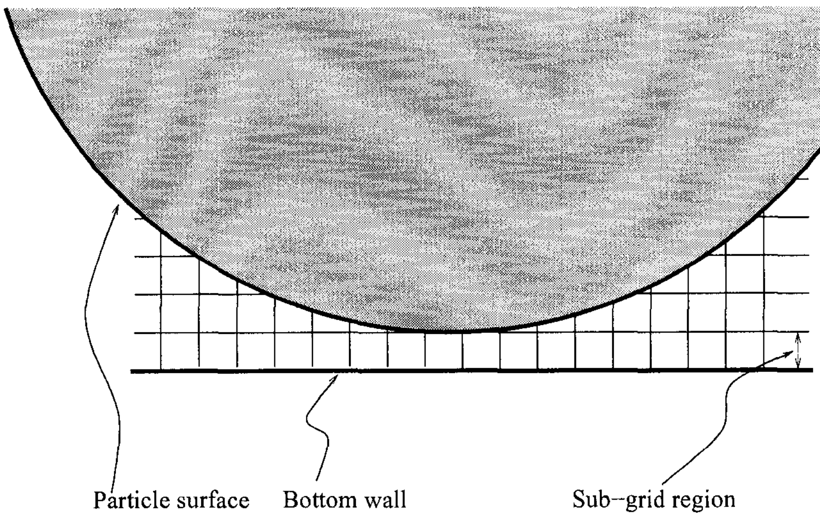


Figure 6.3: Illustration of the sub-grid region where lubrication force is dominant

updated by (3.56). This ensures that at the completion of every time step, no particle domains are overlapping. The collision detection declares collision under the following conditions:

- The gap between the colliding surfaces is taken as:

  $$h_0 = \begin{cases} |\mathbf{X}_1 - \mathbf{X}_2| - (R_1 + R_2), & \text{collision with particle} \\ |\mathbf{X}_1 - \mathbf{X}_w| - R_1, & \text{collision with wall} \end{cases}$$

  where $\mathbf{X}_w$ is the point on the wall closest to the particle, in other words the value of $|\mathbf{X}_1 - \mathbf{X}_w|$ can be obtained by distance of a point (in this case center of the particle) to plane formula. If the gap is less than one grid spacing then the particle(s) are considered for collision correction.

- A first order accurate prediction of the position of particles is made using current particle velocities at next time step and if the gap at these predicted positions is negative, then the particles are considered for collision correction in current time step itself.

The collision detection function returns whether the particle under consideration is colliding with another particle or bounding walls or both. Only in case of collision, the correction of particle velocities is applied following the steps below:

- A local co-ordinate system is defined at the point of contact as shown in figure 6.1. The z-axis or the axis of compression is specified along the normal from particle-1 to particle-2. There are infinitely many ways to specify other two axes in the tangent plane. We resolve the velocity of the first particle along the normal and perpendicular to this normal. This perpendicular direction is taken as x-axis and the y-axis is taken perpendicular to both x and z-axes. This means that particle-1 will never have any velocity component along the y-axis.

- All the translational and angular velocities of both particles are transformed to this local co-ordinate system.
- Now the sliding and compression velocities are found using (6.18). The magnitude of sliding and compression velocities is calculated and the angle of sliding is calculated.
- Next the Stokes number as defined by (6.11) and the value of coefficient of restitution as defined by (6.9) is calculated.
- In the sub-grid region the dominant force that is acting on the particle, before actual contact, is the lubrication force. We assume that once the collision is detected, the sub-grid lubrication force comes into play and continues to decelerate the particle until the contact is made when gap is equal to surface roughness of the particle and/or bottom wall. Note if there is no surface roughness then we run into mathematical difficulty due to lubrication force becoming singular. In order to determine the trajectory of the particle in this sub-grid space the following set of ODEs is solved.

$$\frac{\mathrm{d}h}{\mathrm{d}t} = -v_c \tag{6.60}$$

$$m\frac{\mathrm{d}v}{\mathrm{d}t} = -F_L \tag{6.61}$$

where, $v_c = U_z$ is the compression velocity and the expression for lubrication force F_L is given by Zhang et al. [61] as follows:

$$F_L = \frac{3}{2}\pi\mu R^2 v_c \left(\frac{1}{h} - \frac{1}{h+R}\right) K_1(h)$$

where $K_1(h)$ is the shape correction factor given by

$$K_1(h) = 1.041 - 0.281\ log_{10}\left(\frac{h}{R}\right) - 0.035\ log_{10}^2\left(\frac{h}{R}\right)$$

The above set of equations can be condensed into one equation as follows:

$$\frac{\mathrm{d}v_c}{\mathrm{d}h} = \frac{9}{8}\left(\frac{\mu}{\rho_2 R}\right) K_1(h) \left(\frac{1}{h} - \frac{1}{h+R}\right)$$

In dimensionless form this equation is

$$\frac{\mathrm{d}v_c'}{\mathrm{d}h'} = \frac{9}{4}\frac{\rho_1}{\rho_2}\frac{1}{Re} K_1(2h') \left(\frac{1}{h'} - \frac{1}{h'+0.5}\right) \tag{6.62}$$

This equation is solved using four stage Runge-Kutta method starting with the velocity with which particle enters the sub-grid region as initial condition at $h' = h_0$. A constant step size of dh' is used and time to cover this distance is estimated as

$$dt = \frac{2dh'}{(v_{c,i+1}' + v_{c,i}')}$$

The integration is continued until the gap is equal to surface roughness of the particle. At this point contact is assumed. The total time during this period is stored as say t_1. Now the stereomechanical impact model is applied using the final velocity given by the lubrication model.

- To determine whether the friction acts at its limiting value, the critical angle as defined in (6.28) is calculated.

- If the friction coefficient $f = \tan\alpha > \tan\gamma$, the set of equations (6.33)-(6.38) and (6.39)-(6.44) is applied. Otherwise, the second set (6.48)-(6.53) and (6.54)-(6.59) is applied.

- After impact the particle is still in contact and is now about to leave the sub-grid region. Again lubrication force comes into play to decelerate the particle. Therefore, the equation (6.62) is solved again to determine the final rebound velocity and the time that the particle takes to leave the sub-grid region is say t_2. Then the total impact time is taken as $T_i = t_1 + t_2$.

The collision with wall is treated similarly, with the assumption that the wall is massive i.e. $m_2 \rightarrow \infty$ which implies putting $M = 0$ in all the expressions. The entire collision detection and velocity correction procedure is implemented inside a loop with sub-time stepping to afford better collision detection. The collision model gives us the impulse that acts on the particle during impact. This impulse is a delta force that must be incorporated into the fluid equation as $\mathbf{F}_i^c$ as previously pointed out in chapter 3.

6.4.2 Formulation for collision

In the present formulation, the collision impulse force enters the particle equation (3.28) as follows:

$$\rho_i V_i \frac{\mathrm{D}\mathbf{U}_i}{\mathrm{D}t} = (\rho_i - \rho_1) V_i \mathbf{g} + \mathbf{F}_i + \rho_i V_i \frac{\mathrm{D}W_i}{\mathrm{D}t}, \quad \forall i = 1, \cdots, P_N \tag{6.63}$$

where the velocity of the particle before and after the collision and the contact time T_i is used to evalute the impulsive acceleration term (See §6.5.3 for an example calculation). The derivation with this additional term follow the same sequence of steps already detailed in §3.3.1. The term finally ends up in the final interaction force which now reads

$$\mathfrak{F} = \begin{cases} \frac{1}{Fr}\mathbf{e}_g + \left(\frac{\rho_1}{\rho_i - \rho_1}\right)\mathbf{F} + \left(\frac{\rho_i}{\rho_i - \rho_1}\right)\frac{\mathrm{D}W_i}{\mathrm{D}t}, & \text{in } \Omega_i, \forall i = 1, \cdots, P_N \\ 0 & \text{in } \Omega_1 \end{cases} \tag{6.64}$$

and the fluid equation (3.38) now reads

$$\frac{\mathrm{D}\mathbf{u}}{\mathrm{D}t} = -\nabla P + \frac{1}{Re}\nabla^2\mathbf{u} + \left(\frac{\rho_i - \rho_1}{\rho_1}\right)(\mathbf{G} - \mathfrak{F}) + \left(\frac{\rho_i}{\rho_1}\right)\frac{\mathrm{D}W_i}{\mathrm{D}t} \quad \text{in } \Omega$$

The additional term is added to the right hand side vector when we solve the Stokes problem equation (3.42), but only for the timestep at which collision occurs. Since this addition force is a singular force causing abrupt reversal in the fluid flow it represents a discontinuity in the time stepping scheme. The pre-collision time step fluid velocity field and the particle velocities can not be used to determine future velocities. Therefore, the calculations are restarted at this point using the particle velocities predicted by the collision model and the flow field that we get by solving Stokes problem with the impulse force on the right hand side, as the initial guess. We perform one first order time step before continuing with second order scheme. The convection term must ideally be resolved using an implicit method. But as an approximation, previous time step velocity field with particle velocity imposed on the particle domain is used when restarting the calculations.

6.5 Validation of bouncing particle

6.5.1 Experimental data

The experimental results of bouncing steel ball on glass surface as reported by Gondret et al. [21] is used for validation. The fluid medium used in the experiments vary and a variety of materials were employed for the settling particle.

Unfortunately, the release height have not been reported for most of the experimental runs. Therefore, only those cases can be used for which the impact occurs after the settling particle has reached terminal settling velocity. The case of steel ball impacting with a glass wall is used to demonstrate the capability of the new collision model to capture bouncing motion of the steel ball.

The diameter of the steel ball used is $d_p = 3\ mm$ and is allowed to settle in silicon oil of density $\rho_1 = 935\ kg/m^3$ and viscosity of $\mu = 0.01\ Pa.s$. The density of the steel ball is $\rho_2 = 7800\ kg/m^3$. Using the MATLAB code provided in appendix B.1 the dimensionless parameters are determined at the terminal settling velcity of the steel ball. We get Reynolds number $Re = 162.83$ based on the diameter of the particle, Froude number $Fr = 11.45$, Stokes number $St = 150.93$ and the terminal settling velocity as $U_\infty = 0.5805\ m/s$. These values closely agree with the data provided in [21]. In addition, they have also provided the coefficient of restitution to be $e = 0.78$ and an estimate of contact time equal to $0.01\ ms$, as measured by a piezoelectric sensor.

6.5.2 Numerical Set-up

The experiments were performed in a rectangular tank with dimensions of $10\ cm \times 10\ cm \times 30\ cm$ which when dimensionalized using the diameter of the particle $d_p = 3\ mm$ gives the dimensions to be used in numerical simulations as $34 \times 34 \times 100$. It is not known where the ball was released. Simulations in a tank of these dimensions turned out to be very expensive due to the large number of nodes required. It was found that there is hardly any influence of wall on the trajectory of the steel ball at $Re = 162.83$ and when ball is released at the height of 6 diameters from bottom, it has sufficient time to reach terminal settling velocity well before the bottom wall. Therefore, all the simulations are carried out in a tank of size $4 \times 4 \times 7$ particle diameters with particle released at 6 diameters from bottom. Two meshes with different resolutions were used as shown in figure 6.4.

The distribution of nodes is such that to resolve the particle trajectory near the wall and at the same time reduce the total number of nodes. The position of nodes for these meshes are given in table 6.1. The resulting meshes have 225199 and 813101 nodes respectively. The experiments are reported to be done in a open container, hence in the numerical simulations we treat the top surface as being open to atmosphere i.e. $P = 0$ which is equivalent to the stress free boundary condition in finite elements method and all other boundaries are implemented as

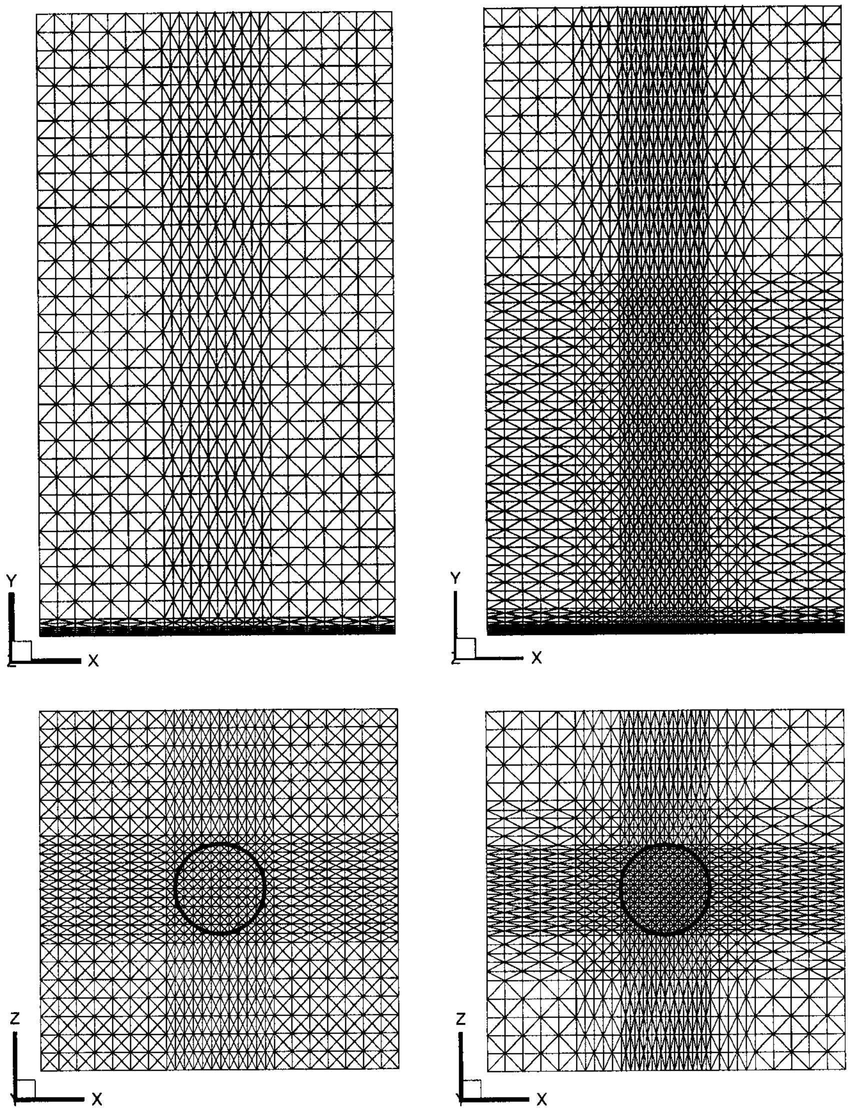


Figure 6.4: Comparison of coarse and fine meshes for bouncing particle validation

wall boundaries.

Table 6.1: Placement of nodes for bouncing particle simulation

Mesh type	Dimension	Placement
coarse	x and z	$-2.0 : 0.2 : -0.6$, $-0.5 : 0.1 : 0.5$, $0.6 : 0.2 : 2.0$
	y	0.0, 0.0025, 0.005, 0.01, 0.015, 0.02, 0.03, 0.05, 0.075, 0.1, 0.15, $0.2 : 0.2 : 7.0$
finer	x and z	$-2.0 : 0.2 : -1.0$, $-0.9 : 0.1 : -0.5$, $-0.45 : 0.05 : 0.45$, $0.5 : 0.1 : 0.9$, $1.0 : 0.2 : 2.0$
	y	$0.0 : 0.0025 : 0.01$, $0.015 : 0.005 : 0.03$, $0.04 : 0.01 : 0.06$, $0.08 : 0.02 : 0.12$, 0.15, 0.2, 0.25, 0.3, $0.4 : 0.1 : 4.0$, $4.2 : 0.2 : 7.0$

6.5.3 Comparison and Convergence

Unlike the experimental observation in [21] where the steel ball impacts the bottom wall with terminal settling velocity, in all our simulations we observe a reduction in velocity when the ball is very near the wall. The following sample excerpt from the output generated by one of the simulations gives good idea of the sequence of steps followed in the collision mechanism. All quantities are dimensionless and we simply point out what these numbers mean. First, we infer that the velocity with which the ball enters the sub-grid region is $U0[2] = -0.937814 = W_b$, it is negative since the particle is settling in gravity direction which is along negative y-axis. If there were no wall this value should be -1.0 i.e. equal to the terminal settling velocity.

```
================================================================
TP = 1111.0: collision type: 2
P = 1: X0 = [-6.37077e-06, 0.503399, 1.59577e-05]
U0 = [-1.32877e-05, -0.937814, 9.27029e-06]
W0 = [8.09729e-06, -9.22936e-06, 2.76478e-06]
P=1<-->W=3:
Compression velocity before lubrication: 0.937814
Gap before lubrication: 0.0024987
Compression velocity after lubrication: 0.929454
Gap after lubrication: 9.86986e-05
St = 140.282, Impact angle = 0.00117411 deg, e = 0.840949
```

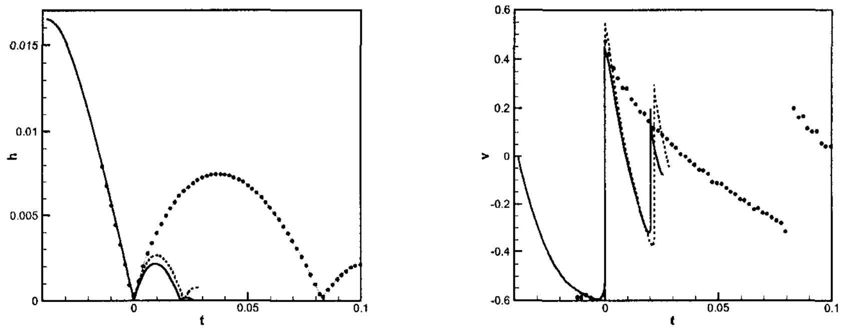
```
f = 0.02 => f > 3.18036e-06: Sufficient friction
Restitution velocity before lubrication: 0.781623
Gap before lubrication: 9.86986e-05
Restitution velocity after lubrication: 0.773166
Gap after lubrication: 0.0025987
Total contact time: 0.00578944
TP = 1111.100: last collision type: 2
P = 1: Xi = [-6.384e-06, 0.502537, 1.59669e-05]
Ui = [-9.46255e-06, 0.773166, 5.39964e-06]
Wi = [2.12959e-05, -9.22936e-06, -1.24533e-05]
================================================================
```

The gap at which lubrication comes into play is $0.0024987 = h_0$ and after lubrication the velocity is reduced to 0.929454. At impact, the Stokes number is 140.282 and then using the empirical relationship between e and St (6.9), we get $e = 0.84$. The velocity predicted by the collision model is 0.781623 and again the lubrication force acts until gap increases to 0.0025987 resulting in final rebound velocity of only $W_a = 0.773166$. The total contact time is $T_i \sim 0.0058$ which on multiplying with time scale $\frac{d_p}{U_\infty} = 0.0052\ s$ gives the real contact time as $\sim 0.03\ ms$. This quantity is of the same order as the contact time measured in experiments ($\sim 0.01\ ms$). The acceleration of the ball due to collision is obtained as follows:

$$\frac{\mathrm{D}W_i}{\mathrm{D}t} = \frac{W_a - W_b}{T_i} = 295.535$$

The contact time estimation is very crucial to get the correct impulse. It depends on the gap width and the velocity with which the particle enters the sub-grid region. As the gap width is decreased by introducing more nodes, the contact time will also decrease provided the entering velocity remains the same. But, the particle is decelerating due to lubrication and therefore reaches the sub-grid region with even lower velocity. Lower velocity implies higher contact time. Thus, according to this analysis there should be an asymptotic convergence towards the correct contact time as we improve the mesh resolution near the wall. Ultimately, one can do away with any empirical modeling if we can resolve the space near the wall right down to the surface roughness of the particle and/or bottom wall. While space resolution is one aspect of the problem, we must accurately capture the moment at which the particle enters the sub-grid region. This requires reducing the time step size or in other words improve the time resolution. Since there is a limit to which we can resolve space and time, we have to implement the collision model that is proposed here. A similar convergence towards the experimental data can be shown for the collision model to prove that it is consistent with the physical reality.

However, there is one more problem with the current numerical set-up, that of multiple length scales. As is evident from the distribution of nodes for meshes near wall, the smallest node spacing and therefore the element size (Δx) is 0.0025 (for second order element used in present work the node spacing between velocity nodes is 0.00125). If the magnitude of the velocity that is to be resolved is around ~ 1.0 (it will be higher in the squeezing flow of fluid near wall) then the time step size that can be used is of the order of $\Delta t \sim 0.001$ (or lesser) in order for the convection routines to work efficiently. In figure 6.5, the bouncing height and velocity obtained from numerical simulations on coarse and fine grids are compared with the experimental data in [21]. These results are far from showing any convergence. However, the collision model seems to predict the correct rebound velocity and it matches with the experimentally observed rebound velocity. Since we are not simulating the exact domain as in the experiments, we are not sure about the wall effects. However, these effects for slightly larger domain where checked and found to be negligible. Once again the correct validation depends on the right resolution for the mesh and the appropriate time step size. In future



(a) Bouncing height

(b) Bouncing Velocity

Figure 6.5: The numerical results (lines) for bouncing height and velocity of the steel ball is compared with experimental data (symbols) [21]. Coarse mesh result is shown in solid line and finer mesh result in dashed line.

more resolved mesh and smaller time steps should be attempted to investigate further convergence. Further detailed experiments in a smaller domain which can more exactly be replicated in numerical simulation should be performed.

6.6 Application of Collision Model

In this section, we have compiled a number of physical scenarios where the collision model has been tested qualitatively but in future must be quantitatively validated with an appropriate experimental set-up.

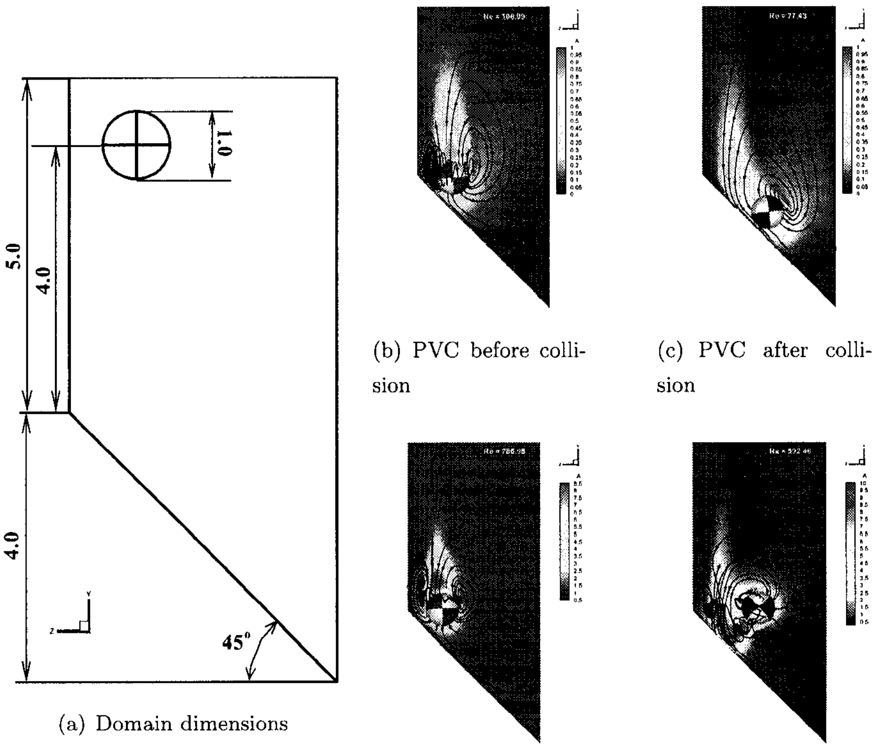
6.6.1 Interaction with slant walls

This physical situation is very common in any particulate flow system including mixing, slurry transport, particle depostion and unit operations like the rake settlers. In order to verify the calculations in case of oblique collisions we set-up a numerical simulation where the settling particle undergoes collision with a plane inclined at 45° to the normal. Two extreme limits of collision were tested. At one extreme, the particle trajectory is simply along the incline of the wall. On the other hand, the particle can bounce off the inclined wall. To enable experimental verification in the future, realistic particle and fluid properties are chosen.

The overall dimensions in terms of particle diameters is shown in figure 6.6. The dimension in third direction is $2.0d_p$. The direction of gravity is always along the negative y-axis. The density of the fluid is $\rho_1 = 1180\ kg/m^3$. The low Reynolds number case corresponds to a PVC particle with density $\rho_2 = 1410\ kg/m^3$ and high Reynolds number case corresponds to a steel particle with density $\rho_2 = 7800\ kg/m^3$. A high resolution unstructured mesh with 589682 nodes and 424271 elements was used. The subfigures in figure 6.6 for a PVC particle and for a steel particle show the contours and streamtraces before and after the collision with slant wall. Note that PVC particle is continuously in contact with wall after initial impact. We found that initially there is a fluid layer between the particle and the wall that makes the particle slip but as the particle sinks through the fluid layer the friction coefficient determines whether the particle will slide or roll.

6.6.2 Drafting-Kissing-Tumbling Mechanism

The drafting-kissing-tumbling mechanism is a well known phenomenon for interaction between two settling particles [16]. This phenomena results from the



(a) Domain dimensions

(b) PVC before collision

(c) PVC after collision

(d) Steel before collision

(e) Steel after collision

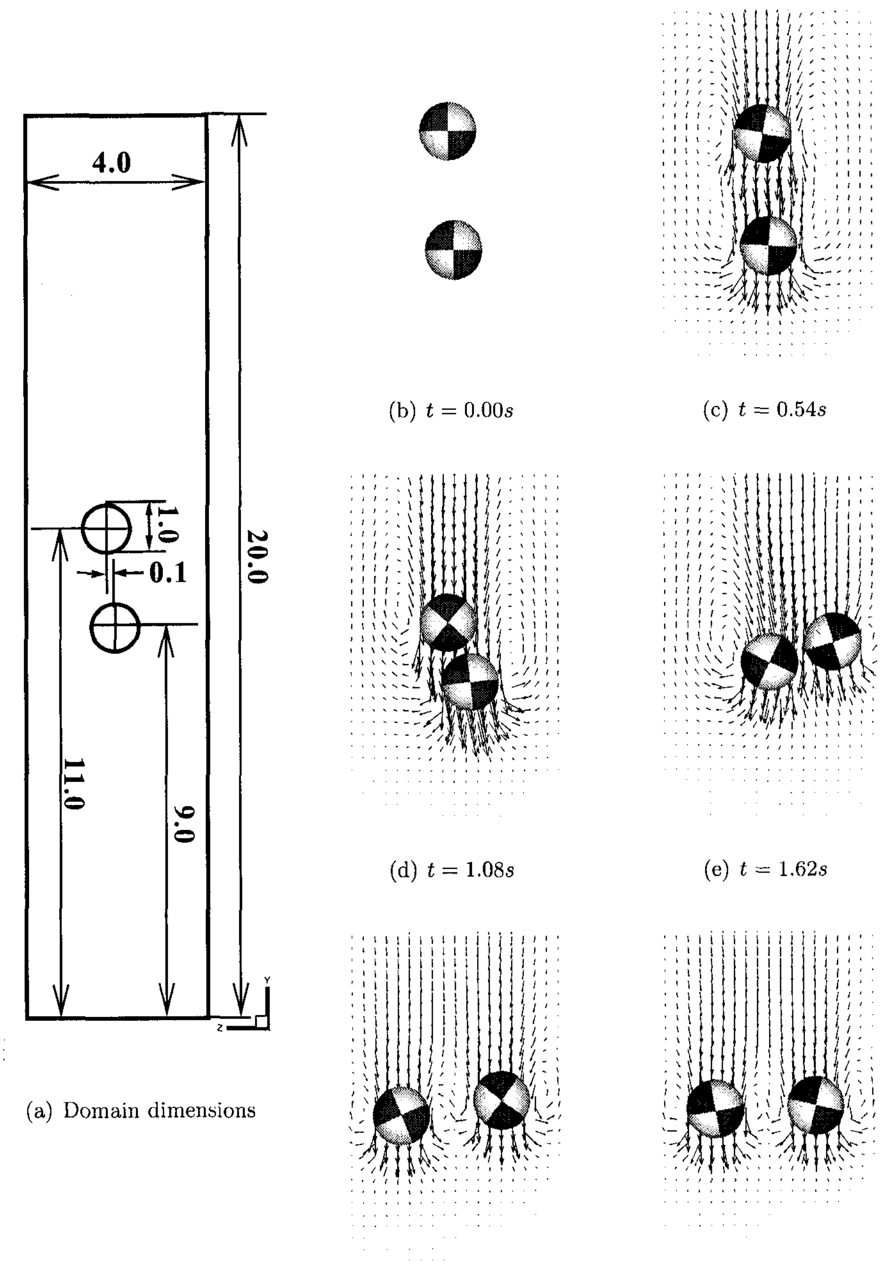
Figure 6.6: Comparison of the velocity contours and streamtraces for a PVC particle sliding or rolling on the slant wall and a steel particle bouncing off a 45° slant wall

interaction of the wake of the leading particle with the lagging particle. Initially both particles accelerate at same rate but as the wake of leading particle reaches the lagging particle, the lagging particle accelerates faster due to low pressure in the wake. This leads to particles colliding with each other vertically. The particles may continue to settle together but even a slight perturbation makes the oblong configuration loose stability. Therefore, the particle on top tumbles sideways. During this time the particles are continously in contact. Once they reach a horizontal configuration, they separate to establish themselves at a distance. Future investigations on the interaction of wakes as they settle in horizontal configuration might shed light on the critical separation distance.

The sequence of drafting, kissing and tumbling is shown in the figure 6.7. The settling particles used in this numerical experiment were PVC particles ($\rho_2 = 1410.0\ kg/m^3$) of diameter 1.27 cm. The fluid medium properties used was same as that of 80% glycerine solution at 21° C. The viscosity of this solution is 57.48 cP and density is 1207.89 kg/m^3. Initially, the particles were placed vertically one d_p apart with the top particle offset by a small distance ($0.1d_p$) in the positive z-direction to predefine the tumbling plane. The center of the bottom and top particle is 11.43 cm and 13.97 cm from the bottom, respectively. The channel used has a square cross-section of side 5.08 cm and length of the channel is 25.4 cm. The direction of gravity is as always set along negative y-axis. The overall dimensions are shown in the figure 6.7.

The four vertical faces are implemented as wall boundaries. The bottom face is implemented as inlet and top face as outlet. The simulations are done using moving reference frame fixed to the centroid of the two particle system. Hence the advection velocity is taken with respect to the velocity of the centroid of the two particles. A uniform mesh was employed, however since the particles are confined to the center of the channel due to the moving reference frame, a non-uniform mesh with higher resolution in the middle and coarser mesh towards the inlet and outlet could also be used. The mesh used had 238641 nodes and 160000 elements. The entire simulation was carried out for 20000 time steps with time step size of 0.005.

The figure 6.8 shows the velocity of the settling particles. These trajectories are qualitatively similar to the simulation results of Glowinski et al. [18].



(a) Domain dimensions

(b) $t = 0.00s$

(c) $t = 0.54s$

(d) $t = 1.08s$

(e) $t = 1.62s$

(f) $t = 3.23s$

(g) $t = 6.46s$

Figure 6.7: Drafting-Kissing-Tumbling sequence of two settling particles.

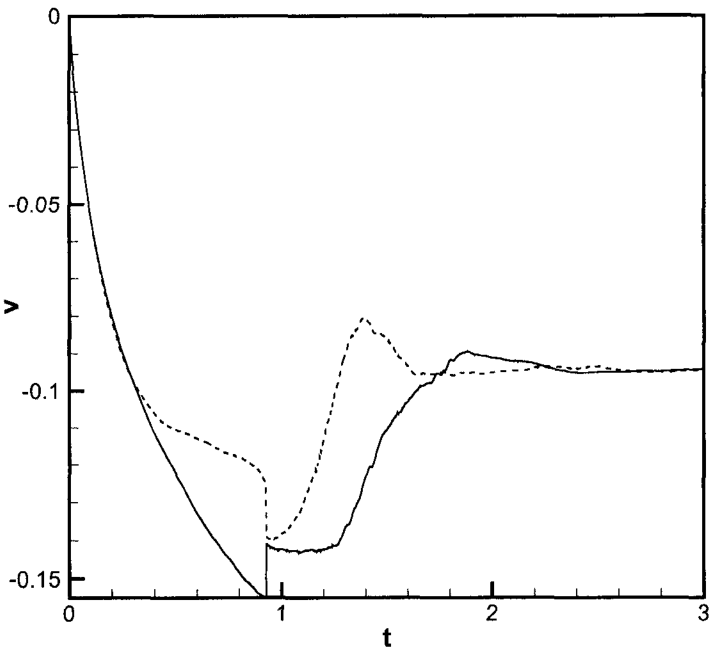


Figure 6.8: Trajectory of the two settling particles: leading particle (dashed line) and lagging particle (solid line), undergoing drafting, kissing and tumbling mechanism.

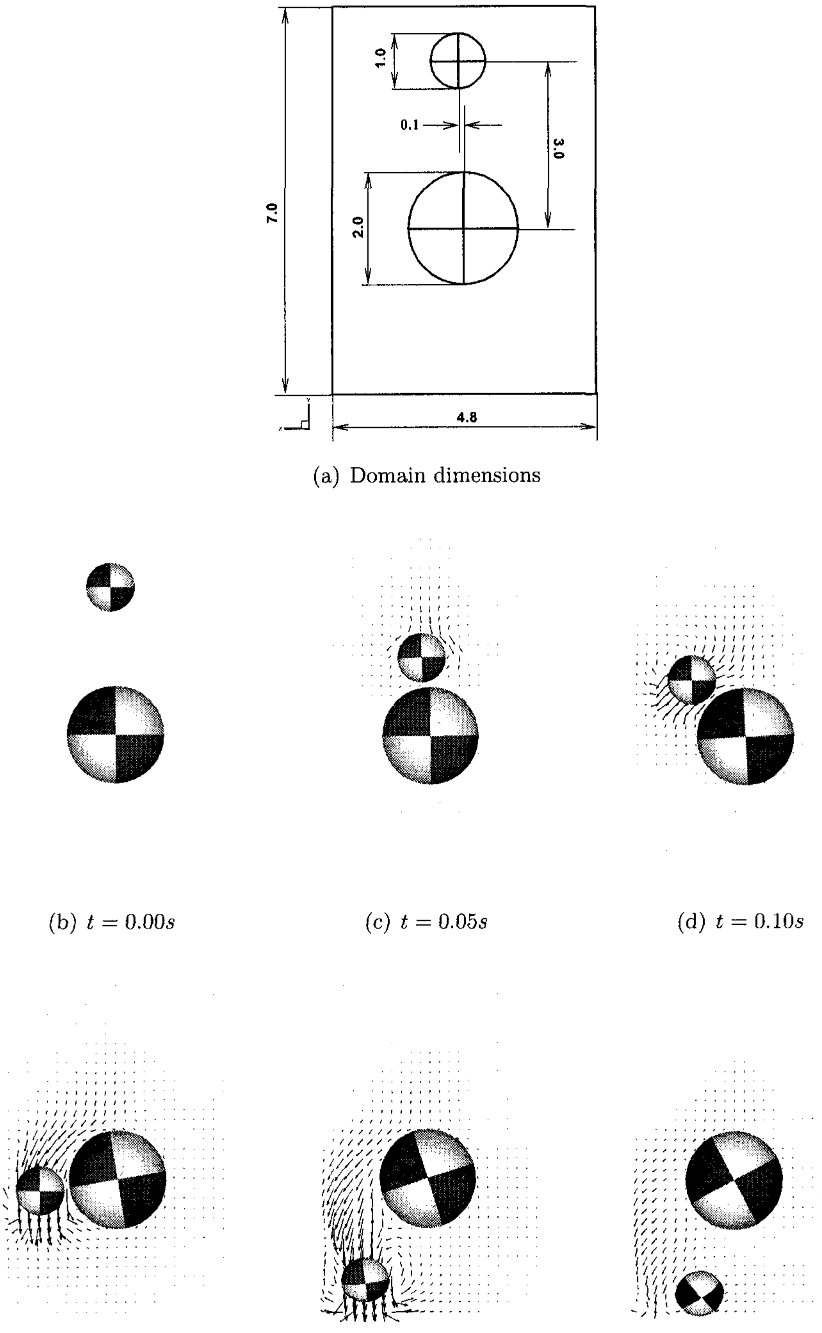
6.6.3 Interaction with neutrally buoyant particle

The simulations discussed here demonstrate the capability of the code to handle interaction between particles with different densities and sizes. This is usually the case of heterogeneous sedimentation systems. Moreover, the collision modeling in this case is challenging due to the prolonged contact between two moving particles.

Settling particles of different materials were used to vary the density ratio with respect to the fixed fluid medium. Table 6.2 gives details of the parameters used in the numerical study. The fluid medium properties used was same as that of 80% glycerine solution at 21° C. The viscosity of this solution is 57.48 cP and density is 1.20789 g/cc. The diameter of the settling particle is 1.27 cm and that of neutrally buoyant particle is 2.54 cm. The dimensions of the container are $8.89 \times 6.096 \times 4.064$ cm. Figure 6.9 shows the sequence of interaction of the settling and neutrally buoyant particle and in figure 6.10, we show the positions of the settling and neutrally buoyant particles as collision proceeds. The figure 6.9 also gives the dimensions and initial positions of the particles. The initial position of settling and neutrally buoyant particles are shown by dashed circles in figure 6.10. The two subfigures show the enlarged trajectory of neutrally buoyant particle and the interaction of the settling particle with bottom wall. The denser settling particles leave contact sooner and the brass particle can be seen to bounce at the bottom wall. The boundary conditions corresponds to that of a open container i.e. four side faces and bottom face are implemented as wall boundary conditions and the top surface stress free boundary condition i.e. $P = 0$. The mesh used had 781485 nodes and 537600 elements. The entire simulation was carried out for 5000 time steps with time step size of 0.005.

Table 6.2: Materials used and corresponding parameters for simulation of settling particle interaction with neutrally buoyant particle

Material	ρ_p	Re	St
PVC	1410.0	31.4648	4.0811
Teflon	2160.0	91.7935	18.2388
Steel	7830.0	311.1846	224.1352
Brass	8530.0	330.5544	259.3714



(a) Domain dimensions

(b) $t = 0.00s$

(c) $t = 0.05s$

(d) $t = 0.10s$

(e) $t = 0.125s$

(f) $t = 0.15s$

(g) $t = 0.20s$

Figure 6.9: Interaction of settling and neutrally buoyant particles

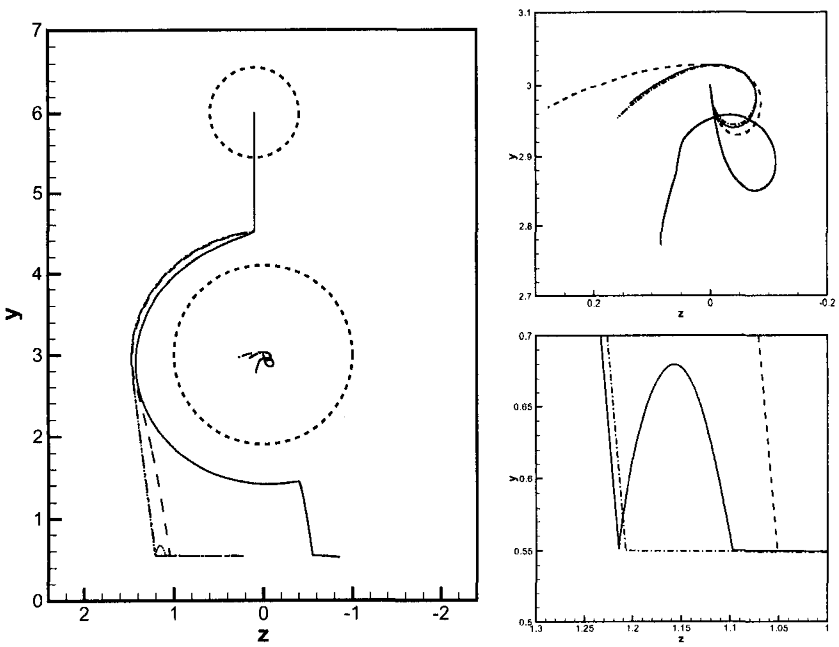


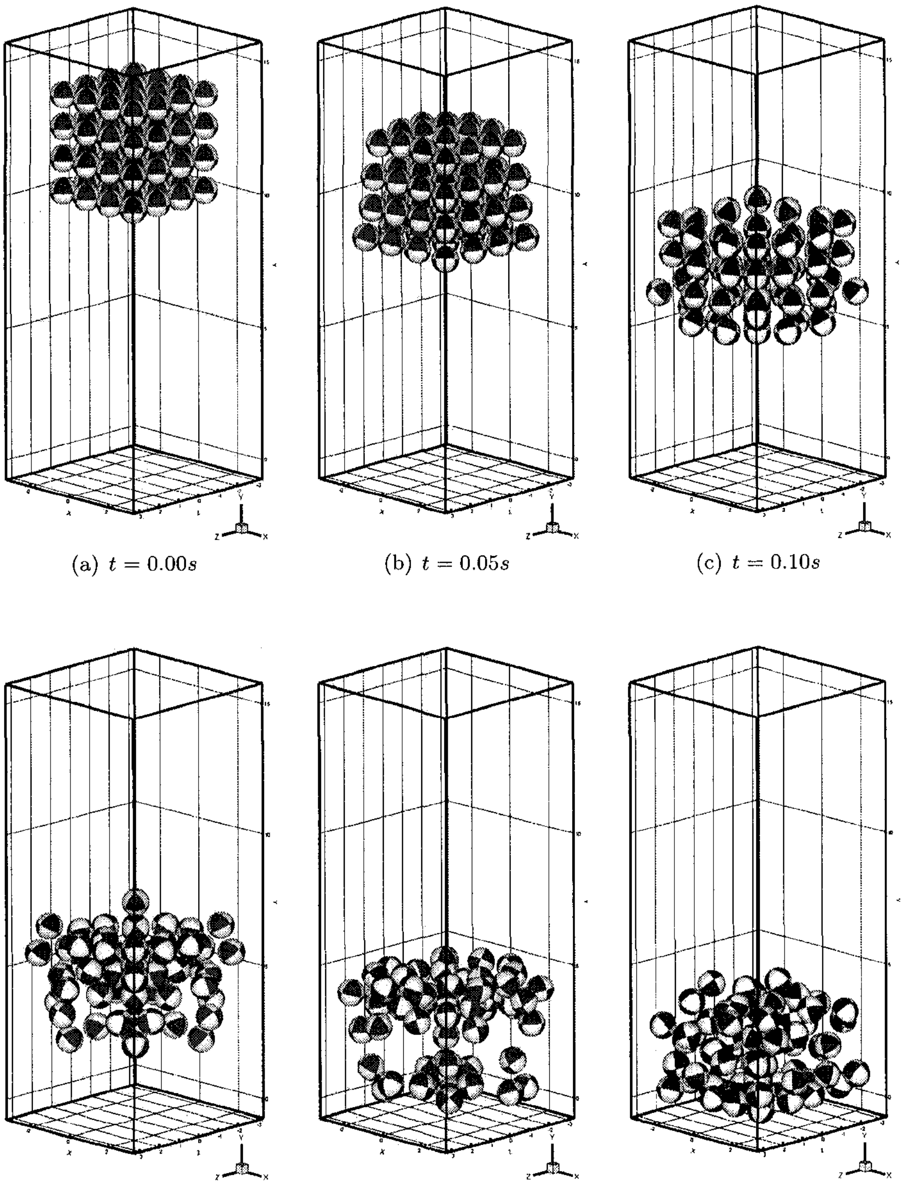
Figure 6.10: Trajectory of the particles: solid line-PVC, dashed-Teflon, dasheddot-steel and dotted-brass particle.

6.6.4 Simulation of 64 particle in a sedimentation cell

The collision model developed is for two particle interaction and particle and wall interaction, however it can be easily extended to large number of particles using time substepping. The figure 6.11 shows frames of the settling of 64 particles in closed box taken at specific time steps. The particles are initially arranged in a cubic array of 4×4×4 particles at the top of the box and then let to settle. Initially the particles settle in a *wave-like* fashion with expansion of the cluster. As the particles at the center start to accelerate, they create low pressure in the wake and other particles are sucked into the wake leading to creation of *swirls*. The higher the density of the particles with respect to the fluid, the sooner the cluster breaks into swirls. The density ratio in the present case is $\rho_2/\rho_1 = 7.06$. The mesh used had 494371 nodes and 337500 elements. The boundary conditions corresponding to a open box were employed. The entire simulation was carried out in 1000 time steps with a time step size of 0.005. The particle density corresponding to brass particles was used ($\rho_2 = 8530.0\ kg/m^3$) and the fluid medium properties used was same as that of 80% glycerine solution at $21^\circ\ C$. The viscosity of this solution is $57.48\ cP$ and density is $1.20789\ g/cc$.

6.7 Conclusion

A new collision model based on a stereomechanical impact model and lubrication theory is developed for numerical implementation of particulate flows. The model was tested for a number of physical situations and validation for the case of bouncing steel ball on a horizontal glass wall was attempted. Convergence on two meshes was shown and it is believed that with further resolution of the mesh more accurate values can be obtained. Although the collision model is based on impact of two spheres in space, simulation of 64 particles in a closed box is also performed with the help of sub-stepping in time.



(a) $t = 0.00s$

(b) $t = 0.05s$

(c) $t = 0.10s$

(d) $t = 0.15s$

(e) $t = 0.20s$

(f) $t = 0.26s$

Figure 6.11: Sedimentation of 64 spherical particles in a 3D closed box

Chapter 7

Application to Progressing Cavity Pump

7.1 Introduction

Progressing cavity pumps (refered to as PCP for the rest of the chapter) were first conceptualized by Rene Moineau in 1930s when he licensed his patent to various companies for production. But only recently their successful application in oil sands industry for pumping fluids containing substantial fraction of solids and fluids with very high viscosity have greatly contributed to their popularity in Canada and for similar applications worldwide. The pump basically consists of a stator which as the name suggests is stationary and a rotor which rotates eccentrically about the stator axis. The mating between stator and rotor is such that there is a continuous seal line from end to end. As the rotor rotates a series of cavities are formed that progress from one end to another carrying the fluid with them and hence the name of the pump. In general, the stator and rotor are designed like gears which can have multiple lobes. However, due to high torque requirement in operating these pumps, it is the single-lobe design that is widely employed. The application of the non-Lagrange multiplier formulation to these single-lobe pumps are the subject of this chapter.

First we will describe the computational domain for PCP as required for the fictitious domain formulation in §7.2. In §7.3, the details of the pumps kinematics and dynamics are discussed to complete the physics of the problem. The theoretical expression for flow rate allows us to validate the rotor kinematics and these are discussed in §7.4. Finally we conclude with suggestions for future work in §7.5.

7.2 Fictitious domain approach

According to the fictitious domain computational domain (chapter 2), we suppose that the fluid occupies a domain Ω_1. This fluid is supposed to have a constant density ρ_1 and constant viscosity μ. The rotor on the other hand is supposed to occupy a domain Ω_2. We will denote quantities associated with fluid with a subscript '1' and those associated with rotor with a subscript '2'. Let us denote the interface between Ω_1 and Ω_2 by Γ_2. This is also the boundary of the rotor. The outer boundary of the fluid domain be denoted by Γ_1. Now the entire domain is denoted by $\Omega = \Omega_1 \cup \Omega_2$. See figure 7.1 for illustration.

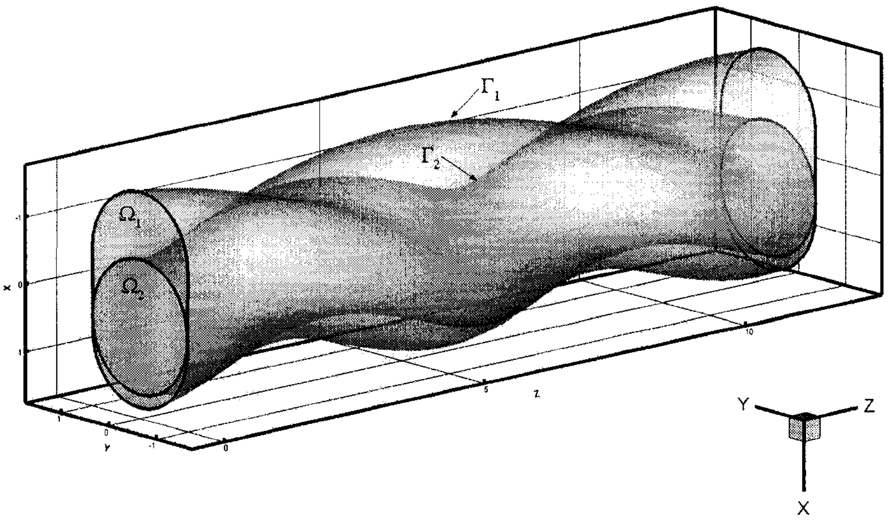


Figure 7.1: Computational domain used for simulation of progressing cavity pump

7.2.1 Formulation

The formulation of non-Lagrange multiplier approach has already been discussed in chapter 3. In order to apply this formulation to progressing cavity pump we need to substitute the particle with the rotor and all the dynamics of free particle are replaced by the prescribed motion of the pump (§7.3). Without loss of generality, the density of the pump can be assumed same as that of fluid. Therefore, we are looking at a neutrally buoyant particle of highly complicated shape. Like in the case of neutrally buoyant particles, there is no role for gravity here and therefore Froude number Fr drops out. The angular velocity of the rotor, can be used to define characteristic time scale $t_c = 30/(\pi N)$ in s, where N is the rotation rate of rotor in rpm. The rotor's cross-section's radius is taken as characteristic length scale $L_c = R_r$ in m. Hence, we can define a characteristic velocity scale $U_c = (\pi R_r N)/30$ in m/s. We define Reynolds number for fluid flow based on these characteristic scales and density and viscosity of fluid as $Re = (\rho U_c L_c)/\mu = (\pi \rho R_r^2 N)/(30\mu)$ where density ρ is in kg/m^3 and viscosity μ is in $Pa \cdot s$.

7.3 Theory and kinematics of progressing cavity pump

In this section we will describe the motion of the rotor and derive velocity at any general point in the rotor domain which can then be used to impose velocity at nodes inside rotor domain.

7.3.1 Geometry

The three essential parameters that decide whether a rotor and stator combination can be coupled together are the stator pitch (P_s), the rotor diameter (D_r) and the eccentricity of rotor axis about the stator axis (ϵ). Of course, the rotor also has a pitch (P_r), but that is always equal to half of the stator pitch. The geometry of a single lobe progressing cavity pump configuration is shown in figure 7.2. This figure shows a section at $z = 0$ of the 3D pump shown in figure 7.1. We will fix the fluid flow direction to be along +z direction and the rotational kinematics of the rotor are accordingly set. Initially the center of the rotor head at inlet or suction end of the pump is at $\{x = 0, y = 2\epsilon, z = 0\}$. This center is denoted by 'C'. The point 'B' is the projection of the rotor axis on the $z = 0$ plane at time $t = 0$ and point 'A' is the projection of the stator axis on the $z = 0$ plane at all times. For our derivations of subsequent motion of rotor, point 'A' is an ideal reference point. All calculations take this point as the origin unless otherwise stated. The pump design illustrated in figure 7.2 is for some arbitrary eccentricity of rotor axis from stator axis. In figure 7.3 we have shown a view from +y direction. The point 'R' is any arbitrary point on the helix that describes the locus of the center of the rotor sections at any z-coordinate and point 'P' is any point on the same rotor

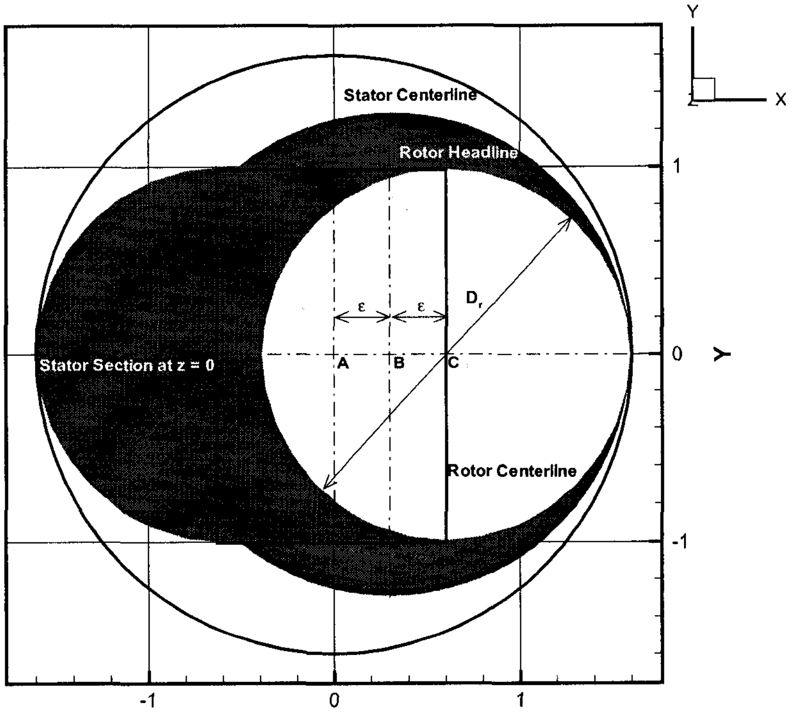


Figure 7.2: Section of a single lobe progressing cavity pump

section (constant z) as point 'R'.

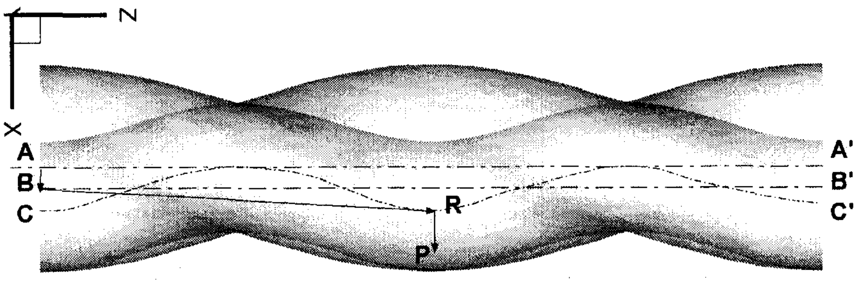


Figure 7.3: View of computational domain from +y direction

7.3.2 Rotor kinematics

In motion, the rotor axis BB' rotates around the stator axis AA' with an angular velocity of $\omega_z = (\pi N)/30$ in rad/s where N is the angular velocity in rpm. Also the whole rotor rotates about its own axis BB' with an angular velocity $-\omega_z$. Suppose that the z-coordinates of the point 'R' be z, then the phase of the helix on which it lies is given by $\gamma = (2\pi z)/P_r$. The position of 'R' with respect to 'B' at any instant is given by:

$$\mathbf{R}_{RB} = \epsilon \cdot cos(-\omega_z t + \gamma)\hat{i} + \epsilon \cdot sin(-\omega_z t + \gamma)\hat{j} + z\hat{k} \tag{7.1}$$

Since rotor axis itself is rotates about stator axis, the position of point 'B' with respect to 'A' at any instant is given by:

$$\mathbf{R}_{BA} = \epsilon \cdot cos(\omega_z t + \beta_0)\hat{i} + \epsilon \cdot sin(\omega_z t + \beta_0)\hat{j} \tag{7.2}$$

where β_0 is the angle that vector BA would make with x-axis at $t = 0$. For simplicity, we can take this to be zero for all simulations. The general point 'P' that lies inside the rotor domain at any time can be located by polar co-ordinates fixed at point 'R' as $\{r_p, \phi\}$. The position of 'P' with respect to 'R' is then given by:

$$\mathbf{R}_{PR} = r_p cos\phi\hat{i} + r_p sin\phi\hat{j} \tag{7.3}$$

Then clearly the position of point 'P' with respect to 'A' is given by:

$$\begin{aligned}\mathbf{R}_{PA} &= \mathbf{R}_{PR} + \mathbf{R}_{RB} + \mathbf{R}_{BA} \\ &= \left[\epsilon\left\{cos(-\omega_z t + \gamma) + cos(\omega_z t)\right\} + r_p cos\phi\right]\hat{\imath} \\ &+ \left[\epsilon\left\{sin(-\omega_z t + \gamma) + sin(\omega_z t)\right\} + r_p sin\phi\right]\hat{\jmath} + z\hat{k} \qquad (7.4)\end{aligned}$$

The velocity of the point 'P' with respect to 'A' is then given by the derivative of equation (7.4) with respect to time.

$$\begin{aligned}\mathbf{V}_{PA} &= \omega_z\epsilon\left\{sin(-\omega_z t + \gamma) - sin(\omega_z t)\right\}\hat{\imath} \\ &+ \omega_z\epsilon\left\{cos(\omega_z t) - cos(-\omega_z t + \gamma)\right\}\hat{\jmath} \qquad (7.5)\end{aligned}$$

This is the rotor domain velocity at any node that is inside rotor domain denoted by **U** in particle equations.

7.3.3 Theoretical volumetric flow rate

The progressing cavity pump can be broadly classified as a screw-type positive displacement pump. The pump achieves positive displacement of fluid by pushing the series of fluid carrying cavities formed as the rotor rotates. The theoretical flow rate from the pump can easily be calculated from the geometry and kinematics described above. At any cross-section of the pump (see figure 7.2) the area of the cavity is simply $A_c = 4\epsilon D_r$, i.e. equivalent to central rectangular region. Since this cavity area is same along the entire length of the pump, the volume of fluid is all cavities in one pitch length of stator P_s is $V = A_c P_s$. If the rate of rotation of rotor is N, then the theoretical volumetric flow rate is $Q_t = VN = 4\epsilon D_r P_s N$.

In order to compare the flow rate simulated to theoretical flow rate, we calculate the flow rate at the outlet plane in simulation as:

$$Q = \int_{A_c} \mathbf{u} \cdot \mathbf{n}_1 \mathrm{d}S \qquad (7.6)$$

where $\mathbf{n}_1$ is normal to the outlet of the fluid domain.

The comparison of the theoretical and simulated flow rate is discussed in §7.4.

Hydraulic force and torque

The torque that is required to rotate the rotor is spent in overcoming the pressure differential against which the fluid has to be pumped and to overcome the frictional resistance because of contact between rotor and stator material. We will not consider the frictional resistance in this article because this is dependent on material properties of rotor and stator. Even if the pressure differential is zero, torque is required to overcome the viscous dissipation of the fluid. Since, we have assumed constant viscosity during the pump operation, we can assume this torque requirement is constant. Of course, it must be pointed out that in real application there is always some heating of the fluid involved due to friction and viscous dissipation which it turn will change the viscosity of the fluid.

The hydraulic torque is a function of pressure differential and volume of fluid displaced $\tau_H = V\Delta P$, where V is the volumetric displacement and ΔP is the pressure differential.

The hydrodynamic force and torque that the fluid exerts on the rotor can be determined in simulations by evaluating the integrals:

$$\mathbf{F} = \int_{\Gamma_2} \boldsymbol{\sigma} \cdot \mathbf{n} \mathrm{d}S \tag{7.7}$$

$$\tau = \int_{\Gamma_2} \mathbf{r} \times (\boldsymbol{\sigma} \cdot \mathbf{n}) \mathrm{d}S \tag{7.8}$$

where, $\mathbf{r}$ is the position of the elemental surface dS on the rotor with respect to the center of the pump $\{x = 0, y = 0, z = (N_s P_s)/2\}$. N_s denotes number of stages in the pump, with each stage of length P_s. Note that the torque calculated in simulation includes the normal and shear stresses from the fluid and hence includes viscous effects of the fluid. As already pointed out this will only require a constant torque (constant in case of constant viscosity) in additional to whatever torque is required to overcome the pressure differential.

7.4 Numerical Results

The results reported in this section are in non-dimensional terms to protect the copyrights of the company who provided us the actual pump specifications to simulate. The radius of the rotor cross-section is used as the characteristic length scale, then the eccentricity is $\epsilon = 0.5$ and diameter of rotor is $D_r = 2.0$. The pitch of the stator is $P_s = 14.3516$. The properties of the fluid are taken as $\rho_1 = 1000 kg/m^3$ and viscosity of $\mu = 4.5 Pa \cdot s$. The Reynolds number as defined before can be calculated as 3.348. The non-dimensional rotation speed is $\omega_z = 1.0$. The volumetric flow rate is fixed once the geometry of the pump is fixed. The non-dimensional flow rate defined in terms of non-dimensional quantities can be calculated as 9.091.

In order to test the flow rate calculation of the code, the geometry as shown in the figure 7.4 is used. The inlet section from where the pump draws fluid and outlet section where the fluid is pumped into, are added to the pump geometry shown in figure 7.1 to facilitate implemention of proper boundary conditions. Both the inlet plane and outlet plane are implemented with open boundary conditions i.e. $P = 0$. This allows us to see whether the rotor motion is implemented properly to get correct flow rate. The pump should develop the pressure required for the flow to take place. Usually the pressure is developed in more than one stator pitch. Each fluid cavity occupies one full stator pitch going around the rotor. Two other half cavities co-exist with this full cavity. Since, the fluid in these cavities is in contact with inlet and outlet sections there will not be any pressure developed in going from inlet to outlet. This is at least true for the incompressible fluid case, where the pressure is supposed to propagate instantaneously across the whole domain. This justifies the use of open boundary conditions at both the inlet and outlet. In a multi-stage (one stage is equal to one stator pitch) pump the pressure will develop in the fluid cavity as it is transported across the pump. Then the boundary conditions will have to be revised and probably the assumption of incompressible fluid might not hold because then we cannot have pressure increase for the fluid with constant volume (or density). Therefore, here we will attempt only the single stage pump and validate the flow rate calculations.

The meshing for this complicated geometry is done using Gambit and is imported into the file format required for the code. The mesh is more refined in the stator section with node spacing of 0.15 as compared to inlet and outlet section where it is progressively coarsened to 0.3 at the inlet and outlet plane. The flow direction is in the positive z direction. The mesh without inlet and outlet sections has 291294 nodes and 205929 elements while the mesh with inlet and outlet sections has 833192 nodes and 589819 elements.

The code is run with different time step size to see the convergence of the calculated flow rate values. As expected the calculations are more accurate with smaller time steps. The tables 7.4 and 7.4 summarizes the calculated flow rate without and with inlet and outlet sections respectively. The theoretical flow rate is 9.091 for the pump configuration used. The table also records the percentage error from theoretical value of flow rate. The values for flow rates without the inlet or outlet sections are reported in comparison to the simulations that includes inlet and outlet sections to highlight the effect of boundary placement

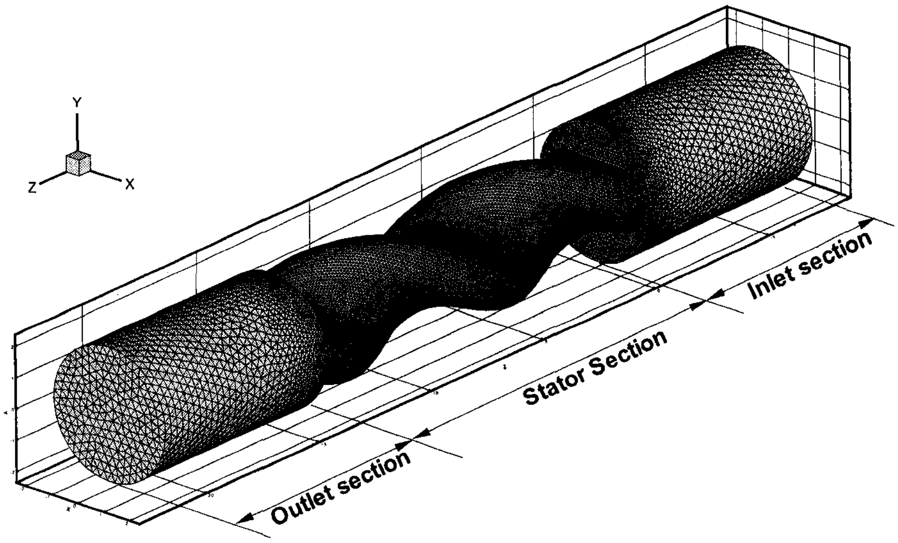


Figure 7.4: Computational domain of progressing cavity pump along with the inlet and outlet sections

Table 7.1: Calculated non-dimensional flow rates without inlet and outlet sections

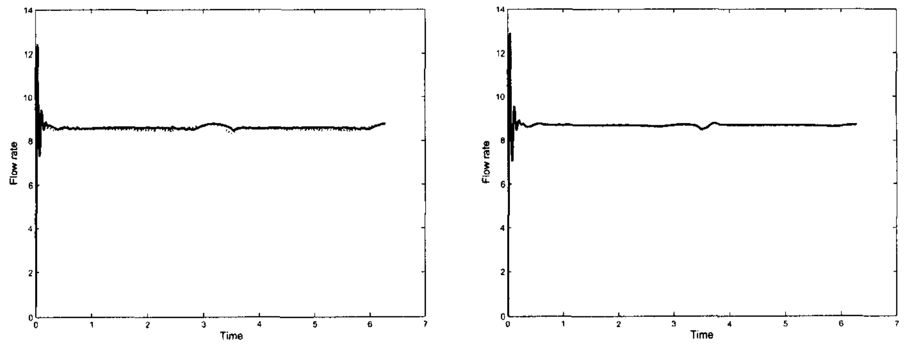
Time step size	Q_{in}	% error	Q_{out}	% error
0.01	8.0378	11.6	8.0953	10.9
0.005	8.3822	7.8	8.4357	7.2
0.0025	8.5526	5.9	8.6031	5.4

Table 7.2: Calculated non-dimensional flow rates with inlet and outlet sections

Time step size	Q_{in}	% error	Q_{out}	% error
0.01	8.0731	11.2	8.0729	11.2
0.005	8.4865	6.6	8.4866	6.6
0.0025	8.6796	4.5	8.6796	4.5

on the calculations. One important improvement that can be inferred from the tables 7.4 and 7.4 is the better conservation of mass when inlet and outlet sections are placed. This could be due to the fact that when there is no inlet and outlet sections used, the flow at the inlet to the pump is varying across the cavity's cross-section and accordingly the pressure across the section is varying due to the movement of the rotor. Hence, using a open boundary condition right at the pump's inlet is not correct. The inlet and outlet sections allows the flow to develop fully at the inlet and open boundary condition i.e. $P = 0$ is applicable uniformly across the outlet.

The figure 7.5 compares the development of the flow rate for both cases. In the later case the flow at the inlet is exact beneath the flow at the outlet. Also note that the flow rate is constant once the steady state is reached.



(a) Without inlet/outlet sections

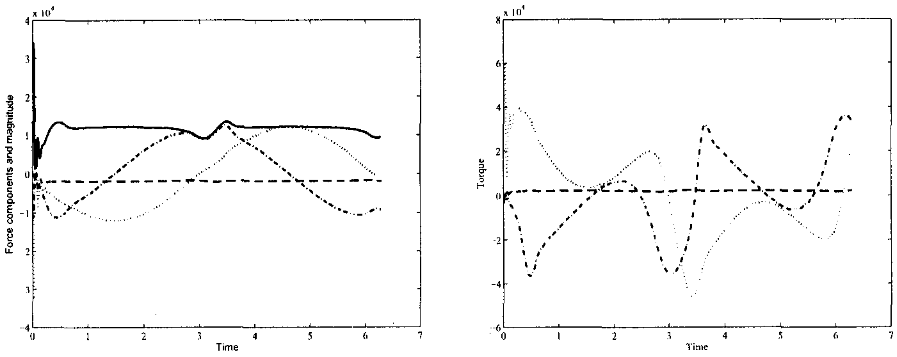
(b) Without inlet/outlet sections

Figure 7.5: Development of flow rates with and without inlet and outlet sections. The flow rate at inlet is shown with dotted line and outlet with solid line

We also calculated the hydrodynamic torque and force as given by the equations 7.7. The figures 7.6 provide the variation of the torque and force components as a function of time for the case when inlet and outlet sections are used and time step size of $dt = 0.0025$ is employed. However, in the absence of experimental data these quantities have not been validated.

7.5 Conclusion

The implementation of the fictitious domain approach with non-Lagrange multiplier method for the case of progressing cavity pumps was discussed. There are



(a) Force components

(b) Torque components

Figure 7.6: The time profile of hydrodynamic force and torque on the PCP rotor. The x-component is shown in dashdot, y-component in dotted and z-component in dashed line. Magnitude of force is shown in solid line

three geometrical parameters that define the volumetric displacement capacity of the pumps: the eccentricity of rotor axis from stator axis (ϵ), the diameter of the rotor (D_r) and the pitch of the stator (P_s). The theoretical volumetric flow rate could be obtained a simple relation based on these parameters. The calculation of incompressible fluid flow rate across the single stage pump was validated with the theoretical value. In future work the torque calculated by the numerical simulations should be validated with experiments and then the numerical simulation of these pumps for any combination of the geometrical parameters can be used to obtain a quick estimate of the real torque requirement and hence the power requirement of the pump for any given fluid. Of course these numerical simulations have only be done for Newtonian fluids. Often in real life applications, one would encounter non-Newtonian fluids and then the algorithm needs to modified for appropriate constitutive relationship between stress and strain.

Another useful next step is the dynamic motion of the rotor as opposed to the prescribed motion as used here. This would enable simulation of the inverse problem where fluid motion would be specified and the motion of the rotor is predicted. This has applications in real world operations like drilling.

Chapter 8

Conclusions

8.1 Major contributions

- We propose a modification of the fictitious domain method of Diaz-Goano et al. [13]. The major improvement is that the interaction force (the Lagrange multiplier) is explicitly resolved in the discretization process and therefore, it is possible to avoid or greatly reduce the number of iterations for the imposition of rigid body motion.

- We introduce a dynamic particle meshing technique for the computation of the interaction terms which greatly improves the accuracy of the calculations.

- We propose a new collision model which combines lubrication and stereomechanical impact theories and allows for a proper resolution of particle-particle and particle-wall interactions at a range of governing parameters. A number of numerical collision experiments are performed and physically meaningful results are obtained. A multiple particle simulation is also performed using sub-stepping in time to detect collision between multiple particles.

- Finally, we simulate the operation of a progressing cavity pump using the fictitious domain approach. The flow rate of the positive displacement pump can be theoretically evaluated. We use the theoretical value to validate the flow rate given by the simulation.

8.2 Future work

In future the first task is to optimize the parallel code. A parallel version has been developed using PETSc (Portable, Extensible Toolkit for Scientific computing) libraries. This implementation uses a simple matrix splitting methodology for distributing the computing load to various processors in shared as well as in distributed memory architectures. Further optimization of the code is required to improve the scalability to larger number of processors. Once satisfactory scalability is achieved, the parallel version must be validated for the settling particle and migrating particle cases.

Using a finer mesh, bouncing particle simulations should be performed to validate the collision model at higher Reynolds numbers. There is also the need to perform experiments to validate the various particle interaction simulations outlined in this thesis.

Once the code is validated and good scalability has been achieved, it can be used to solve large multiple particle interaction problems. With at least 100 particles in a closed box sedimentation set-up, one can further investigate the wavy and swirl regimes observed in this thesis and clearly delineate the limits of these regimes.

Bibliography

[1] J. R. Abbott, A. L. Graham, L. A. Mondy, and H. Brenner. Dispersion of a ball settling through a quiescent neutrally buoyant suspension. *Journal of Fluid Mechanics*, 361:309–331, 1998.

[2] F. Abraham. Functional dependence of drag co-efficient of a sphere on reynolds number. *Physics of Fluids*, 13:2194, 1970.

[3] C. K. Aidun and E-J. Ding. Dynamics of particle sedimentation in a vertical channel: Period-doubling bifurcation and chaotic state. *Physics of Fluids*, 15 (6):1612–1621, 2003.

[4] G. Barnocky and R. H. Davis. Elastohydrodynamic collision and rebound of spheres: experimental verification. *Physics of Fluids*, 31(6):1324–1329, 1988.

[5] J. Bey. Tetrahedral mesh refinement. *Computing*, 55(4):355–378, 1995.

[6] H. Brenner. The slow motion of a sphere through a viscous fluid towards a plane surface. *Chemical Engineering Science*, 16:242–251, 1961.

[7] S. Chen and G. D. Doolen. Lattice boltzmann method for fluid flows. *Annual Review of Fluid Mechanics*, 30:329–364, 1998.

[8] T. Chen, P. D. Minev, and K. Nandakumar. A projection scheme for incompressible multiphase flow using adaptive eulerian grid. *International Journal of Numerical Methods in Fluids*, 45(1):1–19, 2004.

[9] R. H. Davis. Effects of surface roughness on a sphere sedimenting through a dilute suspension of neutrally buoyant spheres. *Physics of Fluids A.*, 4(12): 2607–2619, 1992.

[10] R. H. Davis and N. A. Hill. Hydrodynamic diffusion of a sphere sedimenting

through a dilute suspension of neutrally-buoyant spheres. *Journal of Fluid Mechanics*, 236:513–533, 1992.

[11] R. H. Davis, J. M. Serayssol, and E. J. Hinch. The elastohydrodynamic collision of two spheres. *Journal of Fluid Mechanics*, 163:479–497, 1986.

[12] C. Diaz-Goano. *Direct Numerical Simulations of Particulate Flows using a Finite Element/Fictitious Domain Approach.* PhD thesis, University of Alberta, Edmonton, AB, Canada, 2003.

[13] C. Diaz-Goano, P. D. Minev, and K. Nandakumar. A fictitious domain/finite element method for particulate flows. *Journal of Computational Physics*, 192: 105–123, 2003.

[14] M. L. Ekiel-Jezewska, N. Lecoq, R. Anthore, F. Bostel, and F. Feuillebois. Rotation due to hydrodynamic interactions between two spheres in contact. *Physics Review E.*, 66(051504), 2002.

[15] J. Feng, H. H. Hu, and D. D. Joseph. Direct simulation of initial value problems for the motion of solid bodies in a newtonian fluid. part 1. sedimentation. *Journal of Fluid Mechanics*, 261:95–134, 1994.

[16] A. F. Fortes, D. D. Joseph, and T. S. Lundgren. Nonlinear mechanics of fluidization of beds of spherical particles. *Journal of Fluid Mechanics*, 177: 467–483, 1987.

[17] R. Glowinski, T. W. Pan, T. I. Hesla, and D. D. Joseph. A distributed lagrange multiplier/ fictitious domain method for particulate flows. *International Journal of Multiphase Flows*, 25:755–794, 1999.

[18] R. Glowinski, T. W. Pan, T. I. Hesla, D. D. Joseph, and J. Periaux. A fictitious domain approach to the direct numerical simulation of incompressible viscous flow past moving rigid bodies: Application to particulate flow. *Journal of Computational Physics*, 169:363–426, 2001.

[19] W. Goldsmith. *Impact: The theory and physical behaviour of colliding solids.* London: Edward Arnold ltd., 1960.

[20] P. Gondret, E. Hallouin, M. Lance, and L. Petit. Experiments on the motion of a solid sphere toward a wall: From viscous dissipation to elastohydrodynamic bouncing. *Physics of Fluids*, 11:2803–2805, 1999.

[21] P. Gondret, M. Lance, and L. Petit. Bouncing motion of spherical particles in fluids. *Physics of Fluids*, 14(2):643–652, 2002.

[22] P. M. Gresho. On the theory of semi-implicit projection methods for viscous incompressible flow and its implementation via a finite element method that also introduces a nearly consistent mass matrix. part.1.theory. *International Journal of Numerical Methods in Fluids*, 11:587–620, 1990.

[23] J. Guermond and P. D. Minev. Analysis of a projection/characteristic scheme for incompressible flow. *Communications in Numerical Methods in Engineering*, 19:535–550, 2003.

[24] J. Happel and H. Brenner. *Low Reynolds number hydrodynamics with special applications to particulate media.* The Hague : Martinus Nijhoff, 1973.

[25] H. H. Hu, D. D. Joseph, and M. J. Crochet. Direct simulation of fluid particle motions. *Theoretical Computational Fluid Dynamics*, 3:285–306, 1992.

[26] H. H. Hu, N. A. Patankar, and M. Y. Zhu. Direct numerical simulations of fluid-solid systems using the arbitrary lagrangian-eulerian technique. *Journal of Computational Physics*, 169:427–462, 2001.

[27] M. S. Ingber. Numerical simulation of the hydrodynamic interaction between a sedimenting particle and a neutrally buoyant particle. *International Journal of Numerical Methods in Fluids*, 9:263–273, 1989.

[28] D. J. Jeffrey and Y. Onishi. Calculation of the resistance and mobility functions for two unequal rigid spheres in low-reynolds-number flow. *Journal of Fluid Mechanics*, 139:261–290, 1984.

[29] A. A. Johnson and V. C. Patel. Flow past a sphere upto a reynolds number of 300. *Journal of Fluid Mechanics*, 378:19–70, 1999.

[30] A. A. Johnson and T. E. Tezduyar. Simulation of multiple spheres falling in a liquid-filled tube. *Computer Methods in Applied Mechanics and Engineering*, 134:351–373, 1996.

[31] A. A. Johnson and T. E. Tezduyar. 3-d simulation of fluid-particle interactions with the number of particles reaching 100. *Computer Methods in Applied Mechanics and Engineering*, 145:301–321, 1997.

[32] A. A. Johnson and T. E. Tezduyar. Advanced mesh generation and update methods for 3d flow simulations. *Computational Mechanics*, 23:130–143, 1999.

[33] G. G. Joseph and M. L. Hunt. Oblique particle-wall collisions in a liquid. *Journal of Fluid Mechanics*, 510:71–93, 2004.

[34] G. G. Joseph, R. Zenit, M. L. Hunt, and A. M. Rosenwinkel. Particle wall collisions in a viscous fluid. *Journal of Fluid Mechanics*, 433:329–346, 2001.

[35] A. A. Kantak and R. H. Davis. Oblique collisions and rebound of spheres from a wetted surface. *Journal of Fluid Mechanics*, 509:63–81, 2004.

[36] P. D. Minev and C. R. Ethier. A characteristic/finite element algorithm for the 3-d navier-stokes equations using unstructured grids. *Computer Methods in Applied Mechanics and Engineering*, 178:39–50, 1999.

[37] N. Mordant and J.-F. Pinton. Velocity measurement of a settling sphere. *European Physical Journal B.*, 18:343–352, 2000.

[38] N.-Q. Nguyen and A. J. C. Ladd. Lubrication corrections for lattice-boltzmann simulations of particle suspensions. *Physics Review E.*, 66 (046708), 2002.

[39] H. Nicolai, B. Herzhaft, E. J. Hinch, L. Oger, and E. Guazzelli. Particle velocity fluctuations and hydrodynamic self-diffusion of sedimenting non-brownian spheres. *Physics of Fluids*, 7:12–23, 1995.

[40] H. Nicolai, Y. Peysson, and E. Guazzelli. Velocity fluctuations of a heavy sphere falling through a sedimenting suspension. *Physics of Fluids*, 8:855–862, 1996.

[41] S. Osher and R. P. Fedkiw. Level set methods: An overview and some recent results. *Journal of Computational Physics*, 169:463–502, 2001.

[42] T.-W. Pan and R. Glowinski. Direct simulation of the motion of neutrally buoyant balls in a three-dimensional poiseuille flow. *Comptes Rendus Mecanique*, 333:884–895, 2005.

[43] T.-W. Pan, D. D. Joseph, R. Bai, R. Glowinski, and V. Sarin. Fluidization of 1204 spheres: simulation and experiment. *Journal of Fluid Mechanics*, 451: 169–191, 2002.

[44] P. Ploumhans, G. S. Winckelmans, J. K. Salmon, A. Leonard, and M. S. Warren. Vortex methods for direct numerical simulation of three-dimensional bluff body flows: application to the sphere at re = 300, 500, and 1000. *Journal of Computational Physics*, 178:427–463, 2002.

[45] D. Ruprecht and H. Müller. A scheme for edge-based adaptive tetrahedron subdivision. In H.C. Hege and K. Polthier, editors, *Mathematical Visualization*, pages 61–70. Springer: Heidelberg, 1998.

[46] W. B. Russel, D. A. Saville, and W. R. Schowalter. *Colloidal dispersions.* Cambridge University Press, 1989.

[47] R. Scardovelli and S. Zaleski. Direct numerical simulation of free-surface and interfacial flow. *Annual Review of Fluid Mechanics*, 31:567–603, 1999.

[48] G. Segré and A. Silberberg. Radial poiseuille flow of suspensions. *Nature*, 189:206, 1961.

[49] P. Singh, T. I. Hesla, and D. D. Joseph. Distributed lagrange multiplier method for particulate flows with collisions. *International Journal of Multiphase Flows*, 29:495–509, 2003.

[50] J. R. Smart and D. T. Leighton Jr. Measurement of the hydrodynamic surface roughness of non-colloidal spheres. *Physics of Fluids*, 1:52–60, 1989.

[51] A. ten Cate, C. H. Nieuwstad, J. J. Derksen, and H. E. A. Van den Akker. Particle imaging velocimetry experiments and lattice-boltzmann simulations on a single sphere settling under gravity. *Physics of Fluids*, 14(11):4012–4025, 2002.

[52] G. Tryggvason, B. Bunner, A. Esmaeeli, D. Juric, N. Al-Rawahi, W. Tauber, J. Han, S. Nas, and Y.-J. Janz. A front-tracking method for the computations of multiphase flow. *Journal of Computational Physics*, 169:708–759, 2001.

[53] C. Veeramani, P. D. Minev, and K. Nandakumar. A fictitious domain formulation for flows with rigid particles: non-lagrange multiplier version. *Journal of Computational Physics*, 2006.

[54] C. Veeramani, P. D. Minev, and K. Nandakumar. A ficitious domain method for particle sedimentation. *Lecture Notes in Computer Science*, 3743:544–551, 2006.

[55] D. C. Wan and S. Turek. Direct numerical simulation of particulate flow via multigrid fem techniques and the fictitious boundary method. *International Journal of Numerical Methods in Fluids*, 51(5):531–566, 2006.

[56] B. H. Yang, J. Wang, D. D. Joseph, H. H. Hu, T. W. Pan, and R. Glowinski. Migration of a sphere in tube flow. *Journal of Fluid Mechanics*, 540:109–131, 2005.

[57] F. L. Yang and M. L. Hunt. Dynamics of particle-particle collisions in a viscous liquid. *Physics of Fluids*, 18(121506):1–11, 2006.

[58] Z. Yu, N. Phan-Thein, and R. I. Tanner. Dynamic simulation of sphere motion in a vertical tube. *Journal of Fluid Mechanics*, 518:61–93, 2004.

[59] S. Zeng, E. T. Kerns, and R. H. Davis. The nature of particle contacts in sedimentation. *Physics of Fluids*, 8(6):1389–1396, 1996.

[60] D. Z. Zhang and A. Prosperetti. Averaged equations for inviscid disperse two-phase flow. *Journal of Fluid Mechanics*, 267:185–219, 1994.

[61] W. Zhang, R. Noda, and M. Horio. Evaluation of lubrication force on colliding particles for dem simulation of fluidized beds. *Powder Technology*, 158:92–101, 2005.

Appendix A

Derivations

A.1 Combined equation of motion

The stress divergence term in the eqn.(3.1) is modified in the following steps to cancel the hydrodynamic interaction between fluid and particle.

- Using the tensor identity $\tau : \mathbf{D}[\mathbf{v}] = \boldsymbol{\nabla} \cdot (\tau \cdot \mathbf{v}) - (\boldsymbol{\nabla} \cdot \tau) \cdot \mathbf{v}$

$$\int_{\Omega_1} (\boldsymbol{\nabla} \cdot \sigma_1) \cdot \mathbf{v} \mathrm{d}\Omega = \int_{\Omega_1} \boldsymbol{\nabla} \cdot (\sigma_1 \cdot \mathbf{v}) \mathrm{d}\Omega - \int_{\Omega_1} \sigma_1 : \mathbf{D}[\mathbf{v}] \mathrm{d}\Omega \tag{A.1}$$

- Applying Gauss Divergence Theorem to first term on right:

$$\begin{aligned}\int_{\Omega_1} (\boldsymbol{\nabla} \cdot \sigma_1) \cdot \mathbf{v} \mathrm{d}\Omega = & \int_{\partial P_i} (\sigma_1 \cdot \mathbf{v}) \cdot \mathbf{n}_1 \mathrm{d}S + \int_{\Gamma} (\sigma_1 \cdot \mathbf{v}) \cdot \mathbf{n}_1 \mathrm{d}S \\ & - \int_{\Omega_1} \sigma_1 : \mathbf{D}[\mathbf{v}] \mathrm{d}\Omega\end{aligned} \tag{A.2}$$

  The surface integral on the external fluid boundary Γ drops out due to the weighting function $\mathbf{v} = 0$. Also changing the order of dot product in the remaining surface integral, we get:

$$\int_{\Omega_1} (\boldsymbol{\nabla} \cdot \sigma_1) \cdot \mathbf{v} \mathrm{d}\Omega = \int_{\partial P_i} (\sigma_1 \cdot \mathbf{n}_1) \cdot \mathbf{v} \mathrm{d}S - \int_{\Omega_1} \sigma_1 : \mathbf{D}[\mathbf{v}] \mathrm{d}\Omega \tag{A.3}$$

- Since $\mathbf{v} = \mathbf{V}_i + \boldsymbol{\xi}_i \times \mathbf{r}_i$ on $\partial P_i(t)$, the second term can be written as:

$$\begin{aligned}\int_{\Omega_1} (\boldsymbol{\nabla} \cdot \sigma_1) \cdot \mathbf{v} \mathrm{d}\Omega = & \int_{\partial P_i} (\sigma_1 \cdot \mathbf{n}_1) \cdot \mathbf{V} \mathrm{d}S + \int_{\partial P_i} (\sigma_1 \cdot \mathbf{n}_1) \cdot (\boldsymbol{\xi}_i \times \mathbf{r}_i) \mathrm{d}S \\ & - \int_{\Omega_1} \sigma_1 : \mathbf{D}[\mathbf{v}] \mathrm{d}\Omega\end{aligned} \tag{A.4}$$

- Using the vector identity: $a \cdot (b \times c) = b \cdot (c \times a)$ and the definition of hydrodynamic force (2.15) and torque (2.16), we get

$$\int_{\Omega_1} (\boldsymbol{\nabla} \cdot \sigma_1) \cdot \mathbf{v} \mathrm{d}\Omega = -\mathbf{F}_i \cdot \mathbf{V} - \mathbf{T}_i \cdot \boldsymbol{\xi}_i - \int_{\Omega_1} \sigma_1 : \mathbf{D}[\mathbf{v}] \mathrm{d}\Omega \qquad (A.5)$$

The expression is substituted back into combined variational equation of motion to the final combined equation of motion (3.2)

A.2 Integral Equation for particle domain

The steps for obtaining this form of particle motion is outlined in Glowinski et al. [17], but detailed derivations are not provided. The details of the derivations are presented here.

- The first step is to get the material derivative of the fluid velocity $\mathbf{u}_2$ as defined in particle domain.

$$\begin{aligned}
\frac{\mathrm{D}\mathbf{u}_2}{\mathrm{D}t} &= \frac{\partial \mathbf{u}_2}{\partial t} + (\mathbf{u}_2 \cdot \boldsymbol{\nabla})\mathbf{u}_2 \\
&= \frac{\partial}{\partial t}(\mathbf{U}_i + \boldsymbol{\omega}_i \times \mathbf{r}_i) + (\mathbf{u}_2 \cdot \boldsymbol{\nabla})(\mathbf{U}_i + \boldsymbol{\omega}_i \times \mathbf{r}_i) && \text{by (3.3)} \\
&= \left[\frac{\partial \mathbf{U}_i}{\partial t} + (\mathbf{u}_2 \cdot \boldsymbol{\nabla})\mathbf{U}_i\right] + \frac{\partial}{\partial t}(\boldsymbol{\omega}_i \times \mathbf{r}_i) \\
&\quad + (\mathbf{u}_2 \cdot \boldsymbol{\nabla})(\boldsymbol{\omega}_i \times \mathbf{r}_i) \\
&= \frac{\mathrm{D}\mathbf{U}_i}{\mathrm{D}t} + \frac{\partial \boldsymbol{\omega}_i}{\partial t} \times \mathbf{r}_i + \boldsymbol{\omega}_i \times \frac{\partial \mathbf{r}_i}{\partial t} \\
&\quad + (\mathbf{u}_2 \cdot \boldsymbol{\nabla}\boldsymbol{\omega}_i) \times \mathbf{r}_i + \boldsymbol{\omega}_i \times (\mathbf{u}_2 \cdot \boldsymbol{\nabla})\mathbf{r}_i \\
&= \frac{\mathrm{D}\mathbf{U}_i}{\mathrm{D}t} + \left[\frac{\partial \boldsymbol{\omega}_i}{\partial t} + (\mathbf{u}_2 \cdot \boldsymbol{\nabla}\boldsymbol{\omega}_i)\right] \times \mathbf{r}_i \\
&\quad + \boldsymbol{\omega}_i \times \left[\frac{\partial \mathbf{r}_i}{\partial t} + (\mathbf{u}_2 \cdot \boldsymbol{\nabla})\mathbf{r}_i\right] \\
&= \frac{\mathrm{D}\mathbf{U}_i}{\mathrm{D}t} + \frac{\mathrm{D}\boldsymbol{\omega}_i}{\mathrm{D}t} \times \mathbf{r}_i + \boldsymbol{\omega}_i \times \frac{\mathrm{D}\mathbf{r}_i}{\mathrm{D}t} \\
&= \frac{\mathrm{D}\mathbf{U}_i}{\mathrm{D}t} + \frac{\mathrm{D}\boldsymbol{\omega}_i}{\mathrm{D}t} \times \mathbf{r}_i + \boldsymbol{\omega}_i \times \frac{\mathrm{D}\mathbf{X}_i}{\mathrm{D}t} && \text{(rigid body)} \\
&= \frac{\mathrm{D}\mathbf{U}_i}{\mathrm{D}t} + \frac{\mathrm{D}\boldsymbol{\omega}_i}{\mathrm{D}t} \times \mathbf{r}_i + \boldsymbol{\omega}_i \times \mathbf{U}_i && \text{by (2.13)}
\end{aligned}$$

The final equation after substituting $\mathbf{U}_i = \boldsymbol{\omega}_i \times \mathbf{r}_i$, again true only for rigid body, becomes:

$$\frac{\mathrm{D}\mathbf{u}_2}{\mathrm{D}t} = \frac{\mathrm{D}\mathbf{U}_i}{\mathrm{D}t} + \frac{\mathrm{D}\boldsymbol{\omega}_i}{\mathrm{D}t} \times \mathbf{r}_i + \boldsymbol{\omega}_i \times (\boldsymbol{\omega}_i \times \mathbf{r}_i) \qquad (A.6)$$

- The second step is to get the variational formulation, multiply the above equation by $\rho_i \mathbf{v}$ and integrate over particle domain.

$$\int_{\Omega_i} \rho_i \frac{\mathrm{D}\mathbf{u}_2}{\mathrm{D}t} \cdot \mathbf{v} \mathrm{d}\Omega = \int_{\Omega_i} \rho_i \Bigg(\frac{\mathrm{D}\mathbf{U}_i}{\mathrm{D}t} + \frac{\mathrm{D}\boldsymbol{\omega}_i}{\mathrm{D}t} \times \mathbf{r}_i + \boldsymbol{\omega}_i \times (\boldsymbol{\omega}_i \times \mathbf{r}_i) \Bigg) \cdot \mathbf{v} \mathrm{d}\Omega \tag{A.7}$$

Since, for a spherically symmetric particle $\int_{\Omega_i} \mathbf{r}_i \mathrm{d}\Omega = 0$ and also all first moments about center vanish, it can be easily shown that:

$$\int_{\Omega_i} \rho_i \frac{\mathrm{D}\mathbf{u}_2}{\mathrm{D}t} \cdot \mathbf{v} \mathrm{d}\Omega = M_i \frac{\mathrm{D}\mathbf{U}_i}{\mathrm{D}t} \cdot \mathbf{V}_i + I_i \frac{\mathrm{D}\boldsymbol{\omega}_i}{\mathrm{D}t} \cdot \boldsymbol{\xi}_i + 0 \tag{A.8}$$

where, the mass of the particle $M_i = \int_{\Omega_i} \rho_i \mathrm{d}\Omega$ and moment of inertia for the sphere $I_i = \int_{\Omega_i} \rho_i \mathbf{r}_i^2 \mathrm{d}\Omega$ is used. In addition, any body force on the particle can also be written in weak form as:

$$\int_{\Omega_i} \rho_i \mathbf{g} \cdot \mathbf{v} \mathrm{d}\Omega = M_i \mathbf{g} \cdot \mathbf{V}_i \tag{A.9}$$

- Finally we can combine the equations as $\frac{\rho_1}{\rho_i}$((A.8) - (A.9)) and add the term $-\int_{\Omega_i} \sigma_2 : \mathbf{D}[\mathbf{v}] \mathrm{d}\Omega$, which is zero because of the rigid body assumption $\mathbf{D}[\mathbf{v}] = 0$, to the right hand side we get:

$$\int_{\Omega_i} \rho_1 \left(\frac{\mathrm{D}\mathbf{u}_2}{\mathrm{D}t} - \mathbf{g} \right) \cdot \mathbf{v} \mathrm{d}\Omega - \frac{\rho_1}{\rho_i} \left(M_i \left(\frac{\mathrm{D}\mathbf{U}_i}{\mathrm{D}t} - \mathbf{g} \right) \right) \cdot \mathbf{V}_i - \frac{\rho_1}{\rho_i} \left(I_i \frac{\mathrm{D}\boldsymbol{\omega}_i}{\mathrm{D}t} \right) \cdot \boldsymbol{\xi}_i = - \int_{\Omega_i} \sigma_2 : \mathbf{D}[\mathbf{v}] \mathrm{d}\Omega \tag{A.10}$$

This particle domain equation of motion can then be combined with fluid domain equation to arrive at the fictitious domain combined equation of motion in weak form (3.5).

Appendix B

MATLAB Code

B.1 Finding terminal settling velocity

The following code can be used to find properties of Glycerine-Water solution at various temperatures and compositions. One can use the Glycerine-Water properties chart available online at

http://www.dow.com/PublishedLiterature/dh_003d/09002f138003df2e.pdf ?filepath=glycerine/pdfs/noreg/115-00657.pdf&fromPage=GetDoc

for density and viscosity of the solution at various temperatures and compositions. These charts must be made into excel sheets for the program to load them into workspace. In case, properties of the fluid are already known then the first stage of the program can be skipped and terminal settling velocity along with Reynolds number (Re), Froude number (Fr) and Stokes number (St) can be calculated using the second stage of the program. In the present work only the second stage of the program is used.

```
% Glycerine-Water Density and Viscosity values
disp('*** PROPERTIES.M ***');
disp('STAGE-1: Estimate density and viscosity of solution using
temperature and composition');
disp('STAGE-2: Estimate Re, Fr, St and free settling velocity for
given particle in the solution');
opt = input('Do you want to perform solution property calculations?
(1/0): ');
```

```
if (opt)
disp('----------------------STAGE:1--------------------------------');
disp('Estimate the viscosity and density of glycerine-water solution');
disp('Linear interpolation is used for both cases');
disp('Temperate range for viscosity is 0-100 C');
disp('Temperate range for density is 15-30 C');
disp('-------------------------------------------------------------');
% Read the viscosity chart for glycerine
V = xlsread('Glycerine-Water-Viscosity');
D = xlsread('Glycerine-Water-Density');

% The row and column of viscosity matrix
% Range of temperature is the rows of viscosity chart
viscT = 0:10:100;
% Range of weight% of glycerine is the column of viscosity chart
viscWt = [0:10:60,65,67,70,75,80,85,90:1:100];

% The row and column of density matrix
% Range of temperature is the rows of density chart
denT =  [15, 15.5, 20, 25, 30];
denWt = 100:-1:0;

T = input('Enter the temperature in deg C: ');
wt = input('Enter weight % of glycerine in solution: ');
disp('-------------------------------------------------------------');
% Interpolate viscosity
% Find the row indices from temperature
if (T < viscT(1) | T > viscT(length(viscT)))
    error('Temperature is beyond valid range of 0-100 deg.C!!');
end
if (wt < viscWt(1) | wt > viscWt(length(viscWt)))
    error('Weight percent of glycerine is beyond the range of 0-100%??');
end
rlow = max(find(viscT <= T));
rhigh = min(find(viscT >= T));
```

```
% Find the column indices from weight %
clow = max(find(viscWt <= wt));
chigh = min(find(viscWt >= wt));

% Linear interpolation of viscosity between rows
if (rlow == rhigh)
    % no need to interpolate between rows
    Vlow = V(rlow,clow);
    Vhigh = V(rlow,chigh);
else
    % need to interpolate
    Vlow = V(rlow,clow) + (V(rhigh,clow)-V(rlow,clow))/...
    (viscT(rhigh)-viscT(rlow))*(T - viscT(rlow));
    Vhigh = V(rlow,chigh) + (V(rhigh,chigh)-V(rlow,chigh))/...
    (viscT(rhigh)-viscT(rlow))*(T - viscT(rlow));
end
if (clow == chigh)
    % no need to interpolate
    Vapprx = Vlow;
else
    % Linear interpolation between columns
    Vapprx = Vlow + (Vhigh - Vlow)/(viscWt(chigh)-viscWt(clow))...
    *(wt - viscWt(clow));
end
fprintf('Properties of %5.3f percent glycerine solution at
%5.3f C is: \n',wt,T);
fprintf('Viscosity = %5.2f mPa.s \n',Vapprx);

% Interpolate density
if (T < denT(1) | T > denT(length(denT)))
    error('Temperature is beyond database range of 15-30 deg.C!!');
end
if (wt > denWt(1) | wt < denWt(length(denWt)))
    error('Weight percent of glycerine is beyond the range of 0-100%??');
end
```

```
rlow = max(find(denT <= T));
rhigh = min(find(denT >= T));
% Find the column indices from weight %
chigh = min(find(denWt <= wt));
clow = max(find(denWt >= wt));

if (rlow == rhigh)
    % No need to interpolate
    Dlow = D(rlow,clow);
    Dhigh = D(rlow,chigh);
else
    % Linear interpolation of viscosity between rows
    Dlow = D(rlow,clow) + (D(rhigh,clow)-D(rlow,clow))/...
    (denT(rhigh)-denT(rlow))*(T - denT(rlow));
    Dhigh = D(rlow,chigh) + (D(rhigh,chigh)-D(rlow,chigh))/...
    (denT(rhigh)-denT(rlow))*(T - denT(rlow));
end
if (clow == chigh)
    % no need to interpolate
    Dapprx = Dlow;
else
    % Linear interpolation between columns
    Dapprx = Dlow + (Dhigh - Dlow)/(denWt(chigh)-denWt(clow))...
    *(wt - denWt(clow));
end
fprintf('Density = %7.5f g/cc \n',Dapprx);
end

disp('-----------------------STAGE:2---------------------------------');
disp('Estimate the dimensionless parameters for the settling particle');
disp('Drag co-efficient: Cd = (24/Re)*(1 + (Re^0.5)/9.06)^2');
disp('Ref: Ten Cate et al., Phys. Fluids, Vol. 14, No. 11, November 2002');
disp('Ref: F.Abraham, Phys. Fluid, 13, 2194 (1970).');
disp('---------------------------------------------------------------');
if (~opt)
```

```
    disp('Skipped Stage-1, enter properties of solution!!');
    Dapprx = input('Enter the density of the solution in g/ml: ');
    Vapprx = input('Enter the viscosity of the solution in cP: ');
end
Dp = input('Enter the diameter of the settling particle in m: ');
% acceleration due to gravity, in m/s2
g = 9.81;
rho_p = input('Enter the density of the settling particle in kg/m^3: ');
% density of the solution estimated above, in kg/m^3
rho_f = Dapprx * 1000;
% Viscosity of the solution estimated above, in Pa.s
mu_f = Vapprx * 1e-3;

% Iterate for U_inf
count = 1;
% Initial guess for the settling velocity
U_1 = 0;
% Reynolds Number
Re = (rho_f*Dp*U_1)/mu_f;
U_2 = ((Dp^2*(rho_p-rho_f)*g)/(18*mu_f))*(1/(1+(Re^0.5)/9.06)^2);
while (norm(U_2-U_1) > 1e-6)
  U_1 = U_2;
  Re = (rho_f*Dp*U_1)/mu_f;
  U_2 = ((Dp^2*(rho_p-rho_f)*g)/(18*mu_f))*(1/(1+(Re^0.5)/9.06)^2);
  count = count +1;
  if (count > 100)
    disp(sprintf('The iteration didnot converge in %d steps',count));
    break;
  end
end
Re = (rho_f*Dp*U_2)/mu_f;
U_inf = U_2;
St = (Re/9)*(rho_p/rho_f);
Fr = (U_2)^2/(Dp*g);
fprintf('Reynolds number = %7.5f \n',Re);
fprintf('Froude number = %7.5f \n',Fr);
fprintf('Stokes number = %7.5f \n',St);
fprintf('Free settling velocity = %7.5f m/s \n',U_inf);
```

A sample run of the program to determine terminal settling velocity of the steel ball settling in silicon oil RV10 used for collision model validation is shown below.

```
>> properties
*** PROPERTIES.M ***
STAGE-1: Estimate density and viscosity of solution using
temperature and composition
STAGE-2: Estimate Re, Fr, St and free settling velocity for
given particle in the solution
Do you want to perform solution property calculations? (1/0): 0
-----------------------STAGE:2---------------------------------
Estimate the dimensionless parameters for the settling particle
Drag co-efficient: Cd = (24/Re)*(1 + (Re^0.5)/9.06)^2
Ref: Ten Cate et al., Phys. Fluids, Vol. 14, No. 11, November 2002
Ref: F.Abraham, Phys. Fluid, 13, 2194 (1970).
---------------------------------------------------------------
Skipped Stage-1, enter properties of solution!!
Enter the density of the solution in g/ml: 0.935
Enter the viscosity of the solution in cP: 10
Enter the diameter of the settling particle in m: 0.003
Enter the density of the settling particle in kg/m^3: 7800
Reynolds number = 162.83141
Froude number = 11.45039
Stokes number = 150.93107
Free settling velocity = 0.58050 m/s
```

To determine the terminal settling velocity the program employs the the C_D versus Re relationship given by Abraham [2].