

**A Lagrangian-Lagrangian Framework for the Simulation of Fluid-Solid Interaction
Problems with Rigid and Flexible Components**

By
Arman Pazouki

A dissertation submitted in partial fulfillment of
the requirements for the degree of

Doctor of Philosophy
(Mechanical Engineering)

at the
UNIVERSITY OF WISCONSIN–MADISON
2014

Date of final oral examination: 04/30/2014

The dissertation is approved by the following members of the Final Oral Committee:

Dan Negrut, Associate Professor, Mechanical Engineering

Daniel J. Klingenberg, Professor, Chemical and Biological Engineering

William J. Likos, Associate Professor, Civil and Environmental Engineering

Krishnan Suresh, Associate Professor, Mechanical Engineering

Mario F. Trujillo, Assistant Professor, Mechanical Engineering

© Copyright by Arman Pazouki 2014

All Rights Reserved

This thesis is dedicated to my parents, Masoumeh and Aliasghar Pazouki, my first and best teachers. Thank you for your endless love, support, and encouragement.

ACKNOWLEDGMENTS

I would like to thank my advisor, Professor Dan Negrut, for his guidance and support. I would also like to thank the committee members for sharing their expertise and for their time. I very much enjoyed and benefited from many insightful conversations that I have had with my colleagues and friends in the Simulation-Based Engineering Laboratory, particularly with Dr. Radu Serban, Hammad Mazhar, and Andrew Seidl. Finally, I am so grateful to my family, whose unconditional love and support means so much to me.

DISCARD THIS PAGE

TABLE OF CONTENTS

	Page
LIST OF TABLES	vi
LIST OF FIGURES	vii
NOMENCLATURE	xii
ABSTRACT	xviii
1 Introduction	1
2 Modeling	4
2.1 Smoothed Particle Hydrodynamics (SPH)	4
2.1.1 SPH for fluid dynamics	6
2.2 General rigid body dynamics	9
2.3 Absolute Nodal Coordinate Formulation (ANCF) of flexible objects	10
3 Numerical solution	14
3.1 Boundary conditions	14
3.1.1 Fluid free-surface	14
3.1.2 Periodic BC	15
3.1.3 Inflow and outflow	17
3.1.4 Wall boundary condition	19
3.2 Fluid-solid interaction using Boundary Condition Enforcing (BCE) markers	19
3.3 Solid-solid short range interaction	20
3.3.1 Dry friction model	21
3.3.2 Lubrication force model	22
3.4 Time integration	23
3.4.1 Dual-rate integration	24

	Page
4 Advanced computing aspects	28
4.1 GPU hardware and programming model	28
4.2 Proximity calculation and neighbor search	30
4.2.1 Proximity computation-Approach 1: Optimized memory	30
4.2.2 Proximity computation-Approach 2: Optimized process	30
4.2.3 Proximity computation-concluding remarks	32
5 Validation studies and further numerical experiments	34
5.1 Particle suspension: A validation study	34
5.1.1 Transient Poiseuille flow	36
5.1.2 Particle migration in 2D Poiseuille flow	36
5.1.3 Particle migration in pipe flow	37
5.1.4 Radial distribution of particles in suspension	37
5.1.5 Effect of Reynolds number	41
5.2 Particle suspension: Further numerical experiments	41
5.2.1 Effect of particle rotation	41
5.2.2 Effect of particle concentration	42
5.2.3 Effect of inter-particle distance	43
5.2.4 Effect of particle asymmetry	44
5.2.5 Effect of particle size	46
5.3 Investigation of fluid-flexible beams interaction	46
5.3.1 Floating beam in Poiseuille flow	48
5.3.2 Flexible cantilever immersed in fluid: Effect of viscosity	50
5.3.3 Impulsively started motion of cantilevers in channel flow: Effect of elasticity	50
6 Performance Analysis	55
7 Demonstration of technology	61
7.1 Flow cytometry using microfluidic techniques	61
7.2 Dense particle suspension	63
7.3 Microscale numerical simulation of flow in porous media	64
7.4 Immersed deformable objects	66
7.5 Interacting rigid and flexible objects in channel flow	67
8 Directions of the future work and preliminary results	73
8.1 New numerical methods: Implicit integration for SPH	73

	Page
8.1.1 Incompressible fluid flow on a grid	73
8.1.2 Incompressible SPH	76
8.1.3 Implicit method	77
8.1.4 Boundary conditions	86
8.1.5 Preliminary results	89
8.2 Using SPH to investigate challenging applications	90
8.2.1 Effect of particle shape on the behavior of particle suspension and sedi- mentation	90
8.2.2 Ice-structure interaction	90
8.3 Using MPI-enabled high performance computing for faster solution of larger prob- lems	94
9 Conclusion	96
LIST OF REFERENCES	98
APPENDIX Partial derivatives on staggered grid	108

DISCARD THIS PAGE

LIST OF TABLES

Table	Page
5.1 Flow parameters used for the validation of transient Poiseuille flow	36
6.1 Time required for advancing the rigid body dynamics simulation by one time step as function of problem size (number of rigid bodies, N_r)	55
6.2 Time required for advancing the flexible body dynamics simulation by one time step as function of problem size (number of flexible bodies, N_f)	56
6.3 Time required for advancing a fluid dynamics simulation by one time step as function of problem size (number of SPH markers, N_m)	56
6.4 Time required for advancing the dynamics of FSI problems by one time step. The simulation times are provided for FSI problems with fixed number of SPH markers and increasing number of rigid bodies	58
6.5 Time required for advancing the dynamics of FSI problems by one time step. The simulation times are provided for FSI problems with fixed number of SPH markers and increasing number of flexible bodies, for two different values of the multi-rate integration factor $\tau_m = \Delta t_{SPH} / \Delta t_{ANCF}$	59
6.6 Time required for advancing the dynamics of combined FSI problems by one time step. The simulation times are provided for FSI problems of increasing size	59

DISCARD THIS PAGE

LIST OF FIGURES

Figure	Page
2.1 Illustration of the kernel W and its support domain S . SPH markers are shown as black dots. For 2D problems the support domain is a circle, while for 3D problems it is a sphere. The radius of the support domain is defined as a multiple, κ , of the kernel's characteristic length, h	6
3.1 Fluid free-surface modelled with SPH. The set of missing exterior, boundary, and interior markers are denoted as E , B , and I respectively.	16
3.2 Periodic boundary condition in SPH.	16
3.3 Schematic of the inflow boundary condition [66]. The inflow zone includes a set of ghost markers. The periodic boundary condition is applied to the ghost domain in the flow direction.	18
3.4 Fluid-solid interaction using BCE markers attached to a body: (a) a rigid body with the velocity, $\mathbf{V}$, and angular velocity, $\boldsymbol{\omega}$, shown at its center of gravity, $C.G.$. The markers velocities are represented by $\{\mathbf{v}_a, \mathbf{v}_b, \dots, \mathbf{v}_g\}$; (b) a flexible beam. BCE and fluid markers are represented by black and white circles, respectively. The BCE markers positioned in the interior of the body should be placed on and below the surface up to a depth no larger than the size of the compact support of the kernel W	20
3.5 A typical contact in DEM.	21
4.1 Illustration of the spatial subdivision method used for proximity computation in 2D. The circles represent the domain of influence of each marker; i.e., the support domain. For clarity, a coarse distribution of markers is shown. In reality, the concentration of markers per bin is much larger.	31
5.1 Velocity profile of transient Poiseuille flow obtained from simulation (dots) and series solution (continuous lines).	37

Figure	Page
5.2 Simulation of plane Poiseuille flow: lateral position as a function of dimensionless time of neutrally buoyant circular cylinders released from different initial positions. Results are within 1% relative error of those reported in [54,98].	38
5.3 Migration of rotating and non-rotating neutrally buoyant spheres in pipe flow.	39
5.4 Particle radial distribution as a function of non-dimensional distance L , compared to the experiment [116] at: (a) $L = 0$, (b) $L = 0.08$, (c) $L = 0.16$, (d) $L = 0.32$, (e) $L = 0.69$. Note that rigid bodies cannot be initialized in the region close to the wall; i.e., $0.9 \leq r/R \leq 1$, due to their finite size.	40
5.5 Effect of Re on the tubular pinch effect for $Re \in [1, 1400]$ and two particle size ratios, $\lambda = 0.11$ and $\lambda = 0.15$. The results are compared to data provided by Matas <i>et al.</i> [74], Yang <i>et al.</i> [131], and Shao <i>et al.</i> [120].	42
5.6 Particle radial distribution at $L = 0.69$, normalized by initial concentration, plotted for six different initial concentrations in the range of 4 through 128 particles/cm ³ equivalent to $\phi \in [0.109, 3.488]\%$. Results are compared to a reference experimental distribution obtained in [116] for concentration of 1 particles/cm ³	44
5.7 Particle arrangement at $L = 0.32$ for particle concentrations of (a) 32 particles/cm ³ and (b) 64 particles/cm ³	45
5.8 Blunting of the velocity at $\phi = 6.01\%$ compared to velocity profile at $\phi = 0\%$	45
5.9 Schematic of simulation setup used to investigate the effect of inter-particle distance on radial migration where the fluid flow and periodic boundary are in the x-direction (top). Effect of inter-particle distance (spacing) on radial stable position (bottom).	46
5.10 Particle trajectories as a function of a non-dimensional travel distance along the pipe axis x/R , plotted for several inter-particle distances increasing monotonically from $d1$ to $d13$. The results demonstrate smaller, yet faster, radial migration for larger inter-particle distances.	47
5.11 Effect of particle skewness on the radial stable position.	47
5.12 Effect of sphere size on the radial stable position.	48
5.13 Time snapshots of a flexible cantilever moving in vacuum under the action of gravity. The darker colors denote earlier stages of the motion.	49

Figure	Page
5.14 Convergence test of a soft cantilever beam falling under gravity. Trajectory of the beam tip is shown for different discretization resolution.	49
5.15 Comparison of the dynamics of a rigid cylinder and of a corresponding stiff deformable beam under accelerating channel flow: (a) beam orientation; (b) center velocity.	51
5.16 Motion of a cantilever beam in fluid of different viscosities: (a) tip displacement in x direction; (b) tip displacement in z direction.	52
5.17 Arrays of flexible cantilever beams in laminar channel flow. The beams, laid out in a uniform grid, are anchored at an angle of 30° in the direction of the flow.	53
5.18 Motion of a cantilever beam of different elasticity modulus in laminar channel flow: (a) tip displacement in x direction; (b) tip displacement in z direction.	54
6.1 Scaling analysis of Chrono::Fluid for rigid body dynamics, flexible body dynamics, and fluid dynamics as function of problem size (N_r , N_f , and N_m , respectively). The coefficients of determination, R^2 , are specified to each linear regression. Virtually exact linear trends are indicated since $R^2 \approx 1$	57
6.2 Scaling analysis of Chrono::Fluid for FSI problems: simulation time as function of combined problem size. In this experiment, the volume of the simulation domain is increased, up to 32 times the volume of the initial domain, leading to proportional increases in the number of SPH markers and of solid objects (both rigid and flexible bodies) as shown in the bottom plot (see also Table 6.6). As illustrated by the top graph, the simulation time for an one-step dynamics update varies linearly with problem size. The coefficients R^2 are specified for each linear regression.	60
7.1 Numerical investigation of flow cytometry using asymmetric channel: (a) experiment setup showing the channel and the particle feeding valve; (b) snapshot of the numerical simulation; (c) channel profile used in experiment and simulation. All dimensions are in mm.	64
7.2 Direct numerical simulation of a dense suspension of ellipsoids in 3D square channel flow. Only the mid-section of the flow, which shows the rigid ellipsoids suspended in the fluid, is shown. The coloring scheme represents fluid velocity: from zero (blue) to maximum (red).	65

Figure	Page
7.3 Direct numerical simulation of a dense suspension of ellipsoids in 3D square channel flow. The fluid was removed to show a perspective view of the rectangular channel and the ellipsoids.	66
7.4 Example simulation of flow in porous media. The entire domain is shown in (a). For a clear visualization, the fluid and porous matrix are partially removed in (b) and (c), respectively. The color represents fluid velocity: from zero (blue) to maximum (red). The maximum flow velocity is approximately 6 mm/s, i.e. $Re \approx 6$	69
7.5 Example simulation of flow over a dense array of interacting beams.	70
7.6 Example simulation of flow of rigid bodies within an array of deformable beams. For a clear visualization, only parts of the domain are shown in each picture: (a) beams; (b) beams and rigid bodies; (c) beams, rigid bodies and fluid flow, cut at different sections. The color represents the velocity: from zero (blue) to maximum (red), with $V_{max}^{fluid} = 0.045$ m/s, $V_{max}^{rigid} = 0.041$ m/s, and $V_{max}^{beam} = 0.005$ m/s.	71
7.7 Flow of a suspension of rigid and deformable objects. Parts of the rigid and fluid phase are removed for a clearer illustration.	72
8.1 A typical staggered grid cell: (a) positions of velocities and pressure; (b) indexing convention.	74
8.2 Schematic of the domain indexing in staggered grid taxonomy.	75
8.3 Cavity flow solution using staggered grid at $Re = 100$. The contours show: (a) velocity in x direction; (b) velocity in y direction; (c) magnitude of velocity; (d) pressure.	76
8.4 Velocity profile of transient Poiseuille flow obtained from implicit SPH simulation (dots) and series solution (continuous lines).	89
8.5 Offshore wind turbine designs for shallow and deep water.	91
8.6 Offshore structure design to reduce the ice loading: (a) breaking ice cones installed at offshore substructures (Barker <i>et al.</i> [9]); (b) crushing failure mechanism of ice against a vertical face; (c) flexural failure mechanism of ice against an ice cone.	92
8.7 Comparison of ice pressure to ice strength ratio versus substructure diameter to ice thickness ratio as recommended by DNV [130] to that obtained from field measurements (field data from Timco <i>et al.</i> [127], Kärnä <i>et al.</i> [58], Brown & Määtänen [12].	93

Figure	Page
A.1 Illustration of half indices used for interpolation of velocities in: (a) momentum in x direction; (b) momentum in y direction.	110

DISCARD THIS PAGE

NOMENCLATURE

AABB	Axis Aligned Bounding Box
ALE	Arbitrary Lagrangian-Eulerian [method]
ANCF	Absolute Nodal Coordinate Formulation
B	Byte
BC	Boundary Condition
BCE	Boundary Condition Enforcing [marker]
BDF	Backward Differentiation Formula [integration method]
COV	Coefficient Of Variation
CPU	Central Processing Unit
CUDA	Compute Unified Device Architecture [language extensions for GPU computing]
DAE	Differential Algebraic Equations
DEM	Discrete Element Method
DLM	Distributed Lagrange Multiplier [method]
DOF	Degree(s) Of Freedom
EE	Eulerian-Eulerian
EL	Eulerian-Lagrangian

EOM	Equations Of Motion
FEM	Finite Element Method
FDM	Fictitious Domain Method [method]
Flops	Floating point operation per second
FSI	Fluid-Solid Interaction [problem]
GPU	Graphics Processing Unit [card]
GPC	Graphics Processing Cluster
HPC	High Performance Computing
LBM	Lattice Boltzmann Method
LL	Lagrangian-Lagrangian
LNS	Lagrangian Numerical Simulation
MPI	Message Passing Interface [standard]
SBEL	Simulation Based Engineering Laboratory
SIMD	Single Instruction Multiple Data
SPMD	Single Program Multiple Data
SPH	Smoothed Particle Hydrodynamics [method]
SM	Stream Multiprocessor
SP	Scalar Processor
XSPH	Extended Smoothed Particle Hydrodynamics [method]
a	sphere radius

c_s	sound speed
D	diameter
d	particles distance
E	modulus of elasticity
$\mathbf{e}$	nodal coordinates
$\mathbf{F}$	force applied to a rigid body
$\mathbf{F}^{lub}$	lubrication force
$\mathbf{f}$	fluid body force
$\mathbf{f}_{ij}^k$	partial lubrication force between solids i and j
$\mathbf{G}$	auxiliary matrix constructed from the rotation quaternion
h	kernel function's characteristic length
$\mathbf{I}$	identity matrix
$\mathbf{J}'$	rigid body mass moment of inertia, represented in local reference frame
l	element length in ANCF
M	rigid body mass
m	marker mass
N_m	number of markers
N_r	number of rigid bodies
N_f	number of flexible bodies
$\mathbf{n}$	unit vector normal to the surface or boundary

n_b	number of bins
n_n	number of nodes
n_e	number of elements
p	pressure
$\mathbf{Q}$	generalized element force in ANCF
$\mathbf{q}$	rotation quaternion
R	pipe radius
$\mathbf{r}$	markers relative distance (SPH); global position vector on beam (ANCF)
r	particle offset from the pipe axis
Re	pipe Reynolds number
Re_c	channel Reynolds number
Re_p	particle Reynolds number
$\mathbf{S}$	shape function matrix for ANCF beam element
S	support domain of SPH kernel function
s	minimum distance between the surfaces
$\mathbf{T}'$	torque applied to a rigid body, represented in local reference frame
t	time
t^*	non-dimensional time
$\mathbf{V}$	rigid body velocity
V	mean flow velocity

$V_{\max}$	maximum flow speed
$\mathbf{v}$	flow velocity
$\mathbf{v}_a$	velocity of marker a
$\langle \mathbf{v}_a \rangle$	XSPH approximation of flow velocity for marker a
W	SPH kernel function
w	channel width
$\mathbf{X}$	rigid body position
$\mathbf{x}$	marker position
Δ_c	cut-off value for lubrication force model
Δ_i	cut-off value for dry friction model
ε_{11}	axial strain
κ	magnitude of the curvature vector
λ	sphere to pipe size ratio
μ	fluid viscosity
ϕ	volumetric concentration
Π_{ab}	viscose force between markers a and b
ρ	density of fluid
ρ_s	density of solid
τ	time step size
τ_m	multi-rate integration factor

ω'_i	rigid body angular velocity
ξ	element characteristic distance

ABSTRACT

This work is concerned with formulating and validating a Lagrangian-Lagrangian (LL) approach for the simulation of fully resolved Fluid Solid/Structure Interaction (FSI) problems. In the proposed approach, the method of Smoothed Particle Hydrodynamics (SPH) is used to simulate the fluid dynamics in a Lagrangian framework. The solid phase is a general multibody dynamics system composed of a collection of interacting rigid and deformable objects. While the motion of arbitrarily shaped rigid objects is approached in a classical 3D rigid body dynamics framework, the Absolute Nodal Coordinate Formulation (ANCF) is used to model the deformable components, thus enabling the investigation of compliant elements that experience large deformations with entangling and self-contact. The dynamics of the two phases, fluid and solid, are coupled with the help of Lagrangian markers, referred to as Boundary Condition Enforcing (BCE) markers used to impose no-slip and impenetrability conditions. The BCE markers, which are associated both with the solid suspended bodies and with any confining boundary walls, are distributed in a narrow layer on and below the surface of solid objects. The solid-solid interaction is known to have a crucial effect on the small-scale behavior of fluid-solid mixtures. The dry encounter of solid surfaces is resolved herein through a penalty based approach. However, using this model for the short range interaction of solid surfaces in fluid media does not follow the real physics of wet interaction. To accommodate this, a lubrication force model consistent with SPH is adopted herein to capture the short range interaction of arbitrary shapes in fluid. The ensuing fluid-solid interaction forces are mapped into generalized forces on the rigid and flexible bodies and subsequently used to update the dynamics of the solid objects according to the rigid or flexible body motion.

The software implementation of the proposed LL approach is used to investigate the two- and three-dimensional (2D, 3D) pipe flow of dilute suspensions of macroscopic neutrally buoyant rigid

bodies at flow regimes with Reynolds numbers (Re) between 0.1 and 1400. Several validation studies demonstrated good predictive attributes of the proposed LL approach. The simulation results obtained indicate that (1) rigid body rotation affects the behavior of a particle laden flow; (2) an increase in neutrally buoyant particle size decreases radial migration; (3) a decrease in inter-particle distance slows down the migration and shifts the stable position further away from the channel axis; (4) rigid body shape influences the stable radial distribution of particles; (5) particle migration is influenced, both quantitatively and qualitatively, by the Reynolds number; and (6) the stable radial particle concentration distribution is affected by the initial concentration.

The software implementation of the approach, called Chrono::Fluid, is available as an open-source software package. In the multi-threaded, multi-scale High Performance Computing (HPC) implementation the collective system states are integrated in time using an explicit, multi-rate scheme. To alleviate the heavy computational load, the overall algorithm leverages parallel computing on Graphics Processing Unit (GPU) cards. Performance and scaling analysis are provided for simulations scenarios involving one or multiple phases with up to tens of thousands of solid objects.

The parallel LL simulation framework developed herein does not impose restrictions on the shape or size of the rigid bodies and was used in the numerical investigation of several problems. This included the real scale analysis of flow cytometry using microfluidic channel, dense and colloidal particle suspension, microscale simulation of flow in porous media, deformation and vibration study of immersed deformable objects, and interacting rigid and flexible objects in channel flow.

Chapter 1

Introduction

Engineers commonly rely on prototypes and physical testing when performing design and analysis tasks. Unfortunately, such undertakings can be expensive and time consuming. Because computational hardware continues to advance in terms of both processing speed and memory size, a trend is growing in which computer simulation is used to augment and, in some cases, replace large amounts of experimental work. With increasing computational power, engineers are able to perform faster, larger, and more accurate simulations. Computer simulation has several advantages over physical experiments. Through simulation, engineers may study a range of parameter values that would prove costly or impractical to study experimentally. Moreover, computer simulation can produce representative data that experimental measurements could never achieve. Experimental insights are limited by the position, fidelity, and number of sensors, whereas a simulation inherently tracks the state of every component of the system. For example, simulation can generate, in a non-intrusive fashion, the set of forces acting between all the individual bodies in a suspension flow.

Current simulation capabilities are sometimes inadequate to capture phenomena of interest. This problem is manifest when simulating the dynamics of Fluid-Solid Interaction (FSI) systems, which may contain tens of thousands of rigid and deformable bodies that interact directly or through the fluid media. Solving such large problems will require significant improvements in terms of both algorithms and implementation.

To alleviate computational limitations, numerical simulation approaches devised for the general category of FSI problems usually suppress some physics depending on the specific application. For instance, several approaches have been proposed to study characteristics of the flow of

particle suspensions. These include Eulerian-Eulerian (EE) approaches, where the solid phase is considered as a continuum [30, 35, 133]; Lagrangian particle tracking, also known as Lagrangian Numerical Simulation (LNS), approaches which either consider a one-way coupling of fluid and solid phase, or else introduce a collective momentum exchange term to the fluid equation [2, 77]; Eulerian-Lagrangian (EL) approaches, where the Lagrangian solid phase moves with/within the Eulerian grid used for fluid simulation [36, 50, 64]; and Lagrangian-Lagrangian (LL) approaches, where both phases are modeled within a Lagrangian framework [89, 104, 105]. As in EE methodologies, LNS approaches rely on empirical forms of hydrodynamic fluid-solid forces, determined mostly for dilute conditions where the particle-particle interaction is neglected.

Similar approaches are also applied to the fluid-structure interaction. In this document, the focus is primarily on the LL approaches, particularly those geared towards large deformation favored by the multibody dynamics community. Results of studies on problems involving small structural deformation using a Lagrangian representation of fluid flow are provided in [1, 4, 67].

The body of work on FSI problems using a Lagrangian fluid representation and large structural deformation is very limited. Schörgenhumer *et al.* [114] presented a co-simulation approach for FSI problems. In their approach, they used a heuristic force field to couple the dynamics of the fluid and flexible objects. The suggested force field, which involves some heuristic parameters to enforce the fluid-solid coupling, cannot approximate the FSI interaction at a resolution finer than that of the fluid discretization. In this sense, it is equivalent to all other approaches proposed for the implementation of wall boundary condition with the caveats that: (1) the procedure of finding the minimum distance between fluid markers and solid surfaces can be prohibitively tedious, particularly for complex shapes; and (2) the wrong choice of heuristic parameters at a certain flow condition can result in either an inexact coupling or a stiff force model which can lead to numerical instability. Additionally, little, if any, is said about the performance of the co-simulation approach. Similarly, Hu *et al.* [52] adopted the method of moving boundary to approach the simulation of FSI problem involving flexible components.

This body of work collects all of the previous contributions described mainly in Pazouki and Negrut [104, 105] and Pazouki *et al.* [103, 106, 107], into a unified framework for the simulation of

FSI problems using advanced computing. The proposed approach is more general than alternative FSI simulation frameworks [52, 114] in that it introduces a solid-solid interaction model required for many-body FSI problems. In fact, the main focus of [52] is on the motion of the non-immersed flexible component in a problem involving a cavity as a single rigid body. In the present work, support for many-body FSI problems, such as those encountered in suspension and polymer flow, is provided by incorporating a lubrication force model and scalable rigid and flexible body dynamics. In addition, validation and investigation studies are provided for the suspension of interacting rigid and flexible components. Finally, a high performance implementation that leverages parallel computing on Graphical Processing Unit (GPU) cards is provided. Complete scaling and time analyses performed herein demonstrate roughly a typical ten-fold speedup compared to the results provided in [52] for problems of comparable size.

This document is organized as follows. The modeling components for the simulation of fluid as well as rigid and flexible body dynamics are explained in Chapter 2. The discussion of key numerical algorithms, including the choice of boundary conditions, integration algorithm, fluid-solid coupling, and solid-solid short range interaction makes up Chapter 3. The HPC-based computational methodology is explained in Chapter 4. Physics based validations and investigations studies are presented in Chapter 5. Chapter 6 discusses the performance and scalability analysis of the framework. Chapter 7 describes the application of the proposed framework in the study of several problems including flow in porous media, flow cytometry using microfluidic channel, and dense suspension of rigid and flexible objects. Chapter 8 outlines: (a) several problems that can be approached via the simulation framework; and (b) several research projects that are at different stages of completion. A set of conclusions is provided in Chapter 9.

Chapter 2

Modeling

The proposed FSI simulation framework leverages the method of Smoothed Particle Hydrodynamics (SPH) for the simulation of the fluid flow, a Newton-Euler formulation for rigid body dynamics, and an Absolute Nodal Coordinate Formulation (ANCF) for flexible body dynamics. These algorithmic components are described in more detail in the following subsections.

2.1 Smoothed Particle Hydrodynamics (SPH)

The term *smoothed* in SPH refers to the approximation of point properties via a smoothing kernel function W , defined over a finite support domain S . This approximation reproduces functions with up to 2nd order of accuracy, provided the kernel function: (1) approaches the Dirac delta function, δ , as the size of the support domain tends to zero, that is $\lim_{h \rightarrow 0} W(\mathbf{r}, h) = \delta(\mathbf{r})$, where $\mathbf{r}$ is the spatial distance and h is a characteristic length that defines the kernel smoothness; (2) is symmetric, i.e., $W(\mathbf{r}, h) = W(-\mathbf{r}, h)$; and (3) is normal, i.e., $\int_S W(\mathbf{r}, h) d\mathbb{V} = 1$, where S is the domain of function W and $d\mathbb{V}$ denote the differential volume. A typical spatial function $f(\mathbf{x})$ is then approximated by $\langle f(\mathbf{x}) \rangle$ as

$$\begin{aligned}
 f(\mathbf{x}) &= \int_S f(\mathbf{x}') \delta(\mathbf{x} - \mathbf{x}') d\mathbb{V} \\
 &= \int_S f(\mathbf{x}') W(\mathbf{x} - \mathbf{x}', h) d\mathbb{V} + O(h^2) \\
 &= \langle f(\mathbf{x}) \rangle + O(h^2) .
 \end{aligned} \tag{2.1}$$

To simplify notation, in the remainder of this document $f(\mathbf{x})$ is used to represent $\langle f(\mathbf{x}) \rangle$. Using Eq. (2.1) and the divergence theorem, the spatial derivatives of a function can be mapped to the derivatives of the kernel function. For instance, the gradient of a function can be written as

$$\begin{aligned} \nabla f(\mathbf{x}) = & \int_{\partial S} f(\mathbf{x}') W(\mathbf{x} - \mathbf{x}', h) \cdot \mathbf{n} d\mathbb{A} \\ & - \int_S f(\mathbf{x}') \nabla W(\mathbf{x} - \mathbf{x}', h) d\mathbb{V} , \end{aligned} \quad (2.2)$$

where ∂S is the boundary of S and $d\mathbb{A}$ is the differential area. By imposing an additional property for the kernel function, namely that (4) it approaches zero as $\mathbf{r}$ increases, i.e., $\lim_{\mathbf{r} \rightarrow \infty} W(\mathbf{r}, h) = 0$, the first term on the right hand side of Eq. (2.2) vanishes. Note that some additional considerations, which will be addressed later, are required for the SPH approximation near boundaries.

The term *particle* in SPH terminology indicates the discretization of the domain by a set of Lagrangian particles. To remove the ambiguity caused by the use of the term *rigid particles* in the context of FSI problems, the term *marker* will be used herein to refer to the SPH discretization unit. Each marker has mass m associated to the representative volume $d\mathbb{V}$ and carries all of the essential field properties. As a result, any field property at a certain location is shared and represented by the markers in the vicinity of that location. Accordingly, the value of a certain function at a given location is calculated as a weighted sum of the function values at the location of all nearby markers. The summation weights depend on the distance of the respective markers from the location of interest and the expression of the kernel function W . This leads for the second approximation embedded in SPH, which can be expressed as

$$\begin{aligned} f(\mathbf{x}) = & \int_S \frac{f(\mathbf{x}')}{\rho(\mathbf{x}')} W(\mathbf{x} - \mathbf{x}', h) \rho(\mathbf{x}') d\mathbb{V} \\ \simeq & \sum_b \frac{m_b}{\rho_b} f(\mathbf{x}_b) W(\mathbf{x} - \mathbf{x}_b, h) , \end{aligned} \quad (2.3)$$

where b is the marker index and ρ_b is the fluid density, smoothed at the marker location $\mathbf{x}_b$. The summation in Eq. (2.3) is over all markers whose support domain overlaps the location $\mathbf{x}$. Several other properties of the kernel functions are provided in [71]. For instance, the kernel function should be a positive and monotonically decreasing function of $\mathbf{r}$, which implies that the influence of

distant markers on field properties at a given location is less than that of nearby markers. Moreover, for acceptable computational performance and to avoid a quadratic computational complexity, kernel functions have a compact domain of influence with a radius R_s defined as some finite multiple κ of the characteristic length h , as shown in Figure 2.1. The methodology adopted herein relies on a cubic spline,

$$W(q, h) = \frac{1}{4\pi h^3} \times \begin{cases} (2 - q)^3 - 4(1 - q)^3, & 0 \leq q < 1 \\ (2 - q)^3, & 1 \leq q < 2 \\ 0, & q \geq 2 \end{cases}, \quad (2.4)$$

where $q \equiv |\mathbf{r}|/h$. This kernel function has a support domain with radius $2h$, i.e. $\kappa = 2$.

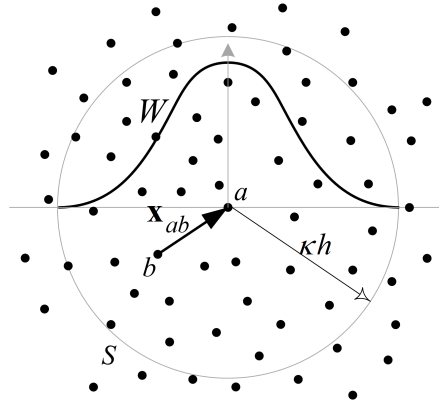


Figure 2.1: Illustration of the kernel W and its support domain S . SPH markers are shown as black dots. For 2D problems the support domain is a circle, while for 3D problems it is a sphere. The radius of the support domain is defined as a multiple, κ , of the kernel's characteristic length, h .

2.1.1 SPH for fluid dynamics

The momentum, i.e. Navier-Stokes, and continuity equations describing the fluid dynamics are given as

$$\frac{d\rho}{dt} = -\rho \nabla \cdot \mathbf{v} \quad (2.5)$$

and

$$\frac{d\mathbf{v}}{dt} = -\frac{1}{\rho} \nabla p + \frac{\mu}{\rho} \nabla^2 \mathbf{v} + \mathbf{f}, \quad (2.6)$$

where μ is the fluid viscosity, $\mathbf{f}$ is the body force, and $\mathbf{v}$ and p are the flow velocity and pressure, respectively. In order to obtain the SPH discretization of Eqs. (2.5) and (2.6), an equivalent and more general set of identities for spatial derivatives will be used herein:

$$\frac{\nabla p}{\rho} = \frac{p}{\rho^\sigma} \nabla \left(\frac{1}{\rho^{1-\sigma}} \right) + \rho^{\sigma-2} \nabla \left(\frac{p}{\rho^{\sigma-1}} \right), \quad (2.7)$$

$$\rho \nabla \cdot \mathbf{v} = \frac{\nabla \cdot (\rho^{\sigma-1} \mathbf{v}) - \mathbf{v} \cdot \nabla \rho^{\sigma-1}}{\rho^{\sigma-2}}. \quad (2.8)$$

Two frequently-used SPH discretizations are associated with $\sigma = 1$ and $\sigma = 2$. While $\sigma = 2$ is used mostly for the simulation of single phase flow, $\sigma = 1$ is favored for multi-phase flow since it results in equal and opposite inter-phase forces. The coherency condition of the discretization requires the same value of σ in Eqs. (2.7) and (2.8). However, as Oger *et al.* suggested [94], this condition can be suppressed. Herein, the values of $\sigma = 1$ and $\sigma = 2$ are used in Eqs. (2.7) and (2.8), respectively. This model has been widely used for the single phase flow simulation, see for instance [84]. As a result, Eqs. (2.5) and (2.6) are discretized at an arbitrary location $\mathbf{x} = \mathbf{x}_a$ within the fluid domain as

$$\dot{\rho}_a = \frac{d\rho_a}{dt} = \rho_a \sum_b \frac{m_b}{\rho_b} (\mathbf{v}_a - \mathbf{v}_b) \cdot \nabla_a W_{ab}, \quad (2.9)$$

and

$$\dot{\mathbf{v}}_a = \frac{d\mathbf{v}_a}{dt} = - \sum_b m_b \left[\left(\frac{p_a}{\rho_a^2} + \frac{p_b}{\rho_b^2} \right) \nabla_a W_{ab} + \Pi_{ab} \right] + \mathbf{f}_a. \quad (2.10)$$

In the above equations, quantities with subscripts a and b are associated with markers a and b (see Figure 2.1), respectively. It is important to note that these quantities are different from the corresponding physical quantities at locations $\mathbf{x}_a$ and $\mathbf{x}_b$. For instance, ρ_a , i.e. the density associated to marker a , is not essentially the same as density at location $\mathbf{x}_a$, the latter being computed through as indicated in Eq. 2.3.

The viscosity term Π_{ab} is defined as

$$\Pi_{ab} = - \frac{(\mu_a + \mu_b) \mathbf{x}_{ab} \cdot \nabla_a W_{ab}}{\bar{\rho}_{ab}^2 (x_{ab}^2 + \varepsilon \bar{h}_{ab}^2)} \mathbf{v}_{ab}, \quad (2.11)$$

where $\mathbf{x}_{ab} = \mathbf{x}_a - \mathbf{x}_b$, $W_{ab} = W(\mathbf{x}_{ab}, h)$, ∇_a is the gradient with respect to $\mathbf{x}_a$, i.e., $\partial/\partial \mathbf{x}_a$, and ε is a regularization coefficient. Quantities with an over-bar are averages of the corresponding quantities for markers a and b . For instance, $\bar{\rho}_{ab} = (\rho_a + \rho_b)/2$.

Alternative viscosity discretizations include:

1. The model suggested in [20]:

$$\Pi_{ab}^* = -\frac{\mu_a \mu_b \mathbf{x}_{ab} \cdot \mathbf{v}_{ab}}{(\mu_a + \mu_b) \rho_a \rho_b (x_{ab}^2 + \varepsilon \bar{h}_{ab}^2)} \nabla_a W_{ab}, \quad (2.12)$$

2. Direct discretization of ∇^2 operator [82, 84],

3. The class of artificial viscosity models introduced in [73, 79, 83].

However, using Eq. 2.11 is preferred since it has the following properties: (i) it ensures that the viscous force is along the shear direction, $\mathbf{v}_{ab}$, instead of the particles center line, $\mathbf{x}_{ab}$; (ii) it is less sensitive to local velocities by avoiding the numerical calculation of second derivatives; (iii) it allows for better computational efficiency by removing the nested loop required for the computation of the ∇^2 operator; and (iv) it is stated in terms of physical properties, rather than model parameters like those in artificial viscosity, which are introduced primarily for numerical stabilization through damping. In the simulation of transient Poiseuille flow discussed in Chapter 5, although virtually exact results were obtained using Eq. 2.11 [105], the error caused by implementing either Π_{ab}^* or artificial viscosity were non-negligible.

In the weakly compressible SPH model, the pressure p is evaluated using an equation of state [82]

$$p = \frac{c_s^2 \rho_0}{\gamma} \left\{ \left(\frac{\rho}{\rho_0} \right)^\gamma - 1 \right\}, \quad (2.13)$$

where ρ_0 is the reference density of the fluid, γ tunes the stiffness of the pressure-density relationship, and c_s is the speed of sound. To reduce the stiffness caused by using Eq. (2.13) and yet keep the compressibility below a threshold value, c_s is adjusted depending on the maximum speed of the flow, $V_{\max}$. For instance, the values of $\gamma = 7$ and $c_s = 10V_{\max}$, were used throughout this work to allow a maximum of 1% flow compressibility [82].

The fluid flow equations (2.9) and (2.10) are solved together with

$$\dot{\mathbf{x}}_a = \frac{d\mathbf{x}_a}{dt} = \mathbf{v}_a \quad (2.14)$$

to update the position of the SPH markers.

The original SPH summation formula calculates the density according to

$$\rho_a = \sum_b m_b W_{ab}. \quad (2.15)$$

Equation (2.9), which evaluates the time derivative of the density, was preferred to the above since it produces a smooth density field, works well for markers close to the boundaries (the free surface, solid, and wall), and does not exhibit the large variations in the density field introduced when using Eq. (2.15) close to the boundaries. However, Eq. (2.9) does not guarantee consistency between a marker's density and the associated mass and volume [11, 81, 84]. The so-called *density re-initialization technique* [21] attempts to address this issue by using Eq. (2.9) at each time step and periodically; i.e., every n time steps, using Eq. (2.15) to correct the mass-density inconsistency. The results reported herein were obtained with $n = 10$. The Moving Least Squares method or a normalized version of Eq. (2.15) could alternatively be used to address the aforementioned issues, see [21, 27].

Finally, the methodology proposed employs the extended SPH approach (XSPH), which prevents extensive overlap of the markers' support domain and enhances flow incompressibility [80]. This correction takes into account the velocity of neighboring markers through a mean velocity evaluated within the support of a nominal marker a as

$$\hat{\mathbf{v}}_a = \mathbf{v}_a + \Delta \mathbf{v}_a, \quad (2.16)$$

where

$$\Delta \mathbf{v}_a = \zeta \sum_b \frac{m_b}{\bar{\rho}_{ab}} (\mathbf{v}_b - \mathbf{v}_a) W_{ab}, \quad (2.17)$$

and $0 \leq \zeta \leq 1$ adjusts the contribution of the neighbors' velocities. All the results reported herein were obtained with $\zeta = 0.5$. The modified velocity calculated from Eq. (2.16) replaces the original velocity in the density and position update equations, but not in the momentum equation [21].

2.2 General rigid body dynamics

Given all external forces and torques including impact, contact, and intra-phase interactions, the dynamics of the rigid bodies is fully characterized by the Newton-Euler equations of motion

(EOM), see for instance [42]:

$$\frac{d\mathbf{V}_i}{dt} = \frac{\mathbf{F}_i}{M_i}, \quad (2.18)$$

$$\frac{d\mathbf{X}_i}{dt} = \mathbf{V}_i, \quad (2.19)$$

$$\frac{d\boldsymbol{\omega}'_i}{dt} = \mathbf{J}'_i{}^{-1} \left(\mathbf{T}'_i - \tilde{\boldsymbol{\omega}}'_i \mathbf{J}'_i \boldsymbol{\omega}'_i \right), \quad (2.20)$$

$$\frac{d\mathbf{q}_i}{dt} = \frac{1}{2} \mathbf{G}_i^T \boldsymbol{\omega}'_i, \quad (2.21)$$

and

$$\mathbf{q}_i^T \mathbf{q}_i - 1 = 0, \quad (2.22)$$

where $\mathbf{F}_i, \mathbf{T}'_i, \mathbf{X}_i, \mathbf{V}_i, \boldsymbol{\omega}'_i \in \mathbb{R}^3$ denote the force, torque, position, velocity, and angular velocity associated to body $i, i = 1, 2, \dots, n_b$, respectively. The quantity $\mathbf{q}_i$ denotes the rotation quaternion, while M_i and $\mathbf{J}'_i$ are the constant mass and moment of inertia, respectively. Quantities with a prime symbol are represented in the rigid body local reference frame. Given $\mathbf{a} = [a_x, a_y, a_z]^T \in \mathbb{R}^3$ and a quaternion for body $i, \mathbf{q}_i = [q_x^i, q_y^i, q_z^i, q_w^i]^T \in \mathbb{R}^4$, the auxiliary matrices $\tilde{\mathbf{a}}$ and $\mathbf{G}_i$ are defined as [42]

$$\tilde{\mathbf{a}} = \begin{bmatrix} 0 & -a_z & a_y \\ a_z & 0 & -a_x \\ -a_y & a_x & 0 \end{bmatrix} \quad \text{and} \quad \mathbf{G}_i = \begin{bmatrix} -q_y^i & q_x^i & q_w^i & -q_z^i \\ -q_z^i & -q_w^i & q_x^i & q_y^i \\ -q_w^i & q_z^i & -q_y^i & q_x^i \end{bmatrix}. \quad (2.23)$$

2.3 Absolute Nodal Coordinate Formulation (ANCF) of flexible objects

The ANCF formulation [119], which allows for large deformations and large body rotations, is adopted herein for the simulation of flexible bodies suspended in the fluid. While extension to other elastic elements is straightforward, the current implementation only supports gradient deficient ANCF beam elements, which are used to model slender flexible bodies composed of n_e adjacent ANCF beam elements. The flexible bodies are modeled using a number $n_n = n_e + 1$ of equally-spaced node beam elements, each represented by 6 coordinates, $\mathbf{e}_j = [\mathbf{r}_j^T, \mathbf{r}_{j,x}^T]^T, j = 0, 1, \dots, n_e$; i.e., the three components of the global position vector of the node, $\mathbf{r}_j$, and the three components of its slope, $\mathbf{r}_{j,x} = d\mathbf{r}_j/dx$, where x is the parametric distance on the beam. This is therefore

equivalent to a model using n_e ANCF beam elements with $6 \times n_n$ continuity constraints, but is more efficient in that it uses a minimal set of coordinates. It is worth noting that formulations using gradient deficient ANCF beam elements display no shear locking problems [34, 115, 118] and, due to the reduced number of nodal coordinates, are more efficient than fully parametrized ANCF elements. However, gradient deficient ANCF beam elements cannot describe a rotation about its axis and therefore cannot model torsional effects.

Consider first a single ANCF beam element of length ℓ . The global position vector of an arbitrary point on the beam center line, specified through its element spatial coordinate $0 \leq x \leq \ell$, is then obtained as

$$\mathbf{r}(x, \mathbf{e}) = \mathbf{S}(x)\mathbf{e}, \quad (2.24)$$

where $\mathbf{e} = [\mathbf{e}_l^T, \mathbf{e}_r^T]^T \in \mathbb{R}^{12}$ is the vector of element nodal coordinates. With $\mathbf{I}$ being the 3×3 identity matrix, the shape function matrix $\mathbf{S} = [S_1\mathbf{I} \ S_2\mathbf{I} \ S_3\mathbf{I} \ S_4\mathbf{I}] \in \mathbb{R}^{3 \times 12}$ is defined using the shape functions [119]

$$\begin{aligned} S_1 &= 1 - 3\xi^2 + 2\xi^3 \\ S_2 &= \ell(\xi - 2\xi^2 + \xi^3) \\ S_3 &= 3\xi^2 - 2\xi^3 \\ S_4 &= \ell(-\xi^2 + \xi^3), \end{aligned} \quad (2.25)$$

where $\xi = x/\ell \in [0, 1]$. The element EOM are then written as

$$\mathbf{M}\ddot{\mathbf{e}} + \mathbf{Q}^e = \mathbf{Q}^a, \quad (2.26)$$

where $\mathbf{Q}^e$ and $\mathbf{Q}^a$ are the generalized element elastic and applied forces, respectively, and $\mathbf{M} \in \mathbb{R}^{12 \times 12}$ is the constant symmetric consistent element mass matrix defined as

$$\mathbf{M} = \int_{\ell} \rho_s A \mathbf{S}^T \mathbf{S} dx. \quad (2.27)$$

The generalized element elastic forces are obtained from the strain energy expression [119] as

$$\mathbf{Q}^e = \int_{\ell} EA \varepsilon_{11} \left(\frac{\partial \varepsilon_{11}}{\partial \mathbf{e}} \right)^T dx + \int_{\ell} EI \kappa \left(\frac{\partial \kappa}{\partial \mathbf{e}} \right)^T dx, \quad (2.28)$$

where $\varepsilon_{11} = (\mathbf{r}_x^T \mathbf{r}_x - 1)/2$ is the axial strain and $\kappa = \|\mathbf{r}_x \times \mathbf{r}_{xx}\|/\|\mathbf{r}_x\|^3$ is the magnitude of the curvature vector. The required derivatives of the position vector $\mathbf{r}$ can be easily obtained from

Eq. (2.24) in terms of the derivatives of the shape functions as $\mathbf{r}_x(x, \mathbf{e}) = \mathbf{S}_x(x)\mathbf{e}$ and $\mathbf{r}_{xx}(x, \mathbf{e}) = \mathbf{S}_{xx}(x)\mathbf{e}$.

External applied forces, in particular the forces due to the interaction with the fluid (see Sect. 3.2), are included as concentrated forces at a BCE marker. The corresponding generalized forces are obtained from the expression of the virtual work as

$$\mathbf{Q}^a = \mathbf{S}^T(x_a)\mathbf{F}, \quad (2.29)$$

where $\mathbf{F}$ is the external point force and the shape function matrix is evaluated at the projection onto the element's center line of the force application point. The generalized gravitational force can be computed as

$$\mathbf{Q}^g = \int_{\ell} \rho_s A \mathbf{S}^T \mathbf{g} dx. \quad (2.30)$$

In the above expressions, ρ_s represents the element mass density, A is the cross section area, E is the modulus of elasticity, and I is the second moment of area.

The EOM for a slender flexible body composed of n_e ANCF beam elements are obtained by assembling the elemental EOMs of Eq. (2.26) and taking into consideration that adjacent beam elements share 6 nodal coordinates. Let $\hat{\mathbf{e}} = [\mathbf{e}_0^T, \mathbf{e}_1^T, \dots, \mathbf{e}_{n_e}^T]^T$ be the set of independent nodal coordinates; then the nodal coordinates for the j -th element can be written using the mapping

$$\begin{bmatrix} \mathbf{e}_l \\ \mathbf{e}_r \end{bmatrix}_j = \mathbf{B}_j \hat{\mathbf{e}}, \text{ with } \mathbf{B}_j = \begin{bmatrix} \mathbf{0} & \mathbf{0} & \dots & \mathbf{I}_3 & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \dots & \mathbf{0} & \mathbf{I}_3 & \dots & \mathbf{0} \end{bmatrix} \quad (2.31)$$

and the assembled EOMs are obtained, from the principle of virtual work, as follows. Denoting by $\mathbf{M}_j$ the element mass matrix of Eq. (2.27) for the j -th ANCF beam element,

$$\mathbf{M}_j = \begin{bmatrix} \mathbf{M}_{j,ll} & \mathbf{M}_{j,lr} \\ \mathbf{M}_{j,rl} & \mathbf{M}_{j,rr} \end{bmatrix}, \quad (2.32)$$

where $\mathbf{M}_{j,lr} = \mathbf{M}_{j,rl}^T$ and all sub-blocks have dimension 6×6 . Here, l denotes the *left* end of the beam element, i.e., the node characterized by the nodal coordinates $\mathbf{e}_{j-1}$, while r corresponds to the node with coordinates $\mathbf{e}_j$. With a similar decomposition of a generalized element force into

$$\mathbf{Q}_j = \begin{bmatrix} \mathbf{Q}_{j,l} \\ \mathbf{Q}_{j,r} \end{bmatrix} \quad (2.33)$$

one obtains

$$\hat{\mathbf{M}}\ddot{\mathbf{e}} = \hat{\mathbf{Q}}^a - \hat{\mathbf{Q}}^e \quad (2.34)$$

where

$$\hat{\mathbf{M}} = \begin{bmatrix} \mathbf{M}_{1,ll} & \mathbf{M}_{1,lr} & & & \\ \mathbf{M}_{1,rl} & \mathbf{M}_{1,rr} + \mathbf{M}_{2,ll} & \mathbf{M}_{2,lr} & & \\ & \mathbf{M}_{2,rl} & \mathbf{M}_{2,rr} + \mathbf{M}_{3,ll} & & \\ & & & \ddots & \\ & & & & \mathbf{M}_{n_e,rr} \end{bmatrix} \quad (2.35)$$

$$\hat{\mathbf{Q}}^a - \hat{\mathbf{Q}}^e = \begin{bmatrix} \sum \mathbf{Q}_{1,l}^a \\ \sum \mathbf{Q}_{1,r}^a + \sum \mathbf{Q}_{2,l}^a \\ \sum \mathbf{Q}_{2,r}^a + \sum \mathbf{Q}_{3,l}^a \\ \vdots \\ \sum \mathbf{Q}_{n_e,r}^a \end{bmatrix} - \begin{bmatrix} \mathbf{Q}_{1,l}^e \\ \mathbf{Q}_{1,r}^e + \mathbf{Q}_{2,l}^e \\ \mathbf{Q}_{2,r}^e + \mathbf{Q}_{3,l}^e \\ \vdots \\ \mathbf{Q}_{n_e,r}^e \end{bmatrix}. \quad (2.36)$$

Inclusion of additional kinematic constraints, e.g., anchoring the beam at one end to obtain a flexible cantilever or fixing its position to obtain a flexible pendulum, can be done either by formulating the EOM as differential-algebraic equations or by deriving an underlying ODE after explicitly eliminating the corresponding constrained nodal coordinates. The latter approach was used in all simulations involving flexible beams anchored at one end, pivoted at one end, and pivoted at both ends as further discussed in Chapters 5 and 6.

Chapter 3

Numerical solution

The modeling components outlined in Chapter 2 describe frameworks for the simulation of fluid, rigid-, and flexible-body dynamics, independently. Several details are left to be addressed regarding the boundary condition treatment, self and intra-phase interactions, and time integration algorithms, which are all addressed in this chapter.

3.1 Boundary conditions

Boundary conditions have significant effect on the solution of the Navier-Stokes equations. The most prevalent boundary conditions, namely free-surface, periodic, inflow and outflow, and rigid wall boundary conditions are discussed herein in conjunction with the weakly compressible, time explicit, SPH model. Alternative boundary conditions, consistent with the implicit, constrained, SPH formalism are discussed in Chapter 8.

3.1.1 Fluid free-surface

Imposing the free-surface boundary condition in SPH is more straightforward than in Eulerian approaches in that there is no need for surface tracking or mesh consideration. Generally, no special treatment is required for free-surface as long as Eq. (2.5) is used exclusively for density update. The fluid surface can be approximated by markers whose partition of unity, $\sum_b (m_b/\rho_b) W_{ab}$, is close to a cut-off threshold. However, some consideration is required if Eq. (2.15) is used to calculate density exclusively or in combination with Eq. (2.5), i.e. the density re-initialization

technique. This is due to the lack of markers close to the surface, which results in incompleteness of the partition of unity (see Figure 3.1). This problem can be easily modified by density normalization as suggested in [109].

Shown in Figure 3.1 is a free-surface boundary, with I and B denoting the set of interior and boundary fluid markers, respectively. To obtain the density of a boundary marker, for instance a , a set of virtual missing markers is assumed outside the boundary. The density of marker a can be obtained accordingly as

$$\rho_a = \sum_{b \in I} m_b W_{ab} + \sum_{b \in B} m_b W_{ab} + \sum_{b \in E} m_b W_{ab}. \quad (3.1)$$

Similarly, the partition of unity at the location of marker a results in

$$1 = \sum_{b \in I} \frac{m_b W_{ab}}{\rho_b} + \sum_{b \in B} \frac{m_b W_{ab}}{\rho_b} + \sum_{b \in E} \frac{m_b W_{ab}}{\rho_b}. \quad (3.2)$$

The last terms in the right hand sides of Eqs. (3.1) and (3.2) are zero due to the lack of markers. Therefore, by assuming a constant density, $\rho = \rho_a$, for the set of boundary markers, multiplying Eq. (3.2) by ρ_a and subtracting it from Eq. (3.1), ρ_a is obtained as

$$\rho_a = \frac{\sum_b m_b W_{ab}}{\sum_b \frac{m_b}{\rho_b} W_{ab}}. \quad (3.3)$$

Equation (3.3) replaces Eq. (2.15) for the free-surface boundary markers, i.e. markers whose partition of unity is smaller than a threshold value, say 95%.

Addressing the free-surface at small scale, e.g. at the scale of droplets and bubbles, requires the inclusion of surface tension. While Morris [85] approached this problem by using a color function to determine the surface curvature, Tartakovsky *et al.* [126] used an inter-marker short-range force to account for the surface tension. These problems are out of the scope of the current work and will not be considered herein.

3.1.2 Periodic BC

The periodic BC follows the simple rule of transferring markers and objects from one side of the domain to the other side as they leave the domain's boundary. The problem arises when markers

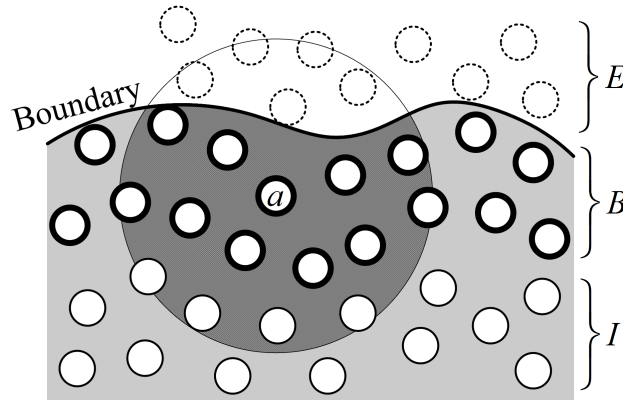


Figure 3.1: Fluid free-surface modelled with SPH. The set of missing exterior, boundary, and interior markers are denoted as E , B , and I respectively.

are close enough to the boundaries such that their domain of influence extends to the outside of the boundary. Therefore, these markers should start influencing the markers of the other side even before they are transferred, see Figure 3.2.

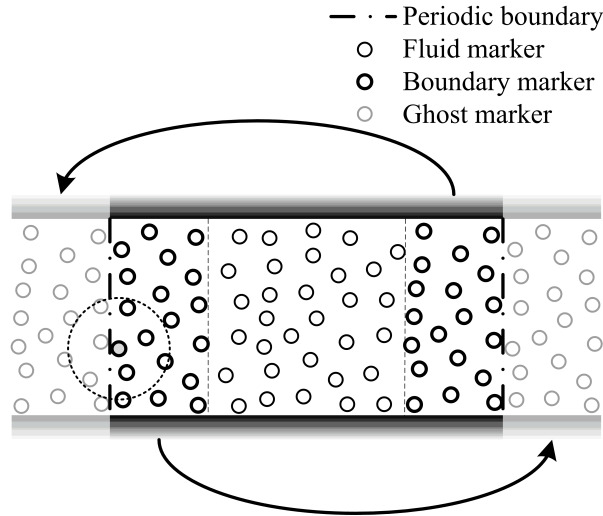


Figure 3.2: Periodic boundary condition in SPH.

One solution to this problem is to include a stripe of ghost markers next to the periodic boundary by copying markers from the alternative side of the domain. This technique is the only feasible solution if the proximity computation of Sect. 4.2.2 is used. The simpler alternative approach can be used in conjunction with the memory-optimized proximity calculation described in Sect. 4.2.1.

Therein, the search for potential interactions is localized to the cubic cells around each markers, i.e. bins. For the markers next to the periodic boundary, bins from the other side of the domain can be accessed by looping around the domain length. For instance, if a periodic boundary condition is applied in the x direction, the index of the neighbor bin, i , may fall out of bound since there is not any bin out of the domain boundary. Therefore the x components of the markers distance, d_x , and bin index, i , are replaced by d_x modulo l_x and i modulo n_x , respectively, where l_x and n_x are the x direction measures of the domain length and the number of bins, respectively. This leads to an access to the opposite side of the domain whenever a neighbor search invocation tries to reach beyond the domain.

Application of periodic BC to rigid bodies requires an extra step since each rigid body is a set of two entities; i.e., BCE markers and rigid body properties. Therefore, the periodic boundary condition should affect both entities. BCE markers follow the same treatment as the other markers. For a rigid object, the object is moved to the other side of the domain as soon as its center is out of bound. Similar treatments are also applied to a flexible object's BCE and nodes, with the only difference being that the beam is moved to the other side of the domain whenever all of its nodes are out of bound.

3.1.3 Inflow and outflow

The procedure of applying inflow and outflow boundary conditions in SPH is more tedious than in Eulerian approaches since these boundaries have an Eulerian, i.e. fixed, nature. The flow characteristics at the boundary are a collective property of SPH markers and need to be distributed to the markers so that they produce similar Eulerian conditions. This is achieved herein by including ghost regions at inflow and outflow. The appropriate pressure and velocity boundary conditions are applied to the markers within the ghost regions. Markers are constantly being added and removed from the physical and ghost regions to account for the fluid flow. The length of the ghost region in the flow direction should be at least equal to the radius of the markers support domain. The rate at which the markers are added or removed from the ghost region needs to be consistent with the physical flow rate across each boundary. Figure 3.3 depicts an inflow boundary condition using

ghost markers. As shown therein, whenever a marker leaves the inflow ghost region, it is added to the physical domain. At the same time, a marker is included to the ghost region according to the periodic boundary condition strategy.

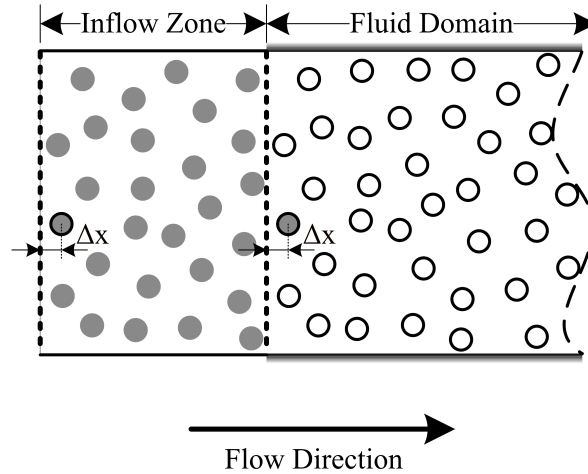


Figure 3.3: Schematic of the inflow boundary condition [66]. The inflow zone includes a set of ghost markers. The periodic boundary condition is applied to the ghost domain in the flow direction.

By coupling fluid and energy equations, Lastiwka *et al.* [66] noted, based on the method of characteristics, that for subsonic flow, two variables must be determined at inflow or outflow while the third variable is determined from the domain interior. Intuitively, velocity and temperature (internal energy) can be determined at inflow, leaving the density or pressure to be determined by the information propagating upstream from the domain interior. Within the SPH framework, temperature and velocity of ghost markers at inflow are determined from the boundary value, while their pressure is assigned through an extrapolation from the domain interior. Since pressure extrapolation can cause poor results, the reproducing kernel particle method [72] can be used to get a first-order consistency in the boundary extrapolation. Setting up the outflow condition is simpler since no new markers need to be created. Specifically, one variable is prescribed to determine the boundary condition. If, for instance, velocity and temperature are determined at inlet, pressure is

determined at outlet. Extrapolation of the other variables is merely the result of the downstream migration of markers.

It is worth mentioning that since the energy equation is not considered in this work, only two variables need to be determined. For instance, velocity and pressure may be determined at inlet and outlet, respectively.

3.1.4 Wall boundary condition

The wall boundary condition emerges in most of the fluid flow applications as a fixed or moving boundary. This is a simplified form of a general fluid-solid coupling, where the two-way dynamic coupling is replaced with a one-way or kinematics interaction. Therefore, further discussion of this type of boundary can be found in Sect. 3.2.

3.2 Fluid-solid interaction using Boundary Condition Enforcing (BCE) markers

The two-way fluid-solid coupling was implemented based on a methodology described in [104]. The state update of any SPH marker relies on the properties of its neighbors and resolves shear as well as normal inter-marker forces. For the SPH markers close to solid surfaces, the SPH summations presented in Eqs. (2.9), (2.10), (2.15), and (2.17) capture the contribution of fluid markers. The contribution of solid objects is calculated using BCE markers placed on and close to the solid surface as shown in Figure 3.4. In the case of flexible beams, the BCE markers are placed on "rigid disks" that are uniformly-spaced along the beam's axis and whose normals always coincide with the local tangent to the beam's axis. In all cases, the BCE marker locations are initialized so that the distance between two neighboring BCE markers is approximately equal to the initial distance between two SPH markers; in particular, this is also the distance between two adjacent disks of BCE markers in Figure 3.4b. This way, the partition of unity stays correct close to the interface, thus preventing any spurious force at the interface.

The velocity of a BCE marker is obtained from the rigid/deformable body motion of the solid and as such ensures the no-slip condition on the solid surface. Including the BCE markers in the

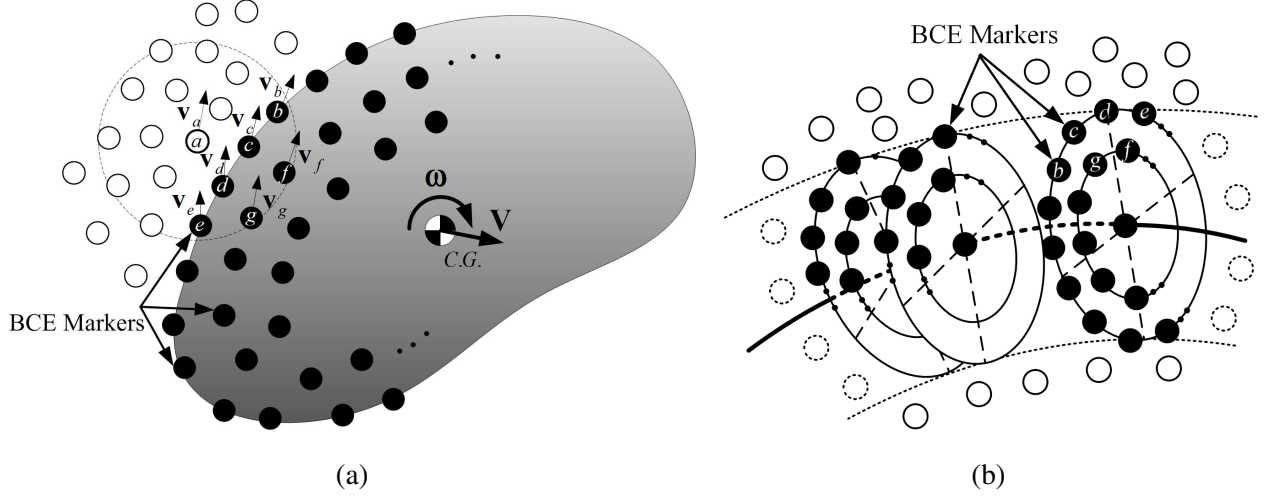


Figure 3.4: Fluid-solid interaction using BCE markers attached to a body: (a) a rigid body with the velocity, V , and angular velocity, ω , shown at its center of gravity, $C.G.$. The markers velocities are represented by $\{v_a, v_b, \dots, v_g\}$; (b) a flexible beam. BCE and fluid markers are represented by black and white circles, respectively. The BCE markers positioned in the interior of the body should be placed on and below the surface up to a depth no larger than the size of the compact support of the kernel W .

SPH summation equations (2.9) and (2.10) thus enforces the solid-to-fluid coupling. Conversely, the fluid-to-solid coupling is realized by applying the quantity in the right-hand side of Eq. (2.10), evaluated at each BCE marker, as an external force on the corresponding rigid or deformable solid body using Eqs. (2.18) and (2.29), respectively.

3.3 Solid-solid short range interaction

Dry friction models, typically used to characterize the dynamics of granular materials [3, 62, 63], do not capture accurately the impact of solid surfaces in hydrodynamics media. In practice, it is infeasible to fully resolve the short-range, high-intensity impact forces arising in wet media due to computational limits on space resolution and time step. Classical lubrication theory predicts no direct contact between solid surfaces in hydrodynamic media. Nevertheless, as many researchers suggested, see for instance [24, 46], the direct impact of solid surfaces is quite possible due to the

surface imperfections. Therefore, a sophisticated lubrication model should take into account both dry and wet interactions. One approach to this problem is a unified model which allows for direct impact of surfaces [6]. The second approach, used herein, is to capture the short range interaction via a lubrication force model, and the compliance encounters via a penalty approach.

3.3.1 Dry friction model

The dry frictional contact between two solid surfaces is handled using a penalty approach relying on the Discrete Element Method (DEM) [23]. In this model, the normal contact force between two rigid bodies is a function of the inter-penetration of the bodies and its time derivative. Many DEM approaches have been proposed in literature, see for instance [62, 110, 121]. In this study, linear and non-linear viscoelastic models are considered for the normal and tangential contact forces, respectively [62, 63].

Figure (3.5) shows a typical collision configuration between two rigid bodies, in this case spheres, in which $\mathbf{r}_i$ and $\mathbf{r}_j$ are the positions of the centers of bodies i and j , respectively, and $\mathbf{F}_n$ and $\mathbf{F}_t$ denote the normal and tangential contact forces. The magnitude of the normal force is computed as [62]

$$\mathbf{F}_{n_{ij}} = k_n \delta_{ij} \mathbf{n}_{ij} - \gamma_n \mathbf{v}_{n_{ij}}, \quad (3.4)$$

where k_n and γ_n are user-selected normal stiffness and damping coefficients, respectively; δ_{ij} and $\mathbf{v}_{n_{ij}}$ are relative inter-penetration distance and velocity, respectively; and $\mathbf{n}_{ij}$ is the unit normal vector at the contact location defined from body j to body i .

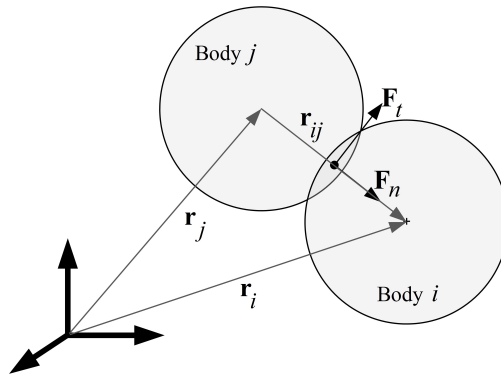


Figure 3.5: A typical contact in DEM.

The magnitude of the tangential force is approximated as [63]

$$\mathbf{F}_{t_{ij}} = -k_t \sqrt{\delta_{ij}} \mathbf{u}_t, \quad (3.5)$$

where

$$\mathbf{u}_t = \int_{t_i}^{t_f} \mathbf{v}_t(\tau) d\tau. \quad (3.6)$$

Here k_t is the coefficient of tangential stiffness, $\mathbf{v}_t$ is the tangential velocity at the contact point, $\mathbf{u}_t$ is the tangential displacement, and t_i and t_f denote the start and end of the inter-penetration time, respectively. Equations (3.4) and (3.5) are usually applied in conjunction with the Coulomb friction model, which constrains the magnitude of the tangential force to satisfy $\|\mathbf{F}_{t_{ij}}\| \leq \mu \|\mathbf{F}_{n_{ij}}\|$.

Handling the frictional contact of rigid bodies with complex, potentially non-convex boundaries is performed using spherical decomposition [60, 76], which takes advantage of BCE markers shown in Figure 3.4.

3.3.2 Lubrication force model

Ladd [65] proposed a normal lubrication force between two spheres that increases rapidly as the distance between spheres approaches zero thus preventing the actual touching of the spheres:

$$\mathbf{F}_{ij}^{lub} = \min \left\{ -6\pi\mu \left(\frac{a_i a_j}{a_i + a_j} \right)^2 \left(\frac{1}{s} - \frac{1}{\Delta_c} \right), 0 \right\} \cdot \mathbf{v}_{n_{ij}}, \quad (3.7)$$

where, a_i and a_j are the sphere radii, $\mathbf{v}_{n_{ij}}$ is the normal component of the relative velocity, s is the distance between surfaces, and Δ_c is a cut-off value that controls the extent of short range interaction: for $s > \Delta_c$, $\mathbf{F}_{ij}^{lub} = 0$, and the spheres are subject only to hydrodynamic forces.

Equation (3.7) provides a basic model for the estimation of the lubrication force in normal direction. The generalization of this model to non-spherical objects requires the calculation of the minimum distance and the curvature of the two contact surfaces. The calculation of the partial lubrication force between non-spherical surfaces follows the approach proposed in [28] for a lattice Boltzmann method yet it is amended to fit the Lagrangian formulation adopted herein.

Accordingly, the force model provided in Eq. (3.7) is modified as

$$\mathbf{F}_{ij}^{lub} = \sum_k \mathbf{f}_{ij}^k, \quad (3.8a)$$

$$\text{with } \mathbf{f}_{ij}^k = -\frac{3}{2}\pi\mu h^2 \mathbf{v}_{n_{ij}}^* \times \begin{cases} \left(\frac{1}{\Delta_i} - \frac{1}{\Delta_c} \right), & 0 \leq s^* < \Delta_i \\ \left(\frac{1}{s^*} - \frac{1}{\Delta_c} \right), & \Delta_i \leq s^* < \Delta_c \\ 0, & s^* \geq \Delta_c \end{cases} \quad (3.8b)$$

where s^* and $\mathbf{v}_{n_{ij}}^*$ denote the markers' relative distance and velocity, respectively, and the summation is over all interacting markers of the two solid objects. The parameter Δ_i is a small cut-off value ($\Delta_i \ll \Delta_c$) introduced to avoid having infinite force at zero distance which is computationally advantageous since it reduces the stiffness of the force model suggested in Eq. (3.7). An extra outcome of this modification is the possibility of the direct impact of solid surfaces, which is in agreement with [24, 46]. Therefore, this model is completed by the force model given in Eqs. (3.4) and (3.5).

3.4 Time integration

The simulation framework, referred to herein as Chrono::Fluid, uses a second order explicit mid-point Runge-Kutta (RK2) scheme [8] for the time integration of the fluid and solid phases, the latter in its rigid or flexible representation.

Algorithm 1 summarizes the steps required for the calculation of the force on the SPH markers, rigid bodies, and deformable beams at time step k . The variables N_m , N_r , and N_f denote the number of markers, rigid bodies, and flexible beams, respectively. The arrays $\boldsymbol{\rho}$, $\mathbf{x}$, $\mathbf{v}$, and $\hat{\mathbf{v}}$ store the density, position, velocity, and modified velocity for all markers, respectively; for example, $\boldsymbol{\rho} = \{\rho_a | a = 0, 1, 2, \dots, N_m - 1\}$.

The external forces on the rigid and flexible bodies include the FSI forces captured via BCE markers at distributed locations on the solid surfaces. The distributed forces need to be accumulated into a single force and torque at the center of each rigid body, or point forces at node locations of each flexible body. The summation of the forces and torques is handled by parallel reduction

operations available through the Thrust library [45], which exposes a scan algorithm that scales linearly.

The RK2 integration scheme requires the calculation of the force at the beginning as well as middle of the time step. Algorithm 2 lists the steps required for the time integration of a typical FSI problem.

To improve the code vectorization and use of fast memory; i.e., L1/L2 cache, shared memory, and registers, each computation task was implemented as a sequence of light GPU kernels. For instance, different computation kernels are implemented to update the attributes of the solid bodies, including force, moment, rotation, translation, linear and angular velocity, and locations of the associated BCE markers. A similar coding philosophy was maintained for the density re-initialization, boundary condition implementation, and mapping of the marker data on an Eulerian grid for post processing.

3.4.1 Dual-rate integration

Stable integration of the SPH fluid equations requires step-sizes which are also appropriate for propagating the dynamics of any rigid solids in the FSI system. However, integration of the dynamics of deformable bodies, especially as their stiffness increases, calls for very small time steps. To alleviate the associated computational cost, a dual-rate integration scheme is employed where intermediate steps for the integration of the flexible dynamics EOMs is performed at a typical rate of $\Delta t_{SPH}/\Delta t_{ANCF} = 10$, although stiffer problems may require ratios of up to 50. This aspect is noteworthy given that typical FSI models involve a number of SPH markers orders of magnitude larger than the number of ANCF nodal coordinates required for the flexible bodies. Without a dual-rate implementation, the numerical solution would visit the fluid phase at each integration step, an approach that led to very large solution times. This aspect is further discussed in Sect. 6.

Algorithm 1 Force Calculation

▷ Calculate modified XSPH velocities

- 1: **for** $a := 0$ to $(N_m - 1)$ **do**
- 2: $\hat{\mathbf{v}}_a^k \equiv \hat{\mathbf{v}}_a(\boldsymbol{\rho}^k, \mathbf{x}^k, \mathbf{v}^k)$
- 3: **end for**

▷ SPH forces

- 4: **for** $a := 0$ to $(N_m - 1)$ **do**
- 5: $\dot{\rho}_a^k \equiv \dot{\rho}_a(\boldsymbol{\rho}^k, \mathbf{x}^k, \hat{\mathbf{v}}_a^k)$
- 6: $\dot{\mathbf{x}}_a^k = \hat{\mathbf{v}}_a^k$
- 7: $\dot{\mathbf{v}}_a^k \equiv \dot{\mathbf{v}}_a(\boldsymbol{\rho}^k, \mathbf{x}^k, \mathbf{v}^k)$
- 8: **end for**

▷ Rigid body forces

- 9: **for** $i := 0$ to $(N_r - 1)$ **do**
- 10: $\dot{\mathbf{V}}_i^k \equiv \dot{\mathbf{V}}_i(\dot{\mathbf{v}}^k)$
- 11: $\dot{\mathbf{X}}_i^k = \dot{\mathbf{V}}_i^k$
- 12: $\dot{\boldsymbol{\omega}}_i^k \equiv \dot{\boldsymbol{\omega}}_i(\dot{\mathbf{v}}^k, \boldsymbol{\omega}_i^k)$
- 13: $\dot{\mathbf{q}}_i^k \equiv \dot{\mathbf{q}}_i(\boldsymbol{\omega}_i^k, \mathbf{q}_i^k)$
- 14: **end for**

▷ ANCF forces

- 15: **for** $j := 0$ to $(N_f - 1)$ **do**
- 16: $\hat{\mathbf{Q}}_j^k = \hat{\mathbf{Q}}_j^a(\dot{\mathbf{v}}^k) - \hat{\mathbf{Q}}_j^e(\hat{\mathbf{e}}^k) + \hat{\mathbf{Q}}_j^g(\hat{\mathbf{e}}^k)$
- 17: $\dot{\hat{\mathbf{e}}}_1^k = \hat{\mathbf{e}}_2^k$
- 18: $\dot{\hat{\mathbf{e}}}_2^k = \hat{\mathbf{M}}^{-1} \hat{\mathbf{Q}}_j^k$
- 19: **end for**

Algorithm 2 RK2 Time Integration

- ▷ Force calculation at beginning of step (Algorithm 1)
- 1: Calculate $\{\hat{\mathbf{v}}^k, \dot{\rho}^k, \dot{\mathbf{x}}^k, \dot{\mathbf{v}}^k, \dot{\mathbf{V}}^k, \dot{\mathbf{X}}^k, \dot{\boldsymbol{\omega}}^k, \dot{\mathbf{q}}^k, \hat{\mathbf{Q}}^k\}$
- ▷ Half-step updates for fluid, rigid body, and flexible body states
- 2: **for** $a \in \{a | a \text{ is a fluid marker}\}$ **do**
- 3: $\psi_a^{k+1/2} = \psi_a^k + \dot{\psi}_a^k \times \Delta t/2$, where $\psi_a \in \{\rho_a, \mathbf{x}_a, \mathbf{v}_a\}$
- 4: **end for**
- 5: **for** $i := 0$ to $(N_r - 1)$ **do**
- 6: $\Psi_i^{k+1/2} = \Psi_i^k + \dot{\Psi}_i^k \times \Delta t/2$, where $\Psi_i \in \{\mathbf{V}_i, \mathbf{X}_i, \boldsymbol{\omega}_i, \mathbf{q}_i\}$
- 7: **end for**
- 8: **for** $j := 0$ to $(N_f - 1)$ **do**
- 9: $\hat{\mathbf{e}}_1^{k+1/2} = \hat{\mathbf{e}}_1^k + \dot{\hat{\mathbf{e}}}_1^k \times \Delta t/2$
- 10: $\hat{\mathbf{e}}_2^{k+1/2} = \hat{\mathbf{e}}_2^k + \dot{\hat{\mathbf{e}}}_2^k \times \Delta t/2$
- 11: **end for**
- ▷ Half-step update for BCE marker positions and velocities
- 12: **for** $a \in \{a | a \text{ is a BCE marker}\}$ **do**
- 13: Obtain $\mathbf{x}_a^{k+1/2}$ and $\mathbf{v}_a^{k+1/2}$ according to associated rigid, flexible, or immersed boundary motion.
- 14: **end for**
- ▷ Force calculation at half-step (Algorithm 1)
- 15: Calculate $\{\hat{\mathbf{v}}^{k+1/2}, \dot{\rho}^{k+1/2}, \dot{\mathbf{x}}^{k+1/2}, \dot{\mathbf{v}}^{k+1/2}, \dot{\mathbf{V}}^{k+1/2}, \dot{\mathbf{X}}^{k+1/2}, \dot{\boldsymbol{\omega}}^{k+1/2}, \dot{\mathbf{q}}^{k+1/2}, \hat{\mathbf{Q}}^{k+1/2}\}$
- ▷ Full-step updates for fluid, rigid body, and flexible body states
- 16: **for** $a \in \{a | a \text{ is a fluid marker}\}$ **do**
- 17: $\psi_a^{k+1} = \psi_a^k + \dot{\psi}_a^{k+1/2} \times \Delta t$
- 18: **end for**

Continued on the next page ...

RK2 Time Integration, continued from the previous page

```

19: for  $i := 0$  to  $(N_r - 1)$  do
20:    $\Psi_i^{k+1} = \Psi_i^k + \dot{\Psi}_i^{k+1/2} \times \Delta t$ 
21: end for

22: for  $j := 0$  to  $(N_f - 1)$  do
23:    $\hat{\mathbf{e}}_1^{k+1} = \hat{\mathbf{e}}_1^k + \dot{\hat{\mathbf{e}}}_1^{k+1/2} \times \Delta t$ 
24:    $\hat{\mathbf{e}}_2^{k+1} = \hat{\mathbf{e}}_2^k + \dot{\hat{\mathbf{e}}}_2^{k+1/2} \times \Delta t$ 
25: end for

    ▷ Full-step update for BCE marker positions and velocities

26: for  $a \in \{a | a \text{ is a BCE marker}\}$  do
27:   Obtain  $\mathbf{x}_a^{k+1}$  and  $\mathbf{v}_a^{k+1}$ .
28: end for

```

Chapter 4

Advanced computing aspects

This chapter discusses a high performance computing framework built around the proposed methodology. The software implementation of the proposed Lagrangian-Lagrangian approach to FSI problems is open-source, available under a BSD3 license, and can be downloaded from [17]. Before explaining the computational aspects of the methodology, a brief overview of the GPU hardware and programming model adopted is provided first. This is followed by a detailed discussion of the key kernels that implement the proposed modeling and solution approach.

4.1 GPU hardware and programming model

To a very large extent, the performance of today's simulation engines is dictated by the memory bandwidth of the hardware solution adopted. Recent numerical experiments conducted in the Simulation Based Engineering Laboratory (SBEL) revealed that only about 5 to 10% of the peak flop rate is reached by the cores present on today's multi-core architectures since these cores most of the time idle waiting for data from global memory or RAM. It is this observation that motivated the selection of the GPU as the target hardware for implementing Chrono::Fluid. At roughly 300 GB/s, the GPU memory bandwidth stands four to five times higher than what one could expect on a fast CPU.

To describe the hardware organization of the GPU, NVIDIA GeForce GTX 680 [78, 93] is considered herein. This GPU is based on the first generation of Kepler architecture, code name GK104, which is also implemented in Tesla K10. The Kepler architecture relies on a Graphics Processing Cluster (GPC) as the defining high-level hardware block. There are a total of 4 GPCs

on the GK104. Each GPC includes two Stream Multiprocessors (SM), each of which has 192 Scalar Processors (SP), for a total of 1536 SPs, and 3.1 TFlops rate of processing single precision.

In addition to the processing cores, the second important aspect of the GPU hardware is that of the memory hierarchy. The memory on the GPU is divided into several types, each with different access patterns, latencies, and bandwidths. In what follows, a summary of the memory hierarchy on Kepler GPU is provided. Further information can be obtained from [78, 88, 93].

Registers (read/write per thread): 65536, 32-bit memory units per SM. Very low latency, high bandwidth ($\simeq 10$ TB/s cumulative) memory used to hold thread-local data.

Shared memory/L1 cache (read/write per block): 64 KB per SM. Low latency, high bandwidth ($\simeq 1.5$ -2 TB/s cumulative) memory divided between shared memory and L1 cache.

Global memory (read/write per grid): 2 GB. Used to hold input and output data. Accessible by all threads, with a bandwidth of 192 GB/s and a higher latency ($\simeq 400$ -800 cycles) than shared memory and registers. All accesses to global memory pass through the L2 cache. The latter is 512 KB large and has a bandwidth of 512 B/clock cycle.

Constant memory (read only per grid): 48 KB per Kepler SM. Used to hold constants, serviced at the latency and bandwidth of the L1 cache upon a cache hit, or those of the global memory upon a cache miss.

The parallel execution paradigm best supported on GPUs is “single instruction multiple data” (SIMD). In this model, if for instance two arrays of length one million are to be added, one million threads are launched with each executing the same instruction; i.e., adding two numbers, but on different data - each thread adding two different numbers. Fortunately, SIMD computing is prevalent in the solution methodology proposed, where each SPH marker is handled by a thread in the same way in which hundreds of thousands of other threads handle their markers using different data. While strongly leveraging the SIMD model owing to the fine grain parallelism that it exposes, the methodology adopted is prone to lead to memory access patterns that do not display high spatial and/or temporal locality. This is because the SPH markers move in time leading

to less structured memory accesses that adversely impact the effective bandwidth reached by the code. This issue will be discussed further in Sect. 4.2 when comparing two algorithms that expose different memory access patterns.

4.2 Proximity calculation and neighbor search

The loop iterations in Algorithms 1 and 2 have no overlap and can be executed in parallel. The computational bottleneck thus becomes the determination of the neighbor lists through proximity calculation, a step that takes about 70% of the entire computational budget and thus critically impacts the overall performance of the simulation. Given the list of neighbors, the calculation of $\hat{\mathbf{v}}_a^k \equiv \hat{\mathbf{v}}_a(\boldsymbol{\rho}^k, \mathbf{x}^k, \mathbf{v}^k)$, $\dot{\rho}_a^k \equiv \dot{\rho}_a(\boldsymbol{\rho}^k, \mathbf{x}^k, \hat{\mathbf{v}}^k)$, and $\dot{\mathbf{v}}_a^k \equiv \dot{\mathbf{v}}_a(\boldsymbol{\rho}^k, \mathbf{x}^k, \mathbf{v}^k)$ is straightforward.

Two different approaches, both relying on spatial subdivision, were implemented and compared in this work. The main difference between the two approaches goes back to a trade off between the memory footprint vs. the computation speed: compared to the second approach, the first approach minimizes the required memory at the cost of increasing the amount of required process by roughly a factor of 2.

4.2.1 Proximity computation-Approach 1: Optimized memory

In the first approach, summarized in Algorithm 3, the computation domain is divided into a collection of bins whose side lengths are equal to the maximum influence distance of a marker, i.e. κh . This localizes the search for the possible interacting markers to the bin and all of its 26 immediate 3D neighbors. Figure 4.1 shows the binning approach in 2D. The neighbor lists are not saved in memory; instead, neighbors are evaluated whenever required.

4.2.2 Proximity computation-Approach 2: Optimized process

Similar to the first approach, a spatial binning algorithm is implemented in the second approach, which is summarized in Algorithm 4 [103]. Nevertheless, the side length associated to the bins is arbitrary. Algorithm 4 takes advantage of the correlation between the influence of marker a on b and that of b on a ; therefore, it reduces the total process by doing the necessary calculations once per

Algorithm 3 Inner loop: accessing neighbor markers (optimized memory)

- 1: Divide the solution space into n_b bins of $(\Delta_x, \Delta_y, \Delta_z)$ dimensions, where $n_b = n_x \times n_y \times n_z$, and (n_x, n_y, n_z) is the number of grid cells along (x, y, z) axis.
 - 2: Construct the hash array: $\mathbf{s} = \{s_a | a = 0, 1, 2, \dots, N_m - 1\}$ according to $s_a = i \times n_y \times n_z + j \times n_z + k$, where (i, j, k) is the location of the bin containing the marker a .
 - 3: Sort $\mathbf{s}$ into $\mathbf{s}_{sorted}$ and obtain corresponding $\boldsymbol{\rho}_{sorted}$, $\mathbf{x}_{sorted}$, $\mathbf{v}_{sorted}$, and $\hat{\mathbf{v}}_{sorted}$.
 - 4: Construct $\mathbf{c}^1 = \{c_e^1 | e = 0, 1, 2, \dots, n_b - 1\}$ and $\mathbf{c}^2 = \{c_e^2 | e = 0, 1, 2, \dots, n_b - 1\}$, where c_e^1 and c_e^2 denote the two indices in $\mathbf{s}_{sorted}$ that bound the sequence of hash values $s_a = e$.
 - 5: Access markers data in bin (i, j, k) by loading $[c_e^1, c_e^2]$ portions of the sorted arrays $\boldsymbol{\rho}_{sorted}$, $\mathbf{x}_{sorted}$, $\mathbf{v}_{sorted}$, and $\hat{\mathbf{v}}_{sorted}$, as needed.
-

inter-penetration event. This results in an almost 2x computation speedup. Another advantage of Algorithm 4 with respect to Algorithm 3 is the capability of processing the proximity of arbitrary

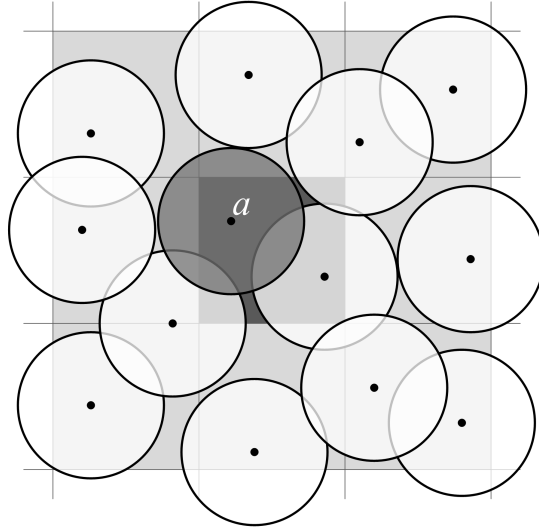


Figure 4.1: Illustration of the spatial subdivision method used for proximity computation in 2D. The circles represent the domain of influence of each marker; i.e., the support domain. For clarity, a coarse distribution of markers is shown. In reality, the concentration of markers per bin is much larger.

shapes. This advantage is irrelevant in the context of SPH, since the domain of influence of all markers are spheres of the same sizes.

4.2.3 Proximity computation-concluding remarks

The data sorting required frequently in Algorithms 3 and 4 is performed using radix sort available in the Thrust library [45]. Since this algorithm scales linearly, it does not affect the overall linear scaling of Algorithms 1 and 2. Working with the sorted arrays for the bulk of the computation has the additional advantage of increasing the memory spatial locality: SPH markers that share a neighborhood in the physical space, do so in memory as well.

Both of the proximity computations described in Sect. 4.2.1 and 4.2.2 were evaluated herein. Although the second algorithm reduced the amount of work by re-using the acceleration terms, it could not be applied efficiently to the SPH method due to the massive amount of memory required to store the data associated with all inter-penetration events. An efficient use of memory in Algorithm 3 allows simulation of domains composed of millions of markers. Switching to Algorithm 4 would drastically reduce the maximum domain size achievable on the same GPU to about 1.0×10^5 . The second advantage of Algorithm 3 is the coalesced memory access achieved by sorting and accessing the data based on the markers location, which translates eventually into a faster computation. The aforementioned two advantages resulted in the choice of Algorithm 3 for the proximity computation performed in Chrono::Fluid.

Algorithm 4 Inner loop: accessing neighbor markers (optimized process)

- 1: Divide the solution space into n_b bins of $(\Delta_x, \Delta_y, \Delta_z)$ dimensions, where $n_b = n_x \times n_y \times n_z$, and (n_x, n_y, n_z) is the number of grid cells along (x, y, z) axis.
 - 2: Construct an Axis Aligned Bounding Box (AABB) for each marker to determine all bin-marker intersections. Save the number of intersections per marker in an array $\mathbf{t}_1$ of the size N_m .
 - 3: Perform an inclusive scan on $\mathbf{t}_1$ to determine the total number of bin-marker intersections, t_2 .
 - 4: Similar to step 2. However, bin-marker intersections are saved in the key-value array $\mathbf{t}_2$ of size t_2 . Key and value are bin and marker indices, respectively.
 - 5: Sort $\mathbf{t}_2$ based on the key.
 - 6: Allocate the key-value array $\mathbf{t}_3$ of size n_b . Each component of $\mathbf{t}_3$ denotes a bin: the index of the first appearance of bin e ($e = i \times n_y \times n_z + j \times n_z + k$), in $\mathbf{t}_{2/\text{key}}$ is stored as key and e is stored as value. The keys associated to the bins with no appearance in $\mathbf{t}_{2/\text{key}}$, i.e. inactive bins, are set to infinity (e.g. $0 \times \text{ffffffff}$).
 - 7: Determine the number of active bins, t_4 by sorting $\mathbf{t}_3$ based on key and determining the first infinity.
 - 8: Determine the number of inter-penetration events per active bin and store them in array $\mathbf{t}_4$ of size t_4 .
 - 9: Perform an inclusive scan on $\mathbf{t}_4$ to determine the total number of inter-penetration events t_5 .
 - 10: Allocate arrays of the size t_5 for each interaction component (e.g. $d\mathbf{v}/dt$, $d\rho/dt$, $\hat{\mathbf{v}}$, etc.).
 - 11: Similar to step 9. However, the interaction components of each inter-penetration event is calculated and saved for the marker with the smaller index of the two into the arrays of step 10. The interactions components can be easily retrieved for the marker with the larger index.
-

Chapter 5

Validation studies and further numerical experiments

The purpose of this section is to validate the predictive attributes of the SPH-enabled Lagrangian-Lagrangian framework and the correctness of its software implementation. To the best of our knowledge, except for the transient Poiseuille flow in subsection 5.1.1, the validation tests discussed next have not been considered in the context of a Lagrangian-Lagrangian formulation via SPH.

5.1 Particle suspension: A validation study

The topic of particle migration has been of great interest since Segre and Silberberg experimentally investigated the pipe flow of a dilute suspension of spherical particles and demonstrated that, at a pipe Reynolds number (Re) between 2 and 700, the particles settle on an annulus with an approximate relative radius of 0.6 with respect to the pipe radius [116, 117]. Subsequent experiments conducted by Oliver [95], Jeffrey and Pearson [55], and Karnis *et al.* [59] confirmed and further investigated the particle radial migration. For dilute suspensions, Matas *et al.* [74] showed experimentally that the radius of stable annulus increases directly with Re . Nevertheless, at a high Reynolds number, $Re > 650$, they observed the formation of an inner annulus of smaller radius that had not been predicted analytically or observed through simulation. Moreover, they showed that the probability of a particle settling on this annulus of smaller radius increases with the Reynolds number. From an analytical perspective, perturbation methods have been widely employed to investigate the lift force responsible for particle migration, see for instance Saffman [111], Ho and Leal [44], Vasseur and Cox [129], Schonberg and Hinch [113], Hogg [47], Asmolov *et al.* [7], and

Matas *et al.* [75]. Particle migration has also been investigated in a number of numerical simulation studies. Feng *et al.* [31] employed a Finite Element Method (FEM) to study the migration of a single circular cylinder in plane Poiseuille flow. Inamuro *et al.* [54] investigated a similar problem using a Lattice Boltzmann Method (LBM). Chun and Ladd employed LBM to investigate the migration of spheres in a square duct at $Re < 1000$ [18]. They showed that the stable lateral position of a single particle moves closer to the duct wall as the Reynolds number increases. For flows containing several particles, a first stable particle configuration forms at $Re < 300$; a secondary stable region nearer to the center of the duct is observed at $Re > 700$. Pan and Glowinski developed the method of Distributed Lagrange Multiplier/Fictitious Domain Method (DLM/FDM) in conjunction with a finite difference approach to investigate the shear induced migration of a circular cylinder [36] and a collection of spheres [100]. Shao *et al.* [120] investigated the motion of spheres in steady Poiseuille flow at moderately high Re using DLM/FDM. Their work confirmed the development of an inner stable annulus at high Re ; i.e., $Re \geq 640$ for specific size and channel length ratio. Yu *et al.* investigated the sphere sedimentation as well as the migration of a sphere in Poiseuille flow at $Re < 400$ via the DLM method. Hu [51] and Hu *et al.* [50] employed the Arbitrary Lagrangian-Eulerian (ALE) method on a body-fitted unstructured finite element grid to simulate fluid-solid systems. Their work influenced that of Patankar *et al.* [101, 102] and Choi and Josef [16] in their study of the lift-off of cylinders in plane Poiseuille flow. Similar techniques have been considered to study the behavior of a non-spherical particle, usually an ellipsoid in fluid flow. Swaminathan *et al.* [125] used ALE based FEM to simulate the sedimentation of an ellipsoid. Pan *et al.* [97] investigated the motion of ellipsoid in Poiseuille flow using DLM/FDM. In several other studies the investigation of flows containing a collection of cylinders (2D) [15, 32, 122] and spheres (3D) [18, 50] was carried out via direct numerical simulation with the LBM [18, 32], Lagrange multiplier based fictitious domain method [15, 99, 122], and ALE-based FEM [50].

All these numerical studies of particle suspension and migration draw on an Eulerian-Lagrangian representation of the fluid-solid system. Nevertheless, the proposed Lagrangian-Lagrangian approach is employed herein to study the particle migration over a wide range of Reynolds numbers.

Table 5.1: Flow parameters used for the validation of transient Poiseuille flow

density	1000 kg/m ³
viscosity	0.001 N s/m ²
volumetric force, x-direction	0.005 N/m ³
channel width	0.002 m

5.1.1 Transient Poiseuille flow

SPH was used in [84] to numerically simulate transient Poiseuille flow at low Reynolds numbers for which an analytical solution is readily available. Although essentially a 2D problem, for validation purposes, the transient Poiseuille flow is simulated herein using a 3D setup: the 2D flow was generated using periodic boundary conditions on the channel side walls; i.e., in the direction perpendicular to x and y , see Figure 5.1. For the set of parameters provided in Table 5.1, results show a virtually exact match between the velocity profiles obtained from the numerical simulation and the analytical solution given in Eq. (5.1):

$$v_x(y, t) = \frac{f_x}{2\mu}y(y - w) + \sum_{n=0}^{\infty} \frac{4f_x w^2}{\mu\pi^3(2n+1)^3} \sin\left(\frac{(2n+1)\pi y}{w}\right) \exp\left(-\frac{(2n+1)^2\pi^2\mu}{w^2}t\right) . \quad (5.1)$$

In this equation, v_x is the flow velocity in the x direction, i.e. channel axis, y is the distance from the channel wall, w is the channel width, f_x is the body force in x direction, μ is the fluid viscosity, and t denotes time.

5.1.2 Particle migration in 2D Poiseuille flow

Although Segre and Silberberg considered particle migration in pipe flow [116, 117], a similar phenomenon occurs in plane Poiseuille flow. Maintaining the setup of the previous subsection, infinitely long cylinders were added into the flow. The same stable lateral positions of the immersed bodies as those reported in [54, 98], with a maximum 1% relative drift with respect to the channel half width, was obtained for a cylinder size $a/w = 0.125$ and $Re_c \approx 12.73$, where a , w , and Re_c are the cylinder radius, channel width, and channel Reynolds number, respectively. Figure

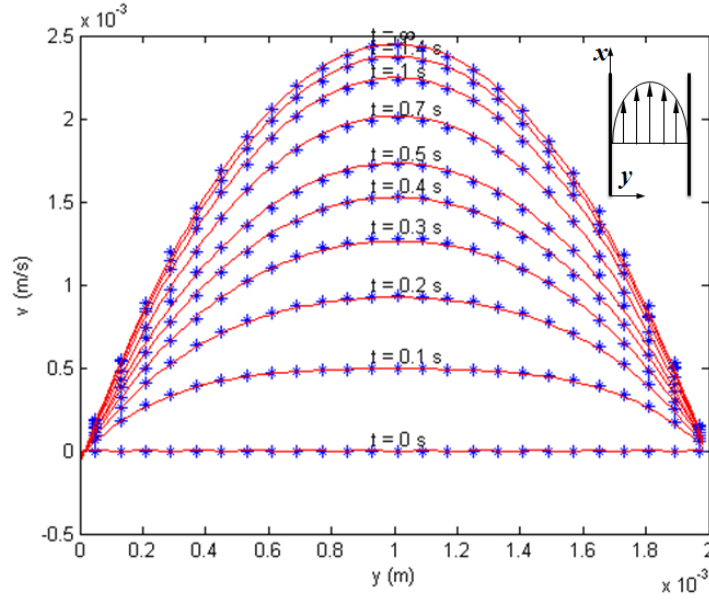


Figure 5.1: Velocity profile of transient Poiseuille flow obtained from simulation (dots) and series solution (continuous lines).

5.2 shows the trajectories of cylinders released from different initial lateral positions, y , versus non-dimensional time, $t^* = t \times V/L$, where V denotes the average flow velocity.

5.1.3 Particle migration in pipe flow

The experiment conducted by Segre and Silberberg on the motion of a sphere in pipe flow demonstrated a final particle stable radial position of $r/R \approx 0.6$, where r and R denote the particle offset from the pipe axis and the pipe radius, respectively [116, 117]. The same results are obtained through simulation at $Re \approx 60$ and particle relative size $a/R = 0.1$ (Figure 5.3). The effects of the Reynolds number and particle size are investigated independently and reported in sub-sections 5.1.5 and 5.2.5, respectively.

5.1.4 Radial distribution of particles in suspension

This test validates the transient behavior of a suspension of several spheres in pipe flow. The simulation parameters are those in Segre and Silberberg's experiment [116, 117], where the

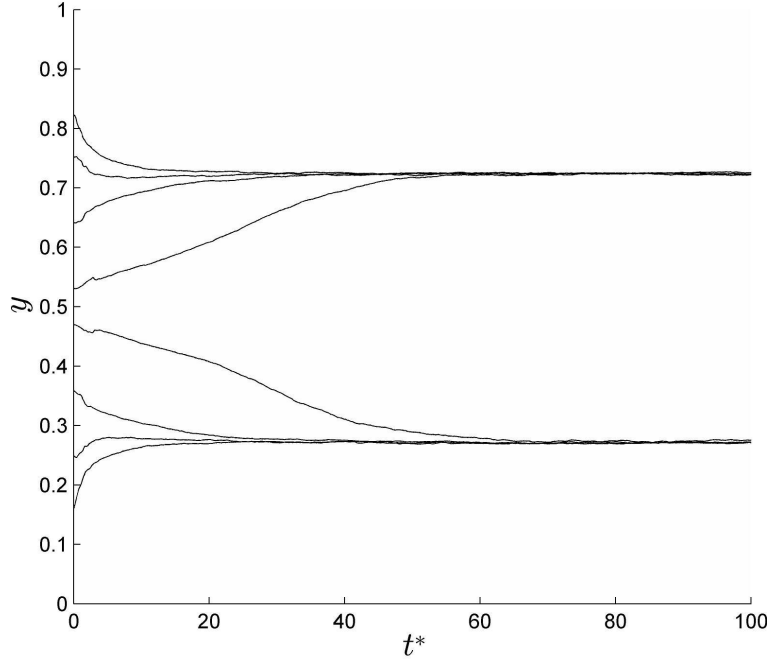


Figure 5.2: Simulation of plane Poiseuille flow: lateral position as a function of dimensionless time of neutrally buoyant circular cylinders released from different initial positions. Results are within 1% relative error of those reported in [54, 98].

sphere's radius, a , and pipe's radius, R , are 0.4 mm and 5.6 mm respectively, and the fluid density, ρ , and viscosity, μ , are 1.18 g/cm³ and 50 cp, respectively. Figure 5.4 shows a comparison of the particle radial distribution at $Re \approx 60$ with experimental results reported in [116]. Here, $L = (a/R)(av\rho/\mu)(l/R)$ is the non-dimensional distance from the pipe inlet and $v = (a/R)^2 V$ is defined based on mean flow velocity V . The experimental setup considered by Segre and Silberberg [116]; i.e., including particle distribution in the range of 1 through 4 particles/cm³ or volumetric concentration of $\phi \in [0.027, 0.109]\%$, is very dilute. As such, generating smooth distribution curves requires a very long channel to include a sufficiently large number of rigid bodies. Reproducing this experiment through simulation requires the flow field to be resolved at a scale fine enough to capture the dynamics of the small rigid bodies suspended in the flow. This translates

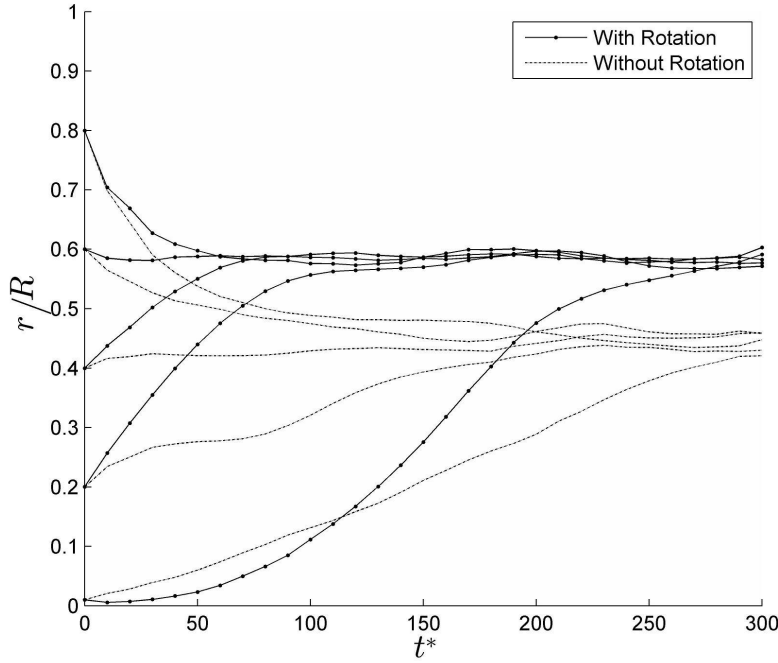


Figure 5.3: Migration of rotating and non-rotating neutrally buoyant spheres in pipe flow.

into a large number of SPH markers. Consequently, the amount of time required to complete a simulation, even when leveraging high performance parallel computing, was prohibitively large. This issue was addressed by performing an ensemble average over a set of smaller channels instead of one single, long channel. Each small channel was subjected to periodic boundary conditions along the channel axis and included between 1 and 16 rigid particles, initialized randomly and tracked independently. In a Monte Carlo framework, a large number of simulations were considered to produce a converged statistical distribution. Yet, this was not an issue since batches of up to 56 simultaneous simulations (one simulation per GPU card) were carried out on the computer cluster available for this study [112]. The distributions reported in Figure 5.4 are the result of a statistical investigation based on 192 ten-hour-long simulations that captured 14 seconds of real time. The simulation results accurately reproduce (1) the stable radial position, and (2) the longitudinal transition distance observed in experimental tests [116].

To generate the distribution curves, the data was sampled at sections located at predefined distances from the pipe entrance. A very fine radial grid was considered to record the particles' radial positions, which resulted in a noisy output. Subsequently, statistical bootstrapping was used to find the confidence zone and distribution curve [29]. The error bars reported with the distribution curves are associated with the 95% confidence interval.

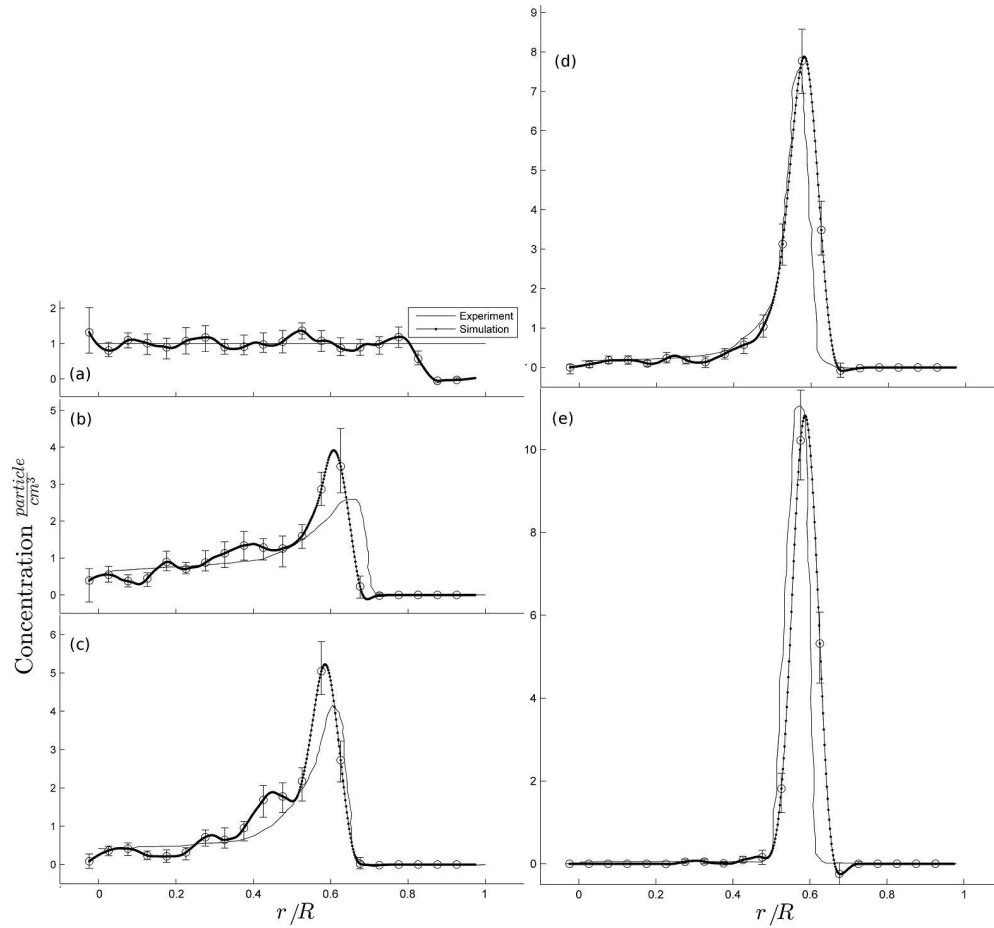


Figure 5.4: Particle radial distribution as a function of non-dimensional distance L , compared to the experiment [116] at: (a) $L = 0$, (b) $L = 0.08$, (c) $L = 0.16$, (d) $L = 0.32$, (e) $L = 0.69$. Note that rigid bodies cannot be initialized in the region close to the wall; i.e., $0.9 \leq r/R \leq 1$, due to their finite size.

5.1.5 Effect of Reynolds number

Matas *et al.* [74] extended the Segre-Silberberg experiment up to $Re \approx 2400$ and demonstrated that the radial stable positions migrate towards the wall as Re increases. However, a second stable region, i.e. an inner annulus which does not comply with the aforementioned trend, forms at high Re , e.g. $Re > 650$, for $\lambda = a/R \in [0.06, 0.11]$. Moreover, the radial distribution of the particles shifts toward the inner annulus as Re increases. Shao *et al.* [120] numerically confirmed the formation of an inner annulus at high Re via the direct-forcing fictitious domain method [132]. Figure 5.5 reports over a wide range of Reynolds numbers results obtained with the proposed approach, experimental results presented in [74], and numerical results from [120, 131]. Our numerical results confirm the emergence at higher Reynolds numbers; i.e., $Re \approx 772$, of a secondary stable annulus in agreement with [74, 120]. The results also show small differences at $Re > 1200$ from those reported in [120]. However, as Matas *et al.* pointed out in their experimental work, particles tend to appear between the two stable annuli effectively everywhere yet with a higher probability around the interior annulus.

5.2 Particle suspension: Further numerical experiments

Unlike the previous section, which focused on validating the proposed approach and its software implementation, this section presents results of several simulations carried out to characterize through direct numerical simulation the effect of particle properties on the radial distribution of suspensions in pipe flow.

5.2.1 Effect of particle rotation

To identify the root cause of particle migration and gauge the influence of the Magnus effect, Oliver [95] carried out experiments with inertia asymmetric spheres in which the center of mass was shifted from the sphere's geometric center. He showed that as the particle rotation was reduced, the stable final position of these particles moved toward the pipe's axis. Similar investigations for the motion of a cylinder in Poiseuille flow were performed numerically by Patankar *et al.* [101] and

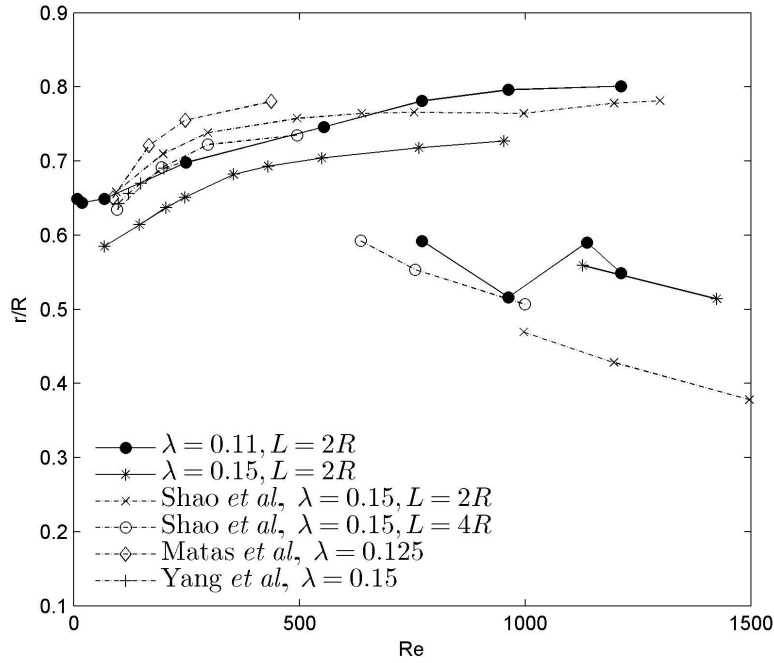


Figure 5.5: Effect of Re on the tubular pinch effect for $Re \in [1, 1400]$ and two particle size ratios, $\lambda = 0.11$ and $\lambda = 0.15$. The results are compared to data provided by Matas *et al.* [74], Yang *et al.* [131], and Shao *et al.* [120].

Joseph and Ocando [57] to demonstrate the influence of the cylinder rotation on the steady state configuration in 2D flows. Herein, the effect of 3D rotation on sphere migration was investigated by eliminating the rotation via imposing extremely large moments of inertia for the immersed spheres. As shown in Figure 5.3, the rigid body migration is noticeably altered when body rotation is artificially removed. Specifically, the stable radial position of non-rotating spheres is closer to the pipe's axis. The conclusion is that the behavior of particles in 3D flow of suspension may exhibit large deviation from the actual dynamics if the body rotation is ignored.

5.2.2 Effect of particle concentration

The experimental results in [116], used herein for validation purposes, only included particle concentrations in the range 1 through 4 particles/cm³, i.e. volumetric concentration of

$\phi \in [0.027, 0.109] \%$. The dynamics of monodisperse particles has been numerically investigated herein for concentrations of up to $\phi = 3.488\%$. Normalized concentrations, defined as the ratio of the particle local concentration to the initial concentration, are plotted for different regimes in Figure 5.6. The results obtained demonstrate a tubular pinch effect even for denser regimes, which suggests low probability of spheres hovering close to the pipe's axis. The particle distribution is, however, more spread in denser flows as the interaction between rigid bodies through drafting, kissing, and tumbling prevents the particles from coalescing into a narrow annulus. Figure 5.7 provides snapshots of particle distribution and resulting annulus formation down the pipe, for concentrations of 32 and 64 particles/cm³.

Figure 5.8 shows the velocity profile at the flow mid-section and illustrates the blunting of the parabolic profile, which is in agreement with a 2D numerical study [98] and experimental results reported in the literature [40, 41, 61, 123].

5.2.3 Effect of inter-particle distance

The effect of a particle's wake on the radial migration of trailing particles is investigated using periodic boundary conditions along the channel axis for spheres of radius a , with $a/R = 0.25$, at $Re \approx 60$. The particle initial position is close to the pipe axis and inter-particle spacing is adjusted by changing the value d , see Figure 5.9. The results suggest that (1) the stable radial position moves closer to the wall as the inter-particle distance decreases; and (2) for very large inter-particle distance; i.e., $d/a \gg 1$, radial migration is independent of inter-particle distance. Finally, Figure 5.10 indicates that decreasing the inter-particle distance slows down the radial migration; i.e., a particle's wake alters the local flow profile around trailing particles. Specifically, while the particles settle further away from the pipe axis, the settling occurs significantly further down the pipe. The result of this test is particularly relevant in the simulation of pipe flow using periodic boundary condition since it suggests a safe value of $d/a > 9$ to avoid the effect caused by the imposition of boundary conditions.

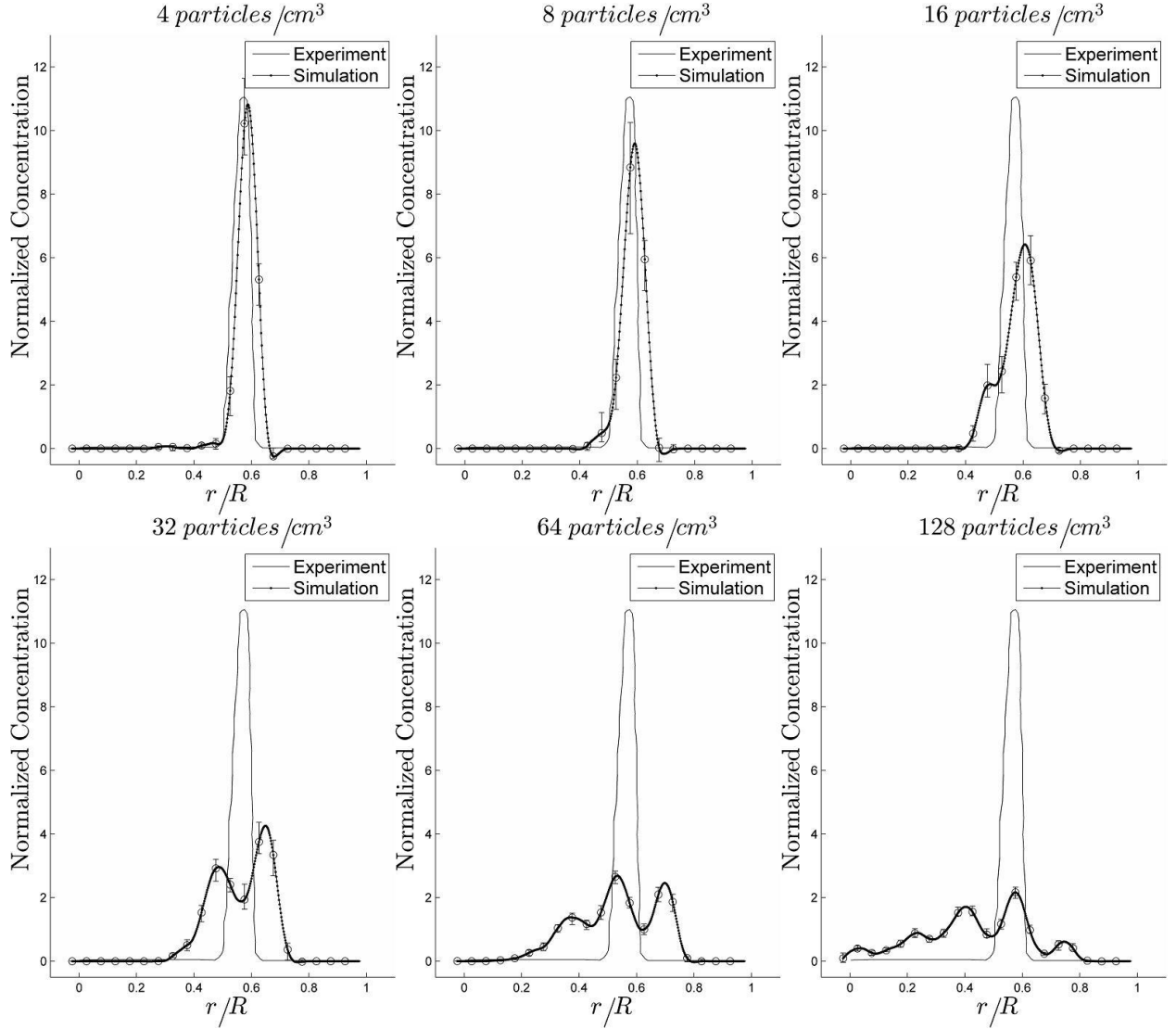


Figure 5.6: Particle radial distribution at $L = 0.69$, normalized by initial concentration, plotted for six different initial concentrations in the range of 4 through 128 particles/cm³ equivalent to $\phi \in [0.109, 3.488]\%$. Results are compared to a reference experimental distribution obtained in [116] for concentration of 1 particles/cm³.

5.2.4 Effect of particle asymmetry

In most of the experiments and numerical simulations of particle migration, the rigid bodies were spherical and the effect of asymmetry was not discussed. A series of numerical simulations were carried out at $Re \approx 60$ by replacing spheres with ellipsoids of radii (a_1, a_2, a_3) , with $a_1 =$

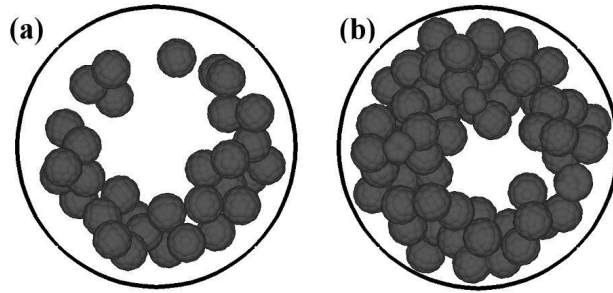


Figure 5.7: Particle arrangement at $L = 0.32$ for particle concentrations of (a) 32 particles/cm³ and (b) 64 particles/cm³.

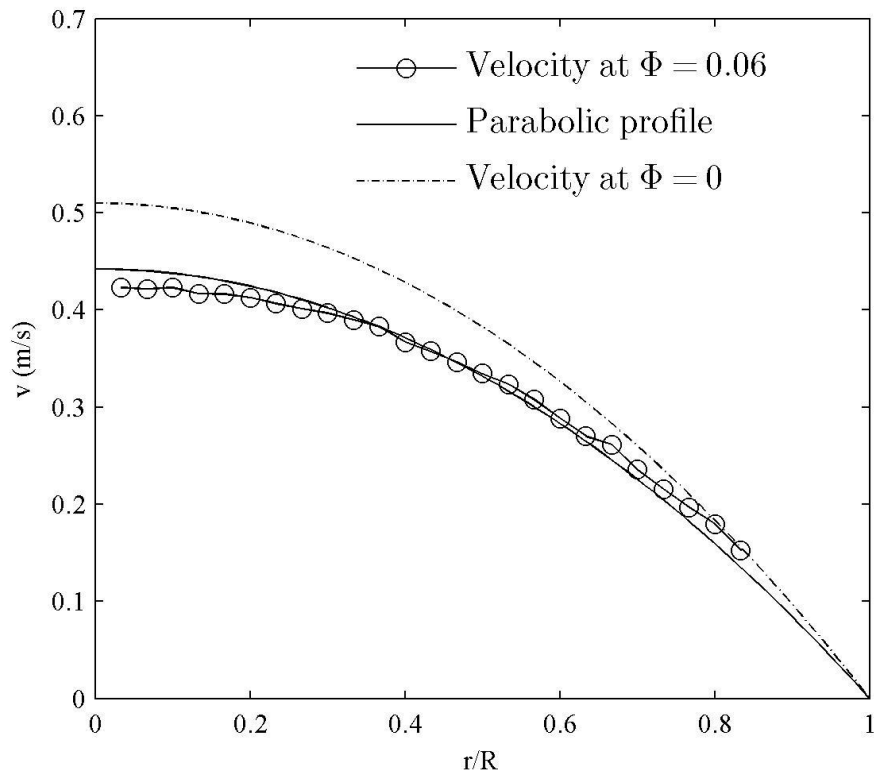


Figure 5.8: Blunting of the velocity at $\phi = 6.01\%$ compared to velocity profile at $\phi = 0\%$.

$a_2 = 0.07R$ and $0.07R < a_3 < 0.43R$. The obtained results suggest that, with an increase in skewness, the stable radial particle position moves toward the pipe axis (Figure 5.11).

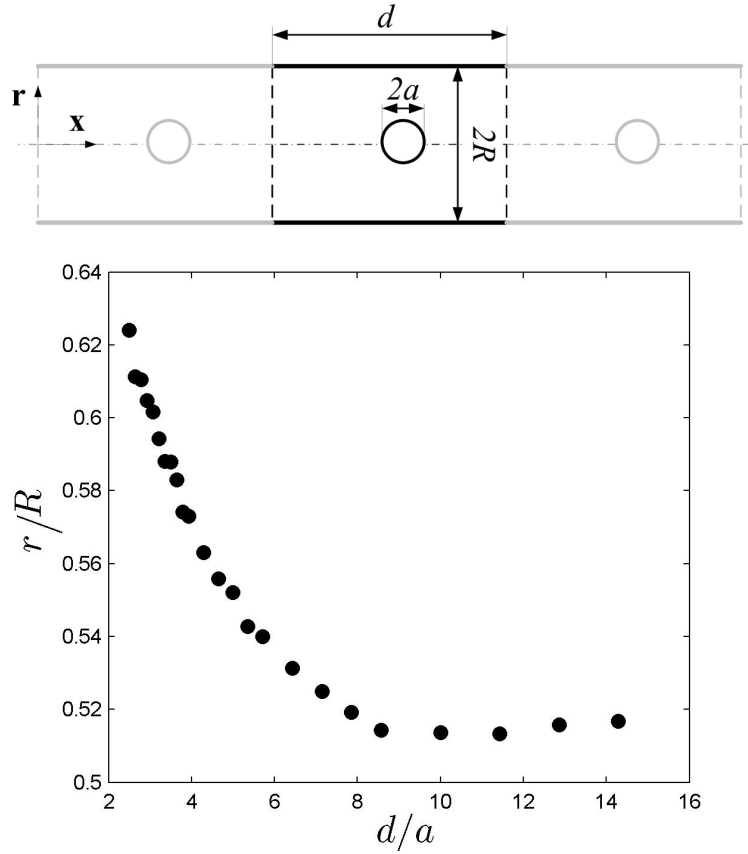


Figure 5.9: Schematic of simulation setup used to investigate the effect of inter-particle distance on radial migration where the fluid flow and periodic boundary are in the x -direction (top). Effect of inter-particle distance (spacing) on radial stable position (bottom).

5.2.5 Effect of particle size

By changing the radius of the neutrally buoyant sphere in the range of $0.07R < a < 0.36R$ at $Re \approx 60$, it is observed that particle radial migration decreases almost linearly as the radius increases (Figure 5.12).

5.3 Investigation of fluid-flexible beams interaction

The simulations presented in this section involve relatively soft beams with a modulus of elasticity $E \leq 20$ MPa that are either unconstrained or else anchored at one end. Since computational efficiency of the FSI code is directly related to the number of nodal coordinates used to model the

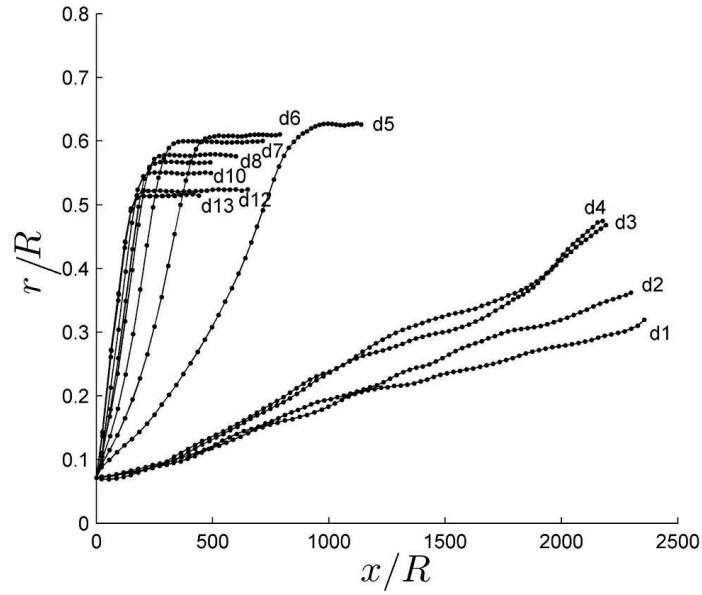


Figure 5.10: Particle trajectories as a function of a non-dimensional travel distance along the pipe axis x/R , plotted for several inter-particle distances increasing monotonically from $d1$ to $d13$. The results demonstrate smaller, yet faster, radial migration for larger inter-particle distances.

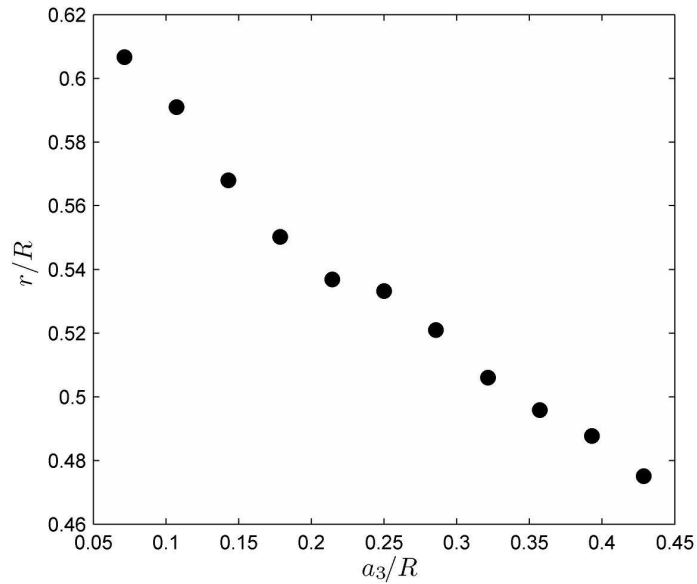


Figure 5.11: Effect of particle skewness on the radial stable position.

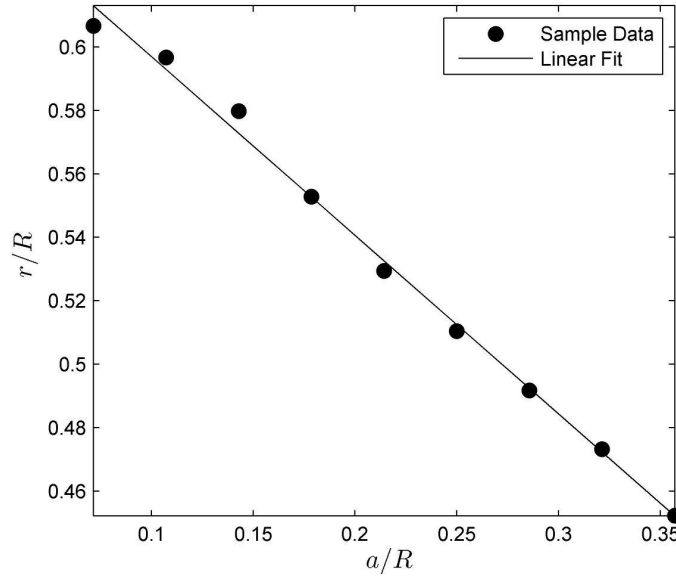


Figure 5.12: Effect of sphere size on the radial stable position.

flexible beams, a parametric study was conducted first to identify the minimum number of ANCF beam elements required to accurately capture the dynamics of interest in the subsequent experiments. In this set of experiments, a cantilever of length $L = 1$ m and diameter $D = 0.04$ m with density $\rho_s = 7200$ kg/m³ and modulus of elasticity $E = 20$ MPa was considered to move under gravity ($g = -9.81$ m/s²) in vacuum or immersed in fluid of various viscosities (see Sect.5.3.2). Figure 5.13 shows a few time snapshots from a dynamic simulation of a cantilever modeled with $n_e = 4$. As shown in Figure 5.14, the tip trajectory of a cantilever moving in a vacuum has an acceptable convergence at all discretizations and virtually identical results for $n_e \geq 4$, the value which was selected for all subsequent simulations.

5.3.1 Floating beam in Poiseuille flow

The validation test was performed using a straight beam with $L = 0.2$ m, $\rho_s = 7200$ kg/m³, $E = 20$ MPa, $d = 0.04$ m, modeled as an ANCF beam with a total of 30 Degrees Of Freedom (DOF), and a rigid cylinder with the same density and geometry, modeled as a 3D rigid body, i.e. 6 DOF. The beam and rigid cylinder were subjected to an accelerating channel flow aligned

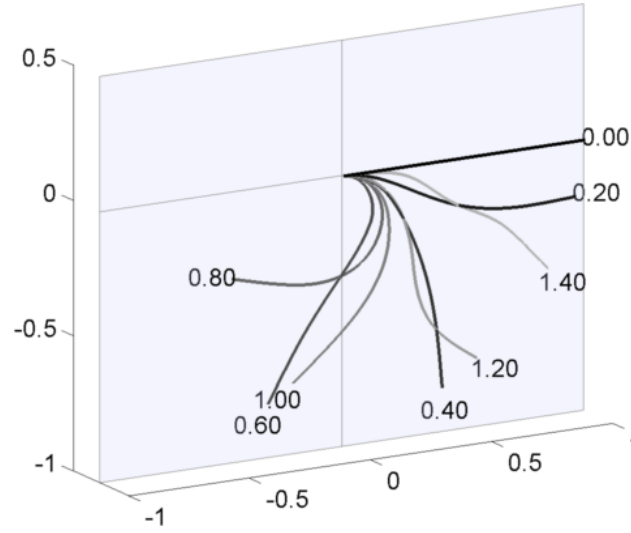


Figure 5.13: Time snapshots of a flexible cantilever moving in vacuum under the action of gravity. The darker colors denote earlier stages of the motion.

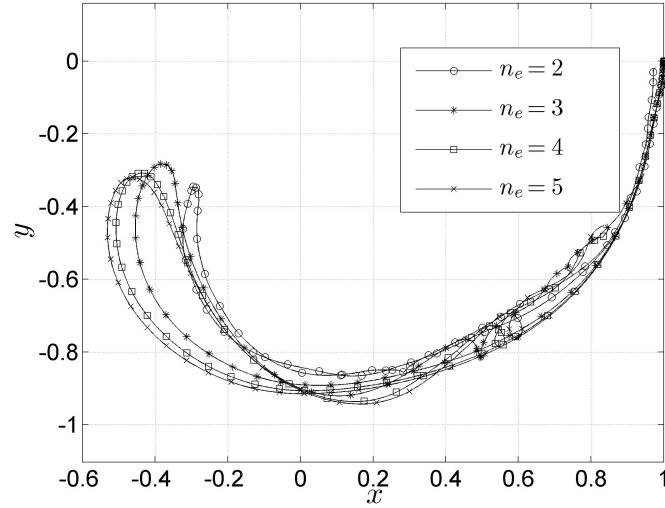


Figure 5.14: Convergence test of a soft cantilever beam falling under gravity. Trajectory of the beam tip is shown for different discretization resolution.

with the global x axis with final steady state Reynolds number $Re_c = \rho V w / \mu = 100$, where $\rho = 1000 \text{ kg/m}^3$, $\mu = 1 \text{ N s/m}^2$, average velocity $V = 0.1 \text{ m/s}$, and channel width $w = 1 \text{ m}$. The beam and cylinder were initially perpendicular to the flow and rotated in the yz plane. Under this

flow condition, the deformation of the unconstrained deformable beam is negligible. Therefore the rigid body mode of the ANCF beam is expected to be similar to the rigid cylinder.

Comparisons of the resulting beam orientation angles, relative to the global x , y , and z axes, and of the time evolution of the velocity in the x direction of the beam center velocity are presented in Figure 5.15. The results show good agreement with differences due to the inability of the gradient deficient ANCF beam element model to capture rotation about the beam's axis.

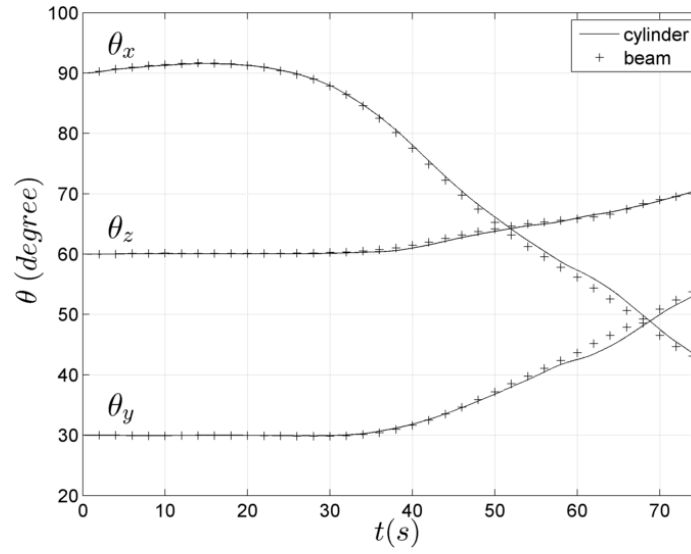
5.3.2 Flexible cantilever immersed in fluid: Effect of viscosity

The effect of viscosity on the motion of the beam's tip was investigated using a cantilever moving under the action of gravity in viscous fluid. As shown in Figure 5.16, the beam motion switches from oscillatory to critically damped motion as the viscosity increases. For the beam parameters used in this study, namely $L = 1$ m, $d = 0.04$ m, $\rho_s = 7200$ kg/m³, and $E = 20$ MPa, the switch between the two behaviors is observed to occur around $\mu \simeq 10$ N s/m². It was also noticed that viscosity has little effect on the trajectory of the beam tip (plots are not provided). Nevertheless, compared to the case of a cantilever moving in vacuum, when immersed in fluid, the tip moves on a much shorter path. This deviation, i.e. having the same trajectory regardless of the fluid viscosity, which is different from that of a cantilever in vacuum, is most probably due to the pressure drag which is added to the viscous drag considered herein.

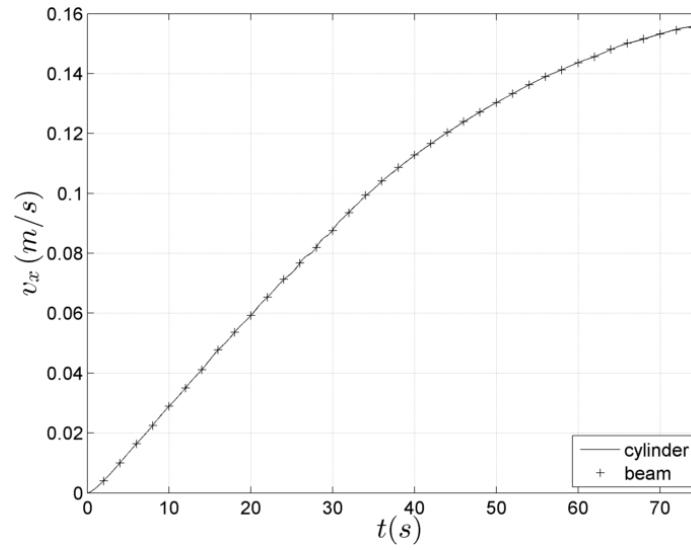
5.3.3 Impulsively started motion of cantilevers in channel flow: Effect of elasticity

The vibration behavior of flexible beams in a viscous fluid was studied by considering an array of cantilevers in channel flow. Unlike the test described in Sect. 5.3.2, here the flexible cantilevers are initially at rest when they are hit by a laminar channel flow. This model can be used to study the effect of horizontal waves on beams submerged in a fluid.

The array of flexible cantilevers is laid out in the xy plane, with $(\Delta x, \Delta y) = (1.2, 0.4)$ m, as shown in Figure 5.17, thus allowing interaction of the beams through the flow. Each beam is anchored in the xz plane with an angle of 30° with respect to the y axis. The fluid, with density



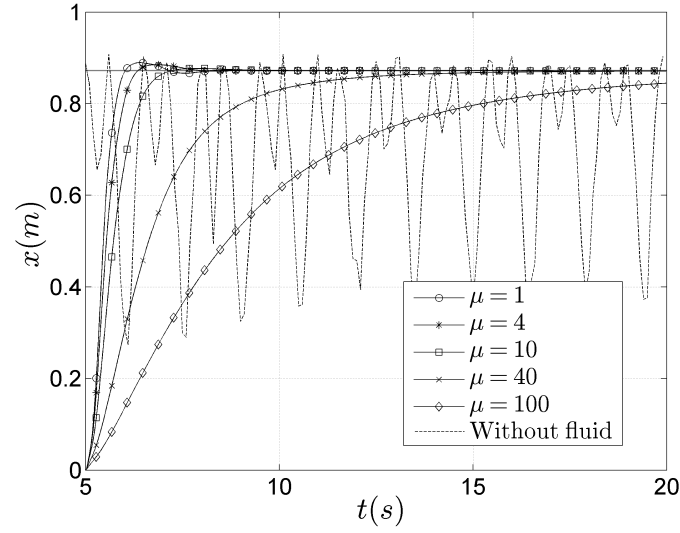
(a)



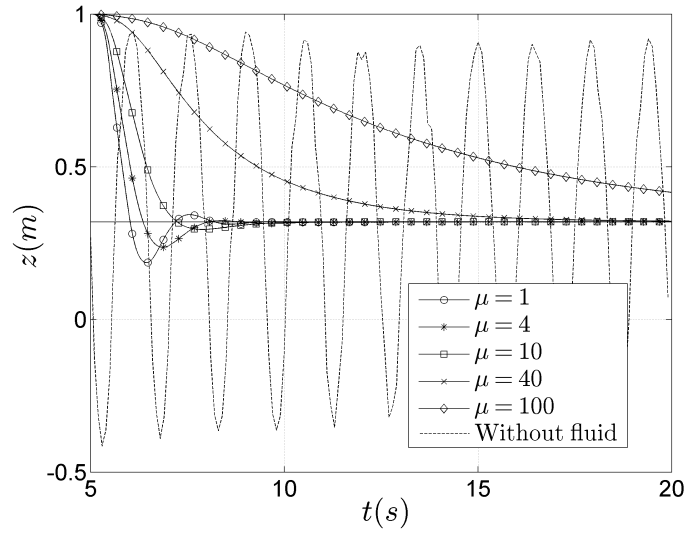
(b)

Figure 5.15: Comparison of the dynamics of a rigid cylinder and of a corresponding stiff deformable beam under accelerating channel flow: (a) beam orientation; (b) center velocity.

$\rho = 1000 \text{ kg/m}^3$ and viscosity $\mu = 1 \text{ N s/m}^2$, flows in the x direction between two planes spaced by $H = 1 \text{ m}$ vertically.



(a)



(b)

Figure 5.16: Motion of a cantilever beam in fluid of different viscosities: (a) tip displacement in x direction; (b) tip displacement in z direction.

Figure 5.18 shows the tip deformation of one cantilever beam for different modulus of elasticity in the range $E \in (0.25, 20)$ MPa. All other beam parameters were kept fixed at $L = 0.7$ m, $d = 0.04$ m, and $\rho_s = 7200$ kg/m³.

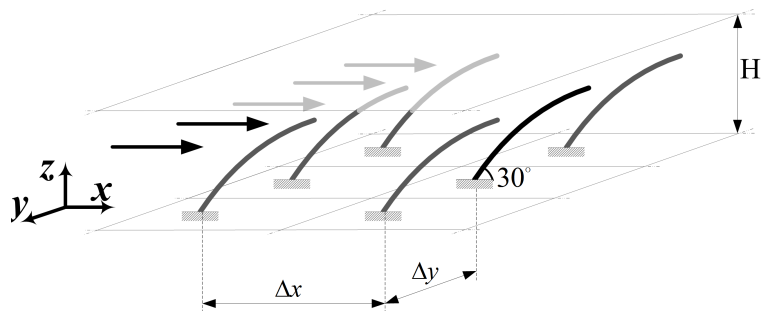
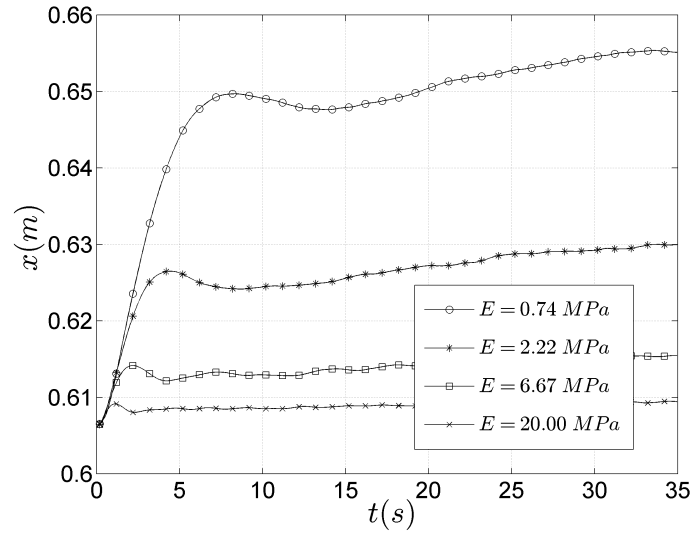
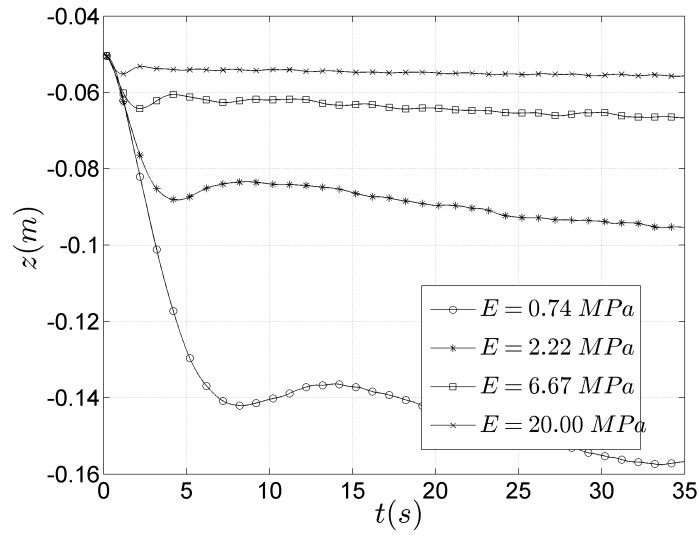


Figure 5.17: Arrays of flexible cantilever beams in laminar channel flow. The beams, laid out in an uniform grid, are anchored at an angle of 30° in the direction of the flow.



(a)



(b)

Figure 5.18: Motion of a cantilever beam of different elasticity modulus in laminar channel flow:
 (a) tip displacement in x direction; (b) tip displacement in z direction.

Chapter 6

Performance Analysis

The simulation approach described in Sect. 4.2 was implemented to execute in parallel on GPU cards using the CUDA programming environment [92]. All simulations reported in this section were run on an NVIDIA GeForce GTX 680 GPU card described in Sect. 4.1. In all simulations, if present, the flexible beams were modeled using $n_e = 4$ ANCF beam elements, while the integrals appearing in the elastic forces $\mathbf{Q}^e$ in Eq. (2.28) were evaluated using 5 and 3 Gauss quadrature points for the axial and bending elastic forces, respectively.

The investigations begin by analyzing the efficiency and scaling attributes of the solution for systems composed exclusively of rigid bodies, flexible bodies, or a fluid phase. In each of these three scenarios, the simulation times for one-step dynamics update is provided for problems of increasing size. Data provided in Tables 6.1, 6.2, and 6.3 and illustrated in Figure 6.1 indicate that updates of the dynamics of each phase scale linearly with problem size; i.e., with the number of rigid bodies, flexible bodies, and SPH markers, respectively.

While the previous results show linear scaling for rigid and flexible body dynamics, in actual FSI problems in which rigid and/or flexible bodies interact with the fluid, the simulation time

Table 6.1: Time required for advancing the rigid body dynamics simulation by one time step as function of problem size (number of rigid bodies, N_r)

$N_m = 0, N_f = 0$					
N_r ($\times 10^3$)	0.49	2.87	16.59	56.77	118.23
t (ms)	5	8	16	44	78

Table 6.2: Time required for advancing the flexible body dynamics simulation by one time step as function of problem size (number of flexible bodies, N_f)

$N_m = 0, N_r = 0$					
N_f ($\times 10^3$)	0.78	3.51	17.55	56.94	115.05
t (ms)	8	14	48	122	238

Table 6.3: Time required for advancing a fluid dynamics simulation by one time step as function of problem size (number of SPH markers, N_m)

$N_r = 0, N_f = 0$					
N_m ($\times 10^6$)	0.06	0.32	0.93	1.79	4.13
t (ms)	27	121	331	538	1150

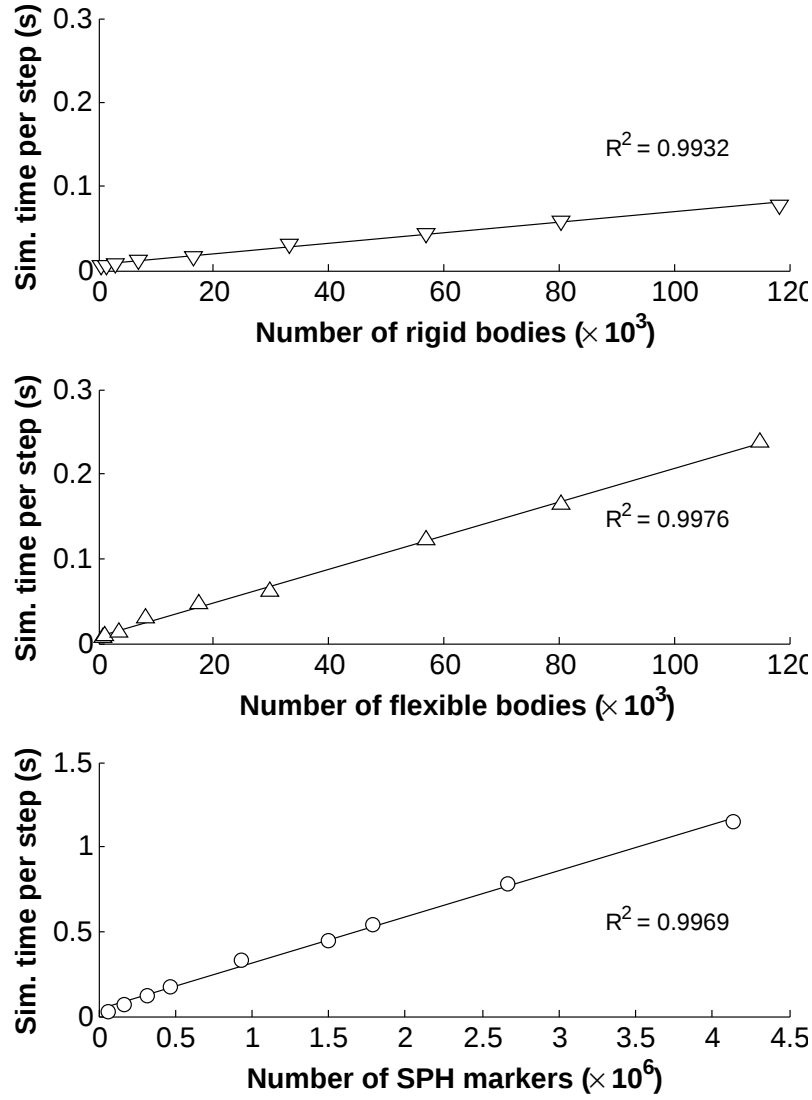


Figure 6.1: Scaling analysis of Chrono::Fluid for rigid body dynamics, flexible body dynamics, and fluid dynamics as function of problem size (N_r , N_f , and N_m , respectively). The coefficients of determination, R^2 , are specified to each linear regression. Virtually exact linear trends are indicated since $R^2 \approx 1$.

is virtually independent of the number of solid objects. The results presented in Tables 6.4 and 6.5 were obtained on a system consisting of approximately 3 million SPH markers by varying the

Table 6.4: Time required for advancing the dynamics of FSI problems by one time step. The simulation times are provided for FSI problems with fixed number of SPH markers and increasing number of rigid bodies

$N_m \simeq 3.0 \times 10^6, N_f = 0$							
N_r	0	36	120	480	1800	8400	33 600
t (ms)	906	919	923	925	926	926	921

number of rigid bodies and flexible bodies. The small sensitivity of the simulation time with respect to the number of solid objects is due to the fact that the number of BCE markers associated with solid bodies represents only a very small fraction of the number of SPH discretization markers, the latter overwhelmingly dictating the required computation time. Nevertheless, as the concentration of solid objects increases, smaller time steps are required since the probability of short-range, high-frequency interactions increases.

The two sets of results provided in Table 6.5 for different values of $\tau_m = \Delta t_{SPH} / \Delta t_{ANCF}$, yet the same Δt_{SPH} , show a small increase in the simulation time when choosing a very small integration step size for flexible body dynamics. This illustrates the efficiency of the multi-rate integration scheme in improving the time integration stability at a small increase in computational cost. The small changes in the simulation times provided in Tables 6.4 and 6.5 are mainly due to the deviations in the magnitude of N_m as the number of solid objects changes. Linear scaling in Chrono::Fluid is also demonstrated in an experiment where a combined FSI problem, i.e. involving rigid and flexible bodies and including a lubrication force model and two-way coupling with fluid, is solved on domains of increasing size (see Figure 6.2). As the simulation domain volume is increased by factors from 2 up to 32, the number of SPH markers varies from about 76,000 to more than 2.5 million. Simultaneously, the number of rigid and flexible bodies grow from 168 to more than 24,000 and from 160 to almost 10,000, respectively (see Table 6.6). As shown in Figure 6.2, Chrono::Fluid achieves linear scaling over the entire range of problem sizes.

Table 6.5: Time required for advancing the dynamics of FSI problems by one time step. The simulation times are provided for FSI problems with fixed number of SPH markers and increasing number of flexible bodies, for two different values of the multi-rate integration factor $\tau_m = \Delta t_{SPH}/\Delta t_{ANCF}$

$N_m \simeq 3.0 \times 10^6, N_r = 0$								
N_f		0	45	140	440	1152	2100	4704
$\tau_m = 10$	t (ms)	906	923	928	916	960	950	921
$\tau_m = 50$	t (ms)	906	973	978	965	1066	1060	1060

Table 6.6: Time required for advancing the dynamics of combined FSI problems by one time step. The simulation times are provided for FSI problems of increasing size

N_m ($\times 10^6$)	0.08	0.16	0.29	0.63	0.95	1.54	2.50
N_r ($\times 10^3$)	0.17	0.52	1.12	4.48	7.84	14.56	24.64
N_f ($\times 10^3$)	0.16	0.42	0.84	2.10	3.36	5.88	9.66
t (ms)	45	74	120	230	343	522	820

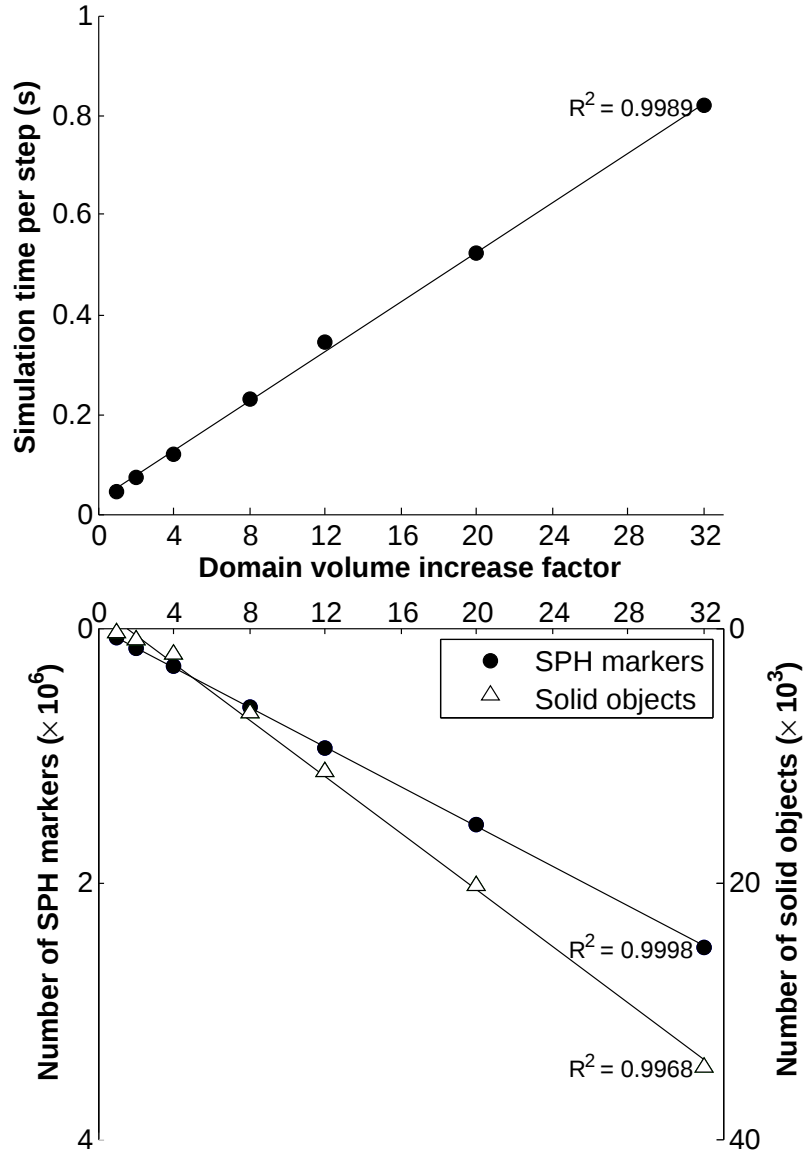


Figure 6.2: Scaling analysis of Chrono::Fluid for FSI problems: simulation time as function of combined problem size. In this experiment, the volume of the simulation domain is increased, up to 32 times the volume of the initial domain, leading to proportional increases in the number of SPH markers and of solid objects (both rigid and flexible bodies) as shown in the bottom plot (see also Table 6.6). As illustrated by the top graph, the simulation time for an one-step dynamics update varies linearly with problem size. The coefficients R^2 are specified for each linear regression.

Chapter 7

Demonstration of technology

The purpose of this chapter is to further demonstrate the capabilities of the proposed framework for the simulation of FSI problems. The problems provided herein are meant to illustrate the type and size of the real-world engineering applications that can be approached via the developed FSI framework for investigation and parametric study purposes.

7.1 Flow cytometry using microfluidic techniques

New flow-cytometry technologies are being developed to rapidly and accurately analyze the composition and function of large particles ($a \approx 25\text{--}500\ \mu\text{m}$) without affecting cell viability [14]. Purification of cell aggregates or 3D engineered tissues would be beneficial for many basic biomedical and clinical applications. Single cells have been efficiently purified using fluidic approaches for several decades. However, purification of 3D microtissues (i.e. large particles with $a > 50\ \mu\text{m}$) and cell aggregates has been difficult. In this context, purification or sorting of large particles and cell aggregates at different conditions continues to demand for an accessible platform.

One of the most common approaches for particle sorting is fluorescence activated cell sorting (FACS), which is accomplished by applying an electromagnetic charge to a droplet containing a single particle, e.g. cell. The particle is deflected accordingly by applying an electric field. This technique has inherent limitations. As the particle size increases, local disturbances can affect the predictability of particle behavior. Additionally, the increased particle inertia due to a size increase can significantly affect its mobility under the electromagnetic field.

One of the few, if not the only, commercially available, large particle FACS systems is the COPAS Bio Sorter [128]. The apparatus is capable of measuring different particle parameters including size, extinction coefficient (optical density), and fluorescence parameters, for particle sizes ranging from 5 to 750 μm . The sorting mechanism is pneumatic and includes a solenoid valve that is actuated/opened in response to defined optical parameters. Therefore, unwanted particles are directed to a waste container. The system is sophisticated in sorting large particles, but is limited in characterizing particles in aggregates.

Microfluidic platforms have been utilized to increase the efficiency and accessibility of systems designed to purify small particles [48, 53, 56, 91] and more recently large particles based on particle size [5]. Particularly, a capillary sorting mechanics was developed using diode laser bars to optically trap or deflect particles up to 100 μm in radius [5, 70]. Theoretically, different size particles are sorted by size based on the supplied laser power. However, the practical application of this approach is limited to particles smaller than 200 μm due to the constraint on the size of the laser bar. Moreover, the impact of the laser power on the function and viability of biologic particles is unknown. Alternatively, physical separation schemes can be leveraged to sort large particles without affecting their viability [13, 70].

A microfluidic design for sorting small particles is proposed in [26]. Therein, particles move with the flow through an asymmetric channel, which intensifies the flow inertial effect and uses hydrodynamic forces to order and separate particles. The design reduces the setup complexity and the reagent consumption while allowing for higher throughput. A similar approach was leveraged in the design of an asymmetric channel for sorting large particles, as shown in Figure 7.1a [13]. In this setup, the line of particles passes the optical interrogation point and upon the detection of a desired event, sorting is triggered. The main and sorting branches, bifurcated at the channel outlet, are controlled by micro-valves to direct the particles to a desired direction. A single phase flow analysis of the setup was provided in [13]. This model will not suffice for the analysis of the drafting, kissing, and tumbling (DKT) of the particles, which has a dominant role in particle sorting via the microfluidic device. Moreover, the sorting efficiency is Reynolds number dependent

and requires a parametric study that accounts for particle size and flow rate. These requirements raised the need for numerical simulation performed herein on a similar setup.

The simulation setup, shown in Figures 7.1b and 7.1c, consists of an asymmetric channel similar to the experimental one. Figure 7.1c shows the channel profile in x - y plane. The height of the channel, i.e. z component, is 1 mm and gravity is applied in the z direction. This geometry was imposed by using BCE markers described in Sect. 3.2. A particle speed between 1.5 and 2 mm/s was obtained by adjusting a flow rate range between 400–1000 $\mu\text{l}/\text{min}$. This flow rate and the channel profile shown in Figure 7.1c accommodate particles in the range of $a = 70\text{--}250\ \mu\text{m}$ [25]. The channel height is the only limiting factor on maximum particle size, therefore it can be modified to accommodate larger particles.

7.2 Dense particle suspension

The model considered in this section is that of a dense suspension composed of about 2.3×10^4 neutrally buoyant rigid bodies in fluid flow. The fluid domain is modeled via approximately 2.0×10^6 markers. Considering all of the intermediate data required for RK2 integration scheme, approximately 1.0×10^8 state variables are required to capture the evolution of the system state.

The rigid particles involved in this simulation are ellipsoids with radii of (1.5, 1.5, 2) cm, flowing with liquid inside a channel of dimensions of (1.1, 1, 1) m, at $Re = 66$. While wall boundary conditions are applied at the bottom and top of the channel in z -direction, periodic boundary conditions are considered in the y -direction to generate a 2D flow. This setup models a dense suspension with volumetric fraction of $\phi \approx 40\%$. The choice of other shapes for rigid bodies is just a matter of initialization as the solution algorithm does not use any assumption in this regards.

Figures 7.2 and 7.3 show snapshots of this simulation. The mid-section of the flow is shown in Figure 7.2 where the color illustrates the velocity profile. Figure 7.3 shows only the rigid particles. The snapshots are taken at about $t^* = 4$, where the non-dimensional time, t^* , is defined based on real time, t , average flow velocity, V , and the channel width, w , as $t^* = t \times V/w$.

The concentration distribution curves obtained for this investigation are noisier than those obtained experimentally by Han *et. al* [41]. Long simulation times (≈ 100 GPU hour per test)

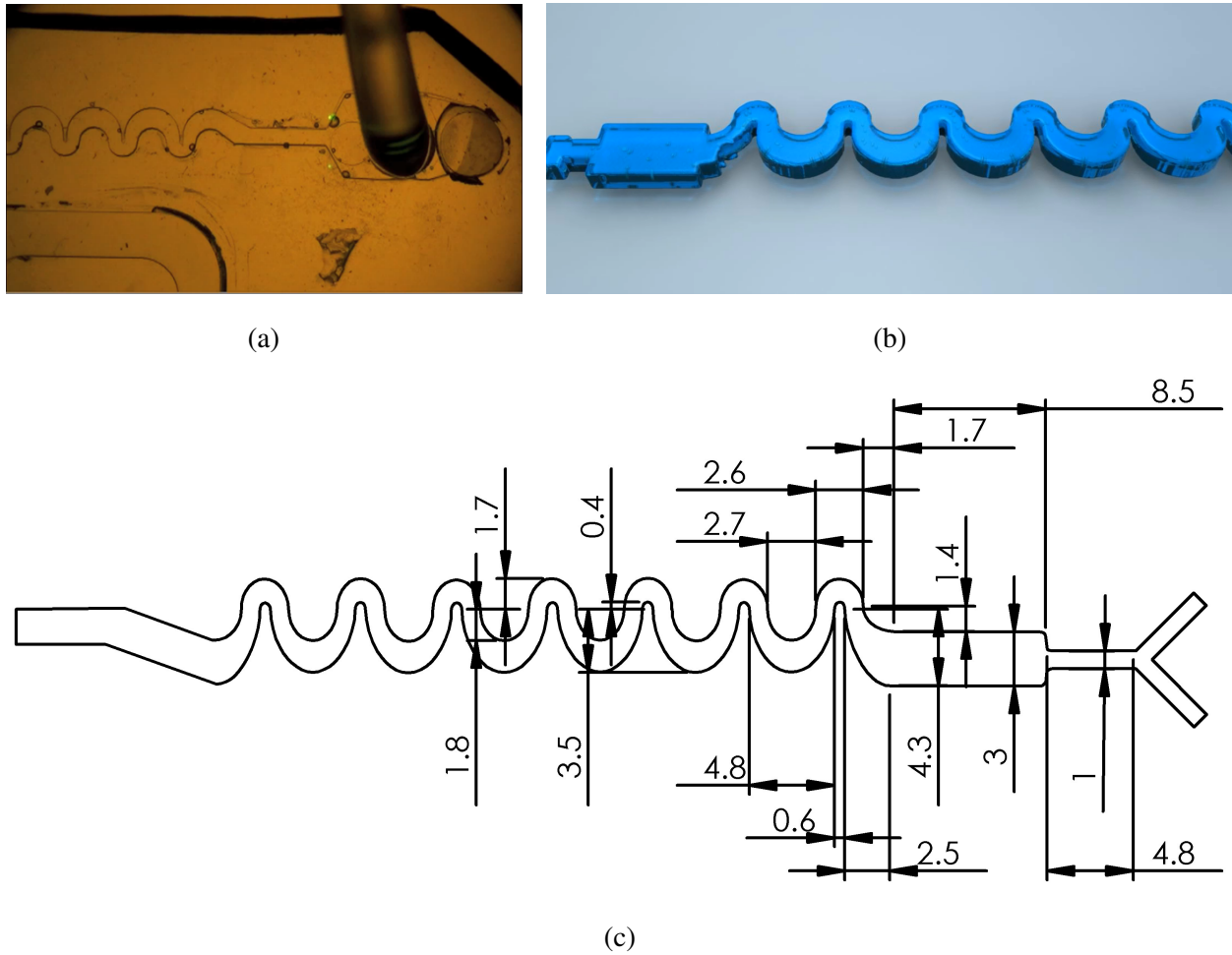


Figure 7.1: Numerical investigation of flow cytometry using asymmetric channel: (a) experiment setup showing the channel and the particle feeding valve; (b) snapshot of the numerical simulation; (c) channel profile used in experiment and simulation. All dimensions are in mm.

prevented us from running a sufficiently large number of simulations required for statistical averaging and bootstrapping. An effort is underway to reduce the simulation times to enable the study of suspensions with moderate to high concentrations.

7.3 Microscale numerical simulation of flow in porous media

There is a class of applications characterized partly by flow within porous media. Examples of such systems include, but are not limited to: (1) oil recovery from deep wells; and (2) biological

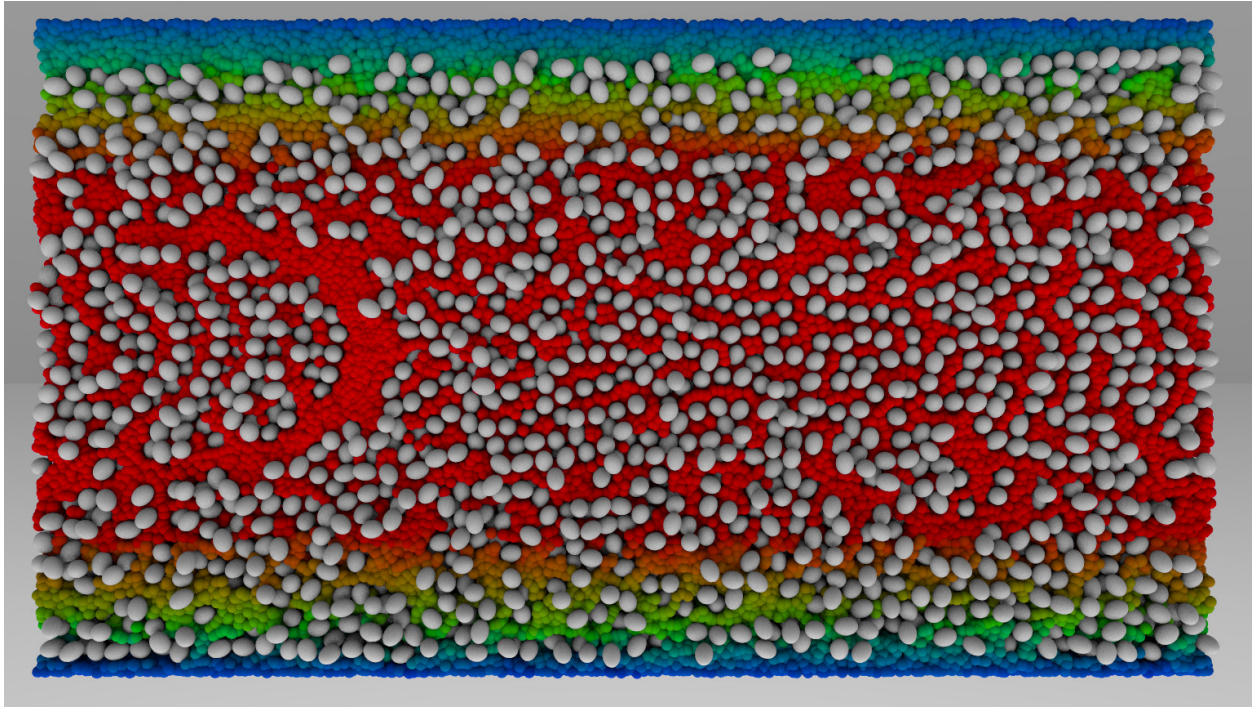


Figure 7.2: Direct numerical simulation of a dense suspension of ellipsoids in 3D square channel flow. Only the mid-section of the flow, which shows the rigid ellipsoids suspended in the fluid, is shown. The coloring scheme represents fluid velocity: from zero (blue) to maximum (red).

applications such as diffusion of nutrients and other macromolecules, e.g. drugs, across and within biological tissues, transport in brain tissues, MRI applications, liquid chromatography, transport of macromolecules in aortic media, and blood flow through muscles. These problems can be approached via the proposed framework, where the porous media is modeled with BCE markers and the rigid/flexible motion is suppressed.

An example simulation of flow within porous media is shown in Figure 7.4. The porous matrix was generated by including overlapping randomly-generated fixed rigid objects. The porosity of the matrix can be adjusted by changing the number or size of rigid objects. Note that a point cloud representing the porous matrix can be generated directly using data from a scanning electron micrograph.

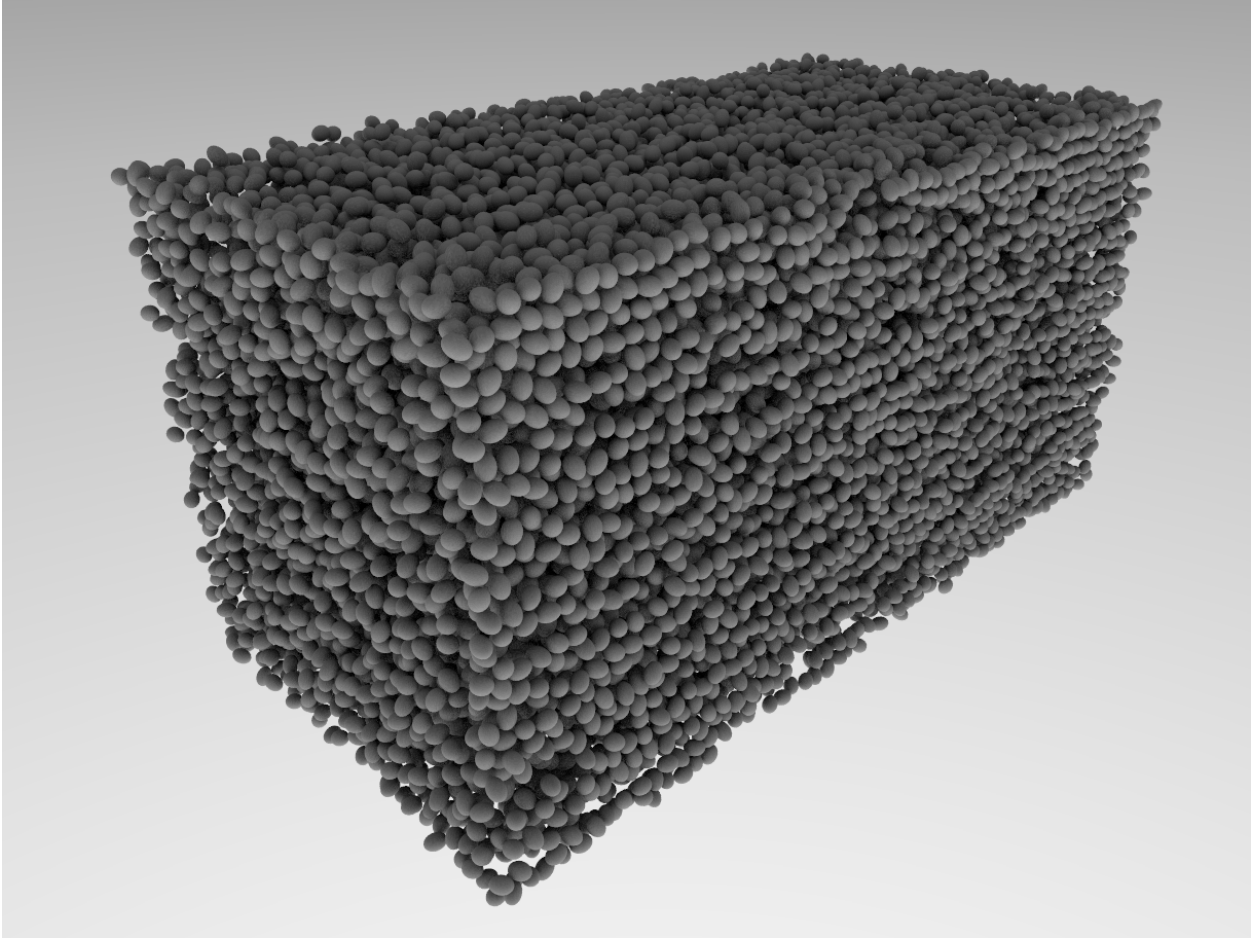


Figure 7.3: Direct numerical simulation of a dense suspension of ellipsoids in 3D square channel flow. The fluid was removed to show a perspective view of the rectangular channel and the ellipsoids.

7.4 Immersed deformable objects

This simulation involves a channel flow passing over an array of beams that are fixed at one end and free to move at the other. The flexible cantilevers can be replaced by flexible pendulums or flexible beams pivoted at both ends via minor modifications performed during the initialization phase.

The simulation involves a similar channel described in Sect. 7.2, with the difference that the fluid flows in both x and y directions, i.e. parallel to the wall, with some induced random behavior.

The beams have $L = 0.8$ m, $\rho_s = 1000$ kg/m³, $E = 0.2$ MPa, and $d = 0.04$ m. The simulation was performed at $Re \leq 20$ using an array of 630 beams and approximately 1.25×10^6 markers. An snapshot of the simulation is shown in Figure 7.5. The simulation animation can be accessed from [112]. This simulation scenario can be modified to tackle specific applications such as structural stresses at geological scales.

7.5 Interacting rigid and flexible objects in channel flow

Two simulations involving fluid flow, rigid bodies, and flexible beams were conducted herein. The solid objects involved in these simulations interact with fluid via a two-way coupling model. Additionally, a solid-solid interaction model is employed to capture rigid-rigid, rigid-beam, and beam-beam interactions.

In the first simulation, an immersed array of cantilever beams is subjected to a flow of a suspension of rigid objects. The rigid phase is composed of approximately 2000 neutrally buoyant ellipsoids of radii $(a_1, a_2, a_3) = (2.25, 2.25, 3)$ cm. The flexible bodies are 64 cm long beams with radius $a = 1.5$ cm, modulus of elasticity $E = 0.2$ MPa, and density $\rho_s = 1000$ kg/m³. The fluid, which has the density and viscosity equal to $\rho = 1000$ kg/m³ and $\mu = 1$ Pa · s, respectively, flows within a channel with dimensions equal to $(l_x, l_y, l_z) = (1.4, 1, 1)$ m. Periodic boundary conditions are applied to both the x and y directions of the flow. The flow and particle Reynolds numbers for this simulation are $Re = 45$ and $Re_p = 2$, respectively, where Re_p is calculated based on the average particle diameter, i.e. $D = 2a = 2(a_1 a_2 a_3)^{1/3}$.

Several snapshots of this simulation are provided in Figure 7.6. Therein, the colors show the velocity, from zero (blue) to maximum (red). Animations showing the transient behavior of this simulation and similar simulations composed of softer and stiffer beams are available online [112].

Problems involving the flow of flexible objects, e.g. polymer, with fluid are usually considered within the framework of non-Newtonian fluid dynamics. Such problems can also be approached via direct numerical simulation, which in turn provides the opportunity of investigating details such as stress distribution in the components. This problem is approached herein using the geometry and material properties similar to those used in the previous problem. The beams, however, are

not constrained and move with the fluid. A snapshot of this simulation is shown in Figure 7.7. Several similar simulations were performed considering extremely soft beams with $E = 0.2$ KPa to moderately soft beams with $E = 0.2$ MPa. Although only straight beam components were studied herein, other micro-structures can be considered by connecting beams to one another. In the current simulation engine, this is possible by reducing the set of independent coordinates as described in Sect. 2.3.

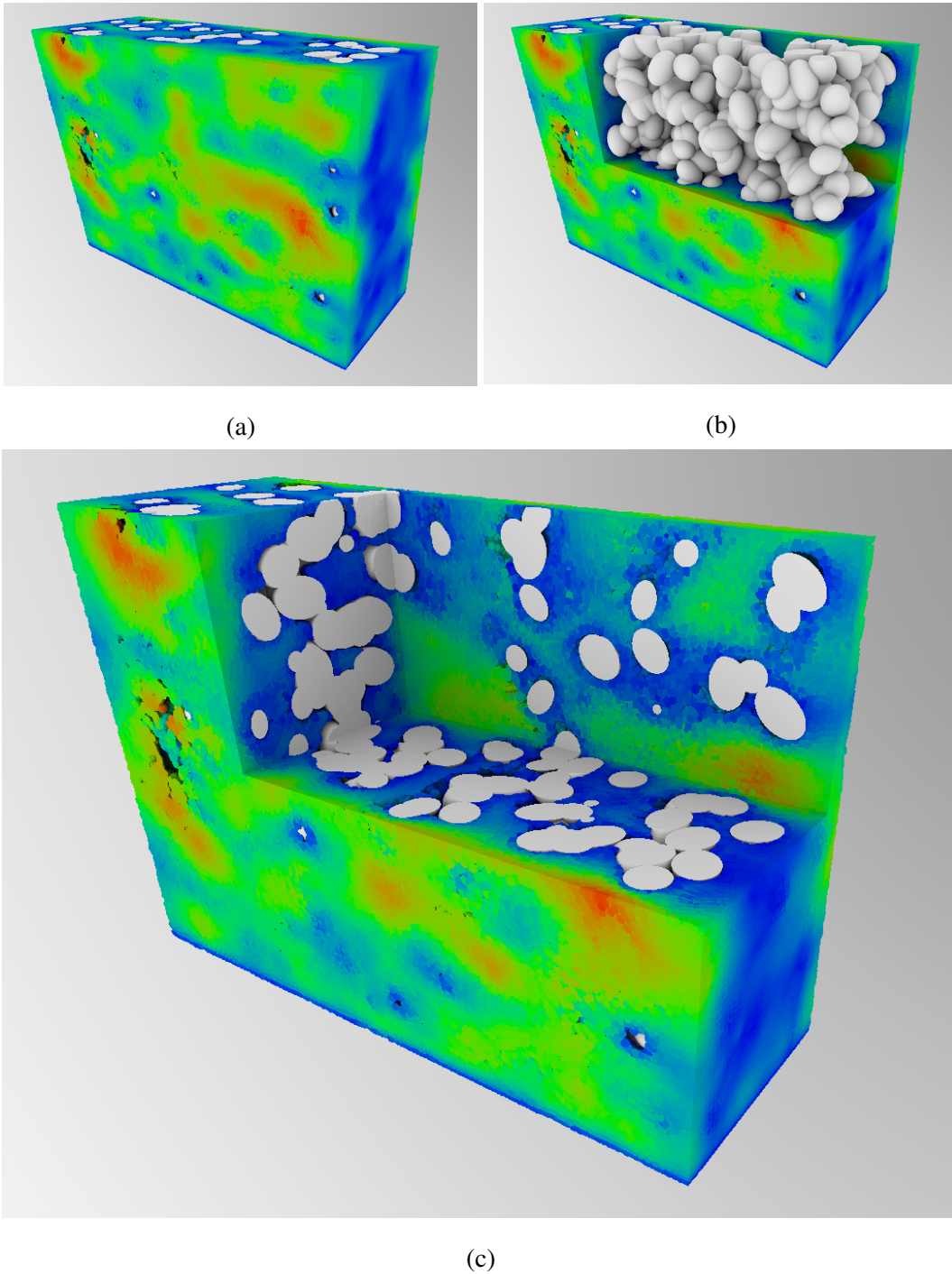


Figure 7.4: Example simulation of flow in porous media. The entire domain is shown in (a). For a clear visualization, the fluid and porous matrix are partially removed in (b) and (c), respectively. The color represents fluid velocity: from zero (blue) to maximum (red). The maximum flow velocity is approximately 6 mm/s, i.e. $Re \approx 6$.

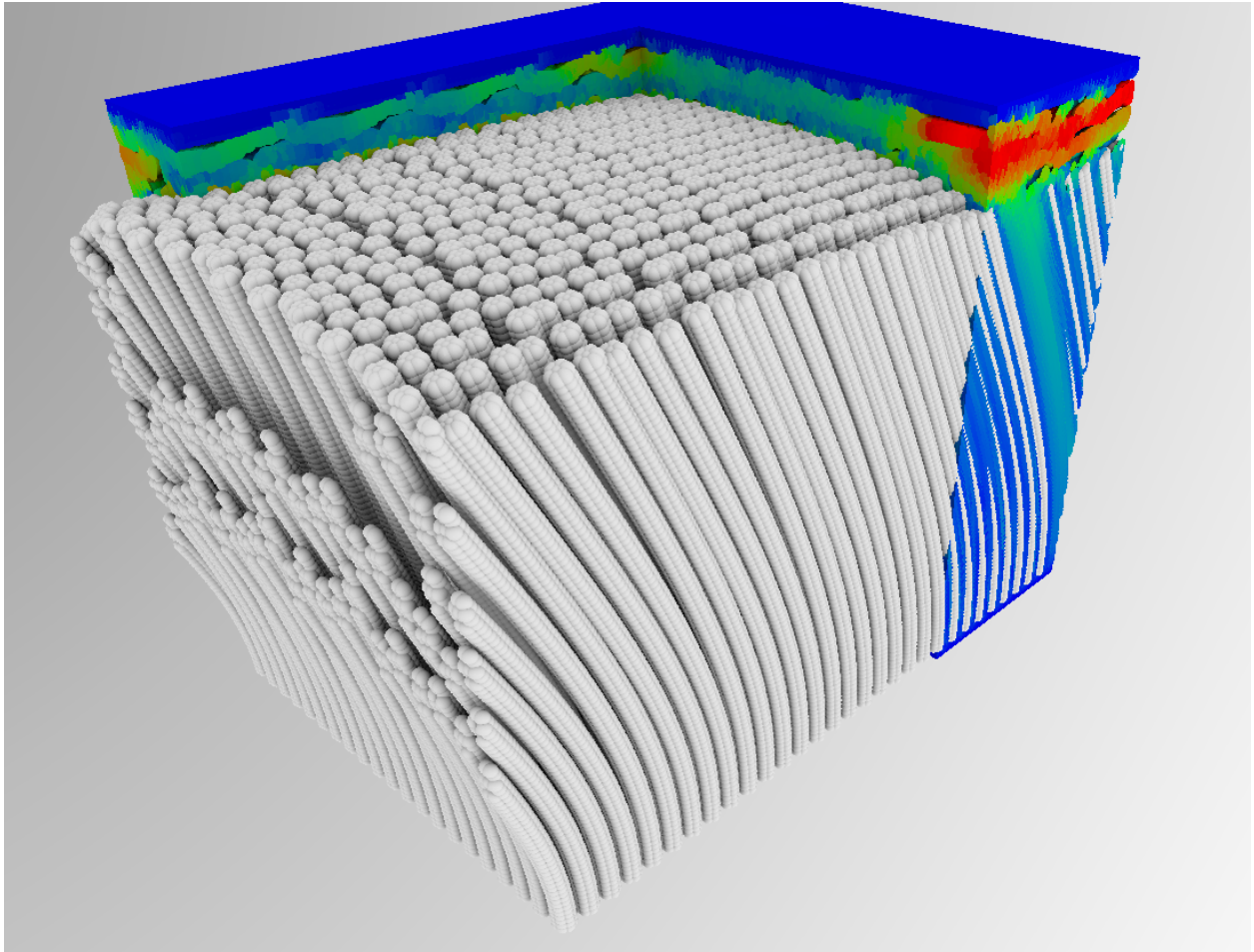


Figure 7.5: Example simulation of flow over a dense array of interacting beams.

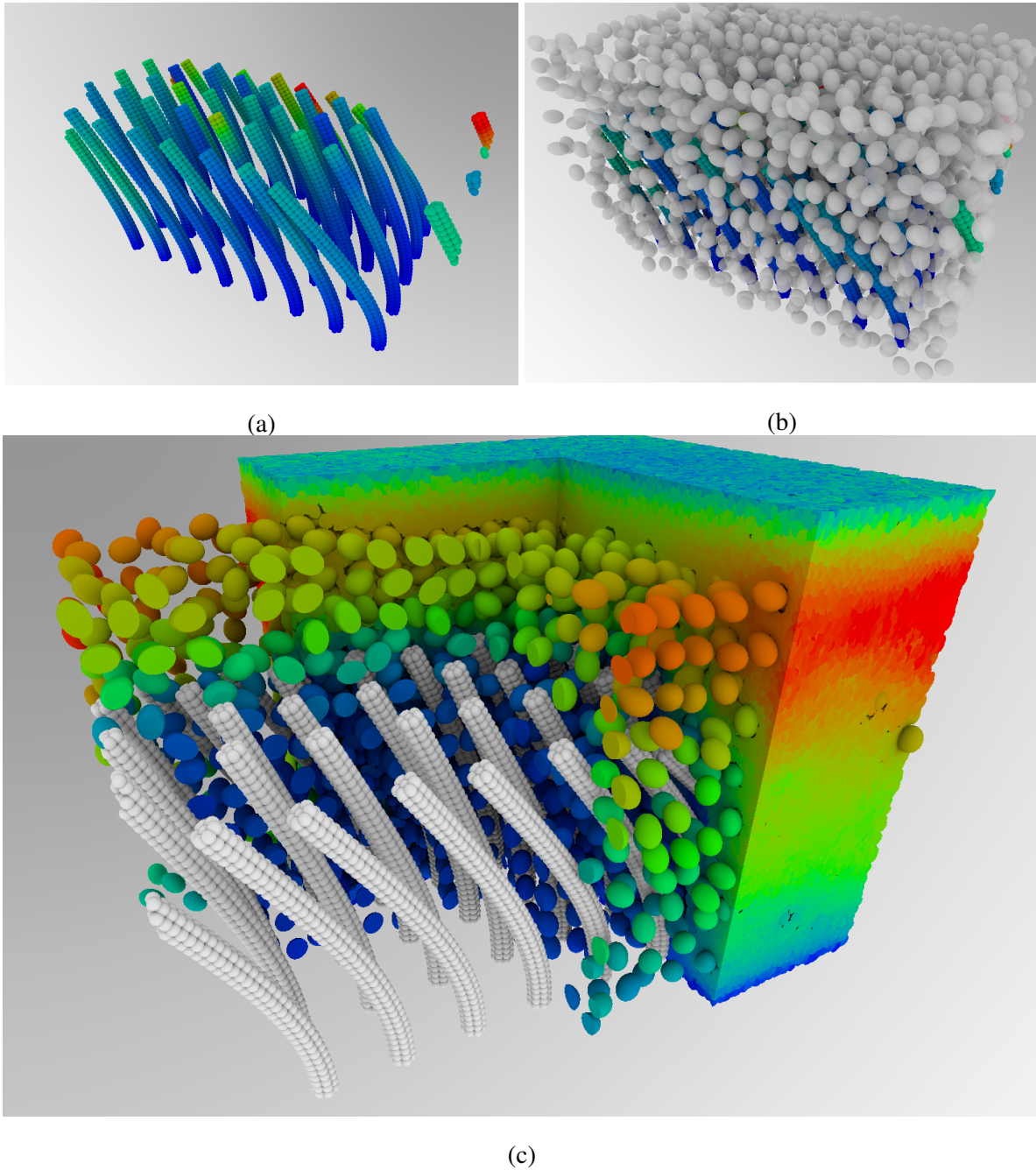


Figure 7.6: Example simulation of flow of rigid bodies within an array of deformable beams. For a clear visualization, only parts of the domain are shown in each picture: (a) beams; (b) beams and rigid bodies; (c) beams, rigid bodies and fluid flow, cut at different sections. The color represents the velocity: from zero (blue) to maximum (red), with $V_{max}^{fluid} = 0.045$ m/s, $V_{max}^{rigid} = 0.041$ m/s, and $V_{max}^{beam} = 0.005$ m/s.

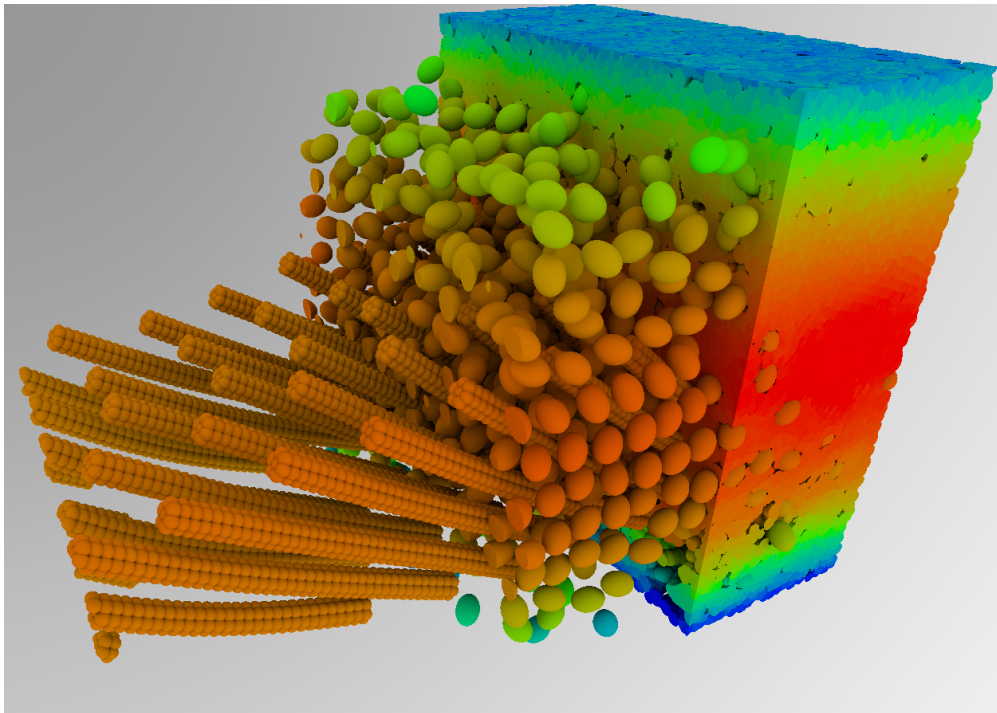


Figure 7.7: Flow of a suspension of rigid and deformable objects. Parts of the rigid and fluid phase are removed for a clearer illustration.

Chapter 8

Directions of the future work and preliminary results

8.1 New numerical methods: Implicit integration for SPH

This section describes ongoing work aimed at producing an implicit integration method for resolving the dynamics of the fluid phase in an SPH framework. In this approach, the Navier-Stokes equations, i.e. Eq.(2.6), are solved by considering the incompressibility condition as a constraint obtained from Eq. (2.5) by setting $d\rho/dt = 0$:

$$\nabla \cdot \mathbf{v} = 0. \quad (8.1)$$

Equations (2.6) and (8.1) combine to form a so-called Hessenberg Index-2 DAE, which can be expressed as

$$\Psi(\mathbf{v}, p) = \mathbf{0}, \quad (8.2)$$

and is to be solved for $\mathbf{v}$ and p . The incompressibility is an outcome of the solution strategy, since this condition was applied as a constraint. This approach is different from the projection method [22] since the pressure is obtained by considering the constraint on the current instead of the predicted velocities. The proposed approach has first been tested on a 2D problem using an Eulerian approach described in Sect. 8.1.1.

8.1.1 Incompressible fluid flow on a grid

In 2D, the Navier-Stokes and incompressibility equations are written as

$$\frac{\partial u}{\partial t} + \frac{\partial(uu)}{\partial x} + \frac{\partial(uv)}{\partial y} = -\frac{1}{\rho} \frac{\partial p}{\partial x} + \frac{\mu}{\rho} \left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} \right) + f_x \quad (8.3a)$$

$$\frac{\partial v}{\partial t} + \frac{\partial(uv)}{\partial x} + \frac{\partial(vv)}{\partial y} = -\frac{1}{\rho} \frac{\partial p}{\partial y} + \frac{\mu}{\rho} \left(\frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2} \right) + f_y \quad (8.3b)$$

$$\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} = 0. \quad (8.3c)$$

Herein, u and v denote the x and y components of the fluid velocity, respectively. A finite difference approach on an staggered grid was adopted to numerically solve the set of equations provided in Eq. (8.3).

A staggered grid is one in which velocities and pressure are located at different positions. A typical staggered grid cell is shown in Figure 8.1a. Therein, the horizontal arrows, vertical arrows, and center dot show the locations of u , v , and p , respectively. A common indexing approach for staggered grid taxonomy is shown in Figure 8.1b. The indices of the left and bottom velocities are associated with the cell index i, j . The result of this indexing convention on the whole domain is shown in Figure 8.2. It is worth noting that the bounds on the i and j indices of each variable, e.g. u , v , and p , is different than the other variables.

With a staggered grid, there is some ambiguity about indexing convention since indices like $i + 1/2$ and $j + 1/2$ may be addressed differently when denoting the momentum equations in x and y directions. To address such ambiguities, Appendix A provides the discretization of the continuity and momentum equations.

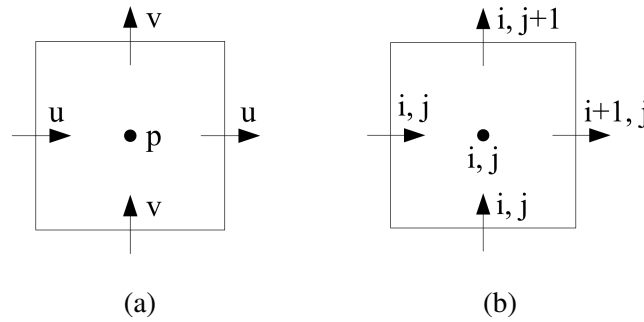


Figure 8.1: A typical staggered grid cell: (a) positions of velocities and pressure; (b) indexing convention.

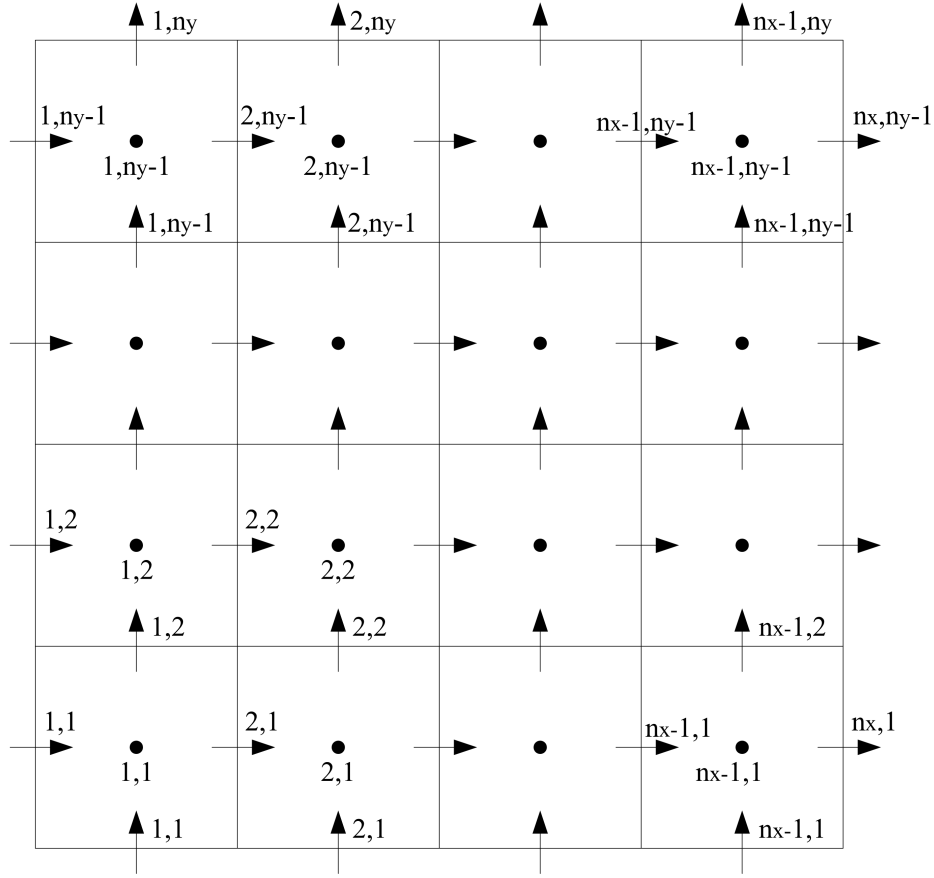


Figure 8.2: Schematic of the domain indexing in staggered grid taxonomy.

A Newmark algorithm [90] was implemented to determine the steady state solution to the system of equations obtained through the discretization of Eq. (8.3). An extra constraint was added to the system by setting the pressure of one node to zero. This is necessary due to the fact that the solution for $\mathbf{p}_0$ is exact up to a constant; i.e. if $\mathbf{p}$ is a solution, then any constant offset of $\mathbf{p}$ is a solution too. Herein $\mathbf{p}$ denotes the array of pressure values at all nodes.

As a case study, a lid driven cavity flow was considered herein. Figure 8.3 shows the results obtained at $Re = 100$ where contours of velocities and pressure are provided.

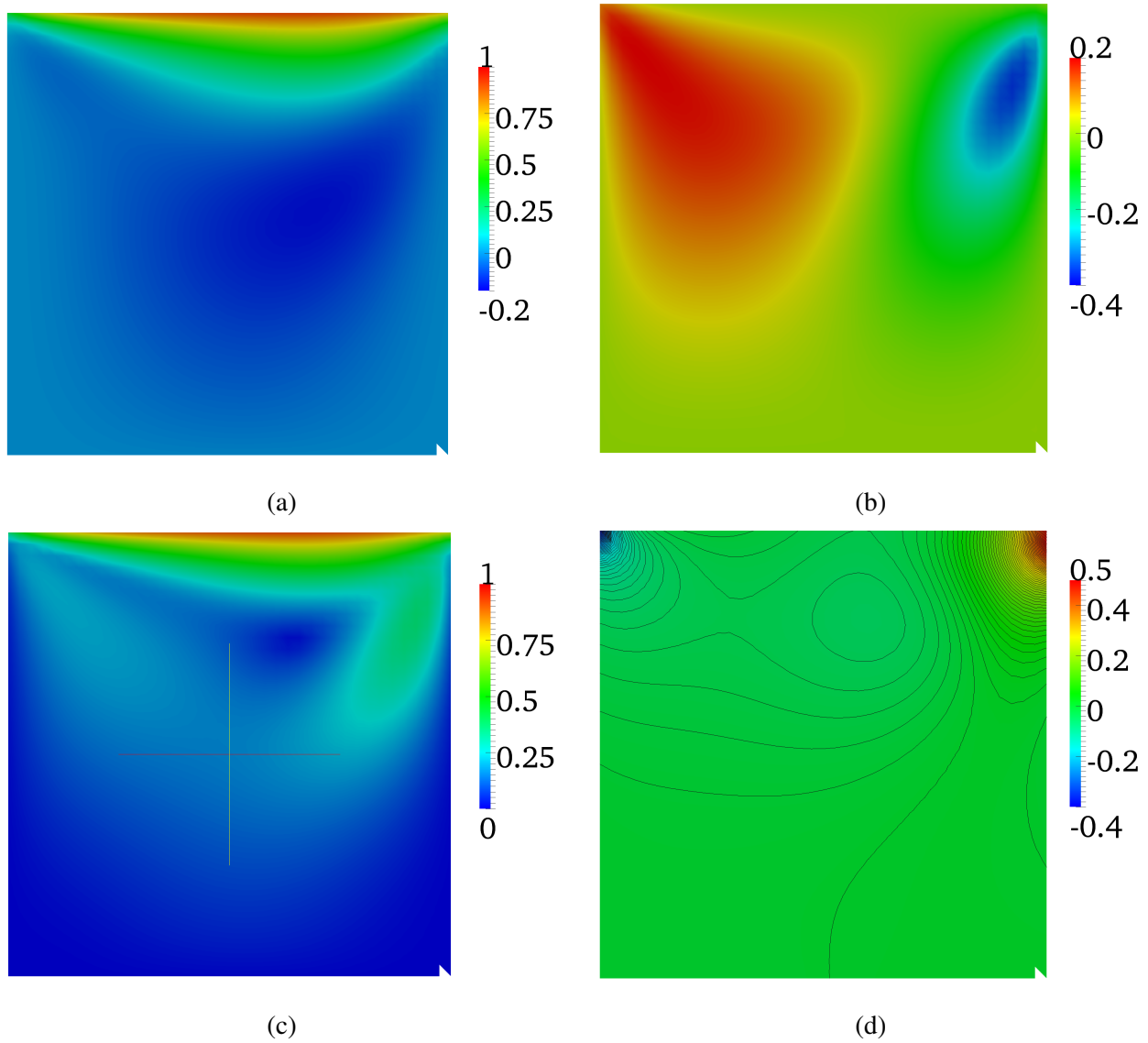


Figure 8.3: Cavity flow solution using staggered grid at $Re = 100$. The contours show: (a) velocity in x direction; (b) velocity in y direction; (c) magnitude of velocity; (d) pressure.

8.1.2 Incompressible SPH

Similar to Sect. 8.1.1, the solution to the incompressible flow is approached herein via a Hessenberg Index-2 DAE formulation where a SPH discretization is implemented instead of a staggered grid. This formulation requires the derivation of sensitivity matrix in conjunction with SPH methodology, which is described in details in this section.

The continuity and Navier-Stokes equations given in Eqs. (2.5) and (2.6) can be described at each Lagrangian point as,

$$\dot{\mathbf{v}}_a + \boldsymbol{\xi}_a - \boldsymbol{\zeta}_a = \mathbf{k}_a \quad (8.4a)$$

$$\chi_a = 0 \quad (8.4b)$$

where a is marker's identifier; and $\boldsymbol{\xi}_a$, $\boldsymbol{\zeta}_a$, and $\mathbf{k}_a$ are pressure, viscosity and body force contributions, respectively. The divergence of the velocity field, χ_a , is enforced to be zero through the constraint equation, Eq. (8.4b). We have used SPH to discretize Eq. (8.4) as,

$$\boldsymbol{\xi}_a = \sum_b m_b \left(\frac{p_a}{\rho_a^2} + \frac{p_b}{\rho_b^2} \right) \nabla_a^T W \quad \in \mathbb{R}^3 \quad (8.5a)$$

$$\boldsymbol{\zeta}_a = \sum_b m_b \frac{2\bar{\mu}}{\bar{\rho}^2} \frac{1}{\mathbf{r}^T \mathbf{r} + \eta^2} (\mathbf{r}^T \nabla_a^T W) \dot{\mathbf{r}} \quad \in \mathbb{R}^3 \quad (8.5b)$$

$$\chi_a = \sum_b \frac{m_b}{\rho_b} \dot{\mathbf{r}}^T \nabla_a^T W = \sum_b \frac{m_b}{\rho_b} (\nabla_a W) \dot{\mathbf{r}} \quad \in \mathbb{R}, \quad (8.5c)$$

where $\dot{\mathbf{r}} = d\mathbf{r}/dt = \mathbf{v}_a - \mathbf{v}_b \in \mathbb{R}^3$, $\bar{\mu} = \mu_a + \mu_b$, $\bar{\rho} = \rho_a \rho_b$ and $\eta = \varepsilon \bar{h}_{ab}^2$.

8.1.3 Implicit method

8.1.3.1 Setup

Given a function $q : \mathbb{R}^3 \rightarrow \mathbb{R}$, $q = q(\mathbf{r})$, define its gradient and Jacobian as:

$$q_{\mathbf{r}} \equiv \nabla q \quad \in \mathbb{R}^{1 \times 3} \quad (8.6a)$$

$$q_{\mathbf{r}\mathbf{r}} \equiv \partial_{\mathbf{r}} [q_{\mathbf{r}}^T] \in \mathbb{R}^{3 \times 3}. \quad (8.6b)$$

Consider a kernel function $W : \mathbb{R} \rightarrow \mathbb{R}$, $W = W(q)$ of sufficient smoothness, whose first and second derivatives with respect to q are scalars denoted as W' and W'' . The kernel function W can be seen as an implicit function of $\mathbf{r}$ through $q(\mathbf{r})$, i.e.

$$W : \mathbb{R}^3 \rightarrow \mathbb{R}, \quad W(\mathbf{r}) \equiv W(q(\mathbf{r})). \quad (8.7)$$

The gradient of W with respect to $\mathbf{r}$ is therefore

$$\nabla W = W' q_{\mathbf{r}} \in \mathbb{R}^{1 \times 3} \quad (8.8)$$

and, for notational convenience, let

$$\nabla^T W \equiv (\nabla W)^T \in \mathbb{R}^3. \quad (8.9)$$

Now let $\mathbf{r}^{ab} = \mathbf{x}_a - \mathbf{x}_b$. For brevity, the “ ab ” superscript will be dropped from $\mathbf{r}$. Then $\partial_{\mathbf{x}_a}[\mathbf{r}] = \mathbf{I}_3$ and $\partial_{\mathbf{x}_b}[\mathbf{r}] = -\mathbf{I}_3$. Reliance of W on q and $\mathbf{r}$, can be observed as

$$W : \mathbb{R}^3 \times \mathbb{R}^3 \rightarrow \mathbb{R}, \quad W(\mathbf{x}_a, \mathbf{x}_b) \equiv W(q(\mathbf{x}_a - \mathbf{x}_b)). \quad (8.10)$$

Using the notation

$$\nabla_a W \equiv \partial_{\mathbf{x}_a} [W(\mathbf{x}_a, \mathbf{x}_b)] \in \mathbb{R}^{1 \times 3} \quad ; \quad \nabla_b W \equiv \partial_{\mathbf{x}_b} [W(\mathbf{x}_a, \mathbf{x}_b)] \in \mathbb{R}^{1 \times 3} \quad (8.11a)$$

$$\nabla_a^T W \equiv (\nabla_a W)^T \in \mathbb{R}^3 \quad ; \quad \nabla_b^T W \equiv (\nabla_b W)^T \in \mathbb{R}^3, \quad (8.11b)$$

results in

$$\nabla_a^T W = W' q_{\mathbf{r}}^T \quad ; \quad \nabla_b^T W = -W' q_{\mathbf{r}}^T. \quad (8.12)$$

Note that, when clear from the context, arguments are omitted from W and/or its derivatives. For example, whenever W' or W'' is present, it is assumed that W is seen as a function of q . Similarly, the notation ∇W implies W a function of $\mathbf{r}$.

The derivation of the necessary Jacobian matrices for the SPH discretized equations relies on the second-order derivatives of $W(\mathbf{x}_a, \mathbf{x}_b)$ with respect to $\mathbf{x}_a$ and $\mathbf{x}_b$. Define

$$\mathbf{J}_W \equiv \partial_{\mathbf{r}} [\nabla_a^T W] \in \mathbb{R}^{3 \times 3}. \quad (8.13)$$

Using the chain rule, Eq. 8.13 is expanded as

$$\mathbf{J}_W = \partial_{\mathbf{r}} [W' q_{\mathbf{r}}^T] = W' \left(\frac{\partial q_{\mathbf{r}}^T}{\partial \mathbf{r}} \right) + q_{\mathbf{r}}^T \left(W'' \frac{\partial q}{\partial \mathbf{r}} \right) = W' q_{\mathbf{r}\mathbf{r}} + W'' q_{\mathbf{r}}^T q_{\mathbf{r}}, \quad (8.14)$$

which implies that

$$\partial_{\mathbf{x}_a} [\nabla_a^T W] = \mathbf{J}_W \in \mathbb{R}^{3 \times 3} \quad (8.15a)$$

$$\partial_{\mathbf{x}_b} [\nabla_a^T W] = -\mathbf{J}_W \in \mathbb{R}^{3 \times 3}. \quad (8.15b)$$

The last necessary ingredient is evaluating the derivatives with respect to $\mathbf{x}_a$ and $\mathbf{x}_b$ of an expression of the form $(\mathbf{r}^T \nabla_a^T W) \dot{\mathbf{r}}$, where $\dot{\mathbf{r}} \in \mathbb{R}^3$. This term appears in the momentum equation in conjunction with the viscosity term. To this end, consider first the problem of finding the Jacobian $\mathbf{y}_{\mathbf{x}}$, where $\mathbf{y} = (\mathbf{a}^T \mathbf{b}) \mathbf{c}$, for $\mathbf{a}, \mathbf{b} \in \mathbb{R}^N$, $\mathbf{c} \in \mathbb{R}^M$, and $\mathbf{x} \in \mathbb{R}^K$. The $M \times K$ desired Jacobian is obtained as:

$$\mathbf{y}_{\mathbf{x}} = \frac{\partial(\mathbf{a}^T \mathbf{b}) \mathbf{c}}{\partial \mathbf{x}} = \mathbf{c} \frac{\partial(\mathbf{a}^T \mathbf{b})}{\partial \mathbf{x}} + (\mathbf{a}^T \mathbf{b}) \mathbf{c}_{\mathbf{x}} = \mathbf{c} (\mathbf{a}^T \mathbf{b}_{\mathbf{x}} + \mathbf{b}^T \mathbf{a}_{\mathbf{x}}) + (\mathbf{a}^T \mathbf{b}) \mathbf{c}_{\mathbf{x}}. \quad (8.16)$$

With this, the partial derivative of the viscosity term with respect to $\mathbf{r}$ can be obtained as

$$\begin{aligned} \partial_{\mathbf{r}} [(\mathbf{r}^T \nabla_a^T W) \dot{\mathbf{r}}] &= \dot{\mathbf{r}} \left(\mathbf{r}^T \frac{\partial \nabla_a^T W}{\partial \mathbf{r}} + (\nabla_a W) \frac{\partial \mathbf{r}}{\partial \mathbf{r}} \right) + (\mathbf{r}^T \nabla_a^T W) \frac{\partial \dot{\mathbf{r}}}{\partial \mathbf{r}} \\ &= \dot{\mathbf{r}} (\mathbf{r}^T \mathbf{J}_W + \nabla_a W), \end{aligned} \quad (8.17)$$

which implies

$$\partial_{\mathbf{x}_a} [(\mathbf{r}^T \nabla_a^T W) \dot{\mathbf{r}}] = \dot{\mathbf{r}} (\mathbf{r}^T \mathbf{J}_W + \nabla_a W) \in \mathbb{R}^{3 \times 3} \quad (8.18a)$$

$$\partial_{\mathbf{x}_b} [(\mathbf{r}^T \nabla_a^T W) \dot{\mathbf{r}}] = -\dot{\mathbf{r}} (\mathbf{r}^T \mathbf{J}_W + \nabla_a W) \in \mathbb{R}^{3 \times 3} \quad (8.18b)$$

8.1.3.2 Example kernel

Consider the particular case

$$q(\mathbf{r}) = \frac{\|\mathbf{r}\|}{h} \quad (8.19)$$

where h defines the kernel's support. Then,

$$h^2 q^2 = \mathbf{r}^T \mathbf{r} \Rightarrow 2h^2 q q_{\mathbf{r}} = 2\mathbf{r}^T \Rightarrow q_{\mathbf{r}} = \frac{1}{h^2 q} \mathbf{r}^T \quad (8.20a)$$

$$q q_{\mathbf{r}}^T = \frac{1}{h^2} \mathbf{r} \Rightarrow q_{\mathbf{r}}^T q_{\mathbf{r}} + q q_{\mathbf{r}\mathbf{r}} = \frac{1}{h^2} \mathbf{I}_3 \Rightarrow q_{\mathbf{r}\mathbf{r}} = \frac{1}{q} \left(\frac{1}{h^2} \mathbf{I}_3 - \frac{1}{h^4 q^2} \mathbf{r} \mathbf{r}^T \right) \quad (8.20b)$$

$$\nabla W = W' q_{\mathbf{r}} = \frac{W'}{h^2 q} \mathbf{r}^T \quad (8.21)$$

Therefore, the $\mathbf{J}_W$ is obtained as

$$\begin{aligned}
 \mathbf{J}_W &= W' q_{\mathbf{r}\mathbf{r}} + W'' q_{\mathbf{r}}^T q_{\mathbf{r}} \\
 &= \frac{W'}{q} \left[\frac{1}{h^2} \mathbf{I}_3 - \frac{1}{h^4 q^2} \mathbf{r} \mathbf{r}^T \right] + W'' \left(\frac{1}{h^2 q} \mathbf{r} \right) \left(\frac{1}{h^2 q} \mathbf{r}^T \right) \\
 &= \frac{W'}{h^2 q} \mathbf{I}_3 + \left(W'' - \frac{W'}{q} \right) \frac{\mathbf{r} \mathbf{r}^T}{h^4 q^2}
 \end{aligned} \tag{8.22}$$

A similar kernel function as that of the explicit approach defined in Eq. 2.4 was used herein. The radius of the kernel function, κh , is proportional to the characteristic length, where $\kappa = 2$ for the kernel function defined by Eq. (2.4).

8.1.3.3 SPH Derivatives

The Jacobian information for implicit integration of the discretized SPH equations can be obtained and assembled by blocks, using the identities derived previously, as follows:

$$\partial_{\mathbf{x}_a}[\boldsymbol{\xi}_a] = \sum_b m_b \left(\frac{p_a}{\rho_a^2} + \frac{p_b}{\rho_b^2} \right) \mathbf{J}_W \tag{8.23a}$$

$$\partial_{\mathbf{x}_b}[\boldsymbol{\xi}_a] = -m_b \left(\frac{p_a}{\rho_a^2} + \frac{p_b}{\rho_b^2} \right) \mathbf{J}_W \tag{8.23b}$$

$$\partial_{\mathbf{v}_a}[\boldsymbol{\xi}_a] = 0 \tag{8.23c}$$

$$\partial_{\mathbf{v}_b}[\boldsymbol{\xi}_a] = 0 \tag{8.23d}$$

$$\partial_{p_a}[\boldsymbol{\xi}_a] = \sum_b m_b \frac{1}{\rho_a^2} \nabla_a^T W \tag{8.23e}$$

$$\partial_{p_b}[\boldsymbol{\xi}_a] = m_b \frac{1}{\rho_b^2} \nabla_a^T W \tag{8.23f}$$

$$\partial_{\mathbf{x}_a}[\zeta_a] = \sum_b m_b \frac{2\bar{\mu}}{\bar{\rho}^2} \frac{1}{\mathbf{r}^T \mathbf{r} + \eta^2} \left[\dot{\mathbf{r}} \mathbf{r}^T \left(\mathbf{J}_W - \frac{2(\mathbf{r}^T \nabla_a^T W)}{\mathbf{r}^T \mathbf{r} + \eta^2} \mathbf{I}_3 \right) + \dot{\mathbf{r}} \nabla_a W \right] \quad (8.24a)$$

$$\partial_{\mathbf{x}_b}[\zeta_a] = -m_b \frac{2\bar{\mu}}{\bar{\rho}^2} \frac{1}{\mathbf{r}^T \mathbf{r} + \eta^2} \left[\dot{\mathbf{r}} \mathbf{r}^T \left(\mathbf{J}_W - \frac{2(\mathbf{r}^T \nabla_a^T W)}{\mathbf{r}^T \mathbf{r} + \eta^2} \mathbf{I}_3 \right) + \dot{\mathbf{r}} \nabla_a W \right] \quad (8.24b)$$

$$\partial_{\mathbf{v}_a}[\zeta_a] = \left[\sum_b m_b \frac{2\bar{\mu}}{\bar{\rho}^2} \frac{1}{\mathbf{r}^T \mathbf{r} + \eta^2} (\mathbf{r}^T \nabla_a^T W) \right] \mathbf{I}_3 \quad (8.24c)$$

$$\partial_{\mathbf{v}_b}[\zeta_a] = -m_b \frac{2\bar{\mu}}{\bar{\rho}^2} \frac{1}{\mathbf{r}^T \mathbf{r} + \eta^2} (\mathbf{r}^T \nabla_a^T W) \mathbf{I}_3 \quad (8.24d)$$

$$\partial_{p_a}[\zeta_a] = 0 \quad (8.24e)$$

$$\partial_{p_b}[\zeta_a] = 0 \quad (8.24f)$$

$$\partial_{\mathbf{x}_a}[\chi_a] = \sum_b \frac{m_b}{\rho_b} \dot{\mathbf{r}}^T \mathbf{J}_W \quad (8.25a)$$

$$\partial_{\mathbf{x}_b}[\chi_a] = -\frac{m_b}{\rho_b} \dot{\mathbf{r}}^T \mathbf{J}_W \quad (8.25b)$$

$$\partial_{\mathbf{v}_a}[\chi_a] = \sum_b \frac{m_b}{\rho_b} \nabla_a W \quad (8.25c)$$

$$\partial_{\mathbf{v}_b}[\chi_a] = -\frac{m_b}{\rho_b} \nabla_a W \quad (8.25d)$$

$$\partial_{p_a}[\chi_a] = 0 \quad (8.25e)$$

$$\partial_{p_b}[\chi_a] = 0 \quad (8.25f)$$

The above blocks have the following dimensions:

$$\begin{aligned} \partial_{\mathbf{x}_a}[\boldsymbol{\xi}_a], \partial_{\mathbf{x}_b}[\boldsymbol{\xi}_a], \partial_{\mathbf{v}_a}[\boldsymbol{\xi}_a], \partial_{\mathbf{v}_b}[\boldsymbol{\xi}_a] &\in \mathbb{R}^{3 \times 3} \\ \partial_{p_a}[\boldsymbol{\xi}_a], \partial_{p_b}[\boldsymbol{\xi}_a] &\in \mathbb{R}^3 \end{aligned}$$

$$\begin{aligned} \partial_{\mathbf{x}_a}[\boldsymbol{\zeta}_a], \partial_{\mathbf{x}_b}[\boldsymbol{\zeta}_a], \partial_{\mathbf{v}_a}[\boldsymbol{\zeta}_a], \partial_{\mathbf{v}_b}[\boldsymbol{\zeta}_a] &\in \mathbb{R}^{3 \times 3} \\ \partial_{p_a}[\boldsymbol{\zeta}_a], \partial_{p_b}[\boldsymbol{\zeta}_a] &\in \mathbb{R}^3 \end{aligned}$$

$$\begin{aligned} \partial_{\mathbf{x}_a}[\chi_a], \partial_{\mathbf{x}_b}[\chi_a], \partial_{\mathbf{v}_a}[\chi_a], \partial_{\mathbf{v}_b}[\chi_a] &\in \mathbb{R}^{1 \times 3} \\ \partial_{p_a}[\chi_a], \partial_{p_b}[\chi_a] &\in \mathbb{R} \end{aligned}$$

8.1.3.4 Implicit integration using BDF

Consider an initial value problem (IVP), denote by t^n and $\mathbf{y}^n = \mathbf{y}(t^n)$ the discrete values of the independent variable t and solution $\mathbf{y}(t)$ at $n\tau$, where $\tau = t^n - t^{n-1}$ be the (constant) step-size. The Backward Differentiation Formulas (BDF) [33] method of order k is obtained by differentiating the solution $\mathbf{y}$ using past values as

$$\mathbf{y}^n = \sum_{m=1}^k \alpha_m \mathbf{y}^{n-m} + \tau \beta_0 \dot{\mathbf{y}}^n, \quad (8.26)$$

where the coefficients α_i and β_0 depend on the method order k . In particular, the simplest BDF method (BDF1) is Backward Euler

$$\mathbf{y}^n = \mathbf{y}^{n-1} + \tau \dot{\mathbf{y}}^n, \quad (8.27)$$

while BDF2 is

$$\mathbf{y}^n = \frac{4}{3} \mathbf{y}^{n-1} - \frac{1}{3} \mathbf{y}^{n-2} + \frac{2}{3} \tau \dot{\mathbf{y}}^n. \quad (8.28)$$

For an ODE in implicit form or a DAE; i.e. $\Psi(t, \mathbf{y}, \dot{\mathbf{y}}) = 0$, the BDF method gives $\mathbf{y}^n$ as the solution of

$$\Psi \left(t^n, \mathbf{y}^n, \frac{\mathbf{y}^n - \sum_{m=1}^k \alpha_m \mathbf{y}^{n-m}}{\tau \beta_0} \right) = \mathbf{0}, \quad (8.29)$$

which, to simplify notation, is rewritten as:

$$\Psi \left(t^n, \mathbf{y}^n, \frac{1}{\tau\beta_0} \mathbf{y}^n + \mathbf{b} \right) = \mathbf{0}. \quad (8.30)$$

Here $\mathbf{b}$ incorporates past values of the solution and remains constant during the solution of the nonlinear system. A Newton iteration at time step n , given as

$$\mathbf{y}^{n,i+1} = \mathbf{y}^{n,i} - \mathbf{G}^{-1} \Psi \left(t^n, \mathbf{y}^{n,i}, \frac{1}{\tau\beta_0} \mathbf{y}^{n,i} + \mathbf{b} \right), \quad (8.31)$$

is used herein, where $\mathbf{y}^{n,0} = \mathbf{y}^{n-1}$ and $\mathbf{G}$ is the iteration matrix

$$\mathbf{G} = \frac{\partial \Psi}{\partial \mathbf{y}} + \frac{1}{\tau\beta_0} \frac{\partial \Psi}{\partial \dot{\mathbf{y}}}, \quad (8.32)$$

where all terms are evaluated at t^n .

In general, the condition number of the iteration matrix for a system with DAE index ν is $O(\tau^{-\nu})$. However, it is often possible to reduce the condition number through appropriate scaling of the DAE system. Herein, the interest is in the systems of the form

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{z}) \quad (8.33a)$$

$$\mathbf{0} = \mathbf{g}(\mathbf{x}). \quad (8.33b)$$

Assuming that the product of Jacobians $\mathbf{g}_x \mathbf{f}_z$ is nonsingular for all t , (8.33) represents a so-called Hessenberg Index-2 DAE. For such systems, the iteration matrix is written as

$$\tau \mathbf{G} = \begin{bmatrix} \frac{1}{\beta_0} \mathbf{I} - \tau \mathbf{f}_x & -\tau \mathbf{f}_z \\ \tau \mathbf{g}_x & \mathbf{0} \end{bmatrix}. \quad (8.34)$$

Solving the unscaled linear system given in Eq. 8.31 introduces roundoff errors proportional to ϵ/τ and ϵ/τ^2 in $\mathbf{x}$ and $\mathbf{z}$, respectively, where ϵ is the unit roundoff, which can be confirmed by inspecting the orders of the bottom row blocks of $(\tau \mathbf{G})^{-1}$. Scaling by $1/\tau$ the bottom rows of $\tau \mathbf{G}$, i.e. those corresponding to the algebraic equations (8.33b), the roundoff errors introduced in $\mathbf{x}$ and $\mathbf{z}$ are reduced to $O(\epsilon)$ and $O(\epsilon/\tau)$, respectively.

The SPH discretized momentum equation and incompressibility conditions in Eq. (8.4), together with

$$\dot{\mathbf{x}}_a = \mathbf{v}_a, \quad a = 1, 2, 3, \dots, N \quad (8.35)$$

can be written as the Hessenberg index-2 DAE

$$\dot{\mathbf{x}} = \mathbf{v} \quad (8.36a)$$

$$\dot{\mathbf{v}} = \mathbf{f}(\mathbf{x}, \mathbf{v}, \mathbf{p}) \quad (8.36b)$$

$$\mathbf{0} = \mathbf{g}(\mathbf{x}, \mathbf{v}), \quad (8.36c)$$

where in the rest of this section as well as Sect. 8.1.3.5, $\mathbf{x}$, $\mathbf{v}$, and $\mathbf{p}$ denote the arrays of positions, velocities, and pressures for all particles, respectively; for example, $\mathbf{x} = [\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \dots, \mathbf{x}_{N_m}]^T$. After rearranging terms, this leads to the following BDF discretization (which includes scaling of the algebraic incompressibility constraints):

$$\mathbf{x}^n - \tau\beta_0\mathbf{v}^n - \sum_{i=1}^k \alpha_i \mathbf{x}^{n-m} = \mathbf{0} \quad (8.37a)$$

$$\mathbf{v}^n - \tau\beta_0\mathbf{f}(\mathbf{x}^n, \mathbf{v}^n, \mathbf{p}^n) - \sum_{i=1}^k \alpha_i \mathbf{v}^{n-m} = \mathbf{0} \quad (8.37b)$$

$$\mathbf{g}(\mathbf{x}^n, \mathbf{v}^n) = \mathbf{0} \quad (8.37c)$$

with a Newton iteration matrix obtained as:

$$\mathbf{G} = \begin{bmatrix} \mathbf{I} & -\tau\beta_0\mathbf{I} & \mathbf{0} \\ -\tau\beta_0\mathbf{f}_{\mathbf{x}} & \mathbf{I} - \tau\beta_0\mathbf{f}_{\mathbf{v}} & -\tau\beta_0\mathbf{f}_{\mathbf{p}} \\ \mathbf{g}_{\mathbf{x}} & \mathbf{g}_{\mathbf{v}} & \mathbf{0} \end{bmatrix}, \quad (8.38)$$

where all terms are evaluated at t^n . Note that the size of the resulting linear systems can be reduced by eliminating $\mathbf{x}^n$ using the first equation in (8.37), which is equivalent to the following Schur complement decomposition:

$$\mathbf{G} = \begin{bmatrix} \mathbf{I} & \mathbf{0} & \mathbf{0} \\ -\tau\beta_0\mathbf{f}_{\mathbf{x}} & \mathbf{I} & \mathbf{0} \\ \mathbf{g}_{\mathbf{x}} & \mathbf{0} & \mathbf{I} \end{bmatrix} \cdot \begin{bmatrix} \mathbf{I} & -\tau\beta_0\mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} - (\tau\beta_0)^2\mathbf{f}_{\mathbf{x}} - \tau\beta_0\mathbf{f}_{\mathbf{v}} & -\tau\beta_0\mathbf{f}_{\mathbf{p}} \\ \mathbf{0} & \tau\beta_0\mathbf{g}_{\mathbf{x}} + \mathbf{g}_{\mathbf{v}} & \mathbf{0} \end{bmatrix}. \quad (8.39)$$

Alternatively, taking into account that the SPH formulation of the Navier-Stokes equations results in a Hessenberg index-2 DAE, a different elimination sequence can be used by solving a nonlinear

system for the accelerations $\dot{\mathbf{v}}^n$ and pressures $\mathbf{p}^n$. Indeed, using

$$\mathbf{v}^n = \tau\beta_0\dot{\mathbf{v}}^n + \sum_{i=1}^k \alpha_i \mathbf{v}^{n-m} \quad (8.40a)$$

$$\mathbf{x}^n = (\tau\beta_0)^2\dot{\mathbf{v}}^n + \tau\beta_0 \sum_{i=1}^k \alpha_i \mathbf{v}^{n-m} + \sum_{i=0}^k \alpha_i \mathbf{x}^{n-m}, \quad (8.40b)$$

the following nonlinear system is obtained upon scaling by $\tau\beta_0$:

$$\dot{\mathbf{v}}^n - \bar{\mathbf{f}}(\dot{\mathbf{v}}^n, \mathbf{p}^n) = \mathbf{0} \quad (8.41a)$$

$$\frac{1}{\tau\beta_0} \bar{\mathbf{g}}(\dot{\mathbf{v}}^n) = \mathbf{0}, \quad (8.41b)$$

where $\bar{\mathbf{f}}(\dot{\mathbf{v}}, \mathbf{p}) \equiv \mathbf{f}(\mathbf{x}(\dot{\mathbf{v}}), \mathbf{v}(\dot{\mathbf{v}}), \mathbf{p})$ and $\bar{\mathbf{g}}(\dot{\mathbf{v}}) \equiv \mathbf{g}(\mathbf{x}(\dot{\mathbf{v}}), \mathbf{v}(\dot{\mathbf{v}}))$. In this case, the Newton iteration matrix is:

$$\bar{\mathbf{G}} = \begin{bmatrix} \mathbf{I} - (\tau\beta_0)^2 \mathbf{f}_{\mathbf{x}} - \tau\beta_0 \mathbf{f}_{\mathbf{v}} & -\tau\beta_0 \mathbf{f}_{\mathbf{p}} \\ \tau\beta_0 \mathbf{g}_{\mathbf{x}} + \mathbf{g}_{\mathbf{v}} & \mathbf{0} \end{bmatrix}, \quad (8.42)$$

which is the same as the Schur complement block in (8.39).

8.1.3.5 Simulation algorithm based on Newmark method

By choosing Newmark method and assuming $\dot{\mathbf{v}}$ and $\mathbf{p}$ as the set of unknowns, markers' position and velocities are obtained as,

$$\mathbf{v}^n = \mathbf{k}_{\mathbf{v}}^{n-1} - \tau\beta_0\dot{\mathbf{v}}^n \quad (8.43a)$$

$$\mathbf{x}^n = \mathbf{k}_{\mathbf{x}}^{n-1} - (\tau\beta_0)^2\dot{\mathbf{v}}^n, \quad (8.43b)$$

where

$$\mathbf{k}_{\mathbf{v}}^{n-1} = \mathbf{v}^{n-1} - \beta_0(1 - \tau)\dot{\mathbf{v}}^{n-1} \quad (8.44a)$$

$$\mathbf{k}_{\mathbf{x}}^{n-1} = \mathbf{x}^{n-1} - \beta_0\mathbf{v}^n + \beta_0^2\tau(1 - \tau)\dot{\mathbf{v}}^{n-1}. \quad (8.44b)$$

Therefore, Eq. (8.31) can be used for the iterative solution of

$$\Psi = \mathbf{0}_{4N_m \times 1}, \quad (8.45)$$

where

$$\Psi \equiv \Psi(\dot{\mathbf{v}}, p) = \begin{bmatrix} \dot{\mathbf{v}} - \mathbf{f}(\mathbf{r}, \mathbf{v}, \mathbf{p}) \\ \frac{1}{\tau\beta_0} \mathbf{g}(\mathbf{x}, \mathbf{v}) \end{bmatrix}. \quad (8.46)$$

Note that $\mathbf{f} = [\mathbf{f}_1, \mathbf{f}_2, \mathbf{f}_3, \dots, \mathbf{f}_{N_m}]^T$ and $\mathbf{g} = [\chi_1, \chi_2, \chi_3, \dots, \chi_{N_m}]^T$ are defined based on Eq. (8.4), where $\mathbf{f}_a = -\boldsymbol{\xi}_a + \boldsymbol{\zeta}_a + \mathbf{k}_a$, and Eqs. (8.23) to (8.25) are used to calculate $\mathbf{f}_x$, $\mathbf{f}_v$, $\mathbf{f}_p$, $\mathbf{g}_x$, and $\mathbf{g}_v$.

8.1.3.6 Parallel implementation

At each iteration, a neighbor list is assembled to indicate the set of markers that fall within the kernel support of a marker. If N_m markers are used in the simulation, N_m lists are generated. The summations appearing on the right hand side of Eqs. (8.5) and (8.23)-(8.25) are subsequently computed based on these neighbors lists. The CUDA library [92] along with several functions provided by Thrust [45] are used for a parallel implementation that scales linearly.

8.1.4 Boundary conditions

The wall boundary condition imposes constraints on velocity and pressure as essential and natural boundary conditions, respectively. These conditions need to be resolved on the boundaries of the solution domain.

8.1.4.1 Velocity boundary conditions

The no-slip boundary condition, i.e. $\mathbf{v} = \mathbf{0}$, imposes essential boundary conditions on velocity. The implementation of essential boundary conditions in SPH needs to resolve the particle deficiency issue near the boundaries, which can be addressed in several ways, e.g. using mirror particles [22, 69], dummy particles with zero [68] or rigid body velocity [104], linear [84] or quadratic [10] extrapolation of the velocity to the dummy particles, and normalization procedure to account for the particle deficiency [109].

8.1.4.2 Pressure boundary conditions

The pressure constraint imposes a natural boundary condition. Since pressure is a scalar, a single condition on each boundary is required for its solution. Claeyssen et al. [19] used the momentum equation,

$$\frac{\partial \mathbf{v}}{\partial t} + (\nabla \mathbf{v}) \mathbf{v} + (\nabla p)^T = \nu \nabla^2 \mathbf{v}, \quad (8.47)$$

to derive the pressure boundary condition. For instance, the Neumann boundary condition can be obtained as

$$\mathbf{n}^T (\nabla p)^T \equiv \frac{\partial p}{\partial n} = \nu \mathbf{n}^T \nabla^2 \mathbf{v} - \left(\mathbf{n}^T \frac{\partial \mathbf{v}}{\partial t} + \mathbf{n}^T (\nabla \mathbf{v}) \mathbf{v} \right), \quad (8.48)$$

where n and $\mathbf{n}$ are the coordinate and unit vector normal to the boundary, respectively. By combining the no-slip boundary condition with Eq. (8.48), the natural boundary condition on the rigid walls simplifies to

$$\frac{\partial p}{\partial n} = \nu \mathbf{n}^T \nabla^2 \mathbf{v}. \quad (8.49)$$

In the case of Dirichlet boundary condition, the values of pressure on the boundary can be used to define the essential boundary condition as

$$\boldsymbol{\varrho}^T (\nabla p)^T \equiv \frac{\partial p}{\partial \boldsymbol{\varrho}} = \nu \boldsymbol{\varrho}^T \nabla^2 \mathbf{v} - \left(\boldsymbol{\varrho}^T \frac{\partial \mathbf{v}}{\partial t} + \boldsymbol{\varrho}^T (\nabla \mathbf{v}) \mathbf{v} \right), \quad (8.50)$$

where $\boldsymbol{\varrho}$ is a vector tangent to the boundary.

Despite the straightforward derivation of pressure boundary condition, Strikwerda [124], Henshaw [43], and Petersson [108] pointed out that Eqs. (8.48) and (8.50) do not add extra information to the system since momentum equation is already satisfied within the entire domain “including the boundaries”, a claim that was disputed due to the fact that momentum equation is not employed for Dirichlet boundary [38]. Nevertheless, the consensus is to implement the incompressibility condition while deriving the required boundary condition. One solution is to directly apply the incompressibility condition,

$$\nabla \cdot \mathbf{v} = 0, \quad (8.51)$$

along with Eq. (8.49) on the boundaries. Petersson [108] pointed out that using Eq. (8.51) in an implicit solution requires a much smaller time step than what is imposed by Eq. (8.49). Therefore

he suggested to use Eq. (8.49) by taking into account the fact that Eq. (8.51) should be satisfied normal to the boundary. As a results, Eq. (8.49) transforms into

$$\frac{\partial p}{\partial n} = -\nu \mathbf{n}^T (\nabla \times \nabla \times \mathbf{v}). \quad (8.52)$$

Different approaches were suggested in [22] and [49] to implement the pressure boundary condition in the SPH framework. In [22], the pressure boundary condition is applied in the weak form of the pressure equation. Equation (8.49) has been implemented using the SPH discretization in [49].

As mentioned earlier, particle deficiency is not an issue for essential boundary conditions since dummy particles can be generated according to boundary values in several different ways [10, 22, 68, 69, 84, 104, 109]. However, properties of the dummy particles should comply with the Neumann boundary condition. Different algorithms were suggested to propagate the pressure field to dummy particles representing the boundary to satisfy the Neumann boundary condition [22, 49, 68]. Cummins et al. [22] set the pressure of the mirror particle to be the same as that of the original particle, i.e. $p_m = p$, where m denote the mirror particle. This enforces $\partial p / \partial n = 0$ which is not essentially correct. Lee et al. [68] suggested copying the value of the particle to the dummy particles in the normal direction. If there is overlap along two different normal directions, for instance convex edges, the values obtained from different directions are averaged. A similar approach is used in [49] to propagate the so-called modified pressure, a property which is also subjected to a natural boundary condition.

8.1.4.3 SPH implementation of boundary conditions

Since an incompressibility constraint is applied in the solution procedure, there is no need for extra effort in terms of applying boundary conditions. However, the SPH implementation of the algorithm needs to resolve the particle deficiency issue close to the boundary. Herein, the coupling method using BCE markers described in Sect. 3.2 [104] is consistently used in both implicit and explicit models. In contrast to the explicit method, the pressure of the BCE markers needs to be

defined as well. To this end, the method described in [68] was used, where the pressure of each BCE marker is set equal to the pressure of the closest fluid marker.

8.1.5 Preliminary results

The developed implicit integration algorithm was investigated through the simulation of transient Poiseuille flow at $Re \simeq 1$. The velocity profile obtained from implicit SPH method using 100 markers is in good agreement with the exact solution, as illustrated in Figure 8.4. The result was obtained using a time step about an order of magnitude larger than the one required by the explicit integration algorithm explained in Sect. 3.4. An average of five iterations is required for convergence of the solution at each time step. A preconditioner, which is currently being developed, is anticipated to further improve the efficiency of the approach.

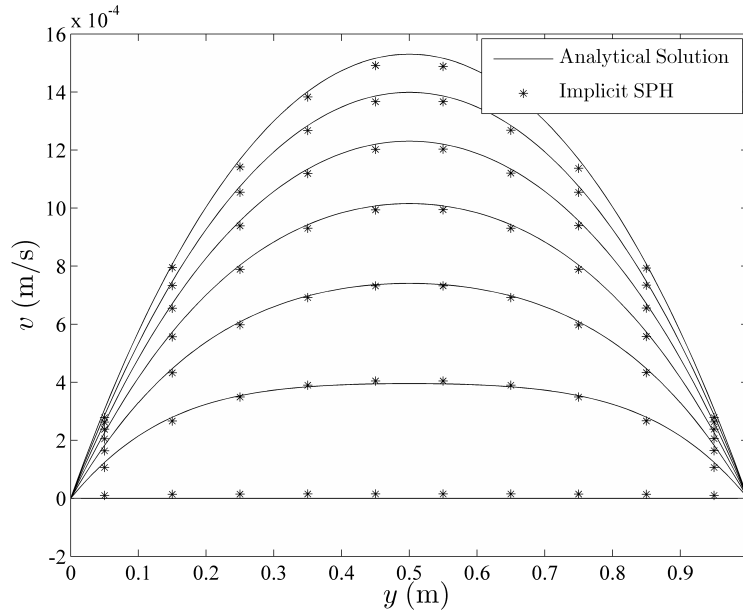


Figure 8.4: Velocity profile of transient Poiseuille flow obtained from implicit SPH simulation (dots) and series solution (continuous lines).

8.2 Using SPH to investigate challenging applications

8.2.1 Effect of particle shape on the behavior of particle suspension and sedimentation

As indicated in Sects. 5.1 and 5.2, experiments and direct numerical investigations have been mostly targeting suspension of spherical particles. The results reported for the migration of single particle in pipe flow demonstrate a close correlation with the distribution of suspension concentration, as suggested by Figures 5.3 and 5.4. Future work can approach the suspension of non-spherical particles to further investigate the migration of single particle of an arbitrary shape as well as suspension distribution, velocity profile, sedimentation, and transient behavior of dense suspensions.

8.2.2 Ice-structure interaction

Offshore wind turbines are being used by several countries to harness the energy of strong, consistent winds that are found over the oceans. In the United States, 53% of the nations population lives in coastal areas, where energy costs and demands are high and land-based renewable energy resources are often limited. Abundant offshore wind resources have the potential to supply immense quantities of renewable energy to major U.S. coastal cities, such as New York City and Boston.

The structural design of offshore wind turbines depends on the conditions imposed locally and can vary according to water depth, as shown in Figure 8.5. In the relatively small wave environment present in most of the Great Lakes, freshwater ice loads have the potential to be the critical design loading case for offshore wind turbines. While offshore wind turbines have been constructed in the North, Baltic and Irish Seas for over 20 years, uncertainty in freshwater ice loads as compared to the saltwater conditions existing for previous developments make the transition of European experience to the Great Lakes uncertain and costly. A project can aim to reduce the uncertainty in costs, as well as realized costs, related to development of offshore wind turbines in the US.

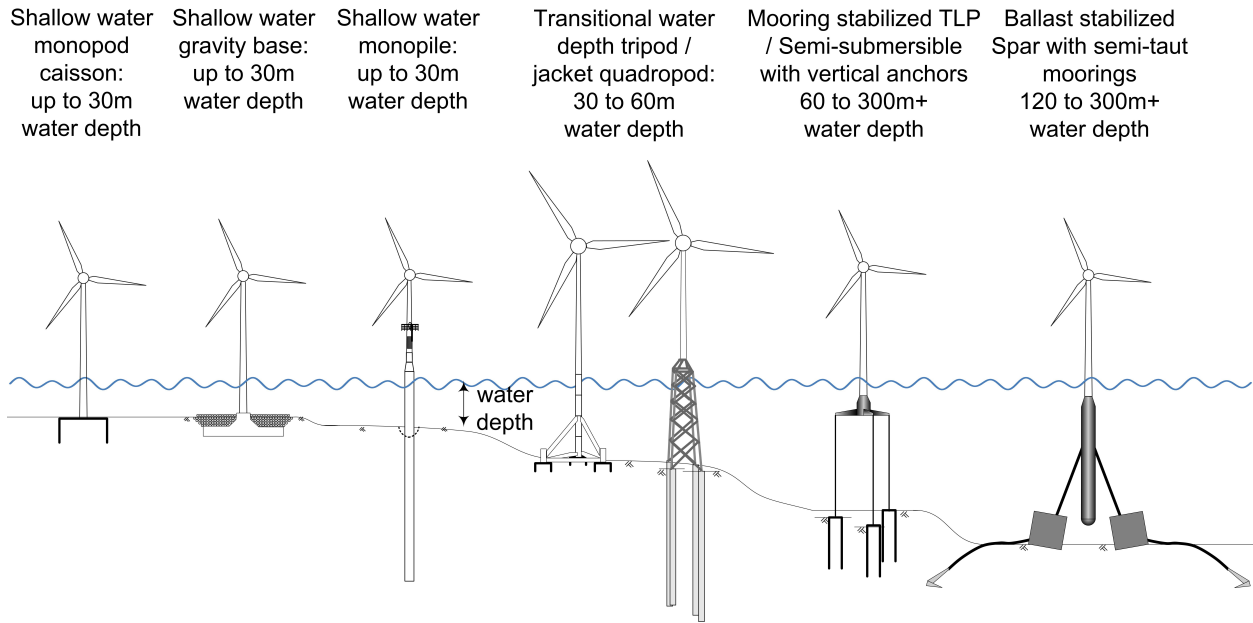


Figure 8.5: Offshore wind turbine designs for shallow and deep water.

Within the Great Lakes, approximately 25% of offshore wind potential occurs for water depths shallower than 30 m, and 60% of offshore wind potential occurs for water depths greater than 60 m [87]. As substructure cost tends to increase with water depth, initial offshore wind turbine developments in the Great Lakes will likely begin in the shallow portions of Lake Erie and western coasts of Lake Michigan, Lake Ontario, and Lake Huron.

Design of ice load-resistant offshore structures is highly uncertain due to rate controlled strength and stiffness, fracture-controlled anisotropic strength, and the wide range of type and internal structures of ice. Successful substructure concepts have been designed for offshore wind turbines to resist ice loads, which often results in fabrication of ice cones in severe environments (e.g., Figure 8.6a). Ice cones are intended to shift the failure mechanisms from that of crushing against a straight shaft (Figure 8.6b) to a flexural failure (Figure 8.6c). Anisotropic ice strength tends to result in lower loads on the turbine tower for the case of flexural failures.

Current design practice in assessing horizontal forces, F , on offshore wind turbines due to impact from moving ice has been applied primarily in the North and Baltic Seas, as outlined in Det Norske Veritas standard DNV-OS-J101 [130]. In this model, an empirical relationship between

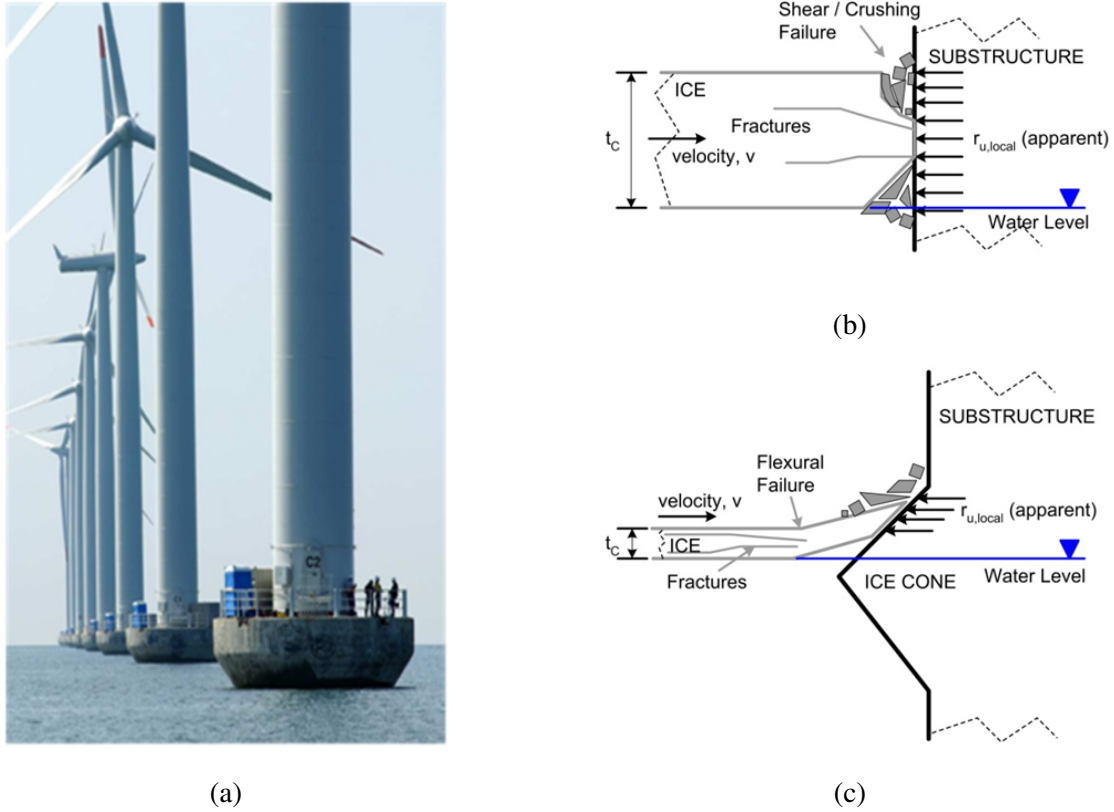


Figure 8.6: Offshore structure design to reduce the ice loading: (a) breaking ice cones installed at offshore substructures (Barker *et al.* [9]); (b) crushing failure mechanism of ice against a vertical face; (c) flexural failure mechanism of ice against an ice cone.

local pressure exerted on a substructure element by ice, $r_{local,C}$, and ice strength, $r_{u,C}$, defined as $\alpha = r_{u,local}/r_{u,C}$, is a function of the ratio of structure diameter, D , to ice thickness, t_c , i.e. $\beta = D/t_c$. The recommended value of α is given by

$$\alpha = \frac{F/(D \cdot t_c)}{r_{u,C}} = \sqrt{1 + 5 \frac{t_c^2}{A_{local}}} = \sqrt{1 + 5 \frac{t_c^2}{t_c \cdot (t_c \cdot D/t_c)}} = \sqrt{1 + \frac{5}{\beta}} \quad (8.53)$$

Equation (8.53) implies that the ratio of local ice pressure to ice strength (α) cannot be less than unity, which is not in agreement with field data for bridge piers and other marine structures, as shown in Figure 8.7. Some of the differences result from ice failing in flexure rather than compression due to the presence of ice cones. However, the major issues highlighted by Figure 8.7 are that (i) most observed α values are less than unity, while the design code specifies values must

be greater than unity; (ii) design ice loads may be overestimated by factors of 2 to 20 for typical tower diameter to ice thickness ratios expected for offshore wind turbines in the Great Lakes; and (iii) there are high levels of uncertainty in design correlations.

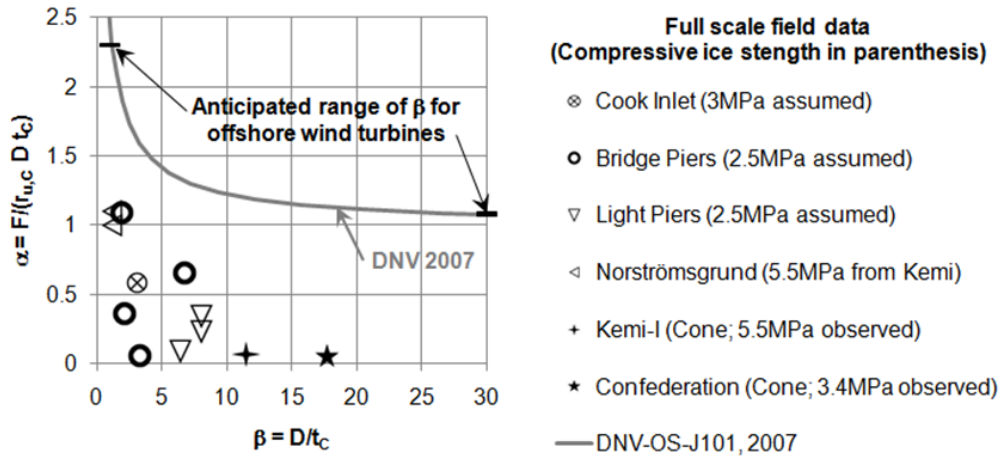


Figure 8.7: Comparison of ice pressure to ice strength ratio versus substructure diameter to ice thickness ratio as recommended by DNV [130] to that obtained from field measurements (field data from Timco *et al.* [127], Kärnä *et al.* [58], Brown & Määttänen [12]).

To reduce uncertainty in ice loads for offshore wind turbine (monopile) substructures, studies have been performed in test tanks with model ice [9, 37]. Data from Barker *et al.* [9] indicate that the ratio of ice pressure to ice strength tended to (i) increase with ice velocity; (ii) be lower for flexural failures on ice cones as compared to crushing failures on straight shafts; and (iii) have relatively high levels of uncertainty (Coefficient of Variation, COV 40% to 60%) when using simple design correlations such as that outlined by Eq. (8.53). It is interesting to note that the ratio α is approximately a factor of 2 to 3 lower for flexural failures on ice cones as compared to crushing failures on straight shafts. The ratio of compressive strength to flexural strength for the model ice used in these tests was also a factor of 2 to 3.5, indicating that local ice pressure is roughly proportional to ice strength for the mode of failure (i.e., crushing or flexure). However, there are still high levels of uncertainties in this empirical relationship (with α varying from 0.05 to 1 in Figures 8.6 and 8.7). A better fundamental understanding of the failure mechanisms controlling ice loads on offshore structures can aid in reducing uncertainty and the high cost of conservatism

illustrated in Figure 8.7. The process of ice impacting the substructure-tower-turbine-foundation system can be better investigated through a dynamic (compared to static) modeling of the entire multi-physics system, which can be approached via fluid-solid interaction scheme.

8.3 Using MPI-enabled high performance computing for faster solution of larger problems

With thousands of scalar processors and giga bytes (GB) of memory, a GPU card provides a very affordable solution to the simulation of complex problems on a personal computer. However, the amount of memory available on a GPU limits the size of the problems that can be tackled by one card. The Message Passing Interface (MPI) provides a scalable solution by relying on a distributed memory architecture in which a collection of nodes are connected, each having its independent processors and memory.

MPI is an agreed-upon syntax and protocol for inter-process communication on parallel computers [39]. An implementation of the MPI standard defines a collection of routines in a library that is used to write message passing parallel programs. Several implementations of the MPI standard exist, including MPICH [86] and OpenMPI [96].

MPI promotes the Single Program, Multiple Data (SPMD) programming paradigm in which the same program is launched independently on numerous cores that execute, as processes, multiple instances of the program. Each process is given a rank, or an identifier, which can be used to differentiate one process from another. The program can be written to do different things depending on the rank of the process.

The most important concept in MPI programming is that of message passing between ranks. The defining attributes of a message are the rank, memory location, data type, and data size for each rank participating in message passing. The simplest type of message passing is called point-to-point communication, when one rank sends a message to a different rank. The send and receive command can be blocking or non-blocking.

The second type of message passing is called collective communication. This type of communication involves a group of ranks, all of which must make the same collective call. For instance,

one data element is broadcast to all ranks, a distributed array is collected in a serial array of a single rank, or one rank performs a reduction operation on distributed data.

MPI will be used to: (1) increase the problem size that can be approached by the FSI simulation engine; (2) accelerate the simulation via leveraging more processing cores. In the MPI-based approach, the simulation domain will be divided into a number of sub-domains. To a great extent, the existing execution model for each sub-domain will stay intact. To accommodate for distributed memory, each sub-domain will be simulated by a separate MPI process, with appropriate communication and synchronization.

Implementing a distributed-memory model using MPI will increase the problem size, restricted currently by GPU hard constraints on the memory size. Nevertheless, GPU programming, i.e. the current parallel execution model, can be leveraged for the computations in each sub-domain, in a so-called heterogeneous computing template. It is worth mentioning that heterogeneous computing is a state-of-the-art computational model; therefore the extent to which the efficiency gain will be compromised by host-GPU data transfer remains to be determined.

Chapter 9

Conclusion

A Lagrangian-Lagrangian (LL) approach for the direct numerical simulation of fluid-solid interaction problems was proposed herein. The simulation framework consists of components to model fluid flow, 3D rigid bodies, and flexible components, where the phases can interact mutually in a fully coupled fashion. While the coupling of the fluid and solid objects is maintained through Lagrangian markers attached to the solid phase, the solid-solid interaction is captured by either a lubrication force model developed for SPH, or a penalty approach.

The LL formalism was employed to study, through direct numerical simulation, the 3D pipe flow of suspensions of macroscopic neutrally buoyant rigid bodies of arbitrary geometry as well as suspended cantilever and free flexible beams. Simulation results obtained for transient Poiseuille flow, migration of cylinders in plane Poiseuille flow, migration of spherical bodies in pipe flow, suspension distribution, blunting of the velocity profile in flow of suspension, and effect of Reynolds number on radial migration are in agreement with previously published experimental and numerical data. Several tests were performed to study different attributes of suspension flows for which results are not available in the literature. Moreover, simulation results provided for free-floating flexible beams in Poiseuille flow, channel flow over a grid of flexible cantilevers, and parametric studies of the effect of fluid viscosity and material elasticity suggest that the adopted approach has good predictive capabilities and is able to capture the dynamics of the systems under consideration.

An HPC approach to the simulation methodology was described and shown to scale linearly in the dynamics of each phase independently as well as in the overall FSI solution implementation. It was shown that the dynamics of the fluid phase, i.e. the time propagation of the SPH markers,

controls the overall simulation time, and neither cross-phase communications nor solid phase simulation have a notable effect on the simulation time. This represents one advantage of the unified FSI methodology over co-simulation and asynchronous approaches. The GPU-based implementation of the approach, called Chrono::Fluid, is available as an open-source software at [17].

Further capabilities of the simulation framework were demonstrated through applications including flow cytometry, dense particle suspension, array of flexible objects immersed in fluid, and a mixture of rigid and flexible objects in fluid flow. Additional examples of Chrono::Fluid simulations can be found at [112].

To enhance the simulation efficiency, an implicit-constrained SPH algorithm was proposed and formulated herein. Further implementation of the approach, validation, and performance analysis will be pursued in the future. Other directions of future work include (a) revisiting the SPH marker numbering scheme in order to improve the spatial and temporal memory access patterns, thus increasing the effective bandwidth of Chrono::Fluid; (b) augmenting the current implementation for use on cluster supercomputers that adopt a distributed memory access model and rely on the Message Passing Interface standard; (c) investigating the rheology and concentration of suspensions of flexible as well as non-spherical objects; (d) approaching problems at macroscale such as ice-structure interaction of offshore structures; and (e) gauge the potential of Chrono::Fluid in biomechanics applications such as blood flow in deformable arteries or heart, channel occlusion in stroke, etc.

LIST OF REFERENCES

- [1] Amini, Y., Emdad, H., Farid, M.: A new model to solve fluid–hypo-elastic solid interaction using the smoothed particle hydrodynamics (SPH) method. *European Journal of Mechanics-B/Fluids* **30**(2), 184–194 (2011)
- [2] Andrews, M., O’rourke, P.: The multiphase particle-in-cell (MP-PIC) method for dense particulate flows. *International Journal of Multiphase Flow* **22**(2), 379–402 (1996)
- [3] Anitescu, M., Hart, G.D.: A constraint-stabilized time-stepping approach for rigid multi-body dynamics with joints, contact and friction. *International Journal for Numerical Methods in Engineering* **60**(14), 2335–2371 (2004)
- [4] Antoci, C., Gallati, M., Sibilla, S.: Numerical simulation of fluid–structure interaction by SPH. *Computers & Structures* **85**(11), 879–890 (2007)
- [5] Applegate, R.W., Marr, D.W., Squier, J., Graves, S.W.: Particle size limits of optical trapping and deflection for sorting using diode laser bars. In: *Conference on Lasers and Electro-Optics*, p. CTuJJ7. Optical Society of America (2010)
- [6] Ardekani, A., Dabiri, S., Rangel, R.: Collision of multi-particle and general shape objects in a viscous fluid. *Journal of Computational Physics* **227**(24), 10,094–10,107 (2008)
- [7] Asmolov, E.S.: The inertial lift on a spherical particle in a plane Poiseuille flow at large channel Reynolds number. *Journal of Fluid Mechanics* **381**, 63–87 (1999)
- [8] Atkinson, K.: *An introduction to numerical analysis*. John Wiley and Sons, USA (1989)
- [9] Barker, A., Timco, G., Gravesen, H., Vølund, P.: Ice loading on Danish wind turbines: part 1: dynamic model tests. *Cold Regions Science and Technology* **41**(1), 1–23 (2005)
- [10] Basa, M., Quinlan, N., Lastiwka, M.: Robustness and accuracy of SPH formulations for viscous flow. *International Journal for Numerical Methods in Fluids* **60**(10), 1127–1148 (2009)

- [11] Benz, W.: Smoothed particle hydrodynamics: A review. In: Proceedings of the NATO Advanced Research Workshop on The Numerical Modelling of Nonlinear Stellar Pulsations Problems and Prospects, Les Arcs, France, March 20-24. Kluwer Academic Publishers (1986)
- [12] Brown, T.G., Mänttinen, M.: Comparison of Kemi-I and Confederation Bridge cone ice load measurement results. *Cold Regions Science and Technology* **55**(1), 3–13 (2009)
- [13] Buschke, D., Resto, P., Schumacher, N., Cox, B., Tallavajhula, A., Vivekanandan, A., Eliceiri, K., Williams, J., Ogle, B.: Microfluidic sorting of microtissues. *Biomicrofluidics* **6**(1), 014 116 (2012)
- [14] Chen, A.A., Underhill, G.H., Bhatia, S.N.: Multiplexed, high-throughput analysis of 3D microtissue suspensions. *Integrative Biology* **2**(10), 517–527 (2010)
- [15] Chen, S., Pan, T., Chang, C.: The motion of a single and multiple neutrally buoyant elliptical cylinders in plane Poiseuille flow. *Physics of Fluids* **24**, 103 302 (2012)
- [16] Choi, H.G., Joseph, D.D.: Fluidization by lift of 300 circular particles in plane Poiseuille flow by direct numerical simulation. *Journal of Fluid Mechanics* **438**(1), 101–128 (2001)
- [17] Chrono::Fluid: An Open Source Engine for Fluid-Solid Interaction (2014). <http://armanpazouki.github.io/chrono-fluid/>
- [18] Chun, B., Ladd, A.: Inertial migration of neutrally buoyant particles in a square duct: An investigation of multiple equilibrium positions. *Physics of Fluids* **18**, 031 704 (2006)
- [19] Claeyssen, J.R., Platte, R.B., Bravo, E.: Simulation in primitive variables of incompressible flow with pressure Neumann condition. *International Journal for Numerical Methods in Fluids* **30**(8), 1009–1026 (1999)
- [20] Cleary, P.W.: Modelling confined multi-material heat and mass flows using SPH. *Applied Mathematical Modelling* **22**(12), 981–993 (1998)
- [21] Colagrossi, A., Landrini, M.: Numerical simulation of interfacial flows by smoothed particle hydrodynamics. *Journal of Computational Physics* **191**(2), 448–475 (2003)
- [22] Cummins, S., Rudman, M.: An SPH projection method. *Journal of Computational Physics* **152**(2), 584–607 (1999)
- [23] Cundall, P., Strack, O.: Discrete numerical-model for granular assemblies. *Geotechnique* **29**(1), 47–65 (1979)
- [24] Davis, R.H., Serayssol, J.M., Hinch, E.: Elastohydrodynamic collision of two spheres. *Journal of Fluid Mechanics* **163**, 479–97 (1986)
- [25] Di Carlo, D.: Inertial microfluidics. *Lab on a Chip* **9**(21), 3038–3046 (2009)

- [26] Di Carlo, D., Irimia, D., Tompkins, R.G., Toner, M.: Continuous inertial focusing, ordering, and separation of particles in microchannels. *Proceedings of the National Academy of Sciences* **104**(48), 18,892–18,897 (2007)
- [27] Dilts, G.: Moving-least-squares-particle hydrodynamics–I. consistency and stability. *International Journal for Numerical Methods in Engineering* **44**(8), 1115–1155 (1999)
- [28] Ding, E.J., Aidun, C.K.: Extension of the lattice-Boltzmann method for direct simulation of suspended particles near contact. *Journal of Statistical Physics* **112**(3-4), 685–708 (2003)
- [29] Efron, B., Tibshirani, R.: *An introduction to the bootstrap*. Chapman and Hall/CRC press (1993)
- [30] Fan, L.S., Zhu, C.: *Principles of gas-solid flows*. Cambridge University Press (2005)
- [31] Feng, J., Hu, H., Joseph, D.: 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)
- [32] Feng, Z., Michaelides, E.: The immersed boundary-lattice Boltzmann method for solving fluid–particles interaction problems. *Journal of Computational Physics* **195**(2), 602–628 (2004)
- [33] Gear, C.W.: *Numerical initial value problems of ordinary differential equations*. Prentice-Hall, Englewood Cliffs, NJ (1971)
- [34] Gerstmayr, J., Shabana, A.: Analysis of thin beams and cables using the absolute nodal co-ordinate formulation. *Nonlinear Dynamics* **45**(1), 109–130 (2006)
- [35] Gidaspow, D.: *Multiphase flow and fluidization: continuum and kinetic theory descriptions*. Academic Press, Boston (1994)
- [36] Glowinski, R., Pan, T., Hesla, T., Joseph, D.: A distributed Lagrange multiplier/fictitious domain method for particulate flows. *International Journal of Multiphase Flow* **25**(5), 755–794 (1999)
- [37] Gravesen, H., Sørensen, S.L., Vølund, P., Barker, A., Timco, G.: Ice loading on Danish wind turbines: Part 2. Analyses of dynamic model test results. *Cold Regions Science and Technology* **41**(1), 25–47 (2005)
- [38] Gresho, P.M., Sani, R.L.: On pressure boundary conditions for the incompressible Navier-Stokes equations. *International Journal for Numerical Methods in Fluids* **7**(10), 1111–1145 (1987)
- [39] Gropp, W., Lusk, E., Skjellum, A.: *Using MPI: portable parallel programming with the message-passing interface*, vol. 1. MIT press (1999)

- [40] Hampton, R., Mammoli, A., Graham, A., Tetlow, N., Altobelli, S.: Migration of particles undergoing pressure-driven flow in a circular conduit. *Journal of Rheology* **41**, 621–640 (1997)
- [41] Han, M., Kim, C., Kim, M., Lee, S.: Particle migration in tube flow of suspensions. *Journal of Rheology* **43**, 1157–1174 (1999)
- [42] Haug, E.: *Computer aided kinematics and dynamics of mechanical systems*. Allyn and Bacon Boston (1989)
- [43] Henshaw, W.D.: A fourth-order accurate method for the incompressible Navier-Stokes equations on overlapping grids. *Journal of Computational Physics* **113**(1), 13–25 (1994)
- [44] Ho, B., Leal, L.: Inertial migration of rigid spheres in two-dimensional unidirectional flows. *Journal of Fluid Mechanics* **65**(02), 365–400 (1974)
- [45] Hoberock, J., Bell, N.: Thrust: C++ template library for CUDA. <http://thrust.github.com/>
- [46] Hocking, L.: The effect of slip on the motion of a sphere close to a wall and of two adjacent spheres. *Journal of Engineering Mathematics* **7**(3), 207–221 (1973)
- [47] Hogg, A.J.: The inertial migration of non-neutrally buoyant spherical particles in two-dimensional shear flows. *Journal of Fluid Mechanics* **272**, 285–318 (1994)
- [48] Holmes, D., Sandison, M., Green, N.G., Morgan, H.: On-chip high-speed sorting of micron-sized particles for high-throughput analysis. In: *Proceedings of IEEE Nanobiotechnology Conference*, vol. 152, pp. 129–135. IET (2005)
- [49] Hosseini, S.M., Feng, J.J.: Pressure boundary conditions for computing incompressible flows with SPH. *Journal of Computational Physics* **230**(19), 7473–7487 (2011)
- [50] Hu, H., Patankar, N., Zhu, M.: Direct numerical simulations of fluid-solid systems using the arbitrary Lagrangian-Eulerian technique. *Journal of Computational Physics* **169**(2), 427–462 (2001)
- [51] Hu, H.H.: Direct simulation of flows of solid-liquid mixtures. *International Journal of Multiphase Flow* **22**(2), 335–352 (1996)
- [52] Hu, W., Tian, Q., Hu, H.: Dynamic simulation of liquid-filled flexible multibody systems via absolute nodal coordinate formulation and SPH method. *Nonlinear Dynamics* pp. 1–19 (2013)
- [53] Huh, D., Bahng, J.H., Ling, Y., Wei, H.H., Kripfgans, O.D., Fowlkes, J.B., Grotberg, J.B., Takayama, S.: Gravity-driven microfluidic particle sorting device with hydrodynamic separation amplification. *Analytical Chemistry* **79**(4), 1369–1376 (2007)

- [54] Inamuro, T., Maeba, K., Ogino, F.: Flow between parallel walls containing the lines of neutrally buoyant circular cylinders. *International Journal of Multiphase Flow* **26**(12), 1981–2004 (2000)
- [55] Jeffrey, R.C., Pearson, J.: Particle motion in laminar vertical tube flow. *Journal of Fluid Mechanics* **22**(4), 721–735 (1965)
- [56] Johansson, L., Nikolajeff, F., Johansson, S., Thorslund, S.: On-chip fluorescence-activated cell sorting by an integrated miniaturized ultrasonic transducer. *Analytical Chemistry* **81**(13), 5188–5196 (2009)
- [57] Joseph, D., Ocando, D., Huang, P.: Slip velocity and lift. *Journal of Fluid Mechanics* **454**, 263–286 (2002)
- [58] Kärnä, T., Qu, Y., Yue, Q.: An extreme value analysis of local ice pressures. In: *Proceedings of the International Conference Exhibition on Performance of Ships and Structures in Ice. ICETECH-06*, pp. 16–19 (2006)
- [59] Karnis, A., Goldsmith, H., Mason, S.: The flow of suspensions through tubes: V. Inertial effects. *The Canadian Journal of Chemical Engineering* **44**(4), 181–193 (1966)
- [60] Kodam, M., Bharadwaj, R., Curtis, J., Hancock, B., Wassgren, C.: Force model considerations for glued-sphere discrete element method simulations. *Chemical Engineering Science* **64**(15), 3466–3475 (2009)
- [61] Koh, C.J., Hookham, P., Leal, L.: An experimental investigation of concentrated suspension flows in a rectangular channel. *Journal of Fluid Mechanics* **266**(1), 1–32 (1994)
- [62] Kruggel-Emden, H., Simsek, E., Rickelt, S., Wirtz, S., Scherer, V.: Review and extension of normal force models for the discrete element method. *Powder Technology* **171**(3), 157–173 (2007)
- [63] Kruggel-Emden, H., Wirtz, S., Scherer, V.: A study on tangential force laws applicable to the discrete element method (DEM) for materials with viscoelastic or plastic behavior. *Chemical Engineering Science* **63**(6), 1523–1541 (2008)
- [64] Ladd, A.J.: Numerical simulations of particulate suspensions via a discretized Boltzmann equation. *Journal of Fluid Mechanics* **271**(1), 285–339 (1994)
- [65] Ladd, A.J.: Sedimentation of homogeneous suspensions of non-Brownian spheres. *Physics of Fluids* **9**, 491–499 (1997)
- [66] Lastiwka, M., Basa, M., Quinlan, N.J.: Permeable and non-reflecting boundary conditions in SPH. *International Journal for Numerical Methods in Fluids* **61**(7), 709–724 (2009)
- [67] Lee, C.J.K., Noguchi, H., Koshizuka, S.: Fluid–shell structure interaction analysis by coupled particle and finite element method. *Computers & Structures* **85**(11), 688–697 (2007)

- [68] Lee, E., Moulinec, C., Xu, R., Violeau, D., Laurence, D., Stansby, P.: Comparisons of weakly compressible and truly incompressible algorithms for the SPH mesh free particle method. *Journal of Computational Physics* **227**(18), 8417–8436 (2008)
- [69] Libersky, L., Petschek, A., Carney, T., Hipp, J., Allahdadi, F.: High strain Lagrangian hydrodynamics: A three-dimensional SPH code for dynamic material response. *Journal of Computational Physics* **109**(1), 67–75 (1993)
- [70] Lillehoj, P.B., Tsutsui, H., Valamehr, B., Wu, H., Ho, C.M.: Continuous sorting of heterogeneous-sized embryoid bodies. *Lab on a Chip* **10**(13), 1678–1682 (2010)
- [71] Liu, M.B., Liu, G.R.: Smoothed particle hydrodynamics (SPH): An overview and recent developments. *Archives of Computational Methods in Engineering* **17**(1), 25–76 (2010)
- [72] Liu, W.K., Jun, S., Li, S., Adee, J., Belytschko, T.: Reproducing kernel particle methods for structural dynamics. *International Journal for Numerical Methods in Engineering* **38**(10), 1655–1679 (1995)
- [73] Lucy, L.B.: A numerical approach to the testing of the fission hypothesis. *The Astronomical Journal* **82**, 1013–1024 (1977)
- [74] Matas, J.P., Morris, J.F., Guazzelli, E.: Inertial migration of rigid spherical particles in Poiseuille flow. *Journal of Fluid Mechanics* **515**(1), 171–195 (2004)
- [75] Matas, J.P., Morris, J.F., Guazzelli, E.: Lateral force on a rigid sphere in large-inertia laminar pipe flow. *Journal of Fluid Mechanics* **621**, 59–67 (2009)
- [76] Mazhar, H., Heyn, T., Negrut, D.: A scalable parallel method for large collision detection problems. *Multibody System Dynamics* **26**(1), 37–55 (2011)
- [77] McLaughlin, J.: Numerical computation of particles-turbulence interaction. *International Journal of Multiphase Flow* **20**, 211–232 (1994)
- [78] Micikevicius, P.: GPU performance analysis and optimization (2014). <http://on-demand.gputechconf.com/gtc/2012/presentations/S0514-GTC2012-GPU-Performance-Analysis.pdf>
- [79] Monaghan, J.J.: An introduction to SPH. *Computer Physics Communications* **48**(1), 89–96 (1988)
- [80] Monaghan, J.J.: On the problem of penetration in particle methods. *Journal of Computational Physics* **82**(1), 1–15 (1989)
- [81] Monaghan, J.J.: Smoothed particle hydrodynamics. *Annual Review of Astronomy and Astrophysics* **30**, 543–574 (1992)

- [82] Monaghan, J.J.: Smoothed particle hydrodynamics. *Reports on Progress in Physics* **68**(1), 1703–1759 (2005)
- [83] Monaghan, J.J., Gingold, R.: Shock simulation by the particle method SPH. *Journal of Computational Physics* **52**(2), 374–389 (1983)
- [84] Morris, J., Fox, P., Zhu, Y.: Modeling low Reynolds number incompressible flows using SPH. *Journal of Computational Physics* **136**(1), 214–226 (1997)
- [85] Morris, J.P.: Simulating surface tension with smoothed particle hydrodynamics. *International Journal for Numerical Methods in Fluids* **33**(3), 333–353 (2000)
- [86] MPICH2: High Performance Portable MPI (2013). <http://http://www.mpich.org/>
- [87] Musial, W., Ram, B.: Large-scale offshore wind power in the United States: Assessment of opportunities and barriers. Tech. rep., National Renewable Energy Laboratory (NREL), Golden, CO. (2010)
- [88] Negrut, D.: GPU memory space (2013). <http://sbel.wisc.edu/Courses/ME964/2013/Lectures/lecture0930.pdf>
- [89] Negrut, D., Tasora, A., Mazhar, H., Heyn, T., Hahn, P.: Leveraging parallel computing in multibody dynamics. *Multibody System Dynamics* **27**(1), 95–117 (2012)
- [90] Newmark, N.M.: A method of computation for structural dynamics. *Journal of the Engineering Mechanics Division, ASCE* pp. 67–94 (1959)
- [91] Nieuwstadt, H.A., Seda, R., Li, D.S., Fowlkes, J.B., Bull, J.L.: Microfluidic particle sorting utilizing inertial lift force. *Biomedical microdevices* **13**(1), 97–105 (2011)
- [92] NVIDIA: CUDA developer zone (2014). <https://developer.nvidia.com/cuda-downloads>
- [93] NVIDIA: NVIDIA GeForce GTX 680 (2014). http://www.geforce.com/Active/en_US/en_US/pdf/GeForce-GTX-680-Whitepaper-FINAL.pdf
- [94] Oger, G., Doring, M., Alessandrini, B., Ferrant, P.: An improved SPH method: Towards higher order convergence. *Journal of Computational Physics* **225**(2), 1472–1492 (2007)
- [95] Oliver, D.: Influence of particle rotation on radial migration in the Poiseuille flow of suspensions. *Nature* **194**, 1269–1271 (1962)
- [96] Open MPI: A High Performance Message Passing Library (2013). <http://www.openmpi.org/>
- [97] Pan, T., Chang, C., Glowinski, R.: On the motion of a neutrally buoyant ellipsoid in a three-dimensional Poiseuille flow. *Computer Methods in Applied Mechanics and Engineering* **197**(25), 2198–2209 (2008)

- [98] Pan, T., Glowinski, R.: Direct simulation of the motion of neutrally buoyant circular cylinders in plane Poiseuille flow. *Journal of Computational Physics* **181**(1), 260–279 (2002)
- [99] Pan, T.W.: Numerical simulation of the motion of neutrally buoyant particles in the plane Poiseuille flow of a Newtonian fluid. *Comptes Rendus de l'Académie des Sciences-Series IIB-Mechanics* **329**(6), 435–438 (2001)
- [100] Pan, T.W., Glowinski, R.: Direct simulation of the motion of neutrally buoyant balls in a three-dimensional poiseuille flow. *Comptes Rendus Mécanique* **333**(12), 884–895 (2005)
- [101] Patankar, N., Huang, P., Ko, T., Joseph, D.: Lift-off of a single particle in Newtonian and viscoelastic fluids by direct numerical simulation. *Journal of Fluid Mechanics* **438**, 67–100 (2001)
- [102] Patankar, N., Ko, T., Choi, H., Joseph, D.: A correlation for the lift-off of many particles in plane Poiseuille flows of Newtonian fluids. *Journal of Fluid Mechanics* **445**, 55–76 (2001)
- [103] Pazouki, A., Mazhar, H., Negrut, D.: Parallel collision detection of ellipsoids with applications in large scale multibody dynamics. *Mathematics and Computers in Simulation* **82**(5), 879–894 (2012)
- [104] Pazouki, A., Negrut, D.: Direct simulation of lateral migration of buoyant particles in channel flow using GPU computing. In: *Proceedings of the 32nd Computers and Information in Engineering Conference, CIE32, August 12-15, Chicago, IL, USA. American Society of Mechanical Engineers* (2012)
- [105] Pazouki, A., Negrut, D.: A numerical study of the effect of particle properties on the radial distribution of suspensions in pipe flow. *Submitted to International Journal of Multiphase Flow* (2014)
- [106] Pazouki, A., Serban, R., Negrut, D.: A high performance computing approach to the simulation of fluid-solid interaction problems with rigid and flexible components. *Archive of Mechanical Engineering* (accepted) (2014)
- [107] Pazouki, A., Serban, R., Negrut, D.: A Lagrangian-Lagrangian framework for the simulation of rigid and deformable bodies in fluid. *Multibody Dynamics: Computational Methods and Applications* (accepted) (2014)
- [108] Petersson, N.A.: Stability of pressure boundary conditions for Stokes and Navier-Stokes equations. *Journal of Computational Physics* **172**(1), 40–70 (2001)
- [109] Randles, P., Libersky, L.: Smoothed particle hydrodynamics: some recent improvements and applications. *Computer Methods in Applied Mechanics and Engineering* **139**(1), 375–408 (1996)

- [110] Rapaport, D.: Radial and axial segregation of granular matter in a rotating cylinder: A simulation study. *Physical Review E* **75**(3), 031 301 (2007)
- [111] Saffman, P.: The lift on a small sphere in a slow shear flow. *Journal of Fluid Mechanics* **22**(02), 385–400 (1965)
- [112] SBEL: Simulation Based Engineering Laboratory, University of Wisconsin-Madison (2014). <http://sbel.wisc.edu>
- [113] Schonberg, J., Hinch, E.: Inertial migration of a sphere in Poiseuille flow. *Journal of Fluid Mechanics* **203**(1), 517–524 (1989)
- [114] Schörgenhumer, M., Gruber, P.G., Gerstmayr, J.: Interaction of flexible multibody systems with fluids analyzed by means of smoothed particle hydrodynamics. *Multibody System Dynamics* pp. 1–24 (2013)
- [115] Schwab, A., Meijaard, J.: Comparison of three-dimensional flexible beam elements for dynamic analysis: finite element method and absolute nodal coordinate formulation. In: *Proceedings of the ASME 2005 IDETC/CIE*, Orlando, FL, USA. American Society of Mechanical Engineers (2005)
- [116] Segre, G., Silberberg, A.: Radial particle displacements in Poiseuille flow of suspensions. *Nature* **189**, 209–210 (1961)
- [117] Segre, G., Silberberg, A.: Behaviour of macroscopic rigid spheres in Poiseuille flow Part 1. determination of local concentration by statistical analysis of particle passages through crossed light beams. *Journal of Fluid Mechanics* **14**(01), 115–135 (1962)
- [118] Shabana, A.: Flexible multibody dynamics: Review of past and recent developments. *Multibody System Dynamics* **1**, 339–348 (1997)
- [119] Shabana, A.: *Dynamics of Multibody Systems*. Cambridge University Press, New York (2005)
- [120] Shao, X., Yu, Z., Sun, B.: Inertial migration of spherical particles in circular poiseuille flow at moderately high reynolds numbers. *Physics of Fluids* **20**, 103 307 (2008)
- [121] Silbert, L., Ert, D., Grest, G., Halsey, T., Levine, D., Plimpton, S.: Granular flow down an inclined plane: Bagnold scaling and rheology. *Physical Review E* **64**(5), 51 302 (2001)
- [122] Singh, P., Hesla, T., Joseph, D.: Distributed Lagrange multiplier method for particulate flows with collisions. *International Journal of Multiphase Flow* **29**(3), 495–509 (2003)
- [123] Sinton, S.W., Chow, A.W.: NMR flow imaging of fluids and solid suspensions in poiseuille flow. *Journal of Rheology* **35**, 735–772 (1991)

- [124] Strikwerda, J.C.: Finite difference methods for the Stokes and Navier-Stokes equations. *SIAM Journal on Scientific and Statistical Computing* **5**(1), 56–68 (1984)
- [125] Swaminathan, T., Mukundakrishnan, K., Hu, H.: Sedimentation of an ellipsoid inside an infinitely long tube at low and intermediate Reynolds numbers. *Journal of Fluid Mechanics* **551**, 357–386 (2006)
- [126] Tartakovsky, A., Meakin, P.: Modeling of surface tension and contact angles with smoothed particle hydrodynamics. *Physical Review E* **72**(2), 026 301 (2005)
- [127] Timco, G.W., Johnston, M., Frederking, R.: The NRC ice load catalogue. In: *Proceedings 15th International Conference on Port and Ocean Engineering under Arctic Conditions, POAC'99*, vol. 1, pp. 444–453 (1999)
- [128] Union Biometrica: Large Particle Flow Cytometry (2014). <http://www.unionbio.com/>
- [129] Vasseur, P., Cox, R.: The lateral migration of a spherical particle in two-dimensional shear flows. *Journal of Fluid Mechanics* **78**(2), 385–413 (1976)
- [130] Veritas, D.D.N.: DNV-OS-J101 Offshore Standard. Design of offshore wind turbine structures (2010)
- [131] Yang, B., Wang, J., Joseph, D., Hu, H.H., Pan, T.W., Glowinski, R.: Migration of a sphere in tube flow. *Journal of Fluid Mechanics* **540**, 109–131 (2005)
- [132] Yu, Z., Shao, X.: A direct-forcing fictitious domain method for particulate flows. *Journal of Computational Physics* **227**(1), 292–314 (2007)
- [133] Zhang, D., Prosperetti, A.: Averaged equations for inviscid disperse two-phase flow. *Journal of Fluid Mechanics* **267**, 185–220 (1994)

APPENDIX

Partial derivatives on staggered grid

Using the staggered grid notation described in Sect. 8.1.1, the discrete continuity equation at location i, j is written as (see Figure 8.1b)

$$\frac{u_{i+1,j}^{n+1} - u_{i,j}^{n+1}}{dx} + \frac{v_{i+1,j}^{n+1} - v_{i,j}^{n+1}}{dy} = 0, \quad (\text{A.1})$$

where dx and dy are discretization lengths in x and y directions, respectively. Similarly, the components of the momentum equations in x direction are written as

$$\frac{\partial p}{\partial x} = \frac{p_{i,j} - p_{i-1,j}}{dx} \quad (\text{A.2a})$$

$$\begin{aligned} \frac{\partial^2 u}{\partial x^2} &= \frac{\left. \frac{\partial u}{\partial x} \right|_{i+1/2,j} - \left. \frac{\partial u}{\partial x} \right|_{i-1/2,j}}{dx} \\ &= \frac{u_{i+1,j} - 2u_{i,j} + u_{i-1,j}}{dx^2} \end{aligned} \quad (\text{A.2b})$$

$$\begin{aligned} \frac{\partial^2 u}{\partial y^2} &= \frac{\left. \frac{\partial u}{\partial y} \right|_{i,j+1/2} - \left. \frac{\partial u}{\partial y} \right|_{i,j-1/2}}{dy} \\ &= \frac{u_{i,j+1} - 2u_{i,j} + u_{i,j-1}}{dy^2} \end{aligned} \quad (\text{A.2c})$$

$$\frac{\partial uu}{\partial x} = \frac{u_{i+1/2,j}^2 - u_{i-1/2,j}^2}{dx} \quad (\text{A.2d})$$

$$\frac{\partial uv}{\partial y} = \frac{u_{i,j+1/2}v_{i,j+1/2} - u_{i,j-1/2}v_{i,j-1/2}}{dy}, \quad (\text{A.2e})$$

where the missing terms are obtained from averaging over the closest nodes, see Figure A.1; i.e.

$$u_{i+1/2,j} = (u_{i,j} + u_{i+1,j}) / 2 \quad (\text{A.3a})$$

$$u_{i-1/2,j} = (u_{i,j} + u_{i-1,j}) / 2 \quad (\text{A.3b})$$

$$u_{i,j+1/2} = (u_{i,j} + u_{i,j+1}) / 2 \quad (\text{A.3c})$$

$$u_{i,j-1/2} = (u_{i,j} + u_{i,j-1}) / 2 \quad (\text{A.3d})$$

$$v_{i,j+1/2} = (v_{i-1,j+1} + v_{i,j+1}) / 2 \quad (\text{A.3e})$$

$$v_{i,j-1/2} = (v_{i-1,j} + v_{i,j}) / 2. \quad (\text{A.3f})$$

Similarly, the following terms are obtained in the y direction,

$$\frac{\partial p}{\partial y} = \frac{p_{i,j} - p_{i,j-1}}{dy} \quad (\text{A.4a})$$

$$\frac{\partial^2 v}{\partial x^2} = \frac{v_{i+1,j} - 2v_{i,j} + v_{i-1,j}}{dx^2} \quad (\text{A.4b})$$

$$\frac{\partial^2 v}{\partial y^2} = \frac{v_{i,j+1} - 2v_{i,j} + v_{i,j-1}}{dy^2} \quad (\text{A.4c})$$

$$\frac{\partial uv}{\partial x} = \frac{u_{i+1/2,j}v_{i+1/2,j} - u_{i-1/2,j}v_{i-1/2,j}}{dx} \quad (\text{A.4d})$$

$$\frac{\partial vv}{\partial y} = \frac{v_{i,j+1/2}^2 - v_{i,j-1/2}^2}{dy}. \quad (\text{A.4e})$$

with the following averaging rules applied to Eq. (A.4)

$$v_{i,j+1/2} = (v_{i,j} + v_{i,j+1}) / 2 \quad (\text{A.5a})$$

$$v_{i,j-1/2} = (v_{i,j} + v_{i,j-1}) / 2 \quad (\text{A.5b})$$

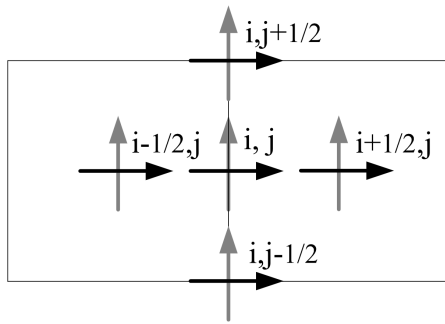
$$v_{i+1/2,j} = (v_{i,j} + v_{i+1,j}) / 2 \quad (\text{A.5c})$$

$$v_{i-1/2,j} = (v_{i,j} + v_{i-1,j}) / 2 \quad (\text{A.5d})$$

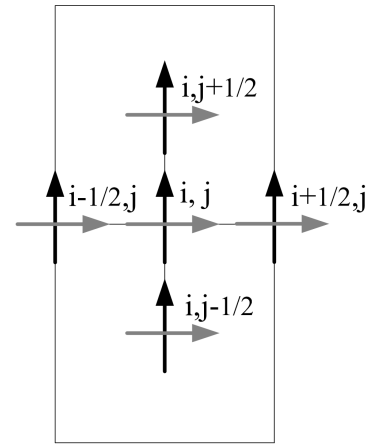
$$u_{i+1/2,j} = (u_{i+1,j-1} + v_{i+1,j}) / 2 \quad (\text{A.5e})$$

$$u_{i-1/2,j} = (u_{i,j} + v_{i,j-1}) / 2. \quad (\text{A.5f})$$

It is worth mentioning that although some terms are indexed similarly in Eqs. (A.2) and (A.4) (for instance $u_{i+1/2,j}$, $u_{i-1/2,j}$, $v_{i,j+1/2}$, and $v_{i,j-1/2}$), they are calculated differently, as illustrated in Eqs. (A.3) and (A.5) as well as Figure A.1.



(a)



(b)

Figure A.1: Illustration of half indices used for interpolation of velocities in: (a) momentum in x direction; (b) momentum in y direction.