MODELLING SOLID/FLUID INTERACTIONS IN HYDRODYNAMIC FLOWS: A HYBRID MULTISCALE APPROACH

Eduardo Ramos Fernández

Tribology Group
Department of Mechanical Engineering
Imperial College London, UK

Submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy

Department of Mechanical Engineering, Imperial College London
November 2020

Declarations

Declaration of originality

The contents of this thesis is the author's own work, except where otherwise stated. The research builds on a wide body of literature, which has been appropriately referenced and paraphrased. All text has been written by the author, with some assistance from Daniele Dini and Edward Smith. All images have been generated by the author, except where credit is given otherwise.

Copyright declaration

The copyright of this thesis rests with the author and is made available under a Creative Commons Attribution Non-Commercial No Derivatives license. Researchers are free to copy, distribute or transmit the thesis on the condition that they attribute it, that they do not use it for commercial purposes and that they do not alter, transform or build upon it. For any reuse or redistribution, researchers must make clear to others the license terms of this work.

Abstract

With the advent of high performance computing (HPC), we can simulate nature at time and length scales that we could only dream of a few decades ago. Through the development of theory and numerical methods in the last fifty years, we have at our disposal a plethora of mathematical and computational tools to make powerful predictions about the world which surrounds us. From quantum methods like Density Functional Theory (DFT); going through atomistic methods such as Molecular Dynamics (MD) and Monte Carlo (MC), right up to more traditional macroscopic techniques based on Partial Differential Equations (PDEs) discretization like the Finite Element Method (FEM) or Finite Volume Method (FVM), which are respectively, the foundation of computational Structural Analysis and Computational Fluid Dynamics (CFD). Many modern scientific computing challenges in physics stem from combining appropriately two or more of these methods, in order to tackle problems that could not be solved otherwise using just one of them alone. This is known as *multi-scale modeling,* which aims to achieve a trade-off between computational cost and accuracy by combining two or more physical models at different scales.

In this work, a multi-scale domain decomposition technique based on coupling MD and CFD methods, has been developed to make affordable the study of slip and friction, with atomistic detail, at length scales otherwise impossible by fully atomistic methods alone. A software framework has been developed to facilitate the execution of this particular kind of simulations on HPC clusters. This have been possible by employing the in-house developed `CPL_LIBRARY` software library, which provides key functionality to implement coupling through domain decomposition.

Acknowledgments

To my supervisor Prof. Daniele Dini, who gave me the encouragement and relentless support throughout this journey and who I consider not only a mentor, but a friend. I will always be in debt for the opportunity you gave me. To Dr. Edward Smith, who has always been my "go to" person on the technical side of things and who I admire profoundly for his brilliance, honesty and hard-work ethos. I wish you a fantastic career as a researcher.

To Prof. David Heyes for the stimulating discussions over coffee. It has been a privilege. To Dr. Christopher Warrens, which mentored the ICAM project I was part of. It was always a pleasure to meet you. I would like also to thank Dr Andrew Powell, my manager at Mathworks, for his support during the thesis writing process while working in his team.

To my girlfriend Nataly Garcia, who has shared the weight of the PhD with me over the last five years. Thank you for every word of encouragement and your continuous support.

Para mi madre Conchi y mi padre José Luis. Por el apoyo incondicional y luchar tanto mis batallas como las suyas sin descanso, os quiero y admiro.

To my "little" brothers Javier & Pablo. Thanks for bringing me joy since you were born. I know I can always count on you.

Finally I would like to thank the Theory and Simulation of Materials CDT, which I have been part of during my postgraduate studies, the EPSRC and most importantly BP-ICAM, for funding the work presented in this thesis.

Contents

Declarations iii

Abstract v

Acknowledgments vii

1. Introduction 1

1.1 Motivation and Project's Aims 1

1.2 Thesis Structure 3

2. Literature Review: Theory and Numerical Methods 5

2.1 Atomistic Simulations Under Shear 6

2.2 Statistical Mechanics 7

2.2.1 Phase Space and Liouville Equations 8

2.2.2 Time Average and Statistical Errors 11

2.2.3 Microscopic-Continuum connection 12

2.2.4 Atomistic Stress and Pressure 13

2.2.5 Transport Coefficients. Green-Kubo Formalism 15

2.3 Molecular Dynamics 16

2.3.1 Lennard-Jones Potential and Reduced Units 18

2.3.2 Molecular Fluids and Force Fields 22

2.3.3 Time Integration of Equations of Motion 25

2.3.4 Bulk Shear-NEMD Simulations and Viscosity Calculation 26

2.4 Fluid Dynamics and CFD 30

2.4.1 Fluid Dynamics Equations 31

2.4.2 Analytical Parallel Flows 35

2.5 Multiscale Domain Decomposition for Fluids 39
2.6 Chapter Summary 47

3. Multiscale CFD-MD Coupling Methodology 49

3.1 Introduction 49
3.2 Molecular Dynamics Domain 52
3.2.1 Modeling the Atomistic/Molecular Fluid 54
3.2.2 Modeling the Atomistic Surface 55
3.2.3 MD Domain Boundaries 58
3.3 Computational Fluid Dynamics Domain 61
3.3.1 Momentum equation. Newtonian and Non-Newtonian solvers 63
3.3.2 Thermal equation 67
3.4 Coupling Methodology 69
3.4.1 The Overlap Region 72
3.4.2 Synchronized Coupling 73
3.4.3 Temporal and Spatial Constraints 75
3.4.4 Averaging of MD Quantities 76
3.4.5 Mass Coupling 77
3.4.6 Momentum Coupling 79
3.4.7 Energy Coupling 81
3.5 Chapter Summary 83

4. Software Framework for Coupling 85

4.1 Introduction 85
4.2 CPL Coupling Library and CPL Applications 88
4.2.1 CPL_LIBRARY Design and API 89
4.2.2 CPL Applications. Object-Oriented CPLSocket Interface 93
4.3 Deployment and Testing of CPL Applications 101
4.3.1 Deploying CPL Applications and Reproducibility 102
4.3.2 Testing CPL Applications 105
4.4 Framework Parallel Scalability 108
4.4.1 LAMMPS scalability 111
4.4.2 Coupled OpenFOAM-LAMMPS scalability 113
4.5 Chapter Summary 121

5. Coupled Simulations: Setup and Results 123

5.1 Introduction 123
5.2 Properties of Liquids from Molecular Dynamics 124
5.2.1 Lennard-Jones Fluids 127
5.2.2 Molecular Fluids: Benzene and Squalane 134
5.3 Coupled Lennard-Jones Simulations 142
5.3.1 Building and Equilibrating the MD Domain 143
5.3.2 Parameter Selection for Density Correction Algorithm 146
5.3.3 Sampling Physical Fields 148
5.3.4 Setup Benchmarking 150
5.3.5 Numerical Validation of Coupled Flows 154
5.4 Coupled Molecular Simulations 161
5.4.1 Building Coupled Molecular Systems 163
5.4.2 Equilibration of Coupled Molecular Systems 166
5.4.3 NEMD Coupled Simulations 178
5.5 Overlap Region Dynamics 185
5.6 Chapter Summary 195

6. Conclusions and Future Work **199**

6.1 Context 199
6.2 Overview of this Work 200
6.2.1 Computational Aspects and Software Development 200
6.2.2 Coupling Methodology 202
6.2.3 Physical Simulations 205
6.3 Future Work 206

Bibliography **209**

Appendix A **237**

Unified Modeling Language 237
Software Versions and Repositories 238

Chapter 1

Introduction

1.1 Motivation and Project's Aims

To date, the field of nanotribology has remained slightly detached from mainstream macro-scale tribology, possibly because it has been developed largely by physicists and has focused primarily on specialised nano-scale applications. For example, layers of molecules of nanometric thickness adsorbed onto a surface can alter the physical properties of the surface-fluid interface, dramatically affecting boundary and hydrodynamic friction [1]. It is important to appreciate that nano-scale phenomena can directly determine much of the macro-scale tribological behaviour. Experimental and computational techniques employed in nanotribology, have now advanced to the stage where they can contribute significantly to our understanding of tribology as a whole. In particular, the use of computational atomistic models to understand mechanisms at length and time-scales which are beyond the resolution of current experimental apparatus, has bloomed in the last decades. An example is the Molecular Dynamics (MD) method [2], which considers the motion of many interacting particles whose trajectories

are computed by solving their individual equations of motion. This technique has been widely used to study the effect of friction modifiers in boundary lubrication [3,4] and hydrodynamic slip in parallel shear-driven flows at the nano-scale [5]. However, even with the advances on modern high performance computing, only rather small time (ns) and length (nm) scales are generally accessible through MD simulations. Consequently, atomistic methods have remained almost exclusive to academic research and are yet to permeate into engineering applications, whose physical scales are currently beyond reach. Therefore, numerical techniques based on continuum models, for instance, computational fluid dynamics (CFD) [6], are still the preferred methodology to study lubrication and friction when fluids are involved. Comparatively, CFD is computationally less expensive than atomistic methods, being able to simulate larger time and length scales. Nevertheless, CFD methods depend on a number of assumptions on the (a) constitutive relations (e.g between stress and rate of deformation) and (b) boundary conditions (e.g no-slip), while atomistic simulations, which rely on accurate descriptions of atomic and molecular interactions, can actually be used to study the suitability of such assumptions. Therefore, it would be great if the strengths of both atomistic and continuum methods could be harnessed to achieve a good trade-off between physical accuracy and computational cost. For example, to simulate bulk fluid behaviour, constitutive relations for CFD can be measured by a set of MD simulations. Then, the CFD would behave rheologically the same as the MD model. Furthermore, if boundary effects exhibiting complex physics are localized, we could in principle limit the application of an expensive atomistic model just to the region of interest close to the boundary and model the rest of the domain with CFD. A family of multi-scale techniques based on *domain-decomposition* [7] are very well suited for this scenario. In this approach, the simulation domain is divided into an atomistic sub-domain containing the boundary and a continuum sub-domain simulated using CFD. The two sub-domains are *coupled* through an overlapping region where consistency between the physical fields on both domains is enforced. Therefore, molecular detail at the surface-fluid interface is retained, but larger length-scales can be reached at virtually no cost by

enlarging the CFD domain, which is orders of magnitude computationally less expensive.

The principal aim of the thesis was to develop and implement a suitable multi-scale MD-CFD coupling methodology to study slip and friction with molecular detail at the solid-fluid interface. This allows simulating flows at length-scales impossible to reach for a full MD simulation. In addition, the method was developed to study tribologically relevant fluids (e.g hydrocarbon molecules present in base oils) at high pressure under incompressibility and laminar flow conditions.

Additionally, the second objective was to implement a parallel software framework to facilitate the setup, efficient execution and reproducibility of coupled MD-CFD simulations in high performance computing (HPC) clusters. This was achieved by adopting modern software development practices like modularization of the software, automated testing and packaging of the coupled software for easy distribution and deployment. Furthermore, the `CPL_LIBRARY` [8] coupling library was used to facilitate the development of the coupled MD and CFD solvers. This library enables geometrical coupling of two solvers using domain decomposition parallelization through `MPI` [9] and takes care of the data transfer between domains in the overlap region. The purpose of this library is two-fold: (i) ensuring efficient data transfer between MD and CFD domains by reducing unnecessary communications, (ii) allowing the development of each coupled solver as a separate application instead of packing them together in a single monolithic piece of software. The latter has profound implications in terms of ensuring the testability, extensibility and future development of the implemented solvers.

1.2 Thesis Structure

This document is organised as follows. In Chapter 2 a review of the the theory to support coupling by domain decomposition is given. Several multi-scale strategies are reviewed with a special focus on coupling molecular dynamics (MD) to computational fluid dynamics (CFD). Furthermore,

different aspects of the MD method like interaction potentials, time-integration, thermostatting and non-equilibrium simulations are also examined and their statistical mechanics foundations revisited. In addition, the three central continuum conservation equations used in fluid dynamics are presented, along with several time-dependent analytical expressions for three different parallel flows. The specifics about the multi-scale methodology implemented in this work are contained in Chapter 3. The chapter is divided into three sections describing the MD and CFD domains implementations and the coupling methodology followed in the overlap region. In Chapter 4 the implementation details of the software framework are discussed. In particular the software architecture, testing and deployment strategy and the results of a range of parallel scalability studies are presented. Chapter 5 contains the results of coupled simulations of Lennard-Jones fluids, molecular fluids (benzene and squalane) and fluid properties computed from bulk MD simulations. Lastly, in Chapter 6 the overall project's achievements are summarised and future lines of work are discussed.

Chapter 2

Literature Review: Theory and Numerical Methods

This chapter reviews the theoretical and numerical background supporting the rest of this work. The topics of *molecular dynamics (MD)* and its grounds in *statistical mechanics*; *computational fluid dynamics (CFD)* and *atomistic-continuum coupling* for fluids are covered. Furthermore, a more specialized section on tribology regarding the study of *hydrodynamic lubrication*, *slip* and *friction* at the nanoscale is also included.

Comprehensive derivations of equations and mathematical expressions which are not original to this work, have not been included but referenced appropriately. Mathematical derivations carried out by the author and other extra content can be found in the appendices.

If the word *molecular* precedes a certain concept (e.g *molecular system*) it refers exclusively to non-monoatomic fluids. This is to make a distinction between simple fluids and more complex ones, with molecules containing internal degrees of freedom.

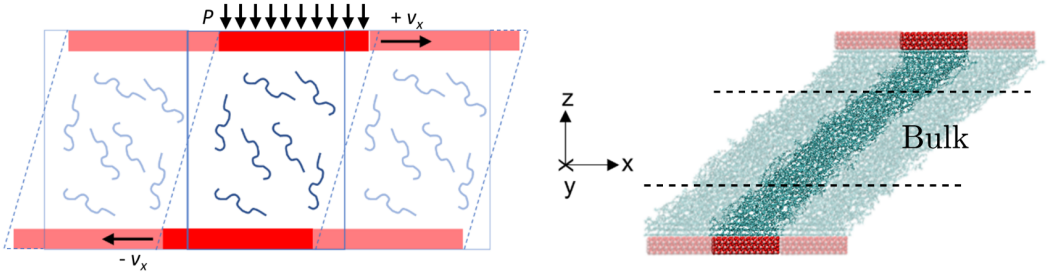
2.1 Atomistic Simulations Under Shear

This section is included to give the reader some perspective about the types of molecular simulations intended to be replicated at larger scales by using MD-CFD coupling, which as stated in Section 1.1, is the primary goal of this thesis. More precisely, the focus is on atomistic systems under shear, from which tribology-related quantities can be measured (e.g friction coefficient, slip, viscosity, etc.). Commonly, non-equilibrium molecular dynamics (NEMD) is employed to simulate such systems. A comprehensive treatment of the method's underlying theory can be found in Evan & Morriss and Todd & Daivis books [10,11].

In early attempts to simulate these systems, shear was induced by applying equal and opposite velocities to the atoms within regions of fluid at the top and bottom of the simulation cell [12]. However, in more recent simulations, it is common to apply shear through two moving parallel atomistic walls, which *confine* the molecular fluid in one direction as shown in Figure 2.1. Span-wise and stream-wise directions are set as periodic. This simulation setup was firstly implemented by Bitsanis *et al.* to study the viscosity and flow properties of liquids confined in micropores [13]. The inclusion of explicit atomistic surfaces enables the study of fluid-solid interfaces under different physical conditions. Some examples are the study of how different types of surface roughness affect slip [14,15], or the impact of friction modifiers (small molecules added to oil formulations) in boundary lubrication [4,16]. However, if interfacial effects are localized and the atomistic fluid exhibit bulk behaviour far from the surface (see Figure 2.1), a domain decomposition strategy could be in principle applied by replacing the bulk atomistic region depicted in Figure 2.1, with a CFD solver.

Alternatively, bulk NEMD simulations can be performed with periodic boundary conditions in all three Cartesian directions. This kind of simulations are commonly used to measure bulk fluid properties (e.g. viscosity) and they have been reviewed in Section 2.3.4. Despite confinement can significantly influence the fluid's behaviour [14], excellent agreement between bulk and confined NEMD simulations has been observed when the

surfaces are sufficiently separated such that there is a negligible influence of the walls on the fluid bulk response.



(a) (b)

Figure 2.1: (a) Schematic of a confined NEMD simulation and (b) an example snapshot from a confined NEMD simulation. Adapted with permission from [1]. Copyright Springer, 2020.

One important consequence of the short accessible timescales in NEMD simulations are that extremely high shear rates, typically $\dot{\gamma} > 10^7$ s^{-1}, are required to reach a non-equilibrium steady-state [18]. Simulating lower shear rates have been a long-sought goal of NEMD simulations in order to enable direct comparison with experiments and real mechanical components. For instance, the viscosity of lubricants at high pressures, can generally only be measured up to shear rates of approximately $\dot{\gamma} = 10^4$ s^{-1} [19], tribology experiments can extend up to around $\dot{\gamma} = 10^6$ s^{-1} [20], while high-performance engine components can reach up to $\dot{\gamma} = 10^8$ s^{-1} [21]. These time-scale limitations are also shared by any MD-CFD coupled simulation, since they are exclusively related to the the size of the MD solver time-step.

2.2 Statistical Mechanics

Statistical mechanics is the branch of physics which studies the description and evolution of physical systems containing a high number of degrees of freedom. Such systems cannot simply be analyzed by following the time-evolution of each degree of freedom individually, but treating all of them in a statistical sense. In particular, this work focuses exclusively on systems composed of a high number of interacting particles (*many-body problem*).

The theory covered in this section is largely based on the books of Landau & Lifshitz [22] and Beale [23], and is intended to provide the underlying statistical framework on which the molecular dynamics method, later reviewed in Section 2.3, is grounded.

2.2.1 Phase Space and Liouville Equations

A 3D system containing N particles can be fully determined at any time by $6N$ degrees of freedom. Mathematically, the system can be described as a $6N$ -dimensional *phase space* which contains all the possible combinations of momenta $\boldsymbol{p} = (p_x(t), p_y(t), p_z(t))$ and positions $\boldsymbol{r} = (x(t), y(t), z(t))$ of all particles. A phase space vector $\boldsymbol{\Gamma} = (\boldsymbol{p}_1, \boldsymbol{p}_2, ..., \boldsymbol{p}_N, \boldsymbol{r}_1, \boldsymbol{r}_2, ..., \boldsymbol{r}_N)$ represents a single microscopic state in the phase space, and an infinitesimal volume of the phase space can be written as $d\boldsymbol{\Gamma} = d\boldsymbol{r}_1 d\boldsymbol{r}_2 ... d\boldsymbol{r}_N d\boldsymbol{p}_1 d\boldsymbol{p}_2 ... d\boldsymbol{p}_N$. The probability of finding a particle in the phase space interval $\boldsymbol{\Gamma} + d\boldsymbol{\Gamma}$ is given by $f(\boldsymbol{\Gamma}; t)\, d\boldsymbol{\Gamma}$ where $f(\boldsymbol{\Gamma}; t)$ is a probability density function (PDF) which satisfies the normalization condition

$$\int_{6N} \dots \int f(\boldsymbol{\Gamma}; t)\, d\boldsymbol{\Gamma} = 1. \tag{2.2.1}$$

The PDF contains all the information about the distribution of the particles in the phase space at any time and it is of pivotal importance in statistical mechanics.

The time evolution of f can be described by the *Liouville's* equations of statistical mechanics, which can be derived in an analogous way to the conservation equations of hydrodynamics but considering a volume in phase space instead of a volume of fluid. The general equation

$$\frac{\partial f}{\partial t} = -\sum_{i}^{N} \left(\frac{\partial f}{\partial \boldsymbol{r}_i} \cdot \dot{\boldsymbol{r}}_i + \frac{\partial f}{\partial \boldsymbol{p}_i} \cdot \dot{\boldsymbol{p}}_i \right), \tag{2.2.2}$$

is expressed in terms of the conjugate position $\dot{\boldsymbol{r}}$ and momenta $\dot{\boldsymbol{p}}$ which are defined by the *canonical equations of motion*

$$\left.\begin{aligned} \dot{\boldsymbol{r}}_i &= \frac{\partial H(\boldsymbol{r_i}, \boldsymbol{p}_i)}{\partial \boldsymbol{p}_i} \\ \dot{\boldsymbol{p}}_i &= -\frac{\partial H(\boldsymbol{r_i}, \boldsymbol{p}_i)}{\partial \boldsymbol{r}_i} \end{aligned}\right\} i = 1, 2, ..., N \tag{2.2.3}$$

where H is the Hamiltonian, which can be written for a closed system of interacting particles as

$$H(\boldsymbol{r}_i, \boldsymbol{p}_i) = K(\boldsymbol{p}_i) + V(\boldsymbol{r}_i) = \sum_i^N \frac{\boldsymbol{p}_i \cdot \boldsymbol{p}_i}{2m_i} + V(\boldsymbol{r}_i), \tag{2.2.4}$$

where $K(\boldsymbol{p}_i)$ is the kinetic energy which depends only on the particle momenta and $V(\boldsymbol{r}_i)$ is the potential (or configurational) energy, which depends exclusively on the particle position. The two contributions are fundamentally different in nature. For a dilute gas, particle interactions are infrequent and most of the energy is kinetic. In the limit of $V(\boldsymbol{r}_i) \to 0$, the the gas can be considered ideal. Conversely, in dense liquids, particles are more packed and the inter-atomic distances are short, therefore the configurational energy is higher and commonly in the order of the kinetic energy [2]. Equation 2.2.2 can now be re-written using Equations 2.2.3 and 2.2.4 as

$$\frac{\partial f}{\partial t} = \sum_i^N \left(\frac{\partial f}{\partial \boldsymbol{p}_i} \cdot \frac{\partial V(\boldsymbol{r}_i)}{\partial \boldsymbol{r}_i} - \frac{\partial f}{\partial \boldsymbol{r}_i} \cdot \frac{\boldsymbol{p}_i}{m_i} \right). \tag{2.2.5}$$

Note this equation holds for equilibrium and non-equilibrium systems and the only assumption made is on the expression for the Hamiltonian H.

Let's now define a *dynamical variable* $A(\boldsymbol{\Gamma})$ as a physical quantity that is a function of $\boldsymbol{\Gamma}$, hence a function of the mechanical state of the system. The expectation value of a dynamical variable A at a time t, can be computed as

$$\langle A; f \rangle = \int_{6N} \dots \int A(\boldsymbol{\Gamma}) f(\boldsymbol{\Gamma}; t)\, d\boldsymbol{\Gamma}, \tag{2.2.6}$$

where $\langle A; f \rangle$ denotes the *ensemble average* and A depends on particle momenta and positions, but not explicitly on time. The rate of change of the expectation value of A can be computed from

$$\frac{\partial}{\partial t} \langle A; f \rangle = \sum_{i}^{N} \left\langle -\frac{\partial A}{\partial \boldsymbol{p}_i} \cdot \frac{\partial V(\boldsymbol{r}_i)}{\partial \boldsymbol{r}_i} + \frac{\partial A}{\partial \boldsymbol{r}_i} \cdot \frac{\boldsymbol{p}_i}{m_i}; f \right\rangle, \tag{2.2.7}$$

by applying Green's theorem over $\boldsymbol{r}_i$ and $\boldsymbol{p}_i$ space, provided f is bounded or decays sufficiently fast as $\boldsymbol{r}_i \to \infty$ and $\boldsymbol{p}_i \to \infty$.

The functional form of f defines the probability of the system of particles to be in a particular microscopic state $\boldsymbol{\Gamma}$. An equilibrium ensemble can be defined mathematically as one whose phase space density function f has no explicit time-dependency, $\partial f / \partial t = 0$. Under this condition the Liouville's equation for the Hamiltonian H in Equation 2.2.4 can be written as

$$\sum_{i}^{N} \left(\frac{\partial f}{\partial \boldsymbol{p}_i} \cdot \frac{\partial V(\boldsymbol{r}_i)}{\partial \boldsymbol{r}_i} - \frac{\partial f}{\partial \boldsymbol{r}_i} \cdot \frac{\boldsymbol{p}_i}{m_i} \right) = 0, \tag{2.2.8}$$

which can be expressed using the Poisson bracket notation as $\{f, H\} = 0$. This implies that f must be a pure function of the Hamiltonian $f(\boldsymbol{\Gamma}) = F(H(\boldsymbol{\Gamma}))$. There are two important ensembles to consider for closed system with a fixed volume V and number of particles N:

a) The *microcanonical ensemble* (NVE) , where the energy of the system is constant $H(\boldsymbol{\Gamma}) = E$. Given the postulate of *a priori equal probabilities* [23], f is constant for any microscopic state $\boldsymbol{\Gamma}$

$$f(\boldsymbol{\Gamma}) \propto \delta \left(H(\boldsymbol{\Gamma}) - E \right). \tag{2.2.9}$$

b) The *canonical ensemble* (NVT), where the system of particles is in thermal equilibrium with a thermodynamic reservoir at temperature T. The probability density function f has the form [23]

$$f(\boldsymbol{\Gamma}) \propto \exp\left(-\frac{H(\boldsymbol{\Gamma})}{K_b T} \right). \tag{2.2.10}$$

Other ensembles where different physical quantities are fixed do exist, like the chemical potential, pressure or stress [11].

2.2.2 Time Average and Statistical Errors

In a molecular dynamics simulation, the mean value of a physical quantity A is computed by averaging over space and time. Sufficient samples in time and space must be taken in order to achieve good statistical significance. The *ergodic hypothesis* [11] states that over long periods of time, the time spent by a system in some region of the phase space of microstates with the same energy, is proportional to the volume of this region. This hypothesis is always assumed to hold in molecular dynamics simulations and the direct implication is that it links the time average of a quantity A, with the ensemble average (defined in Equation 2.2.6)

$$\langle A; f \rangle \approx \frac{1}{\Delta t} \int_{t_0}^{t_0+\Delta t} A(t)\, dt \approx \frac{1}{N_s} \sum_{i=0}^{N_s-1} A(t_i), \tag{2.2.11}$$

with $t_i = t_0 + i\Delta t/N_s$, where N_s is the number of samples. The ergodic hypothesis is crucial as it connects the molecular dynamics method with the ensemble theory of statistical mechanics.

If all instantaneous values selected for a quantity A are statistically independent (uncorrelated samples), their distribution would be Gaussian [2] with the variance

$$\sigma^2(A) = \frac{1}{N_s} \sum_{i=1}^{N_s} \left(A(t_i) - \langle A \rangle\right)^2, \tag{2.2.12}$$

and the variance of the simulation average could be estimated from

$$\sigma^2(\langle A \rangle_{\text{gauss}}) = \frac{\sigma^2(A)}{N_s} = \frac{1}{N_s^2} \sum_{i=1}^{N_s} \left(A(t_i) - \langle A \rangle\right)^2. \tag{2.2.13}$$

In this case, the standard deviation $\sigma(\langle A \rangle_{\text{gauss}})$ would be a good measure for the statistical uncertainty of the simulation average, also known as the standard error of the mean (SEM).

2.2.3 Microscopic-Continuum connection

In hydrodynamics, mass density $\rho(\boldsymbol{r},t)$, momentum density $\rho(\boldsymbol{r},t)\boldsymbol{u}(\boldsymbol{r},t)$ and energy density $\rho(\boldsymbol{r},t)\varepsilon(\boldsymbol{r},t)$ fields, are considered to be continuous functions in space and time. This is known as the *continuum hypothesis* [24], which is consistent with the macroscopic behaviour of matter. As later discussed in Section 2.4.1, mass, momentum and energy are always conserved, which is expressed mathematically, in a differential form, by Equation 2.4.1. In reality, the continuum hypothesis is just a macroscopic manifestation of the statistical collective behaviour of the discrete particles that matter is composed of. Thus, there has to be a group of mathematical relations which connect (in a statistical sense) continuum and atomistic representations of the conserved fields. In 1950 Irving and Kirkwood [25] addressed this problem, providing the link between continuum fields and microscopic variables. To accomplish this, they introduced the Dirac delta function which, for an arbitrary function $F(\boldsymbol{r})$, satisfies the property

$$F(\boldsymbol{r}) = \int_{-\infty}^{\infty} \delta(\boldsymbol{r}_i - \boldsymbol{r}) F(\boldsymbol{r}_i) d\boldsymbol{r}_i, \tag{2.2.14}$$

where $\boldsymbol{r}$ is a spatial location, $\boldsymbol{r}_i$ is the position vector of atom i. Using the the properties of the Dirac delta function, the definition for the expectation value of a variable A (Equation 2.2.6) and the expression for the rate of change of A (Equation 2.2.7), Irving and Kirkwood derived three different relations for mass, momentum and energy. The reader is referred to their work for a full step by step explanation [25]. For the mass density, the two descriptions are linked by

$$\rho(\boldsymbol{r},t) \equiv \left\langle \sum_{i=1}^{N} m_i \delta(\boldsymbol{r}_i - \boldsymbol{r}); f \right\rangle, \tag{2.2.15}$$

where m_i is the mass of particle i located at $\boldsymbol{r}_i$. The relation for the momentum density can expressed as

$$\rho(\boldsymbol{r},t)\boldsymbol{u}(\boldsymbol{r},t) \equiv \left\langle \sum_{i=1}^{N} \boldsymbol{p}_i \delta(\boldsymbol{r}_i - \boldsymbol{r}); f \right\rangle, \tag{2.2.16}$$

where $\boldsymbol{p}_i = m_i \boldsymbol{v}_i$ is the particle momenta in the *laboratory frame*, which is related to the *peculiar* momenta $\bar{\boldsymbol{p}}_i$ as

$$\bar{\boldsymbol{p}}_i \equiv m_i \left(\frac{\boldsymbol{p_i}}{m_i} - \boldsymbol{u}(\boldsymbol{r}_i) \right), \tag{2.2.17}$$

where $\boldsymbol{u}(\boldsymbol{r}_i)$ is the *streaming* velocity (mean velocity of the fluid) at $\boldsymbol{r}_i$. Lastly the connection between the continuum and microscopic energy density is written as

$$\rho(\boldsymbol{r}, t)\varepsilon(\boldsymbol{r}, t) \equiv \left\langle \sum_{i=1}^{N} e_i \delta(\boldsymbol{r}_i - \boldsymbol{r}); f \right\rangle, \tag{2.2.18}$$

where the particle's energy e_i is the sum of its kinetic and potential energy

$$e_i \equiv \frac{\boldsymbol{p}_i^2}{2m_i} + \frac{1}{2} \sum_{j \neq i}^{N} V_{ij}, \tag{2.2.19}$$

where $V_{ij} = V_{ji}$ for a pair-wise potential and the $\frac{1}{2}$ factor is included in the potential energy to avoid double counting.

2.2.4 Atomistic Stress and Pressure

The hydrodynamic stress tensor can be decomposed on its hydrostatic and deviatoric components [26],

$$\boldsymbol{\sigma} = -p\boldsymbol{I} + \boldsymbol{\tau}. \tag{2.2.20}$$

If the fluid is at rest, velocity gradients are zero at every point in space and the stress tensor can be written as

$$\begin{aligned} \boldsymbol{\tau} &= f(\nabla \boldsymbol{v}) = 0, \\ \boldsymbol{\sigma} &= -p\boldsymbol{I}, \end{aligned} \tag{2.2.21}$$

meaning the only stresses in a fluid at rest are related to the hydrostatic pressure p. The mechanical definition of pressure from a continuum perspective, is the force per unit area defined in the limit of an infinitesimal area A as

$$p = \lim_{A \to 0} \frac{\boldsymbol{F} \cdot \boldsymbol{n}}{A}, \tag{2.2.22}$$

where $\boldsymbol{n}$ is the unit normal vector to the area A and $\boldsymbol{F}$ is the net force acting on it. Conversely from thermodynamics, the equilibrium pressure $p \equiv p_0$, is defined by the relation [23]

$$p_0 \equiv \left(\frac{\partial H}{\partial V} \right)_{N,T}, \tag{2.2.23}$$

where H is the Helmholtz free energy, T is the temperature, N is the number of particles and V is the volume. This pressure is defined in the thermodynamic limit as $N \to \infty$, $V \to \infty$. Even though this definition can be used to approximate the pressure in molecular dynamics [27], it is common to use the expression derived from the Virial theorem due to its convenience in terms of implementation [2]

$$p_0 = \left\langle \frac{N k_B T}{V} \right\rangle + \left\langle \frac{1}{3V} \sum_i \boldsymbol{r}_i \cdot \boldsymbol{f}_i \right\rangle, \tag{2.2.24}$$

where N is the number of particles, V is the volume considered, k_B is the Boltzman constant, $\boldsymbol{r}_i$ is the position vector and $\boldsymbol{f}_i$ the total force acting on atom i. For the particular case of pair-wise particle interactions, a more convenient expression can be rewritten as [27]

$$p_0 = \left\langle \frac{N k_B T}{V} \right\rangle + \left\langle \frac{1}{3V} \sum_i \sum_{j<i} \boldsymbol{r}_{ij} \cdot \boldsymbol{f}_{ij} \right\rangle, \tag{2.2.25}$$

where $\boldsymbol{r}_{ij}$ and $\boldsymbol{f}_{ij}$ are the inter-atomic position and force vectors of the atomic pair ij. More generally, the microscopic pressure tensor, which is equivalent to the Cauchy stress tensor [28] can be written as

$$\boldsymbol{P} = \frac{1}{V} \sum_{i=1} \frac{\bar{\boldsymbol{p}}_i \bar{\boldsymbol{p}}_i}{m_i} + \frac{1}{V} \sum_i \boldsymbol{r}_i \boldsymbol{f}_i, \tag{2.2.26}$$

where $\bar{\boldsymbol{p}}_i$ is the peculiar momentum as defined in Equation 2.2.17. Note products between vectors are tensor products in the expression above. The connection between the hydrodynamic and atomistic definition of the stress tensor can be found in the landmark work of Irving & Kirkwood [25], in which Equation 2.2.26 is equivalent to the IK1 stress approximation. This

definition is valid for both equilibrium and non-equilibrium conditions. The non-equilibrium pressure can be identified with the continuum definition of hydrostatic pressure $p = \frac{1}{3}tr(\boldsymbol{P}) = \frac{1}{3}\left(P_{xx} + P_{yy} + P_{zz}\right)$. Conversely if the liquid is in equilibrium, then $p \equiv p_0 = P_{xx} = P_{yy} = P_{zz}$.

2.2.5 Transport Coefficients. Green-Kubo Formalism

Diffusive processes are ubiquitous in nature. Coefficients associated with diffusion of physical quantities in hydrodynamics can be computed from Linear Response Theory (LRT) and the fluctuation-dissipation theorem [11], using the famous *Green-Kubo* relations [29]. The mathematical expressions of these relations are presented here without derivation. The reader is referred to [11] for a comprehensive treatment of the subject. The three coefficients defining the mass flux (self-diffusion D), the momentum flux (viscosity $\eta_{\alpha\beta}$) and heat flux (heat conductivity λ) can be defined, respectively, as

$$D = \frac{1}{3N}\sum_{i=1}^{N}\int_0^\infty \langle \boldsymbol{v}_i(t)\cdot\boldsymbol{v}_i(0)\rangle dt, \tag{2.2.27}$$

where $\boldsymbol{v}_i$ is the particle velocity and N is the number of particles,

$$\eta_{\alpha\beta} = \frac{V}{K_bT}\int_0^\infty \langle P_{\alpha\beta}(t)\cdot P_{\alpha\beta}(0)\rangle dt, \tag{2.2.28}$$

where $P_{\alpha\beta}$ is the component $\alpha\beta$ of the microscopic pressure tensor, and

$$\lambda = \frac{V}{3K_bT}\int_0^\infty \langle \mathbf{J}^q(t)\cdot\mathbf{J}^q(0)\rangle dt \tag{2.2.29}$$

where $\mathbf{J}^q$ is the microscopic heat flux that for a pair-wise potential can be written as

$$\mathbf{J}^q = \frac{1}{V}\left[\sum_i e_i\boldsymbol{v}_i + \sum_i\sum_{i<j}\left(\boldsymbol{F}_{ij}\cdot\boldsymbol{v}_j\right)\boldsymbol{r}_{ij}\right] \tag{2.2.30}$$

where e_i is the per-atom energy, which for pair-wise interactions is defined by Equation 2.2.19. If the molecular potential includes many-interactions, the expression for the heat flux $\mathbf{J}^q$ can be written as described in [30].

2.3 Molecular Dynamics

The Molecular Dynamics (MD) method was initially developed, in an independent manner, by Alder & Wainwright [31] in the 1950s and by Rahman [32] in the 1960s. The method is deterministic and consists on generating phase space trajectories by integrating in time the equations of motion of a group of N classical particles from an initial configuration. Thermodynamic properties of the system can be derived from the trajectories by exploiting the *ergodic hypothesis* under equilibrium and non-equilibrium conditions [11].

Other types of popular atomistic techniques are based on stochastic processes, which are commonly known as Monte-Carlo (MC) methods. In MC a series of trial moves in phase space are either accepted or rejected based on the ensemble considered, which has a specific probability distribution associated to it. However, Monte-Carlo simulations do not provide any dynamical information, hence they cannot be used to model the time-evolution of atomistic fluid flows [33]. Therefore, this family of methods are not useful for the purpose of this work and are not reviewed here.

The equations of motion used in MD can be derived using the Euler-Lagrange equation which express the mathematical conditions under which the principle of least action is valid (variational calculus) [34]

$$\frac{d}{dt}\left(\frac{d\mathcal{L}}{\partial \dot{\boldsymbol{q}}_i}\right) - \left(\frac{d\mathcal{L}}{\partial \boldsymbol{q}_i}\right) = 0, \tag{2.3.1}$$

where $\boldsymbol{q}_i$ and $\dot{\boldsymbol{q}}_i$ are, respectively, the canonical coordinate and time-derivative of the canonical coordinate of particle i and $\mathcal{L}$ is the Lagrangian defined as

$$\mathcal{L} = K - V = \sum_{i=1}^{N} \frac{1}{2} m_i \dot{\boldsymbol{q}}_i \cdot \dot{\boldsymbol{q}}_i - V(\boldsymbol{q}_1, \boldsymbol{q}_2, ..., \boldsymbol{q}_N), \tag{2.3.2}$$

where K and V are the kinetic and potential energy of the particle system respectively. By inserting Equation (2.3.2) into Equation (2.3.1) with

Cartesian coordinates $\boldsymbol{q}_i \rightarrow \boldsymbol{r}_i$ we can obtain the Newtonian equations of motion for an unconstrained system of particles,

$$\begin{aligned} \boldsymbol{F}_i &= m_i \ddot{\boldsymbol{r}}_i = -\frac{\partial V}{\partial \boldsymbol{r}_i}, \\ \dot{\boldsymbol{r}}_i &= \frac{\dot{\boldsymbol{p}}_i}{m_i}, \\ \boldsymbol{v}_i &= \dot{\boldsymbol{r}}_i, \\ \boldsymbol{a}_i &= \ddot{\boldsymbol{r}}_i, \end{aligned} \tag{2.3.3}$$

where V is the configurational or potential energy, which depends exclusively on the particle positions. Lagrangian and Hamiltonian mechanics are linked through a *Legendre transformation* [34] and we could have equally obtained the above equations from Equation 2.2.3. In the case of constrained dynamics, *holonomic* and *non-holonomic* constraints can be imposed on the system using the Gauss' principle of least constraint [34].

In general the the functional form of V can be written as a sum of interactions of increasing order

$$\begin{aligned} V &= \sum_i V_1(\boldsymbol{r}_i) + \sum_i \sum_{j>i} V_2(\boldsymbol{r}_i, \boldsymbol{r}_j) + \sum_i \sum_{j>i} \sum_{k>j>i} V_3(\boldsymbol{r}_i, \boldsymbol{r}_j, \boldsymbol{r}_k) + ... \\ &\approx \sum_i V_1(\boldsymbol{r}_i) + \sum_i \sum_{j>i} V_2^{eff}(\boldsymbol{r}_{ij}) \end{aligned} \tag{2.3.4}$$

where V_1 is the the energy due to the interaction of the particles with an external field and V_m with $m = 2, 3, ...$ are the energy terms from the interaction of pairs, triplets, … of particles. Including non-bonded interactions of order $m > 2$ in atomistic simulations is computationally very expensive, specially in dense fluids, and it is a common practice to replace all m-body interactions with an effective pair-wise term V^{eff} [2]. The most widely used (by far) expression for V^{eff} in molecular simulations is the Lennard-Jones potential, which accounts for short-ranged non-bonded interactions. Coulombic interactions, if considered, are treated separately. On the other hand, bonded interactions which involve pairs, triplets or quadruplets of atoms from the same molecule (such as those defined in bond stretching, angle bending and torsional rotations) can be efficiently computed, so they are usually not included in V^{eff}. Potentials (*force-fields)*

which include bonded interactions, are used to model molecular fluids and are reviewed in Section 2.3.2. In addition, the concept of *Lennard-Jones fluid* is commonly used to refer to mono-atomic fluids interacting with Lennard-Jones potential and it is employed as such in this manuscript.

In this work, the stable version of `LAMMPS` (August 2017) molecular dynamics software package has been employed to perform all MD simulations (see Software Versions and Repositories in Appendix A). Any description of the MD algorithms employed here, refer to the ones implemented in such version or added by the author using the existing base code, in which case it is mentioned explicitly. There are several reasons why `LAMMPS` was chosen as a MD engine: 1) it is open-source so the code can be is easily accessed; 2) its architecture makes easy to add new functionality in a plugin-like fashion; 3) it contains lots of pre-packaged algorithms useful to perform NEMD simulations; 4) it can be parallelized by domain decomposition using `MPI`, which makes it compatible with the `CPL_LIBRARY` used in this work to perform coupling. For a description of the particular core algorithms in `LAMMPS`, like the evaluation of forces using neighbors lists and parallelization strategies, refer to [35].

2.3.1 Lennard-Jones Potential and Reduced Units

This section reviews the Lennard-Jones (LJ) inter-atomic interaction model, which is a very popular potential used in molecular dynamics from which inter-molecular forces are derived (see Equation 2.3.3). This interaction model has been successfully used in numerous applications of statistical thermodynamics and to predict, by computer simulations, properties of real fluids [36–39]. The full original functional form of the LJ potential [40] can be written as

$$V_{LJ}(r) = 4\epsilon \left[\left(\frac{\sigma}{r}\right)^{12} - \left(\frac{\sigma}{r}\right)^{6} \right], \tag{2.3.5}$$

where ϵ is the depth of the energy "well" and σ is the characteristic length scale of the interaction. The Lennard-Jones potential has a strong, repulsive term and a long, weakly attractive tail. In order to understand the functional

form of Equation 2.3.5 we have to examine the origin and nature of forces appearing between molecules. Insightful reviews of the inter-molecular forces topic, are provided in Hirschfelder *et al.* [41] and Maitland *et al.* [42].

Inter-molecular forces have their origin in the electric and magnetic interactions between their constituent fundamental charged particles: protons (positive charge) within the nuclei and electrons (negative charge) within the molecular orbitals. The strength of the interactions depends directly on the distance between the molecules, which are repulsive at short distances and attractive at large distances. Repulsive forces appear when two nuclei are close enough so that their electronic clouds overlap, creating by the Pauli exclusion principle, a reduction in the negative charge density in this region. Therefore, this effect reduces the *shielding* of the positively charged nuclei causing them to repel each other. Conversely, attractive long range forces can have contributions from three different mechanisms:

(a) *Electrostatic interactions between permanent dipoles.* Polar molecules have permanent non-zero dipole moments as a consequence of their electrostatic charge distribution.

(b) *Induced interactions.* When a polar molecule interacts with a non-polar molecule, a distortion is created in the charge distribution of the uncharged molecule, inducing an effective non-zero dipole moment on it. In addition, the newly induced dipole, interacts with the polar molecule resulting in a force between the two molecules.

(c) *Dispersion interactions.* Electron clouds are not static, therefore, fluctuations in the electronic clouds surrounding a molecule occur, which cause the appearance of dispersion forces. Every molecule possesses an instantaneous electric dipole moment due to the charge fluctuation. This instantaneous dipole, induces itself instantaneous dipoles on other molecules, causing molecules to interact with each other by this mechanism.

Short-ranged (≈ 0.5 nm) interactions between two non-bonded and uncharged atoms are known as *Van der Waals*. A commonly adopted

potential energy function to describe these type of interactions is the Exponential-6 function [43]

$$U_{X6} = Ae^{-Br} - \frac{C}{r^6}, \tag{2.3.6}$$

which have a close resemblance to the Lennard-Jones potential. The $1/r^6$ tail, present in both Equation 2.3.5 and Equation 2.3.6, agrees qualitatively with the leading term of the theoretical result for the dispersion energy between two non-polar spherical molecules [42]. On the other hand, the repulsive $1/r^{12}$ term in the LJ model, has no strict physical foundation and it is equivalent to the exponential term in Equation 2.3.6. However, the former is significantly more efficient to compute since it can be calculated as the square of the $1/r^6$ repulsive term.

Other more simplistic interaction potentials, like the hard sphere model, were used in the early days of MD, but they are known to fall short in reproducing certain essential physics [42]. Conversely, more complex potentials, like the Tersoff [44] potential, which includes higher order interactions can be used to model both bonded and non-bonded interactions.

A slightly different modification of the LJ potential known as the Lennard-Jones truncated and shifted (LJTS) potential [45] is commonly used in practice. In this model, interactions are not considered further than a cut-off radius r_c and it is written as

$$V_{LJTS}(r) = \begin{cases} V_{LJ}(r) - V_{LJ}(r_c) & r \leqslant r_c \\ 0 & r > r_c, \end{cases} \tag{2.3.7}$$

where V_{LJ} is the full LJ functional form defined in Equation 2.3.5. The LJTS potential is computationally cheaper than the original Lennard-Jones potential, but still reproduce the fundamental physical features of matter like the existence of a critical point, soft repulsive/attractive interactions, phase equilibria etc. [46,47]. Therefore, the LJTS potential frequently used for testing new algorithms and physical theories. Hence, the LJTS potential has been employed instead of the "full" LJ potential in this work.

Lastly, MD simulations are often performed in reduced units, which are chosen such that they are around the order of unity. Since the characteristic

time and length scales are very small in the conventional SI units, these values are often inconvenient to work with and might even result in calculations with numbers of the same order as the machine precision. This is undesirable from a numerical standpoint since it drastically affect the accuracy of results. To work in reduced units, each quantity is reduced with combinations of the length scale σ, the energy scale ϵ and the atomic mass M, which are taken for a reference material, typically Argon [48]. A summary of the reduced quantities used in this work can be found in Table 2.1, which can always be converted back into real units at the end of the simulation.

Quantity	**Reduced**	**S.I**
Length	$L^* = L/\sigma$	3.405×10^{-10} m
Mass	$m^* = m/M$	6.69×10^{-26} kg
Energy	$E^* = E/\epsilon$	1.65×10^{-21} J
Time	$t^* = t/\sigma\sqrt{\epsilon/M}$	2.168×10^{-12} s
Mass density	$\rho^* = \rho\ \sigma^3/M$	1678 kg/m^3
Temperature	$T^* = T\ K_b/\epsilon$	119.51 K
Pressure	$P^* = P\ \sigma^3/\epsilon$	41.79 MPa
Velocity	$\boldsymbol{v}^* = \boldsymbol{v}\ \sqrt{M/\epsilon}$	159.112 m/s
Force	$f^* = f\ \sigma/\epsilon$	4.84×10^{-12} N
Shear rate	$\dot{\gamma}^* = \dot{\gamma}\ \sigma\sqrt{M/\epsilon}$	4.61×10^{11} s^{-1}
Dynamic viscosity	$\eta^* = \eta\ \sigma^2/\sqrt{\epsilon M}$	9.06×10^{-5} Pa $\cdot$ s
Kinematic viscosity	$\nu^* = \nu/\sigma\sqrt{M/\epsilon}$	5.35×10^{-8} m^2/s
Thermal conductivity	$\lambda^* = \lambda\ \sigma^2/K_b\sqrt{M/\epsilon}$	0.019 W $\cdot$ m^{-1} $\cdot$ K^{-1}
Self-diffusion coefficient	$D^* = D/\sigma\sqrt{M/\epsilon}$	5.35×10^{-8} m^2/s
Heat capacity	$C_v^* = C_v/K_b$	1.38×10^{-23} J/K

Table 2.1: Relevant physical quantities in reduced units. The length scale σ, energy scale ϵ and the atomic mass M reference values have been taken for Argon [48].

2.3.2 Molecular Fluids and Force Fields

Lennard-Jones (LJ) fluids have traditionally been the most widely used model for atomistic simulations and even after more than fifty years, this fluid model is still used in many active areas of research [47,49–51]. Although the LJ model is complex enough to capture essential physics of some gases and simple fluids, poly-atomic compounds exhibit more complicated behaviour due to their internal molecular topology, which needs to be accounted for in the interaction model. Rotational and vibration degrees of freedom affect dramatically how energy is stored within a molecule and its response to any external field. This has a direct impact in equilibrium and non-equilibrium properties of molecular fluids.

Molecular potentials, are a set of parametrized energy functions which model intra- and inter-molecular atomic interactions. These interaction models are also known as *force-fields*. These potentials can be empirically parametrized to reproduce important experimental properties for a set of target compounds, derived from first principles calculations, or computed using a combination of the two. Force-field parameterization is a complex and time-consuming process, and traditionally, most of the force fields has been originally developed for biological applications [52]. An few examples of these potentials are AMBER [53], MM3/MM4 [54], the CHARMM [55] family and OPLS [56] among others. A comprehensive review can be found in Ponder *et al.* [57] and a more recent one focused on tribological applications in Ewen *et al.* [52] . These potentials usually do not consider bond breakage, which is crucial in modeling chemical reactions. The ReaxFF potential was developed to help bridging the gap between classical force-fields and quantum mechanical (QM) energy calculations. ReaxFF casts the empirical inter-atomic potential within a bond-order formalism, thus implicitly describing chemical bonding without the need of expensive QM computations [58].

The accuracy of any MD simulation is heavily dependent on the force field being employed, therefore, significant effort should be spent choosing and testing its performance before counting production simulations. The

functional forms of the majority of classical molecular force fields are quite similar, and in general can be described by Equation 2.3.8. Most of them include energy terms to account for bond stretching, angle bending, dihedral torsions (bonded interactions), van der Waals and electrostatic forces (non-bonded interactions). Bonded interactions are commonly computed from quadratic or similar potentials, while non-bonded terms are modelled by Lennard-Jones (LJ), for short-range weak interactions, and Coulombic potential for long-range (see Table 2.2). LJ interactions are usually cut-off at a distance no larger than 10 Å, whereas long-ranged Coulombic interactions are commonly resolved using a particle–particle, particle–mesh (PPPM) algorithms which do not perform potential truncation [59]. The functional common classical potentials can be written as

$$V = V_{\text{bond}} + V_{\text{angle}} + V_{\text{torsion}} + \sum_i \sum_{j>i} \chi_{ij} \left\{ V_{LJ} + \frac{q_i q_j}{4\pi\varepsilon_0 r_{ij}} \right\} \quad (2.3.8)$$

where the exact form for each energy term can be seen in Table 2.2. The factor χ_{ij} is known as the "fudge factor", which scales down interactions between atoms in the same molecule which are a certain number of bonds apart. In the case of OPLS, $\chi_{ij} = 0.5$ for interactions between atoms one or two bonds apart, and $\chi_{ij} = 1$ otherwise. The presence of angle bending and torsional energy terms, introduces higher order interactions, which need to be accounted appropriately on the calculation of the pressure tensor $\boldsymbol{P}$. Details on how to do this for both pairwise or many-body energy terms, and including the effects of periodic boundary conditions, can be found in [60].

In order to obtain meaningful results from NEMD hydrodynamic simulations of tribological systems, the force-field of choice needs to reproduce not only thermodynamic properties but also the rheology of the material under study. The computational expense of classical NEMD simulations is usually dominated by computing the inter-atomic forces derived from non-bonded interactions. Therefore, historically, united-atom (UA) potentials has been employed, where the non-polar hydrogens are grouped with carbon atoms to form CH, CH2 and CH3 pseudo-atoms [52]. This *coarse-graining* strategy, can speedup by up to an order of magnitude

the force evaluation in molecular dynamics, compared to all-atom (AA) [61]. However, UA force fields are known to severely underpredict viscosity for linear, long-chain alkanes (50% error for n-hexadecane) compared to experiments [52]. For such molecules, all-atom (AA) force fields are required in order to accurately reproduce viscous behaviour [52,62]. For example, the L-OPLS (AA) force field [63] is known to reproduce viscosities within 10% of error for n-hexadecane at ambient conditions but also at high-temperature–high-pressure. Therefore, for accurate viscosity prediction, AA force fields are required for longer, linear molecules, while for shorter molecules, UA force fields may be an acceptable trade-off.

Non-bonded Interactions		
Lennard-Jones	V_{LJ} (Equation 2.3.5) / V_{LJTS} (Equation 2.3.7)	
Mixing Rules	$\sigma_{ab} = \sqrt{\sigma_{aa} \cdot \sigma_{bb}}$ $\epsilon_{ab} = \sqrt{\epsilon_{aa} \cdot \epsilon_{bb}}$	(2.3.9)
Bonded Interactions		
Bond stretching	$V_{\text{bond}} = K_b \left(l - l_0\right)^2$	(2.3.10)
Angle bending	$V_{\text{angle}} = K_a \left(\theta - \theta_0\right)^2$	(2.3.11)
Torsions	$V_{\text{torsion}} = \frac{1}{2} \sum_{3}^{n=1} A_n f_n(\phi)$ $f_n(\phi) = 1 + s_n \cos(n\phi)$ $s_n = (-1)^{n-1}$	(2.3.12)

Table 2.2: Common functional form of energy terms for classical force-fields. In this case, the expressions are exact for the OPLS potential. The Lennard-Jones term can be either the full potential V_{LJ} or the the truncated and shifted version V_{LJTS}. Mixing rules are used to compute cross-type atomistic interaction and length parameters for the Lennard-Jones energy term.

2.3.3 Time Integration of Equations of Motion

In order to advance a molecular simulation in time, the atomistic equations of motion (Equation 2.3.3) have to be integrated. The time-integrator computes the positions and velocities of particles for the next time step $t + \Delta t$ based on the positions and velocities at the current step t. We want the integration scheme to be *efficient*, the fewer the number of floating point operations the better; *stable,* in the sense that it conserves the energy of the system, which is commonly directly related to the step size Δt and *accurate,* so particle's trajectories resemble the realistic ones. One of the simplest algorithms is the Verlet integrator [2] which can be derived through Taylor expansion of the particle's position vector leading to the update rule

$$\boldsymbol{r}_i(t + \Delta t) = 2\boldsymbol{r}_i(t) - \boldsymbol{r}_i(t - \Delta t) + \boldsymbol{a}_i(t)\Delta t^2 + O(\Delta t^4). \quad (2.3.13)$$

Since the velocity is not needed to update $\boldsymbol{r}_i$, it can be calculated by a central difference as

$$\boldsymbol{v}_i(t) = \frac{\boldsymbol{r}_i(t + \Delta t) - \boldsymbol{r}_i(t - \Delta t)}{2\Delta t} + O(\Delta t^2). \quad (2.3.14)$$

It can be seen that particles' positions are locally fourth order accurate whereas velocities are second order. One drawback of this method is that velocities are not available at the same time as positions. This is a weakness shared with the Leap-Frog scheme, another popular algorithm based on Verlet [2]. To overcome this limitation, the velocity Verlet was developed. In this integrator positions, velocities and accelerations are obtained at the same time. This can be achieved by propagating the velocities in two stages

1. Velocities are computed at $t + \Delta t/2$:

$$\boldsymbol{v}_i\left(t + \frac{\Delta t}{2}\right) = \boldsymbol{v}_i(t) + \boldsymbol{a}_i(t)\frac{\Delta t}{2}. \quad (2.3.15)$$

2. Positions are computed at $t + \Delta t$:

$$\boldsymbol{r}_i(t + \Delta t) = \boldsymbol{r}_i(t) + \boldsymbol{v}_i(t)\Delta t + \boldsymbol{a}_i(t)\frac{\Delta t^2}{2}. \quad (2.3.16)$$

3. Accelerations are calculated from the interaction potential as shown in Equation 2.3.3.

4. The velocities can be now computed at $t + \Delta t$ by using steps 1 and 3 as:

$$\boldsymbol{v}_i\left(t+\Delta t\right) = \boldsymbol{v}_i\left(t+\frac{\Delta t}{2}\right) + \boldsymbol{a}_i\left(t+\Delta t\right)\frac{\Delta t}{2}. \tag{2.3.17}$$

This is one of the most common algorithms utilized in molecular dynamics since, similarly to the Verlet algorithm, it preserves time-reversal of the Newton's equations of motion and preserves energy, provided accelerations are derived from a potential. This type of integrator is called *symplectic* or phase space conserving [64]. The algorithms employed in this work are slightly modified versions of the velocity Verlet which are included with `LAMMPS` [35] molecular dynamics engine.

2.3.4 Bulk Shear-NEMD Simulations and Viscosity Calculation

In fluid dynamics simulations, the physical phenomena of interest is inherently out of equilibrium. This is of paramount importance when the properties and boundary conditions in the fluid vary depending on how far the system is from equilibrium. An example are shear-rate dependent boundary conditions [65] or non-Newtonian effects like shear thinning in polymeric fluids [66]. In the last few decades, there has been a rise in the development of non-equilibrium molecular dynamics (NEMD) methods to study this type of systems which have become a very active field of research in the fields of nanotribology or nano-fluidics [1]. Furthermore, NEMD simulations indeed form the majority of molecular simulation methods in continuum-molecular hybrid schemes, which is the central topic of this thesis. For the particular case of computing transport coefficients from MD, the Green-Kubo formulas can be found in Section 2.2.5. However, these relations are only valid for equilibrium ensembles. Evans and Morriss [11] argue that a non-equilibrium system requires the addition of a perturbing field to its

statistical mechanical description. This perturbing field will always do work to drive the system out of equilibrium. The second law of thermodynamics tells us doing work on the system always generates heat, but this heat must be removed in order for the system to reach a steady-state. Therefore, a suitable thermostatting strategy it is always needed in order to remove the excess of heat created by the perturbing field [11].

In this section we are particularly interested in NEMD systems where the perturbing field induces shear, creating a velocity profile across the fluid. Different methods exist to drive the system to a steady-state under shear. Within the most popular ones are the Muller-Plathe algorithm [67], the SLLOD equations of motion [68] combined with Lees-Edwards boundary conditions [69] and confined systems where the fluid is sheared by sliding atomistic walls [52] as the ones discussed in Section 2.1. Once the steady-state is reached, different quantities can be computed. For instance, in order to compute shear viscosity, these three methods rely on attaining a linear steady-state velocity profile. Other NEMD viscosity methods based on imposing non-linear velocity profiles do equally exist, although they are less established. For example, in the momentum impulse relaxation method, the viscosity is measured from the decay of an imposed initial Gaussian velocity profile [70].

In the Muller-Plathe method, the liquid is divided into an even number of slabs across the y direction and momentum is exchanged between pairs of atoms inside the outermost top and bottom slabs and the middle slab. This is accomplished by finding the atoms with highest momentum but opposite direction to the the desired slab movement. For example, in a slab moving in the $-x$ direction, the atom with the largest momentum component in the $+x$ direction is selected for exchange. If both atoms have the same mass, the momentum swap conserves both linear momentum and kinetic energy. The potential energy is equally conserved since atom positions are not changed [67]. As a consequence of this exchange of momentum, a velocity profile builds up over time. Once the linear regime is attained, as shown in Figure 2.2, the viscosity of the fluid at the shear rate $\dot{\gamma} = \partial \langle v_x \rangle / \partial y$, can be computed from the definition of the Newtonian viscosity as

$$\eta = -\frac{P_x \dot{\gamma}^{-1}}{2\Delta t A_{xz}}, \tag{2.3.18}$$

where P_x is the total accumulated momentum exchanged between slabs within a time Δt, A_{xz} is the cross section area of the slabs and $\langle v_x \rangle$ is the mean velocity field computed at the steady-state. A drawback of the Muller-Plathe algorithm is that the target shear rate at which to calculate the viscosity cannot be easily controlled.

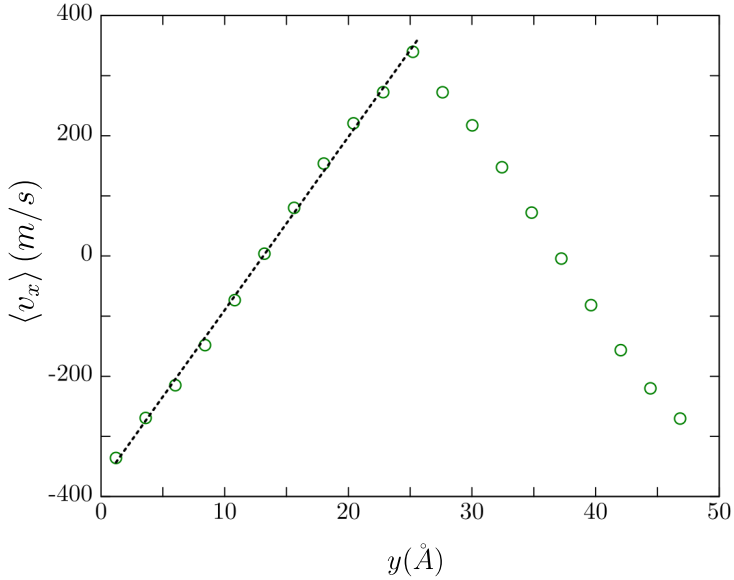


Figure 2.2: Velocity profile obtained from a Muller-Plathe simulation. A fitting line allows to calculate the velocity gradient $\partial \langle \Delta v_x \rangle / \partial y$ necessary to compute the Newtonian viscosity from Equation 2.3.18.

A linear velocity profile can be also generated by the so called Lees-Edwards periodic boundary conditions [69]. The use of the SLLOD equations of motion along with Lees-Edwards periodic boundary conditions (PBCs), guarantees a constant velocity gradient in the steady-state when the Reynolds number is low. The thermostatted SLLOD equations are defined as [11]

$$\begin{aligned} \dot{\boldsymbol{r}}_i &= \frac{\bar{\boldsymbol{p}}_i}{m} + \boldsymbol{r}_i \cdot \nabla \mathbf{u} \\ \dot{\bar{\boldsymbol{p}}}_i &= \boldsymbol{F}_i^{\phi} - \bar{\boldsymbol{p}}_i \cdot \nabla \mathbf{u} - \alpha \bar{\boldsymbol{p}}_i, \end{aligned} \tag{2.3.19}$$

where $\boldsymbol{r}_i$ is the particle position, $\bar{\boldsymbol{p}}_{\boldsymbol{i}}$ is the *peculiar* particle momenta (with respect to the velocity profile [71]), $\mathbf{u}$ is the streaming velocity, $\boldsymbol{F}_i^{\phi}$ is the force on atom i due to inter-atomic interactions with the rest of atoms and α is the is the constraint parameter which fixes the thermal kinetic energy (via Gauss's principle) and can be expressed as [11]

$$\alpha = \frac{\sum \left(\boldsymbol{F}_i^{\phi} \cdot \bar{\boldsymbol{p}}_i - \bar{\boldsymbol{p}}_i \cdot \nabla \mathbf{u} \cdot \bar{\boldsymbol{p}}_i \right)}{\sum \bar{\boldsymbol{p}}_i^2}. \tag{2.3.20}$$

In order to compute the thermostat relaxation parameter α, the velocity profile is either assumed, leading to the *profile biased* family of thermostats (PBTs) or computed by sampling directly the velocity field, in which case are known as *profile unbiased* thermostats (PUTs) [11]. PUTs are subject to statistical noise because the streaming velocity profile is measured statistically from the fluctuating velocities of a finite number of particles.

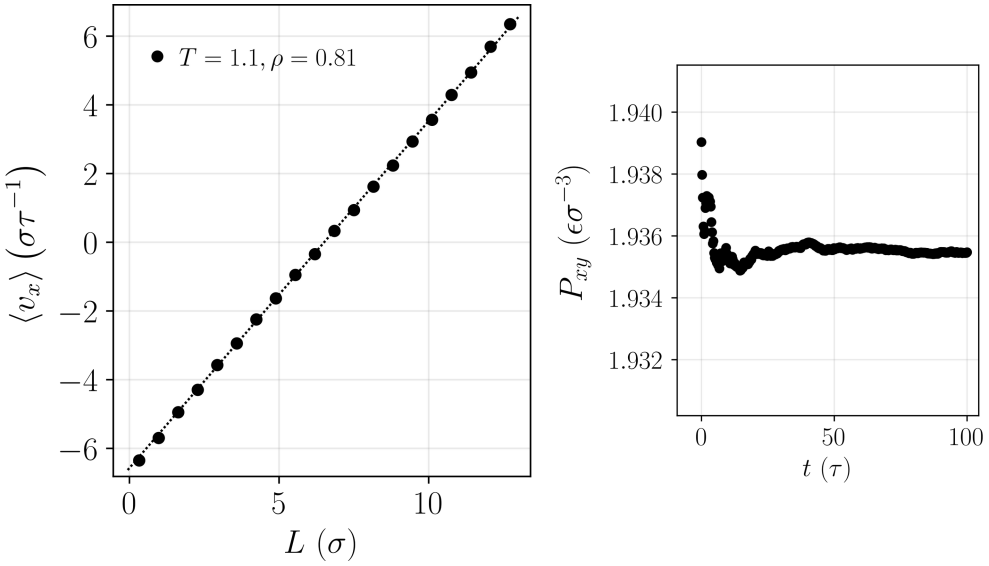


Figure 2.3: Steady state linear velocity profile for a LJ fluid at $\dot{\gamma} = \partial \langle \Delta v_x \rangle / \partial y \approx 1.0$ and accumulated value of the shear component of the pressure tensor P_{xy} computed from Equation 2.2.25. Dotted line is the fitting to the velocity profile.

The viscosity of the fluid can be computed from the slope of the linear steady-state velocity profile and the atomistic shear stress P_{xy} (see Figure 2.3) by using the constitutive relation for the Newtonian viscosity

$$\eta = \frac{\partial \langle v_x \rangle / \partial y}{P_{xy}}. \tag{2.3.21}$$

Lastly, confined NEMD simulations with moving solid surfaces can be used as well to compute viscosity using Equation 2.3.21. These simulations have periodic boundary conditions applied just in stream-wise and span-wise directions. They have the disadvantage of the system being measured is a confined liquid and not bulk as in the previous two methods described.

2.4 Fluid Dynamics and CFD

Fluid dynamics is a sub-discipline of fluid mechanics that describes the flow of fluids liquids and gases. Three fundamental quantities can be transported by a fluid: *mass*, *momentum* and *energy*. The purpose of this branch of physics is to describe the time-evolution of these three quantities. Consequently, three conservation laws are employed in order to solve fluid dynamics problems. The conservation laws may be applied to a *control volume* V. This is a discrete volume in space through which the fluid flow is considered and it is surrounded by a boundary Ω. Section 2.4.1 is devoted to discuss the three conservation laws written in their differential form, where considered V and Ω are infinitesimally small.

Often, fluid dynamics problems are too complicated to be solved analytically and approximate methods are necessary. The field of numerical analysis, dedicated to tackle fluid dynamics problems, is known as Computational Fluid Dynamics (CFD). This is not only an active area of research but also a well-established engineering modeling tool widely used in the industry. The literature on the field of fluid dynamics and CFD is vast. In this section, it has been strictly reviewed the theory which has been directly used as part of the MD-CFD coupling methodology. For further information, the reader is referred to specialized texts like [6,26].

Particularly, the content of this section is largely based on the specialized `OpenFOAM` [72] book from Holzmann [73]. `OpenFOAM` is a popular open source software framework which provides many configurable CFD solvers based on the Finite Volume Method (FVM) and it has been used in this work to implement the CFD domain within the MD-CFD coupled scheme. The advantage of using a package like `OpenFOAM` is that the solvers it provides are extensively tested by the community, they can be executed in parallel using domain decomposition by using MPI [9] and can be extended, as the source code is available and free to use. Regarding the numerical aspects of the FVM implementation used by `OpenFOAM` (operator discretisation, time integration, linear solvers, preconditioners, interpolators, etc.) are not reviewed here, and the reader is directed to Jasak [74] and Ferziger & Peric [6]. These algorithms have not been implemented as part of this work but simply adapted to use them in coupling. Furthermore, as discussed later, CFD is not the critical technique in the MD-CFD coupled scheme, and simulated flows, from the CFD perspective are very simple. Lastly, Section 2.4.2 contains analytical solutions for three different parallel flows of interest.

2.4.1 Fluid Dynamics Equations

In general the differential form of a conservation equation with Neumann boundary conditions, for a quantity $\boldsymbol{\phi}$, can be written for a volume V as

$$\begin{aligned} \frac{\partial}{\partial t}\boldsymbol{\phi} + \nabla \cdot \boldsymbol{J}_{\phi} = \boldsymbol{S}_{\phi}, \quad \boldsymbol{x} \in V \\ \boldsymbol{\phi} = \boldsymbol{f}(\boldsymbol{x}, t), \quad \boldsymbol{x} \in \Omega \end{aligned}, \tag{2.4.1}$$

where, $\boldsymbol{J}_{\phi}$ is the flux of $\boldsymbol{\phi}$, $\boldsymbol{S}_{\phi}$ is a source or sink term for the quantity $\boldsymbol{\phi}$ and f is an arbitrary function prescribing the value of $\boldsymbol{\phi}$ over the enclosing domain boundary Ω (boundary condition). This expression is general and valid for any scalar or vectorial quantity which is conserved through time. A particularization of Equation 2.4.1 can be constructed for mass, momentum and energy, given the quantity fluxes defined in Table 2.3.

Quantity (ϕ)	**Flux** ($\boldsymbol{J}_\phi$)
Mass (ρ)	$\rho\boldsymbol{u}$ (vector)
Momentum ($\rho\boldsymbol{u}$)	$\rho\boldsymbol{u}\boldsymbol{u} + \boldsymbol{\sigma}$ (2^{nd} order tensor)
Energy (ρe)	$\rho e\boldsymbol{u} + \boldsymbol{\sigma}\cdot\boldsymbol{u} + \boldsymbol{q}$ (vector)

Table 2.3: Fluxes for mass, momentum and energy density. The $\rho\boldsymbol{u}\boldsymbol{\phi}$ term represents convection of $\boldsymbol{\phi}$. The $\boldsymbol{\sigma}$ symbol is the local stress tensor defined in Equation 2.4.5 and $\boldsymbol{q}$ is the heat flux as defined in Equation 2.4.10.

The mass conservation equation, commonly known the continuity equation, can be written in the absence of mass sources (e.g closed system) by substituting $\boldsymbol{\phi}$ with ρ in Equation 2.4.1 as

$$\frac{\partial}{\partial t}\rho = \nabla\cdot(\rho\boldsymbol{u}), \tag{2.4.2}$$

which for an incompressible fluid reduces to

$$\nabla\cdot\boldsymbol{u} = 0. \tag{2.4.3}$$

Similarly, the momentum equation can be constructed by replacing $\boldsymbol{\phi}$ with $\rho\boldsymbol{u}$. In absence of body forces acting on the volume element, the source term $\boldsymbol{S}_\phi$, can be neglected giving the equation

$$\frac{\partial}{\partial t}\rho\boldsymbol{u} = -\nabla\cdot(\rho\boldsymbol{u}\boldsymbol{u}) + \nabla\cdot\boldsymbol{\sigma}, \tag{2.4.4}$$

where $\boldsymbol{\sigma}$ is the stress tensor which can be written in terms of its hydrostatic and deviatoric components as

$$\boldsymbol{\sigma} = -p\boldsymbol{I} + \boldsymbol{\tau}, \tag{2.4.5}$$

where p is the scalar *hydrostatic pressure* and $\boldsymbol{\tau}$ is the *viscous stress tensor*. For a general compressible Newtonian fluid the constitutive relation for $\boldsymbol{\tau}$ can be written as

$$\boldsymbol{\tau} \equiv \eta\left[\nabla\boldsymbol{u} + (\nabla\boldsymbol{u})^T - \nabla\cdot\boldsymbol{u}\right] - \lambda\nabla\cdot\boldsymbol{u}, \tag{2.4.6}$$

where η is the *dynamic shear viscosity* [75] and λ is the *bulk viscosity* (dilatational) [50]. If the fluid is considered incompressible, Equation 2.4.3 apply and $\boldsymbol{\tau}$ becomes the *shear stress tensor* which can be rewritten as

$$\boldsymbol{\tau} = \eta \left[\nabla \boldsymbol{u} + (\nabla \boldsymbol{u})^T \right]. \tag{2.4.7}$$

By expanding the convective term in Equation 2.4.4 and combining it with Equations 2.4.5 and 2.4.7 we arrive to the well-known Navier-Stokes equation for momentum transport

$$\frac{\partial \boldsymbol{u}}{\partial t} + (\boldsymbol{u} \cdot \nabla)\, \boldsymbol{u} = -\frac{1}{\rho} \nabla p + \nu \nabla^2 \boldsymbol{u}, \tag{2.4.8}$$

where $\nu = \eta/\rho$ is the momentum diffusion coefficient known as the *kinematic viscosity*.

The total energy of the fluid can be written as the sum of the internal Ψ and kinetic energy e_k, $e = \Psi + e_k = \Psi + |\boldsymbol{u}|^2/2$. The energy equation is written in its conservative form by replacing in Equation 2.4.1, $\boldsymbol{\phi}$ with ρe and substituting $\boldsymbol{J}_\phi$ with the energy flux from Table 2.3. We assume no source terms $\boldsymbol{S}_\phi$ are present and write

$$\begin{aligned} \frac{\partial}{\partial t} \left(\rho \Psi + \frac{1}{2} |\boldsymbol{u}|^2 \right) = &- \nabla \cdot \left(\rho \boldsymbol{u} \left[\Psi + \frac{1}{2} |\boldsymbol{u}|^2 \right] \right), \\ &+ \nabla \cdot (\boldsymbol{\sigma} \cdot \boldsymbol{u}) + \nabla \cdot \boldsymbol{q} \end{aligned} \tag{2.4.9}$$

where $\boldsymbol{q}$ is the heat flux which is in most cases given by the Fourier's Law [76] as

$$\boldsymbol{q} = -\lambda \nabla T, \tag{2.4.10}$$

with λ being the thermal conductivity of the fluid and T the local temperature field. The term $\nabla \cdot (\boldsymbol{\sigma} \cdot \boldsymbol{u})$ in Equation 2.4.9 can be expanded using Equation 2.4.5 to give

$$\nabla \cdot (\boldsymbol{\sigma} \cdot \boldsymbol{u}) = -\nabla \cdot (p \boldsymbol{I} \cdot \boldsymbol{u}) + \nabla \cdot (\boldsymbol{\tau} \cdot \boldsymbol{u}). \tag{2.4.11}$$

		Changes e_k	Changes Ψ	
		Pressure work	**Expansion**	
Reversible	$\nabla \cdot (p\boldsymbol{I} \cdot \boldsymbol{u}) = \boldsymbol{u} \cdot \nabla p + p\nabla \cdot \boldsymbol{u}$			(2.4.12)
		Internal friction	**Viscous dissipation**	
Irreversible	$\nabla \cdot (\boldsymbol{\tau} \cdot \boldsymbol{u}) = \boldsymbol{u} \cdot (\nabla \cdot \boldsymbol{\tau}) + \boldsymbol{\tau} : \nabla \boldsymbol{u}$			(2.4.13)

Table 2.4: Table with the contributions to the energy equation due to the local stress tensor $\boldsymbol{\sigma}$ through the $\nabla \cdot (\boldsymbol{\sigma} \cdot \boldsymbol{u})$ term. Contributions can be categorized as reversible (related to p) and irreversible (related to $\boldsymbol{\tau}$), if energy is transformed into heat.

The kinetic energy equation can be derived by taking the dot product of the momentum equation (Equation 2.4.4) with the velocity field $\boldsymbol{u}$, which after some algebraic manipulation and neglecting body forces, we get the expression

$$\frac{\partial}{\partial t}\left(\rho \frac{1}{2}\left|\boldsymbol{u}\right|^2\right) + \nabla \cdot \left(\rho \boldsymbol{u} \frac{1}{2}\left|\boldsymbol{u}\right|^2\right) = -\boldsymbol{u} \cdot \nabla p + \boldsymbol{u} \cdot (\nabla \cdot \boldsymbol{\tau}). \quad (2.4.14)$$

By substituting Equation 2.4.11 into Equation 2.4.9 (total energy equation) and subtracting Equation 2.4.14, we arrive to the internal energy equation

$$\frac{\partial}{\partial t}(\rho\Psi) + \nabla \cdot (\rho \boldsymbol{u}\Psi) = -\nabla \cdot \boldsymbol{q} - p\nabla \cdot \boldsymbol{u} + \boldsymbol{\tau} : \nabla \boldsymbol{u}. \quad (2.4.15)$$

The internal energy is a thermodynamic state variable that is seldom used in practice. A more commonly used quantity is enthalpy h, which is related to the internal energy as

$$\Psi = h - \frac{p}{\rho}, \quad (2.4.16)$$

which by inserting it into Equation 2.4.15, the energy equation can be rewritten in terms of h as

$$\rho \frac{\partial h}{\partial t} + \rho \boldsymbol{u} \cdot \nabla h = -\nabla \cdot \boldsymbol{q} + \frac{\partial p}{\partial t} + \boldsymbol{u} \cdot \nabla p + \boldsymbol{\tau} : \nabla \boldsymbol{u}. \quad (2.4.17)$$

For an incompressible fluid $dh = C_p dT$, where C_p is the specific heat capacity at constant pressure. By substituting this thermodynamic relation and Equation 2.4.10 into Equation 2.4.17 we arrive to the temperature or thermal equation, which is yet another way to express conservation of energy

$$\rho C_p \frac{\partial T}{\partial t} + \rho C_p \boldsymbol{u} \cdot \nabla T = \nabla \cdot (\kappa \nabla T) + \boldsymbol{\tau} : \nabla \boldsymbol{u}. \tag{2.4.18}$$

2.4.2 Analytical Parallel Flows

In this work we have limited ourselves to the study of non-turbulent parallel nanoflows similar to the ones described in Section 2.1, which are either 1D pressure-driven or shear-driven [26]. The general setup of the system is a slab of fluid in between two infinite parallel plates separated by a length L as shown in Figure 2.4. These type of flows are commonly found in the NEMD nanofluidics literature [77,78] for the study surface-liquid inter-facial phenomena like adsorption or slip [79–81]. Despite their apparent simplicity, parallel flows are of significant engineering interest, particularly in the study of thin-film lubrication layers found in between mechanical components withstanding shear (bearings, reciprocating contacts, etc.) or small tubular structures in microfluidics [82].

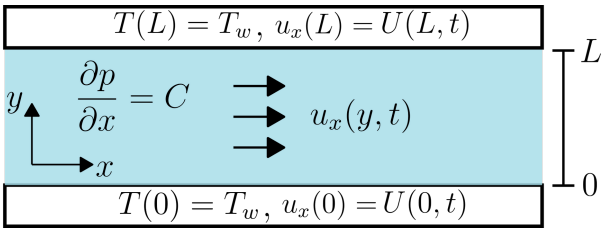


Figure 2.4: Diagram of a slab of fluid between two infinite parallel plates with an internal constant pressure gradient and temperature and velocity boundary conditions.

For a Newtonian incompressible fluid flowing just along the x direction, if convection can be neglected, the momentum equation (Equation 2.4.8) takes the 1D simplified form (see derivation in [26])

$$\frac{\partial u_x}{\partial t} = \frac{1}{\rho}\frac{\partial p}{\partial x} + \nu\frac{\partial^2 u_x}{\partial y^2}, \tag{2.4.19}$$

where the first and second term on the right-hand-side are, respectively, responsible to the change of momentum due to the presence of a pressure gradient along x direction and diffusion. Similarly, the thermal equation (Equation 2.4.18) can be simplified to (see derivation in [26])

$$\rho C_p \frac{\partial T}{\partial t} = \eta\left(\frac{\partial u_x}{\partial y}\right)^2 + \frac{\partial}{\partial y}\left(\lambda\frac{\partial T}{\partial y}\right), \tag{2.4.20}$$

in the absence of convection and considering viscous dissipation to be significant.

There are three widely studied types of flows which originate from Equation 2.4.19 by imposing three different sets of boundary conditions: the Couette flow (wall-driven), the Poiseuille flow (pressure-driven) and the Rayleigh-Stokes flow (wall-driven). All of them are simple enough to be described by analytical expressions (Equations 2.4.21, 2.4.26 and 2.4.31 respectively), hence their popularity as validation models. Therefore, these flows has been employed in Section 5.3.5 to validate the coupling MD-CFD framework by employing the Lennard-Jones fluid model. The analytical expressions given below are derived for the setup depicted in Figure 2.4 with both walls (plates) at temperature T_w and the fluid initially at rest and in thermal equilibrium with the walls.

The Couette flow is generated by a moving wall (top plate in Figure 2.4) at a constant velocity U_0 and imposing no-slip boundary conditions $u_x(L) = U_0,\ u_x(0) = 0$ and a zero pressure gradient $\frac{\partial p}{\partial x} = 0$ with initial condition $u_x(y,0) = 0,\ 0 < y < L$. The exact time-evolution of the velocity field in the x direction can be expressed then as (see [26,76] for a derivation)

$$\begin{aligned} u_x(y,t) &= U_0\frac{y}{L} - \frac{2U_0}{\pi}\sum_{n=1}^{\infty} C_n(y,t), \\ C_n(y,t) &= \frac{1}{n}e^{-n^2\pi^2\nu t/L^2}\sin\left[n\pi\left(1-\frac{y}{L}\right)\right], \end{aligned} \tag{2.4.21}$$

and the steady-state solution when $t \to \infty$ reduces to a linear velocity profile

$$u_x(y, t \to \infty) = \frac{U_0}{L} y. \tag{2.4.22}$$

It is also of interest to find the steady-state temperature solution for each type of flow in order to estimate the maximum temperature attained in the fluid for boundary conditions $T(0) = T(L) = T_w$ and initial condition $T(y, 0) = T_w,\ 0 < y < L$. For a derivation of time-dependent temperature expression see Pai [83]. The expression giving the temperature profile in the Couette's steady-state can be derived by solving Equation 2.4.19 with $\partial T/\partial t = 0$ (see derivation in [84])

$$T\left(y\ , t \to \infty\right) = \frac{\eta U_0^2}{2\lambda}\left(\frac{y}{L} - \frac{y^2}{L^2}\right) + T_w, \tag{2.4.23}$$

and the maximum temperature occurring at the center of the channel $y = L/2$ (parabolic profile)

$$T_{max}\left(y = L/2, t = \infty\right) = \frac{\eta U_0^2}{8\lambda} + T_w, \tag{2.4.24}$$

which can be used to determine the quasi-isothermal condition, useful to setup coupled simulations, as

$$\frac{\eta U_0^2}{8\lambda T_w} \ll 1. \tag{2.4.25}$$

This condition resembles the expression for the Nahme-Griffith (Na) number, which determines when viscous heating effects can influence the fluid's viscosity (for $0.1 < \mathrm{Na} < 0.5$ is not affected) [85].

Secondly, the Poiseuille flow, is obtained by applying an non-constant pressure gradient $\partial p/\partial x = C$ within the fluid, with steady no-slip walls $u_x(L) = u_x(0) = 0$ (see diagram in Figure 2.4). The analytical time-dependent expression for the velocity field has been taken from [86] for a shifted domain $-L/2 < y < L/2$ in order to simplify the mathematics

$$\begin{aligned} u_x\left(y, t\right) &= \frac{|\boldsymbol{F}|}{2\nu}\left(y^2 - d^2\right) + \sum_{\infty}^{n=0} \frac{16\left(-1\right)^n d^2\,|\boldsymbol{F}|}{\nu\pi^3\left(2n+1\right)^3} C_n(y, t), \\ C_n(y, t) &= \cos\left[\frac{\left(2n+1\right)\pi y}{2d}\right] \exp\left[-\frac{\left(2n+1\right)^2 \pi^2 \nu t}{4d^2}\right], \end{aligned} \tag{2.4.26}$$

where $d = L/2$, λ is the thermal conductivity and $|\boldsymbol{F}| = \frac{1}{\rho}\partial p/\partial x$, which is a force per unit of mass. The steady-state velocity is then found to be

$$u_x(y, t \to \infty) = \frac{1}{2\nu} |\boldsymbol{F}| \left(y^2 - d^2\right), \tag{2.4.27}$$

which depends on the viscosity of the fluid, contrary to the solution for the steady-state Couette flow given by Equation 2.4.22. Similarly, the steady-state temperature profile can be obtained by solving Equation 2.4.19 as before (see derivation in [87])

$$T(y, t \to \infty) = \frac{1}{12\eta\lambda} |\boldsymbol{F}|^2 \left(d^4 - y^4\right) + T_w, \tag{2.4.28}$$

with the maximum temperature also found at the center of the domain $y = 0$

$$T_{max}\left(y = 0, t \to \infty\right) = \frac{|\boldsymbol{F}|^2 d^4}{12\eta\lambda} + T_w, \tag{2.4.29}$$

which can be used to write the quasi-isothermal condition for the Poiseuille flow as

$$\frac{|\boldsymbol{F}|^2 d^4}{12\eta\lambda T_w} \ll 1. \tag{2.4.30}$$

Lastly, the Rayleigh-Stokes flow is defined for an unbounded domain where $u_x(y \to \infty) = 0$ and generated by an oscillating plate with no-slip taking place between the fluid and the wall $u_x(0) = U_0 \cos(\omega t)$. No pressure gradient is present $\partial p/\partial x = 0$. The analytical solution of Equation 2.4.19 for this set of boundary conditions is [76]

$$u(y, t) = U_0 e^{-\sqrt{\omega/2\nu}y} \cos\left[\sqrt{\frac{\omega}{2\nu}}y - \omega t\right]. \tag{2.4.31}$$

For practical purposes, this expression is not valid if the system is bounded, which is true if the condition $L/\delta > 1$ is not fulfilled. The symbol δ is the thickness of the oscillating layer, defined as the position were $u/U_0 = e_{tol}$ and can be computed from $e^{-\delta\sqrt{\omega/(2\nu)}} = e_{tol}$ (a sensible value

would be $e_{tol} = 0.01$). In that case, an analytical expression derived in [76] can be employed to get the correct time-dependent solution as

$$\begin{aligned} u_x(y,t) = u_0(t) + \frac{u_1(t) - u_0(t)}{L}\, y + \\ \sum_{n=1}^{n=\infty} \left(\int_0^t e^{-\nu\lambda_n^2 (t-\tau)} \hat{S}_n(\tau)\, d\tau + C_n e^{-\nu\lambda_n^2 t} \right) \sin\lambda_n\, y, \\ \hat{S}_n = 2\frac{\dot{u}_1(t)(-1)^n - \dot{u}_0(t)}{\pi n}, \\ C_n = 2\frac{u_1(0)(-1)^n - u_0(0)}{\pi n}, \end{aligned} \tag{2.4.32}$$

where $\lambda^n = n\pi/L$, $u_0 \equiv u_x(0,t) = U_0 \cos(\omega t)$ and $u_1 \equiv u_x(L,t) = 0$ for the Stokes-Rayleigh flow. The inner integral can be either solved analytically or simply integrated numerically using e.g. a simple trapezoidal quadrature scheme [88]. For this type of flow, the quasi-isothermal condition cannot be found from the steady-state since the solution is not convergent as $t \to \infty$. Therefore, Equation 2.4.25 has been taken as an approximation which would be exact in the limit of $\omega \to 0$ (infinite oscillating period) and Equation 2.4.24 would represent an upper bound for the maximum temperature T_{max} within the fluid.

2.5 Multiscale Domain Decomposition for Fluids

In Section 2.1 it was made evident that NEMD methods are of paramount importance to the understanding of tribological mechanisms in nano-scale lubrication. However, computational fluid dynamics (CFD) remains the *de facto* method for modelling hydrodynamic lubrication. By solving the Navier–Stokes equations or, most commonly, their thin film approximation (the Reynolds equation) [89], a wide range of tribological problems can be studied. As previously mentioned, continuum simulation of tribological systems requires accurate constitutive relations, transport coefficients and

boundary conditions for the system of interest. In some cases, this information may not be readily available nor even measurable with the current experimental technology. In addition, many physical processes that can be modelled explicitly at the atomic level (e.g. adsorption, phase changes, slip, etc.) cannot be easily incorporated into continuum methods. On the other hand, these aspects arise naturally in atomistic simulations from the inter-atomic interactions defined in the model [24]. Why then not pervasively use NEMD? The big drawback of atomistic simulations is their enormous computational cost. Despite the huge increase in available computer power, since the advent of high-performance computing (HPC), length scales over a few nanometers and timescales beyond nanoseconds are currently unfeasible to simulate. This limitation can be found in the type of systems reviewed in Section 2.1, where lubricant films are not thicker than a few tens of nanometers. In the case where the scales of interests are beyond our computational power, a multi-scale method can, for certain cases, be employed as a trade-off between computational efficiency and accuracy. A comprehensive overview of tribological modelling at different scales, including multi-scale techniques, can be found in Vakis *et al.* [89].

A multi-scale method is a combination of two or more methods which operate at distinct scales. It is useful to categorize multi-scale methods into three types [90]:

1. *Parametrization* of macroscopic models using data from microscopic scales, see Figure 2.5.a. Constitutive relations like stress-velocity in fluids can be computed and tabulated from microscopic simulations and later incorporated into a macroscopic model to describe e.g. the rheological behaviour of fluids. This strategy is also feasible to obtain non-trivial boundary conditions (e.g. slip condition) for PDE-based macroscopic models, if time-scales are decoupled between models [91,92].
2. *Heterogeneous Multi-scale Method* (HMM) [93], also known as embedded method, is a general strategy for tackling multi-scale problems, see Figure 2.5.b. In this method, the macroscopic solver is

used throughout the whole simulation domain and microscopic solvers perform "local refinement" on demand for the macroscopic model parameters. An example is computing viscosity using the SLLOD-NEMD algorithm [94] for a particular shear rate local to the mesh nodes in a CFD solver, which is known as a *point-wise coupling* method [95]. An alternative similar strategy, referred to as *field-wise coupling*, instead of performing coupling at mesh nodes, does it over what are defined as distributed heterogeneous fields by Borg *et al.* [96]. The HMM methodology is in general useful when the parameter space is too big to pre-compute and tabulate the data with the parametrization strategy.

3. Methods dealing with *isolated defects*, see Figure 2.5.c. In this context, a defect is a region of the simulation domain where the macroscopic model is inaccurate. Several methods can be found in this category [90], among them, the most relevant to this work are continuum-NEMD coupling strategies through domain decomposition (DD), which are comprehensively reviewed in Bian *et al.* [97]. For the case of fluids, the defect within the continuum domain is often at an interface between two flow regimes where the continuum becomes invalid, for example, a solid-liquid or liquid-vapour interface or a moving shock wave between fast and slow moving fluid. A recent example of a MD-CFD domain decomposition study where thermal and momentum transport is considered for a Lennard-Jones fluid, can be found in [98]. For a study case employing poly-atomic molecules with DD coupling see [99].

The rest of the section briefly reviews the domain decomposition multiscale technique for the particular case of coupling MD and CFD methods. This is the strategy chosen in this thesis to study molecular systems under shear akin to the ones described in Section 2.1. In these systems the solid-fluid interface is treated as the isolated defect and it is modelled by employing molecular dynamics. This section is complemented with the contents of Chater 3, where a more exhaustive review of the theory supporting the particular implementation of the method can be found.

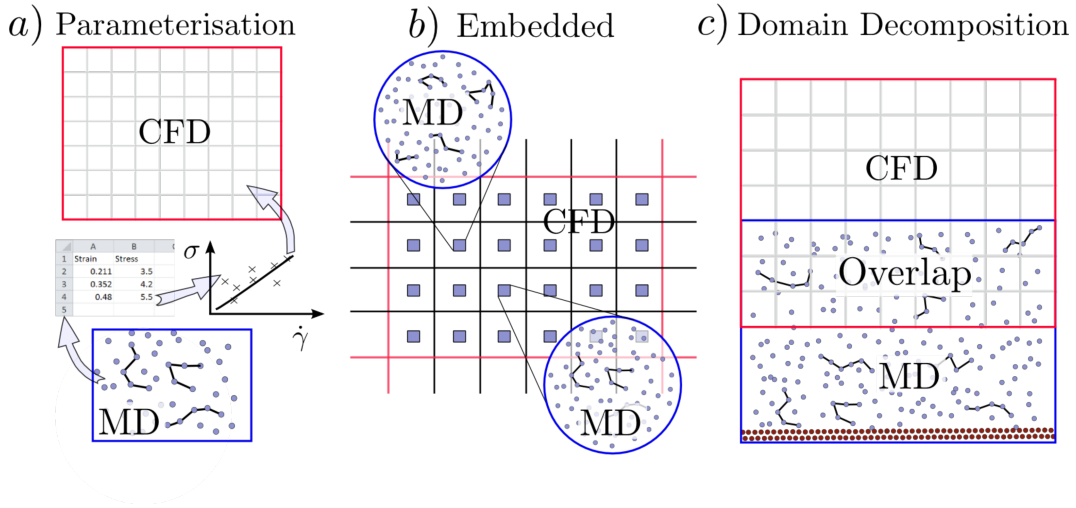


Figure 2.5: Figure depicting three types of coupling multiscale techniques by a) parametrized simulations to generate data by running MD simulations to create lookup tables or fit to physical models; b) embedded coupling using MD simulations on-the-fly driven by the local CFD field, with resulting results passed back to the CFD; c) domain decomposition where both CFD and MD share a single simulation domain, exchanging information concurrently at the overlapping boundaries. Reproduced with permission from [1]. Copyright Springer, 2020.

The first example of domain decomposition for fluid dynamics is given in the work of O'Connell & Thompson [100], which established the key features of this form of two-way concurrent coupling. Many of these features have a historical precedent from techniques used in solid mechanics for coupling of particles and continuum, dating back to the 1970s [101]. The simulation domain is decomposed into two subdomains overlapping in a region called the hybrid solution interface (HSI). The bottom subdomain, where the solid-interface is contained, is modelled atomistically. In the top subdomain, CFD is employed to model the bulk of the fluid. The left and right boundaries are set to periodic boundaries in both the CFD and MD domains. In the MD subdomain, a force is applied within the constrained region in order to guide the molecular fluid to a physical state which agrees with the CFD subdomain. In the region close to the bottom of the CFD, the data from the MD system is accumulated and averaged to provide the CFD

with boundary conditions. Figure 2.6 shows a schematic representation of a Couette flow using domain decomposition. The challenges for domain decomposition coupling include the following:

1. A termination of the MD subdomain. A restraint mechanism is required at the boundary of the molecular region to prevent molecules escaping. This can be a generic force [100,102], one based on a previous simulation or calculated from the radial distribution function [103]. Another method uses a reflective boundary with a correction for density fluctuations based on exponential smoothing of the unbalanced forces caused by the missing fluid at the termination boundary [104]. Alternatively, a buffer zone of molecules can be used as described in [105,106]. This matter is further addressed in Section 3.2.3, where the reflective boundary with density correction method is described in detail.
2. A method of inserting and removing molecules is required to match the mass flux from the continuum. For simple molecules, the most common method is a steepest descent energy search approach called USHER [107]. For complex molecules, energy insertion location can be nearly impossible to find. Therefore, techniques exist where the molecule insertion is facilitated by dynamically changing the coarse-graining level of the molecules, depending on the distance to the HSI [108]. This point is of no concern for this work, since mass fluxes are always zero for parallel flows of incompressible fluids (the type flows considered in this work) and it is mentioned here just for completeness.
3. A procedure for averaging within the MD region in order to obtain continuum boundary conditions (bottom "CFD Boundary" region in Figure 2.6). This requires summing, over time and space, atomistic variables to establish averaged values at discrete locations [2,25]. The removal of statistical noise is a key issue in this process [109], which is directly influenced by the ratio of timesteps used in the two solvers and size of averaged region in terms of statistics [110]. The domain decomposition coupling literature is divided into *state coupling* (mass,

momentum, energy) and *flux coupling* (mass flux, stress, energy flux) [7]. State coupling has been used in this work to provide boundary conditions to the CFD since noise in the calculation of fluxes can hinder their applicability [110]. For a description of the averaging procedure followed in this work see Section 3.4.4.

4. A constraint must be applied to the molecular region to match properties to the continuum ("Constrained Region" in Figure 2.6), which can be done in terms of state variables or fluxes [7]. This process is the most involved part of the coupling strategy since there is no unique way to do so, as many extra degrees of freedom on particles are under-determined [97]. For state variables, the usual strategy is to apply constrained dynamics on each particle, by using the variational principle formulation of mechanics [111] within the constrained region. For example, the velocity state variable can be constrained as

$$\frac{1}{N}\sum_{i=1}^{N} \boldsymbol{v}_i = \boldsymbol{u}_c, \tag{2.5.1}$$

where N is the instantaneous number of particles, $\boldsymbol{v}_i$ is the particle velocity (inside the MD domain) and $\boldsymbol{u}_c$ is the continuum value of the velocity within the constrained region (provided by the CFD domain). For flux coupling, fluxes obtained from the continuum solution need to be spread (few degrees of freedom) to the particles within the constrained region (many degrees of freedom). Considering an unidirectional flow (in the x direction) for illustrative purposes, the equation of motion for each particle can be written a

$$\begin{aligned} \dot{v}_i &= \frac{F_i}{m} + F^c(y_i),\ F^c(y_i) = \tau_{xy} A_{xz} \lambda(y_i), \\ \lambda(y_i) &= \frac{g(y_i)}{\sum_j^N g(y_j)}, \end{aligned} \tag{2.5.2}$$

where and $F^c(y_i)$ is the net force computed from the continuum stress τ_{xy}, $\lambda(y_i)$ is the normalized $g(y_i)$ function, which distributes F^c across the constrained region, and F_i accounts for the sum of rest of forces

acting on the particle. A summary of different popular algorithms with particular implementations of Equation 2.5.1 for state coupling and Equation 2.5.2 for flux coupling, are listed in Table 2.5. Similarly, quantities other than momentum can be equally constrained. The exact procedure followed in this work can be found in Sections 3.4.5, 3.4.6, and 3.4.7.

5. A strategy to implement the connection between the implementation of the microscopic (MD) and continuum (CFD) solvers. In general, the two will be separate codes with their own set of library dependencies and particular intricacies. In addition, in order to simulate meaningful systems they are also likely to implement parallel capabilities to enable high performance computing. Despite being possible to combine the two of them in a monolithic fashion, by integrating them together in a single application, this is far from ideal as explained in Section 4.1. Therefore, it is advisable to use a software layer to connect the two solvers, ideally designed for distributed simulations on high-performance computers (HPC), in order to keep each solver a separate application. This has been achieved in this work by using and performing further developments of the coupling library `CPL_LIBRARY` [112], which provides an interface for efficient communication between two `MPI` codes coupled by domain decomposition. Chapter 4 is devoted entirely to describe the software architecture, parallel scalability and several other computational aspects of the coupling library and the CFD and MD applications developed in this thesis.

Lastly, a constraint algorithm by Smith et al. [113], in which momentum is constrained based on the sum of molecular fluxes on a control volume (CV), provides unification of a number of schemes in the continuum-molecular coupling literature. Constrained dynamics are applied together with an explicit localisation function based on the integral of the Dirac delta function over a CV (a cube in space) in order to derive the constraint

State Variable Coupling		References
Relaxation Dynamics	$\dot{v}_i = \frac{F_i}{m} + \frac{\varepsilon}{\Delta t}\left(u_c \frac{1}{N} - \sum_{j=1}^{N} v_j\right)$ ε : relaxation parameter	O'Connell & Thompson (1995) [100]
Maxwell Buffer	$v_i = u_i + \delta v_i,$ $p(\delta v_i) = \sqrt{\frac{m}{2\pi k_B T}} \exp\left[\frac{-m\left(\delta v_i\right)^2}{2k_B T}\right]$ $u_i \equiv u(y_i)$: interpolated continuum velocity δv_i : stochastic velocity component	Hadjiconstantinou & Patera (1997) [115]
Least Constraint Dynamics	$\dot{v}_i = \frac{F_i}{m} - \frac{1}{N}\sum_{j=1}^{N} \frac{F_j}{m}\frac{1}{\Delta t}\left(u_c \frac{1}{N} - \sum_{j=1}^{N} v_j\right)$	Nie *et al.* (2004) [102]
Flux Coupling		**References**
Equation 2.5.2 with $g(y_i) = 1$		Buscalioni *et al.* (2003) [114]
Equation 2.5.2 with $g(y_i) = 2\left[(L-2y_i)^{-1} - L^{-1} - 2y_i L^{-2}\right],\ 0 \leqslant y_i \leqslant L/2$ L : constrained cell size		Flekkoy *et al.* (2000) [34]

Table 2.5: Mathematical expressions for state variable and flux coupling within the constrained region. The equations are written for a one dimensional flow and have been extracted from [97], where the reader is referred for a more comprehensive review.

$$\dot{v}_i = \frac{F_i}{m} - \frac{m_i \vartheta_i}{M_I}\left[-\frac{d}{dt}\int_V \rho \boldsymbol{u} dV + \sum_{i=1}^{N}\sum_{j\neq i}^{N} \boldsymbol{f}_{ij}\vartheta_{ij} - \sum_{i=1}^{N} m_i \dot{\boldsymbol{r}}_i \dot{\boldsymbol{r}}_i \cdot d\mathbf{S}_i\right] \quad (2.5.3)$$

where $M_I = \sum_{i=1}^{N} m_i \vartheta_i$ is the mass in a control volume with ϑ_i selecting molecules inside the volume, ϑ_{ij} selecting forces acting over the surface of the volume and $d\mathbf{S}_i$ selecting molecules crossing the surfaces at the current time. This constraint included an extra term for the surface flux of momentum which was missing from Nie *et al.* (see Table 2.5). This extra term is shown to be essential for correct implementation of the principle of least action to a local volume. The algorithm iterates each timestep to ensure the time

evolution of a constrained volume in the molecular system is identical (to machine precision) to the corresponding volume in the continuum system.

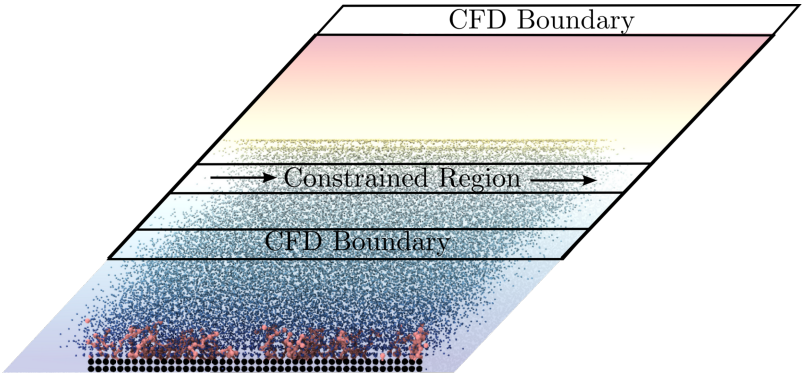


Figure 2.6: A schematic of a coupled Couette flow with polymer brushes. The top wall of the continuum is driven at a velocity of one, and the colors in the background show the velocity field in the coupled system (with molecules colored by the same field). The constrained and CFD boundary regions are denoted on the figure to show the way the coupling is applied. Reproduced with permission from [1]. Copyright Springer, 2020.

2.6 Chapter Summary

The majority of the theory supporting this work has been reviewed in this chapter. The remaining theoretical background not covered here, specific to the multi-scale method by domain decomposition, can be found in Chapter 3. A description of the atomistic systems under shear studied in this thesis is given in Section 2.1. These systems are simulated by molecular dynamics (MD) and are comprised of a slab of fluid confined by, and sheared between two moving atomistic surfaces. The ultimate goal of this thesis is to enable the study of these systems at larger lengh-scales by replacing the bulk region of the fluid (far from the surface-fluid interface) with a CFD solver. This is achieved by using a multiscale technique known as domain decomposition to effectively lower the computational cost of the simulation with respect to the full MD case.

In Section 2.2 the statistical mechanics foundations of the MD method have been reviewed. The concept of phase space and ensemble average are

presented and connected to the time-averaging of particle's trajectories employed by the MD method through the ergodic hypothesis. In addition, the mathematical connection between microscopic variables and hydrodynamic quantities has been provided in Section 2.2.3, the definition of the microscopic pressure tensor in Section 2.2.4 and the Green-Kubo relations to compute transport coefficients from equilibrium MD simulations have been introduced in Section 2.2.5.

The molecular dynamics method, with a focus to its application to fluids, is presented in Section 2.3. A discussion on the Lennard-Jones pair-wise potential to model weak short-ranged inter-atomic interactions can be found in Section 2.3.1. Additionally, force-fields to model interactions for poly-atomic molecules, are discussed in Section 2.3.2. The OPLS potential which is well-suited to model organic molecules and its variant L-OPLS, specifically parametrized for long-chained hydrocarbons, have been also examined. In order to compute viscosity from bulk MD systems under non-equilibrium shear conditions, two NEMD methods, Muller-Plathe and SLLOD, has been examined in Section 2.3.4.

The mass, momentum and energy equations of fluid dynamics have been reviewed in Section 2.4.1 and several time-dependent analytical solutions for the planar Couette, Poiseuille and Rayleigh-Stokes flows are provided in 2.4.2. These expressions has been later used as baseline solutions to check the validity of coupled flows in Section 5.3.5 and 5.4.3.

Lastly, the multiscale domain decomposition method for fluids has been described in Section 2.5, presenting different strategies available for coupling MD and CFD techniques through an overlap region where the two domains are enforced to be consistent. This method is the foundation of the work carried out in this thesis.

Chapter 3

Multiscale CFD-MD Coupling Methodology

3.1 Introduction

Computational fluid dynamics (CFD) methods, where the fluid is described as a set of discretised continuum equations, has been for decades the workhorse for simulating fluids at the macro-scale in many fields of engineering [116–118]. CFD methods are computationally "inexpensive" with respect to other simulation techniques operating at smaller length and time-scales such as molecular dynamics (MD), which has been reviewed in Section 2.3. However, the accuracy of CFD methods depends on providing physically meaningful boundary conditions and accurate constitutive relations and fluid properties which not always are readily available. Effects like velocity and thermal slip at the solid-liquid boundary, which are of paramount importance in nano- and microfluidics, are always part of the inputs CFD models depend upon (e.g. Navier slip boundary condition [119]). On the other hand, molecular methods model explicitly the motion of many

interacting particles, whose sizes are determined by the relevant length-scales of the problem: from atoms, molecules, to more "coarse-grained" descriptions of matter [108,120]. For instance, in molecular dynamics, the principal inputs of the model are the interaction energies between particles (force-field parameters). Therefore, properties and effects at the boundaries in liquids, arise from the complex collective behaviour of their constitutive atoms, which is determined by the force-field parameters provided. Due to the ability to capture complex fluid phenomena, MD has increasingly gained popularity in the scientific computing community, and specifically in the field of nanofluidics [24], where conducting experiments is difficult or simply impossible with the current instrumentation technology. However, despite the appeal of molecular techniques, going beyond the nano-scale into larger time- and length-scales is severely hindered by the computational cost involved, even using parallel algorithms executed in HPC clusters [121].

MD-CFD coupling by *domain decomposition* is a multi-scale technique which offers a strategy to mitigate length-scale limitations of fully atomistic models. If the requirement for molecular detail is limited to a certain region in space (e.g. close to the liquid-surface interface), a cheaper continuum model might be used elsewhere in the domain (in the bulk of the fluid) in order to decrease the simulation time. On the other hand, time-scale limitations are bound to the small time-steps needed to integrate the equations of motions of the atoms in MD, which are in the order of femtoseconds (10^{-5} s) for all-atom/united-atom models [121,122].

The multi-scale strategy employed in this work is based on partitioning the simulation domain into two sub-domains which overlap in a region which acts as a "handshake" zone, also called the hybrid simulation interface (HSI) in the specialised literature [100,102]. Inside this region, the physical fields (typically momentum, density and energy) are enforced to be consistent between the two domains. This type of coupling is categorized as a *two-way concurrent coupling* [90], where data is exchanged between the two sub-domains as the simulation advances in time.

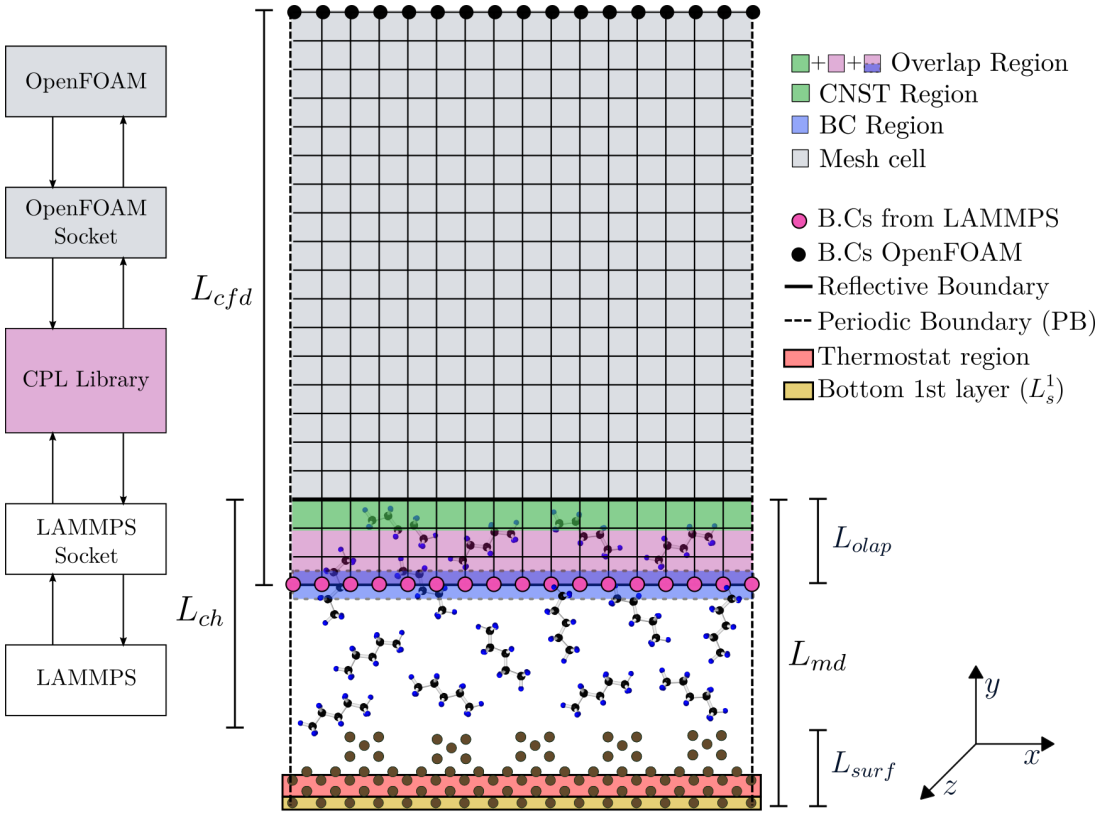


Figure 3.1: Schematic of the setup for a MD-CFD domain decomposition strategy. The simulation domain is divided into the CFD (`OpenFOAM`) and the MD (`LAMMPS`) sub-domains, which physically overlap in an intermediate region, where information is exchanged by employing the `CPL_LIBRARY` described in Section 4.2.1. The simulation domain is periodic in the x and z directions. The lengths defined over the y direction are the length of the CFD domain L_{cfd}, the MD domain L_{md}, the overlap region L_{olap}, the surface L_{surf} and the MD portion of the flow channel L_{ch}. The molecular liquid in the MD is bounded in the y direction by a reflective wall at the top and the atomistic surface at the bottom, where the thermostatting and the first bottom layer of surface atoms L_s^1 is indicated, respectively, in red and yellow. The CNST and the BC regions contain the coupling cells where coupling algorithms from Sections 3.4.5, 3.4.6 and 3.4.7 are applied.

This strategy is well-suited for the simulation domain setup and the cases of interest for this thesis described in Section 2.1. Interfacial effects like molecular slip, local phase changes and friction at the boundary, are localized near the surface, hence the molecular domain has to be sufficiently big to capture them. The diagram in Figure 3.1 gives an overview of the method

implementation followed. The bottom domain, where the liquid interacts with the surface, is modeled atomistically (represented with molecules in the figure). In the top sub-domain, CFD is employed and the black grid in the same figure represents the discretisation mesh used by the solver. Two sets of boundary condition are shown in Figure 3.1 within the CFD domain: bottom boundary conditions provided by the MD domain (pink dots) and top boundary conditions specified as part of the initial conditions for the CFD solver (black filled dots). The characteristic dimension of the domain is set to be always parallel to the y axis and periodic boundaries (PBs) are defined on the left and right faces (perpendicular to x axis), and front and back faces (perpendicular to z axis) of both MD and CFD domains. The constrained region (labeled as CNST) and the boundary condition region (labeled as BC) are defined inside the overlap region as described in Section 3.4.1.

In order to implement the coupling methodology, the molecular dynamics engine `LAMMPS` [35] and several solvers within the `OpenFOAM` [72] CFD package were modified to implement a communication `C++` interface (`CPLSocket`) provided by the in-house developed `CPL_LIBRARY` [111], which is `MPI`-based. This library allows the concurrent exchange of data, related to the coupled physical fields, within the overlap region. A box diagram in Figure 3.1 shows schematically the data flow between the two solvers. The implementation details of the modifications to each solver and the library itself are described in Chapter 4.

In this chapter, the theoretical and numerical aspects of the coupling strategy implemented in this work are presented. The implementation details for the MD and CFD domains can be found, respectively, in Section 3.2 and Section 3.3 and the mathematical apparatus describing the coupling methodology is presented in Section 3.4.

3.2 Molecular Dynamics Domain

The molecular dynamics domain is the most critical building block of the coupling scheme. If we could harness enough computational power to perform a full MD simulation in a reasonable time for the case of interest, no

multi-scale method would be necessary. In the view of the author, in domain-decomposition, the CFD becomes simply a computationally "cheap" replacement model for the portion of the MD domain whose behaviour could be described by a set of continuum equations describing the time evolution of the fluid as explained in Section 2.4.1. This perspective is not often the one shared in the literature [92,100,123–125], where MD is portrayed as a method of adding "molecular detail" to CFD simulations, as in the point-wise coupling technique [95]. Conceptually, this an important distinction since, in my view, it is the MD domain which imposes the majority of implementation constraints. For our purpose, the molecular domain establishes three important restrictions:

1. Maximum accessible time-scales in the order of nanoseconds. Integrating atomic equations of motion for molecules containing hydrogen bonds (all atom potentials) at room temperature, requires time-steps of the order of $\Delta t_{md} \approx 1$ fs [126]. United atoms potentials and techniques imposing constraints over bonds and angles like SHAKE [127], allow increasing the time-step, but computational gains are marginal and the accuracy of rheological properties can be degraded.
2. Geometry requires periodic boundary conditions (PBCs) along span-wise directions (X-Z plane). More complicated geometry configurations would considerably increase the difficulty of coupling the CFD mesh to the molecular domain and are not considered here.
3. Flows with rates below $\dot{\gamma} < 10^7\ \mathrm{s}^{-1}$ become extremely expensive to simulate using MD at room temperature, due to a poor signal-to-noise in the velocity field (the higher the temperature the worse). This requires extremely long averaging windows to filter the noise and recover the mean local value [66,129].

Three important elements of the MD domain can be identified in Figure 3.1: (1) the fluid (2) the bottom atomistic surface (or wall) and (3) the top termination boundary. These three elements are individually examined in Sections 3.2.1, 3.2.2 and 3.2.3 respectively.

3.2.1 Modeling the Atomistic/Molecular Fluid

Two different type of fluids were studied in this work: Lennard-Jones (mono-atomic) and molecular (poly-atomic). The former is comprised of point particles of mass m_i and interactions defined by the LJ potential according to Equation 2.3.7. Parameters for the LJ fluid used in this work can be found in table Table 5.6. LJ simulations and results derived from them, are presented in reduced units [2]. For conversions between reduced and S.I units see Table 2.1. On the other hand, molecular fluids are comprised of molecules with internal degrees of freedom and more complex potentials are used to model their interactions [52]. The all-atom OPLS [56] and L-OPLS [63] potentials were used in this work, respectively, for *benzene* and *squalane*. United atom potentials are computationally less expensive but viscosity is in most cases under-predicted [52]. Details about the functional form of the OPLS potential can be found in Section 2.3.2 and values of the parameters for benzene and squalane molecules are listed in Table 5.3.

Long-range electrostatic interactions (Coulombic) have not been included for two reasons: (a) they do not affect the properties of the molecular fluids studied since hydrocarbons exhibit non-polar behaviour and (b) atomistic surfaces employed in this work do not have partial charges which could potentially polarize fluid molecules. For other fluids like water and surfaces like FeO, which contain partial charges, these interactions cannot be disregarded [129,130]. By not including Coulombic interactions, a significant speedup is achieved since it is the most expensive energy contribution to compute in an MD simulation [131]. We can write the equations of motions for the fluid atoms in a coupled simulation as

$$m_i \frac{d\boldsymbol{r}_i}{dt} = \nabla U^f\left(\boldsymbol{r}_i\right) + \nabla U^{fs}\left(\boldsymbol{r}_i\right) + F_T^{CNST}\left(\boldsymbol{r}_i; T^{th}\right) + F_{\boldsymbol{\tau}}^{CNST}\left(\boldsymbol{r}_i\right) \quad (3.2.1)$$

where U^f/ U^{fs} are the fluid-fluid/fluid-surface interaction energy for particle i, F_T^{CNST} (parametrized by the thermostat temperature T^{th}) is the coupling thermostatting force (see Section 3.4.7) and $F_{\boldsymbol{\tau}}^{CNST}$ is the momentum coupling force (see Section 3.4.6) for particles within the CNST region. Equations are solved employing the velocity Verlet algorithm described in

Section 2.3.3. Fluid-surface interaction energy U^{fs} is also implemented employing a LJ pair-wise potential following Equation 2.3.7.

An important aspect of non-equilibrium MD simulations of confined fluids is the choosing a correct thermostatting strategy [132]. There is strong evidence that thermostatting the fluid particles introduce simulation artifacts such as the appearance of the so called *string phase,* where atoms are found to be arranged in a configuration resembling strings perpendicular to the flow direction [33]. Conversely, thermostatting just the walls shows best agreement in reproducing experimental mechanical and thermal flow properties [133].

A quantitative study conducted (summarized in Table 5.8), suggests that thermostatting only surface particles (see Section 3.2.2) and fluid particles in the perpendicular direction (y direction) within the CNST region, leads to more accurate coupled simulations. See Section 5.3.4 for the details of the study and Figure 5.13 for the results. Therefore, coupled simulations presented in Section 5.3.5 and 5.4.3 use this thermostatting strategy.

3.2.2 Modeling the Atomistic Surface

Including atomistic detail by employing MD at the boundary between the fluid-surface interface allows us to model explicitly realistic inter-facial phenomena (fluid slip, wettability, phase changes, etc.). Otherwise, a phenomenological model for the set of boundary conditions for the liquid-solid boundary would need to be provided in a full CFD scenario (e.g Navier BCs for slip [119]). The thermal, chemical and mechanical behaviour of the fluid-solid interface, is determined by the local thermodynamic conditions and by the nature of the surface and the liquid. However, the atomistic models employed in this work to simulate surfaces, are not meant to be a high fidelity physical representations. Doing so would have required a very careful study to match all the thermo-mechanical (e.g bulk modulus) and thermo-chemical (e.g adsorption energies) properties of the solid surface and the surface-liquid interface [134,135]. The most important physical requirements for the atomistic surfaces used in this work were (i) mechanical

stability, (ii) good thermal dissipation and (iii) the ability to reproduce no-slip/slip behaviour (see Figure 5.24 and Figure 5.25).

The bottom atomistic surface in the MD domain is tethered to its initial position, in order to fix the length of the fluid channel. The channel is defined by the space between the top of the atomistic surface and the top reflective wall, and its length along the y direction is $L_{ch} = L_{md}–L_{surf}$. This is shown in Figure 3.1. A harmonic potential is used to tether the first bottom layer of the atomistic wall to the initial lattice positions

$$U_i^{fix} \equiv U^{fix}\left(\boldsymbol{r}_i\right) = K^{fix}\left|\boldsymbol{r}_i - \boldsymbol{r}_i^0\right|^2 \ , \ i \in L_s^1, \tag{3.2.2}$$

where K^{fix} is the interaction strength of the tethering, $\boldsymbol{r}_i$ and $\boldsymbol{r}_i^0$ are the current and initial position of atom i and L_s^1 is the group of atoms in the first bottom layer of the atomistic wall as shown in Figure 3.1.

For coupled simulations of Lennard-Jones (LJ) presented in Section 5.3, a simple mono-atomic LJ surface was utilized. Surface-surface atomic interactions $U^s\left(\boldsymbol{r}_i\right)$ were modeled by Equation 2.3.7, which is defined by its σ^s (length), ϵ^s (interaction) and r_c^s (cut-off radius) parameters. The LJ interaction model only depends on three parameters, and it is easy to implement since no topology information is required (e.g explicit bonds between atoms).

In the case of coupled molecular simulations presented in Section 5.4, a Cu (copper) mono-atomic surface with harmonic bonds was employed to contain the molecules at the bottom of the MD domain. This is a representative surface which has been used in prior experimental and computational tribological studies [79,136,137]. By using harmonic interactions, we reduce the computational cost compared to other potentials tailored to model solid metals like the Embedded Atom Model (EAM) [138], which are known to be significantly expensive (more than ten times compared to harmonic [139]). Interactions between pairs of surface atoms are defined by the surface bond topology created as part of the simulation setup. The surface-surface interaction energy U_{ij}^n between atom i and atom j (the index n indicates the shell of atoms as explained later) used to model the Cu surface is mathematically defined as

$$U_{ij}^{n} = K_{s}^{n}\left(r_{ij} - r_{ij}^{0}\right)^{2},\ \forall j \in \mathbb{N} \mid R_{in}^{n} \leqslant r_{ij}^{0} \leqslant R_{out}^{n}, \tag{3.2.3}$$

where K_s^n is the interaction strength (harmonic constant), r_{ij} and r_{ij}^0 are, respectively, the instantaneous and initial inter-atomic distance and R_{in}^n, R_{out}^n are the inner and outer radius for the shell n of atoms centered around the initial position $\boldsymbol{r}_i^0$ of atom i. The set of potential parameter's values used are listed in Table 5.12. The value for the harmonic constant for copper was estimated from $K_s^1 = K_s^2 \equiv k \approx E_{Cu} \times a$ [140], where $E_{Cu} = 128$ GPa [141] is the Young modulus and $a = 2.56$ Å the copper's lattice parameter. The total surface-surface energy contribution for particle i at position $\boldsymbol{r}_i$ becomes

$$U_{i}^{s} \equiv U^{s}\left(\boldsymbol{r}_i\right) = \sum_{n}^{N^l}\sum_{j\neq i}^{M^n} U_{ij}^{n}, \tag{3.2.4}$$

where N^l is the number of shells considered to compute the interaction and M^n is the number of atoms in shell n. Two shells ($N^l = 2$) were considered.

Given the previous definitions for the interaction energies, the equations of motions for the surface atoms can be written, in general, as

$$m_i \frac{d\boldsymbol{r}_i}{dt} = \nabla U^{s}\left(\boldsymbol{r}_i\right) + \nabla U^{fs}\left(\boldsymbol{r}_i\right) + F_{T}^{s}\left(\boldsymbol{r}_i; T^{h}\right) + F_{L_s^1}^{s}\left(\boldsymbol{r}_i\right), \tag{3.2.5}$$

where U^s and U^{fs} are the surface-surface and surface-fluid particle interaction energies respectively, F_T^s is the thermostatting force (applied to atoms in the thermostatting region in Figure 3.1) and $F_{L_s^1}^s \in \left\{\nabla U^{fix}, F^{bar}\right\}$ is the force applied to the atoms in the first bottom layer of the surface L_s^1, which can take the form of Equation 3.2.2 (tethered surface) or Equation 5.4.2 (barostatting force).

During a coupled simulation the fluid is not thermostatted, except for the CNST region as described in Section 3.4.7. Therefore, the surface acts as a heat bath for the system which needs to be able to remove any spurious temperature build-up over time [134]. This is achieved by thermostatting the atoms located over the first bottom layer of atoms L_s^1, but not too close to the atoms interacting with the fluid (see Figure 3.1), in order to minimize the particle dynamics disturbance at the surface-fluid interface [133]. A

Langevin thermostatting force applied in the three spatial directions F_T^s was employed, similar to the one described in [142] in order to maintain the surface's temperature constant. The exact form of the Langevin force (implemented in `LAMMPS`) is not important, and any other thermostat with efficient heat removal properties, could have equally been employed instead.

3.2.3 MD Domain Boundaries

In Section 2.5 methods for removal and insertion of particles in molecular dynamics were briefly discussed. Simulations employing them are categorized as "open boundary" [143]. Such type of methods are interesting to simulate physical process where molecules need to leave from or enter to the molecular domain through a simulation box boundary. These methods are particularly useful for low density fluids and gases, where the thermal expansion and compressibility effects and local density changes are significant [98]. Open boundary algorithms are not trivial to implement for non-monoatomic fluids in a physically consistent manner. However, fluids with low thermal expansion coefficients, or under close to isothermal conditions, will always experience negligible density changes. Therefore, in those circumstances, the number of molecules in the MD simulation box can be considered constant and "close boundary" methods can be used instead. Since every simulations in this work has been performed close to isothermal and incompressible conditions, a "close boundary" approach was used, with reflective walls placed to avoid molecules crossing non-periodic simulation boundaries.

The MD domain topology consists of PBs (Periodic Boundaries) along the x and z directions and it is terminated by two, bottom and top, reflective walls along the y spatial dimension (see Figure 3.1). The bottom reflective wall is never in contact with any molecule from the fluid as the atomistic surface holds the molecules contained. On the other hand, the top reflective wall is responsible for avoiding the fluid molecules leaving the simulation domain through the top boundary. The reflective walls employed operate by modifying atoms' velocities and positions when they cross a reflective boundary R following the equations

$$v_y^i\left(t_i\right) = -v_y^i(t_i), \tag{3.2.6}$$

$$y^i(t_i) = y^R - \Delta_y^R, \tag{3.2.7}$$

where for a particle i at time t_i, $v_y^i(t_i)$ is the velocity in the y direction, $y^i(t_i)$ is the y coordinate and $\Delta_y^R = y^i(t_i) - y^R$ is the distance the particle moved outside the boundary in the y direction from $t_{i-1} \rightarrow t_i$.

By removing periodicity along the reflection axis, fluid atoms that are within the potential cutoff distance from the reflective wall, do not receive force contributions via the minimum image convention [144] as in the case of PBs. This causes the appearance of a region of missing interactions, which leads to spurious density fluctuations which decrease with the distance to the reflective wall. This change in the fluid structure might, in principle, affect the local properties of the fluid [104].

The coupling strategy developed in this work, relies heavily on liquid properties being locally consistent between the MD and the CFD domain as later explained in Section 3.4; therefore, this scenario could in principle be detrimental for the accuracy of the method. A density correction algorithm FOREV developed by Issa & Poesio [104] was implemented in `LAMMPS` to correct the liquid density profile close to the top reflective wall. The region close to the reflective wall is divided into N_b equal sized bins along the y direction (reflective direction) over a length L_c. This is the length over which density oscillations are significant and it is related to the cut-off distance of the potential. For each bin k, an instantaneous force can be computed as

$$\boldsymbol{F}_{bin,k} = \sum_{i}^{N_k} \sum_{j \neq i} \frac{\boldsymbol{F}_{ij}}{N_k}, \tag{3.2.8}$$

where N_k is the number of atoms inside bin k and F_{ij} is the force between a pair of atoms within the cut-off distance of the potential. For periodic boundaries (boundary planes perpendicular to the x and z directions in our setup), the equilibrium density profile is always flat and the time-averaged bin forces for x and z directions are $\langle F_{x;bin,k} \rangle = \langle F_{z;bin,k} \rangle = 0$. This is attributed to the symmetry of the region of missing interactions about these

two axes and is explained by the existence of a strong correlation of $\langle F_{y;bin,k} \rangle$ and the local density gradient [104]. The FOREV algorithm is based on zeroing the average force over time on each bin, $\langle \boldsymbol{F}_{bin,k} \rangle = 0$, which is the condition fulfilled for PBs.

The algorithm was implemented to support efficient domain decomposition of the MD domain by parallelization [35]. In principle, a single bin could span multiple processors along the x and z directions. Since the instantaneous value of $\boldsymbol{F}_{bin,k}$ is needed on each processor intersecting a bin, if bins are shared between processors, computing $\boldsymbol{F}_{bin,k}$ could turn out to be extremely expensive due to the inter-process communication involved in resolving the summation in Equation 3.2.8. To mitigate this, each of the N_b bins defined in FOREV along the y direction, is divided in the x and z directions, respectively, by the number of processors Np_x and Np_z. Consequently, the total number of individual bins in the three directions are then $N_b^{av} = Np_x \times Np_z \times N_b$. Furthermore, the binning distance L_c, is always sufficiently large so all the N_b bins can be fitted on a single processor along the y direction without any drop in performance. This setup, ensures that no inter-process communication is needed to compute $\boldsymbol{F}_{bin,k}$ at each time-step. The zeroing condition can be then rewritten as $\left\langle F^P_{y;bin,k} \right\rangle = 0$, where P indicates the processor in which the bin is located. The number of particles within each processor is always reasonably large to ensure good `LAMMPS` scaling. The P index is dropped for clarity in Equation 3.2.9 and further references. To fulfill the zeroing condition over time, a first order exponential smoothing [145] algorithm is applied per bin in the y direction as

$$F^{n+1}_{y;b,k} = \alpha F^{n+1}_{y;bin,k} + (1-\alpha)\, F^{n}_{y;b,k}, \tag{3.2.9}$$

where $-F^{n+1}_{y;b,k}$ becomes the boundary force applied to particles within the bin k and α is the smoothing parameter. At time-step $n = 1$, the initial value is $F^1_{y;b,k} = 0$. In Section 5.3.2 a parametric study to determine sensible values for N_b and α is presented.

3.3 Computational Fluid Dynamics Domain

At the beginning of the chapter, the perspective adopted in this work in which the CFD domain acts, conceptually, as a less expensive replacement for part of the MD fluid was introduced. This is true provided the time evolution the physical fields of interest (momentum, energy and density) can be described by the set of PDEs presented in Section 2.4.1 within the portion of the MD domain to be replaced by the CFD. By placing the CFD domain sufficiently far from the atomistic surface (see Figure 3.1) the molecular fluid exhibit bulk behaviour as solid-liquid inter-facial effects fade out [146]. Therefore, the aim is to limit the size of the MD domain as much as possible to minimize the computational cost of the coupled simulation. This is not always straightforward to achieve. In all atom molecular fluids a great amount of spatial sampling is required to reduce noise in the boundary conditions computed from the MD domain within the BC region. Therefore, larger MD regions are required in that case. See Section 3.4 for more details about the coupling procedure.

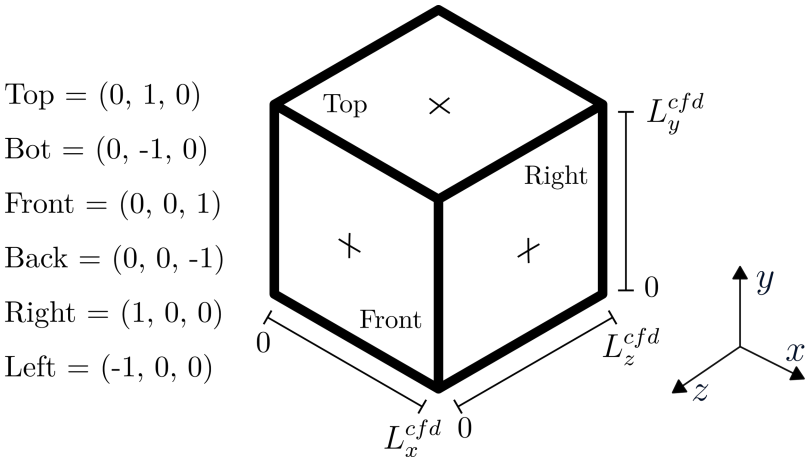


Figure 3.2: CFD domain configuration. The domain is a brick with faces (`OpenFOAM` patches) labeled as depicted. The size of the CFD domain along the x an z directions L_x^{cfd} and L_z^{cfd} are equal to the atomistic surface dimensions L_x^{surf} and L_z^{surf}.

The MD-CFD coupling methodology was developed focusing on viscous liquids for tribological applications at high-pressure. In those conditions, the viscosity of molecular fluids is high and non-Newtonian effects are common. Furthermore, the scope of this work has been limited to *quasi-isothermal*, *incompressible* simulations although there has been some preliminary effort to incorporate thermal effects due to shear heating. This is discussed in detail by Yasuda *et al.* [123]. Details of the work done so far on that aspect and potential challenges are presented in Section 3.4. Furthermore, the simulations has been limited to parallel flows and CFD meshes employed are 1D grids graded along the y direction as depicted in Figure 3.3.

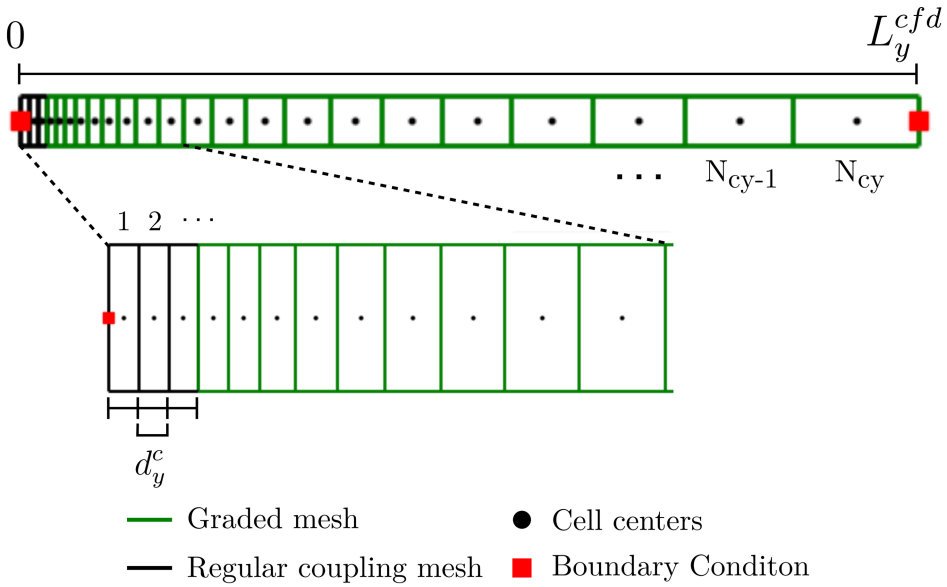


Figure 3.3: The CFD mesh is composed of two blocks: (i) in black a regular mesh which is the portion inside the overlap region with a cell size d_y^c in the y direction (ii) in green, a graded mesh which is finer close to the MD domain. This is desirable in order to resolve near-wall fields in the MD when the CFD domain is orders of magnitude larger than the MD. The meshes employed in this work are 3D meshes with $1 \times N_{cy} \times 1$ cells.

Two different solvers implemented for coupling are described in this section: (a) Newtonian and (b) non-Newtonian. The solvers were implemented by adapting, respectively, the `icoFoam` and

`nonNewtonianIcoFoam` solvers available in `OpenFOAM` [72]. See Section 4.2.2 for an explanation of the modifications made to the solver's code in order to couple them to `LAMMPS`. The advantages of using `OpenFOAM` is that existing solvers are well-tested and support out-of-the box parallelization by domain-decomposition employing `MPI` [9]. Specific details about the the discretization for the PDEs and the implementation of the solvers in `OpenFOAM` are given in [73,74].

3.3.1 Momentum equation. Newtonian and Non-Newtonian solvers

For an incompressible Newtonian fluid, the time-evolution of the velocity field can be computed by solving the ubiquitous Navier-Stokes equations (see Equation 2.4.8). This set of equations represent the workhorse of CFD techniques and `OpenFOAM` provides several explicit and implicit transient solvers with multiple spatial and time discretization methods implemented in `C++`. Table 3.2 lists the discretization technique employed for each mathematical differential operator for a particular physical field. To solve for the coupled velocity-pressure momentum equation, the explicit PISO algorithm [74] was employed.

The isotropic Newtonian shear stress in the CFD domain is defined according to Equation 2.4.7 as

$$\boldsymbol{\tau} = \eta^{cfd}\left[\nabla \boldsymbol{u} + (\nabla \boldsymbol{u})^{T}\right], \tag{3.3.1}$$

where η^{cfd} is the Newtonian dynamic viscosity of the CFD fluid. Momentum transport in coupled simulations using the Lennard-Jones fluid model (Section 5.3.5) and molecular benzene (Section 5.4.3) exhibiting Newtonian bulk behaviour, were modeled with Equation 2.4.8. Nevertheless, in general, the isotropic dynamic viscosity of a fluid can be a function of the spatial coordinate $\boldsymbol{x}$ through the explicit dependency on density $\rho(\boldsymbol{x})$, temperature $T(\boldsymbol{x})$ and shear-rate $\dot{\gamma}(\boldsymbol{x})$ fields (non-Newtonian effects)

$$\eta^{cfd}(\boldsymbol{x}) \equiv \eta^{cfd}\Big(\rho(\boldsymbol{x}), T(\boldsymbol{x}), \dot{\gamma}(\boldsymbol{x})\Big). \tag{3.3.2}$$

We can modify the stress tensor in Equation 2.4.5 by inserting Equation 3.3.2, leading to

$$\boldsymbol{\sigma} = -p\boldsymbol{I} + \eta^{cfd}(\boldsymbol{x})\left[\nabla\boldsymbol{u} + (\nabla\boldsymbol{u})^T\right], \tag{3.3.3}$$

and substitute $\boldsymbol{\sigma}$ in Equation 2.4.4 to obtain a modified set of momentum equations with spatially-dependent isotropic viscosity

$$\frac{\partial\boldsymbol{u}}{\partial t} + (\boldsymbol{u}\cdot\nabla)\,\boldsymbol{u} = -\frac{\nabla p}{\rho} + \nu^{cfd}(\boldsymbol{x})\,\nabla^2\boldsymbol{u} + \nabla\boldsymbol{u}\nabla\nu^{cfd}(\boldsymbol{x}). \tag{3.3.4}$$

Equation 3.3.4 is identical to Equation 2.4.8 but with the extra term $\nabla\boldsymbol{u}\nabla\nu^{cfd}(\boldsymbol{x})$ due to the spatial dependency of ν^{cfd}. This equation was used to model momentum transport in coupled simulations of benzene and squalane in Section 5.4.3 (cases 2 to 5 in Table 5.15). Both molecules exhibit non-Newtonian effects at high pressures in the shear-rate range accessible by MD simulations (10^7-10^{11} s^{-1}) as Figure 5.9 and Figure 5.10 show.

icoFoam	**nonNewtonianIcoFoam**
Kinematic Viscosity (ν^{cfd})	
`dimensionedScalar nu;`	`singlePhaseTransportModel fluid(U, phi);` (`CrossPowerLawLog`, see Equation 5.2.4)
Shear Stress (τ)	
`volSymmTensorField tau =` ν^{cfd}`*density*2*dev(symm(fvc::grad(U)));`	
Momentum Equation Representation	

`fvVectorMatrix UEqn(`
`fvm::ddt(U) +`
`fvm::div(phi, U) -`
`fvm::laplacian(nu, U)`
`);`
`//Solve:`
`UEqn == -fvc::grad(p);`
	`fvVectorMatrix Ueqn(`
	`fvm::ddt(U) +`
	`fvm::div(phi, U) - fvm::laplacian(nu, U)`
	`- (fvc::grad(U) & fvc::grad(nu))`
	`);`
	`//Solve:`
	`UEqn == -fvc::grad(p);`

Table 3.1: Object-oriented representation in `OpenFOAM` for Equation 2.4.8 (implemented in the `icoFoam` solver) and Equation 3.3.4 (implemented in `nonNewtonianIcoFoam` solver). Definitions of kinematic viscosity and shear stress $\boldsymbol{\tau}$ are given in `OpenFOAM` operator notation.

The aim of the multi-scale technique employed in this work is to accelerate shear-driven MD simulations of the like presented in Section 2.1, which commonly fall either in the category of 1D pressure-driven (Poiseuille) or shear-driven (Couette) flows, or a combination of both (see Section 2.4.2). Therefore, in these scenarios, it is not strictly necessary to solve for all the terms in Equation 2.4.8 and Equation 3.3.4 in three dimensions. However, solvers were implemented to always solve the full equation for two reasons: (i) `OpenFOAM` is developed to solve only for 3D geometry (although pseudo-2D is possible by using empty patches [72]) and (ii) MD systems coupled to the CFD are inherently 3D, therefore, solvers were to certain extent, future-proofed should more complex flows be studied in the future with the current software framework. The overhead of solving for the extra terms and dimensions is most likely negligible in comparison with MD solving times. In addition, their contributions will be, if not zero, numerically much more smaller than the convergence tolerance, which is set to 10^{-5} for any quantity.

OpenFOAM operator	**Math. Operator**	**Discretization**
`fvm::ddt(U)`	$\frac{\partial \boldsymbol{u}}{\partial t}$	Crank-Nicolson
`fvm::div(phi, U)`	$(\boldsymbol{u} \cdot \nabla)\, \boldsymbol{u}$	Gauss Linear
`fvm::laplacian(nu, U)`	$\nu^{cfd} \nabla^2 \boldsymbol{u}$	Gauss Linear Orthogonal
`fvc::grad(p)`	∇p	Gauss Linear
`fvc::grad(U)`	$\nabla \boldsymbol{u}$	Gauss Linear
`fvc::grad(nu)`	$\nabla \nu^{cfd}(\boldsymbol{x})$	Gauss Linear

Table 3.2: Discretization schemes available in `OpenFOAM` employed for each differential operator in the momentum equation. Operators in the `fvm:: C++` namespace construct implicit equations which are solved through an iterative process. The `PCG` (preconditioned conjugated gradient) with diagonal incomplete-Cholesky preconditioning was employed as iterative solver [74]. Linear cell-center to face interpolation was used.

Boundary conditions (BCs) defined for the momentum equation are applied over the labeled boundary planes in the rectangular prismatic

domain in Figure 3.2. The same way the geometry between the MD and CFD domains has to be consistent for coupling, boundary conditions need to be as well between both domains. Hence, periodic boundary conditions along the x and z directions in the MD has to be respected. This was achieved in `OpenFOAM` by connecting the `Right` and `Left` and `Front` and `Back` pairs of patches, making the boundaries defined by those planes effectively periodic [72]. The CFD boundary conditions currently implemented for the momentum equation by the coupled framework are listed in Table 3.3, which allow the generation of various coupled shear-driven and pressure driven flows (see Section 5.3.5).

Incidentally, the use of the Crank-Nicholson implicit time-integrator (see Table 3.2) is chosen here for its unconditional stability. This is the most important aspect in a CFD integrator for us, since it minimizes the chance for the solver "crashing" due to the appearance sudden local spikes on the boundary fields caused by the fluctuating BCs provided by the MD domain.

Face	**Velocity ($\boldsymbol{u}$)**		**Pressure (p)**	
	Math Expr.	**OpenFOAM**	**Math Expr.**	**OpenFOAM**
`Top`	$u_x^{Top}(t), u_y^{Top}(t), u_z^{Top}(t)$	`fixedValue`	$\frac{\partial p}{\partial y} = 0$	`zeroGradient`
`Bot`	Equation 3.4.8	`fixedValue`	$\frac{\partial p}{\partial y} = 0$	`zeroGradient`
`Front`	-	`empty`	-	`empty`
`Back`	-	`empty`	-	`empty`
`Right`	$\nabla \boldsymbol{u} = 0$	`zeroGradient`	$p = P_{Right}$	`fixedValue`
`Left`	$\nabla \boldsymbol{u} = 0$	`zeroGradient`	$p = P_{Left}$	`fixedValue`

Table 3.3: Boundary conditions for the momentum equation applied to the six faces (boundaries) of the CFD domain as depicted in Figure 3.2. The mathematical expression and the `OpenFOAM` name for each boundary condition are given. `Top` boundary conditions for velocity can be a time-dependent function and `Bot` boundary conditions are provided by the MD domain following Equation 3.4.8.

3.3.2 Thermal equation

The internal energy equation (Equation 2.4.15) reduces to the thermal equation (Equation 2.4.18) if the compressibility condition (Equation 2.4.3) holds. The form of the thermal equation employed in this work is

$$\frac{\partial T}{\partial t} + \boldsymbol{u} \cdot \nabla T = \alpha^{cfd} \nabla^2 T + \boldsymbol{\tau} : \nabla \boldsymbol{u}, \tag{3.3.5}$$

where $\alpha^{cfd} = \kappa^{cfd} / \left(\rho C_p^{cfd}\right)$ is the thermal diffusivity of the CFD which is considered constant. Table 3.4 contains the `OpenFOAM` implementation of the thermal equation and Table 3.5 provides the CFD boundary conditions.

icoFoam	**nonNewtonianIcoFoam**
Thermal Conductivity (λ^{cfd})	
`dimensionedScalar lambda;`	
Specific Heat Capacity (C_p^{cfd})	
`dimensionedScalar Cp;`	
Kinematic Viscosity (ν^{cfd})	
`dimensionedScalar nu;`	`singlePhaseTransportModel fluid(U, phi);` `(CrossPowerLawTempLog,` see Equation 5.2.5)
Shear Stress ($\boldsymbol{\tau}$)	
`volSymmTensorField tau =` ν^{cfd}`*density*2*dev(symm(fvc::grad(U)));`	
Thermal Equation Representation	

`fvVectorMatrix UEqn(`
`fvm::ddt(T) +`
`fvm::div(phi, T) - fvm::laplacian(lambda, T));`
`//Solve:`
`UEqn == (1/Cp)*(tau && fvc::grad(U));`	

Table 3.4: Object-oriented representation in `OpenFOAM` for Equation 3.3.5, which is solved in `icoFoam` and `nonNewtonianIcoFoam` solvers along with the momentum equation. The definition of the momentum ν^{cfd} and thermal α^{cfd} transfer coefficients, the specific heat capacity C_p^{cfd} and shear stress $\boldsymbol{\tau}$ in `OpenFOAM` operator notation are given.

Despite the thermal equation is presented here, the set of coupled simulations carried out in this work has been all conducted under the *quasi-isothermal* condition. Hence, given a specific coupled simulation setup and set of boundary conditions, this condition needs to be checked prior to running a coupled case. The quasi-isothermal condition holds if the local variation of temperature, is small enough so that transport coefficients α, ν and density ρ remain roughly constant throughout the simulation. The condition could be checked in several ways: (a) by using analytical expressions to find an estimate of the maximum temperature in the domain for a particular flow type (see Section 2.4.2); (b) by using a definition of the Nahme-Griffith number Na which includes variation of viscosity with temperature and shear rate [123]

$$Na = \frac{\eta_0 L^2 \dot{\gamma}_0^2}{\lambda \left| \partial \log(\eta_0)/\partial T_0 \right|^{-1}}, \tag{3.3.6}$$

where L is the characteristic domain length, $\dot{\gamma}_0 = u_x^{Top}/L$ is the characteristic shear rate, $\eta_0 \equiv \eta(\dot{\gamma}_0, T_0)$ is the characteristic viscosity and λ is the thermal conductivity. This expression is particularly useful for non-Newtonian fluids if the relation $\eta(\dot{\gamma}, T)$ is known. For $Na \leqslant 0.1$ it is considered temperature effect on the fluid's rheological properties can be disregarded. Lastly, (c) by solving, in a full CFD simulation, the thermal and momentum equations with no-slip conditions as part of the set of pre-coupling simulations (see Figure 3.4). From this simulation the maximum temperature developed across the domain can be found. Strategies (a) and (c) has been used prior running simulations presented in Section 5.3.5 and Section 5.4.3.

Simulations where the momentum and thermal equations are coupled simultaneously are left for future work (Section 6.3). However, a preliminary strategy is briefly described in Section 3.4.7 and some initial work has been carried out to provide a temperature-dependent coupled solver within the framework developed.

Face	Temperature ($\boldsymbol{T}$)	
	Math Expr.	**OpenFOAM**
`Top`	$T = T_{Top}$	`fixedValue`
`Bot`	Equation 3.4.24	`fixedValue`
`Front`	-	`empty`
`Back`	-	`empty`
`Right`	$\nabla \boldsymbol{T} = 0$	`zeroGradient`
`Left`	$\nabla \boldsymbol{T} = 0$	`zeroGradient`

Table 3.5: Boundary conditions for the temperature equation applied to the six faces (boundaries) of the CFD domain as depicted in Figure 3.2. The mathematical expression for the boundary condition and its `OpenFOAM` name are given.

OpenFOAM operator	Math. Operator	Discretization
`fvm::ddt(T)`	$\frac{\partial T}{\partial t}$	Crank-Nicolson
`fvm::div(phi, T)`	$(\boldsymbol{u} \cdot \nabla) T$	Gauss Linear
`fvm::laplacian(lambda, U)`	$\lambda^{cfd} \nabla^2 \boldsymbol{u}$	Gauss Linear Corrected
`fvc::grad(U)`	$\nabla \boldsymbol{u}$	Gauss Linear

Table 3.6: Discretization schemes from `OpenFOAM` employed for each differential operator in the thermal equation. Operators in the `fvm::` namespace construct implicit equations, which are solved through an iterative process. The PCG (preconditioned conjugated gradient) with diagonal incomplete-Cholesky preconditioning was employed as iterative solver [74]. Linear cell-center to face interpolation was used.

3.4 Coupling Methodology

The methodology presented in this section is conceptually based on the HSI (Hybrid Solution Interface) introduced by O'Connell & Thompson [100]. See Section 2.5 for a review on different methods based on this strategy. The domain decomposition method can be classified as a two-way concurrent

coupling technique [90]. In this approach, the macroscopic density $\rho(\boldsymbol{x})$, momentum $\rho(\boldsymbol{x})\boldsymbol{u}(\boldsymbol{x})$ and energy $\rho(\boldsymbol{x})e(\boldsymbol{x})$ fields, must evolve in time consistently in both domains inside the HSI (*overlap region* from now on). In the MD domain, the time-evolution of the system is achieved by solving the equations of motion of fluid's atoms given by Equation 3.2.1 and surface's atoms by Equation 3.2.5. The mathematical connections between atomistic variables and density, momentum and energy macroscopic fields, are provided respectively by Equations 2.2.15, 2.2.16 and 2.2.18. In the CFD domain, the time-evolution of each physical fields is dictated by the set of PDEs given by Equations 2.4.2, 2.4.8, and 3.3.5. In order to close the three PDEs, it is necessary to provide: (i) constitutive relations for the transport coefficients, the equation of state and the caloric energy density; (ii) boundary conditions. Hence, consistency between the MD and CFD domains within the overlap region is achieved by means of (i) and (ii) The process, which is depicted in Figure 3.4 entails:

1. Computing from MD fluid transport coefficients, the equation of state and the caloric energy density. This is done by tabulating results of a series of MD simulations prior to coupling (pre-coupling), parametrized over the range of temperatures, pressures and shear-rates of interest. For instance, results for equilibrium and non-equilibrium liquid properties for Lennard-Jones and molecular benzene and squalane are provided in Section 5.2.
2. Implementing a strategy to impose boundary conditions from the MD to the CFD domain in the BC region and from the CFD to the MD in the CNST region. The former involves coarse-graining the many degrees of freedom arising from particle dynamics to provide continuum boundary conditions. In the latter, a ‘‘generalized boundary condition’’ [112] needs to be constructed for the particle dynamics which conforms with the continuum description.

Mathematical and algorithmic aspects to tackle point 2, are discussed in this section, while software development and computational aspects, like the parallel efficiency of the coupling framework, are addressed in Chapter 4.

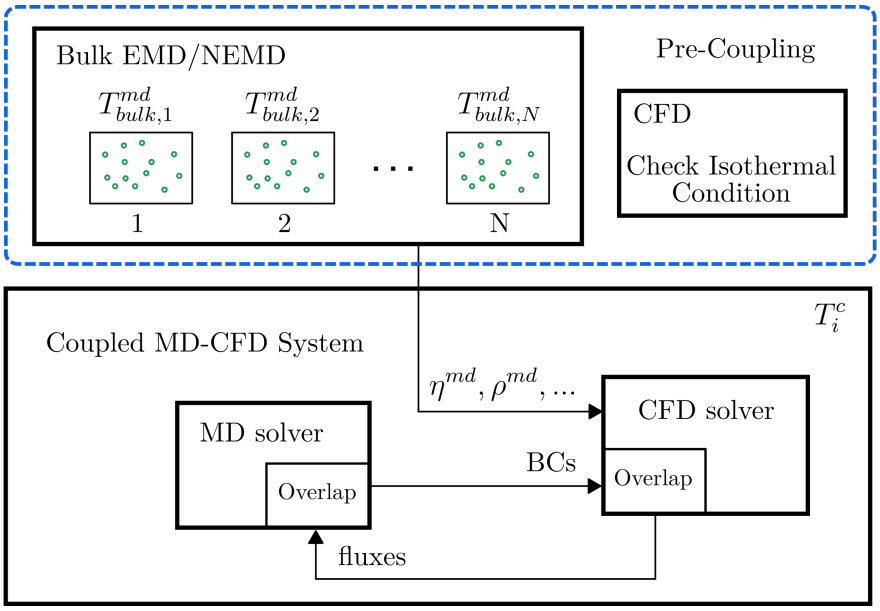


Figure 3.4: Diagram representing a full coupled study S. The CFD solver receives as inputs (a) boundary conditions (BCs) from the MD solver and (b) constitutive relations for the transport coefficients and the equation of state from pre-coupling equilibrium and non-equilibrium bulk MD simulations. The MD domain receives density, momentum and energy fluxes from the CFD. In addition, the quasi-isothermal condition is checked either by using one of the analytical expressions given in Section 2.4.2 or by performing a full CFD temperature-dependent (no-slip) simulation for an equivalent system setup to the one to be studied by coupling. The time taken by the full CFD simulation is negligible compared to $T^{md}_{bulk,i}$, which is the time taken by a pre-coupling bulk MD simulation. T^c_i is the time taken by an instance of a coupled simulation.

The CFD-MD domain decomposition method allows simulating a system which is equivalent to a very large MD but with less computational effort. This directly imposes a constraint in the computational time which is practical to spent in pre-coupling simulations (see Figure 3.4). Take a system S where (a) the total time needed to perform a full MD study is $T^{md} = \sum_i^S T^{md}_i$, T^{md}_i being the time taken by a single MD simulation; (b) the total time taken by an equivalent study employing coupling is $T^c = \sum_i^S T^c_i$, being T^c_i the time taken by a single instance of a coupled simulation and (c) the total time taken by essential pre-coupling simulations

is $T^{md}_{bulk} = \sum_i^N T^{md}_{bulk,i}$, being $T^{md}_{bulk,i}$ the time taken by a single pre-coupling simulation. For the coupling method to be effective, $T^{md} \gg T^{md}_{bulk} + T^c$ has to be true. A sensible gain would be a factor of $T^{md}/T^c = O(10)$ in computational time reduction, for the coupling methodology to be worthwhile.

3.4.1 The Overlap Region

The overlap region, also branded as the (HSI interface in the specialized literature [7]), is the central concept in a coupled scheme by domain decomposition. It defines, on the physical side, the volumes of the MD and CFD sub-domains which are geometrically coupled; on the computational side, the group of MD and CFD processes which intervene in the concurrent exchange of data throughout a coupled simulation (see Figure 4.2). In this work, the overlap region is defined in terms of a portion of the CFD mesh. In general, the CFD mesh can be unstructured, but the mesh portion delimiting the overlap region, has to be always structured. This is a current limitation of the `CPL_LIBRARY` [111] implementation. This is shown in Figure 3.3 where the black cells constitute the *coupling cells*, with a defined volume $V^c = d^c_x \times d^c_y \times d^c_z$, where d^c_x, d^c_y, d^c_z are the cell dimensions along the three spatial directions.

Within the overlap region we can find two subregions named the boundary condition (BC) and the constrained region (CNST), whose combined volume does not necessarily have to occupy the whole overlap region (see Figure 3.1). The size of the CNST and BC regions are always a multiple number of the coupling cell sizes d^c_x, d^c_y and d^c_z. The main purpose of these two regions, is to define the volumes in which the numerical coupling algorithms from Sections 3.4.5, 3.4.6 and 3.4.7 are applied. Two main approaches exists on how to impose boundary conditions to achieve coupling: in the form of state variables (*Dirichlet*) or in terms of fluxes (*Neumann*). This leads to four different schemes based on the possible combinations. These schemes are VV, FV, VF and FF where V accounts for the state variable (e.g flow velocity) and F is a flux (e.g momentum flux). The first

and second letter refer, respectively, to the strategy employed in the CNST and the BC region. An insightful analytical and numerical study of the convergence and stability of these four schemes was carried out by Ren [147]. It was found that FV scheme has superior convergence rate unrelated to the size of the overlap region, only depending on the relative size of the atomistic region with respect to the continuum one. Thus, using the FV scheme does not introduce any further constrains on the size of the overlap region. Furthermore, fluxes are orders of magnitude noisier than state variables in MD. Therefore, the amount of spatio-temporal sampling needed to improve the signal-to-noise ratio to fulfill condition C3 in Table 3.7 would increase dramatically with all-atoms potentials. However, a scheme fully based on fluxes (FF) would be, *a priori* desirable from the mathematical standpoint, since it is the only one fully conservative [124]. The FV scheme is chosen in this work as a coupling strategy since it represents a good compromise between accuracy and aspects of the implementation, including convergence and stability, which are loosely defined in Section 2.3.3.

Section 5.5 is devoted to the study of a positive *feedback-effect* consequence of the particular structure and dynamics of the overlap region. The phenomenon is characterized by the amplification over time of the noise in the velocity field sampled within the BC region in the MD domain. This leads eventually to an unphysical oscillatory steady-state of the velocity field across the simulation domain. This effect, which has not been reported before in the literature, has been found to be particularly harmful for the coupling methodology in high viscosity fluids. Some guidance on how to mitigate this effect are given at the end of Section 5.5, but a solid strategy to fully overcome it is yet to be devised and it is proposed as future work in Section 6.3.

3.4.2 Synchronized Coupling

An important aspect of the coupled strategy is how and when the CFD and MD domains will interchange data in the overlap region. A *synchronized coupling* approach [148], in which both domains advance in time independently until they reach a fixed instant in time (squares and circles in

Figure 3.5), has been selected. Data transfer inside the BC and CNST regions takes place at those synchronization points. This is a scheme that benefits from parallelization since it allows for the MD and CFD to be solved concurrently [148]. Synchronized, concurrent data exchange has been implemented by utilizing the `CPL_LIBRARY`, which provides an interface to couple two solvers based in domain-decomposition parallelization through `MPI` communication. Implementation details are given in Section 4.2.

The *time-step ratio* (TSR) between the MD and the CFD is defined as a positive dimensionless integer number

$$TSR \equiv \frac{\Delta t_{cfd}}{\Delta t_{md}}, \tag{3.4.1}$$

where Δt_{cfd} and Δt_{md} are, respectively, the time-step magnitude of the CFD and MD solver. It is assumed that $\Delta t_{cfd} > \Delta t_{md}$ always and TSR represents the number of MD steps run per CFD step. Averaging of a quantity q in the MD domain is performed over the duration of a CFD time-step $\Delta t_{avg} = \Delta t_{cfd}$ with a sampling interval

$$\Delta t_s = K_s^q \times \Delta t_{md}, \tag{3.4.2}$$

where $K_s^q \in \{1, 2, ..., TSR\}$ is the number of MD steps between samples. A flaw of this strategy is a lag of $\Delta t_{avg}/2$ between the CFD and the MD domain as MD quantities are time-averaged around the mid-point time $t_{i-1} + \Delta t_{avg}/2$ instead of t_i. This has a negligible impact on the coupling between both domains as long as Δt_{avg} is small enough.

A large TSR is, in principle, always desirable in order to minimize the frequency of data exchange. This reduces the amount of `MPI` communication performed by the `CPL_LIBRARY` and the amount of coupling-related operations executed in both domains as a consequence of the boundary conditions update. Both aspects have a positive impact on the parallel efficiency of the coupled framework. This has been studied in Section 4.4, which contains scalability studies analyzing the parallel performance of different coupling scenarios.

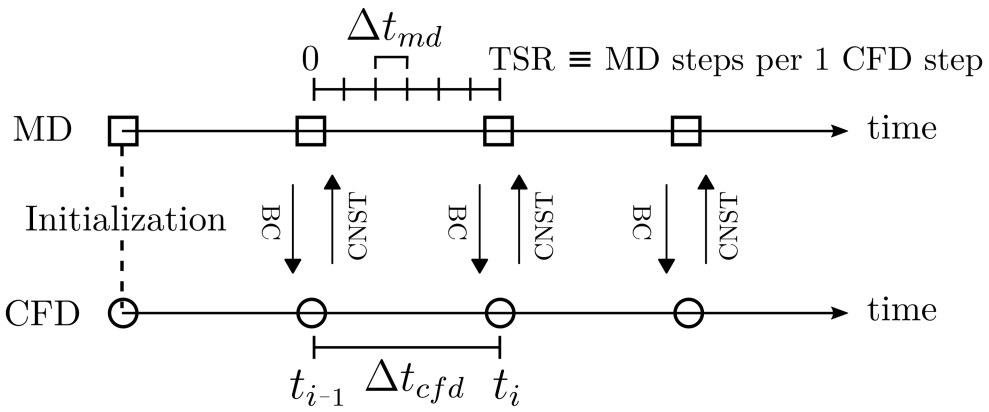


Figure 3.5: Synchronized time-coupling strategy for a MD-CFD concurrent hybrid scheme. Squares and circles represent synchronization points in time for the CFD and MD domains. After an initialization stage, MD and CFD processes exchange data in the CNST and BC regions (see Figure 3.1) every $\Delta t_{cfd} = TSR \times \Delta t_{md}$.

3.4.3 Temporal and Spatial Constraints

In general, CFD mesh cell dimensions Δx and $\Delta^i y$ (depends on cell i), TSR and Δt_{cfd} are not free simulation parameters. A set of constraints arising from numerical and physical considerations listed in Table 3.7 need to be satisfied. It is assumed the MD time-step, Δt_{md}, is a fixed parameter since its size is bound to the particular force-field employed and the system temperature (the higher the temperature the smaller the time-step). From a molecular perspective, to fulfill the continuum fluid dynamics local equilibrium assumption, the size of the cells has to be bigger than the mean free path of the molecules (condition C1.2) and the CFD time-step greater then the particle velocity auto-correlation time (condition C1.1) [112]. Furthermore, in order to time-resolve the flow, we need to ensure that the averaging time of MD quantities Δt_{avg} is much smaller than the characteristic flow time τ_{flow} (condition C2.1). τ_{flow} can be defined as the diffusion time $\tau_{flow} \equiv L_y^2/\nu^{cfd}$ for Couette-like flows (Figure 5.14) and $\tau_{flow} \equiv \tau_{osc}$ for oscillating flows (Figure 5.17). The appropriate spatial resolution is guaranteed with condition C2.2, by choosing an appropriate level of mesh refinement. Since the CFD meshes employed can be graded

along the y direction (see Figure 3.3), the mesh cell size $\Delta^i y$ is labeled with the cell index i. In order to provide robust BCs values to the CFD from the MD, the signal to noise ratio (condition C3) of the quantities computed in the BC region has to be sufficiently high. By looking at Equation 3.4.4, the SNR can be improved by incrementing both the averaging time Δt_{avg} or the averaging volume V_i. Lastly, for explicit numerical solvers the CFL condition [6] needs to be fulfilled for convective C4(1) and diffusive C4(2) terms to be solved.

Physical condition	**Temporal constraint**		**Spatial constraint**	
Local equilibrium	C1.1	$\Delta t_{cfd} > \tau_v$	C1.2	$\Delta x > \mu$
Flow resolution	C2.1	$\Delta t_{avg} < O(0.1)\tau_{flow}$	C2.2	$\left\|\boldsymbol{\chi}^{-1} d\boldsymbol{\chi}/dy\right\| \Delta^i y < 1$
SNR	C3	$\langle \bar{\boldsymbol{\chi}} \rangle / \sigma_{\boldsymbol{\chi}>1}$		
Courant condition	C4	(1) $\Delta t_{cfd} < \left(\Delta^i y \Delta x\right) / \nu_i^{cfd}$ (2) $\Delta t_{cfd} < \Delta x / u^i_{flow}$		

Table 3.7: Constraints imposed in a coupled simulation on the CFD time-step Δt_{cfd}, the averaging time Δt_{avg} and CFD mesh cell dimensions Δx and $\Delta^i y$. Diffusion dominates over the y dimension and convection over the x direction. SNR stands for the Signal to Noise ratio of a particle variable $\boldsymbol{\chi}$ (see Section 3.4.4). τ_v and μ are respectively the velocity auto-correlation time and the mean free path of the molecules/particles and τ_{flow} is the characteristic flow time. The CFD kinematic viscosity ν_i^{cfd} and the local CFD flow u^i_{flow} are defined cell-wise by the cell index i.

3.4.4 Averaging of MD Quantities

In order to transfer information from faster time-scales and shorter length-scales, characteristic of particle dynamics, to a continuum coarse-grained description, an averaging procedure is employed. This is utilized in the BC region to provide boundary conditions from the MD to the CFD domain. Furthermore, to deal with transient, non-equilibrium scenarios, averages has to be taken with respect to the slower time scale and in the coarse-grained spatial coordinates. Since we employ $\Delta t_{avg} = \Delta t_{cfd}$, temporal averaging is

performed over the CFD time step Δt_{cfd}. Spatial averaging over volumes V_i is performed around coordinates $\boldsymbol{X}_i$ as shown in Figure 3.6. For any particle variable $\boldsymbol{\chi}_k$ we define the following averages:

$$\bar{\boldsymbol{\chi}}\left(\boldsymbol{X}_i, t_{cfd}\right) \equiv \frac{1}{N_i} \sum_{k \in V_i}^{N_i} \boldsymbol{\chi}_k, \tag{3.4.3}$$

$$\langle \bar{\boldsymbol{\chi}} \rangle \left(\boldsymbol{X}_i, t_{cfd}\right) \equiv \frac{1}{\Delta t_{cfd}} \int_{t_{cfd}}^{t_{cfd}+\Delta t_{cfd}} \bar{\boldsymbol{\chi}}\left(\boldsymbol{X}_i, t\right) dt, \tag{3.4.4}$$

where the first equation defines a spatial average over a volume V_i containing N_i particles and the second equation performs a temporal averaging $\langle ... \rangle$ over Δt_{cfd}. The times at which the averages are evaluated are the discrete times $t_{cfd} = K \times \Delta t_{cfd}$, with $K \in \{1, 2, ...\}$. In a FV coupling scheme, the mathematical condition to be fulfilled within the BC region is

$$\boldsymbol{\chi}^{cfd}\left(\boldsymbol{X}_i, t_{cfd}\right) = \left\langle \bar{\boldsymbol{\chi}}^{md} \right\rangle \left(\boldsymbol{X}_i, t_{cfd}\right), \tag{3.4.5}$$

where $\boldsymbol{\chi}^{cfd}$ is the continuum field in the CFD and $\left\langle \bar{\boldsymbol{\chi}}^{md} \right\rangle$ is the time averaged field in the MD following Equation 3.4.4. Averaging coordinates within the CNST and BC regions are labeled $\boldsymbol{X}_i^{CNST}$ and $\boldsymbol{X}_i^{BC}$ respectively in the following sections.

3.4.5 Mass Coupling

As explained in Section 3.2, the atomistic wall remains tethered to the initial lattice sites during a coupled simulation, so the volume of the combined coupled domain is always constant. Therefore, if a non-zero mass flux $\rho \boldsymbol{u} \cdot \boldsymbol{n}^{CNST}$ exists across the top boundary of the MD domain (see Figure 3.6), particles inside the CNST region must be removed (negative flux) or added (positive flux) to fulfill mass conservation, which can be done by employing a suitable particle insertion algorithm like USER [107] or FADE [149]. The number of particles/molecules per unit time crossing the top MD boundary at time t_{cfd} can be written as

$$s\left(\boldsymbol{X}_i^{CNST} + \frac{d_y^c}{2}\hat{\mathbf{j}}, t_{cfd}\right) = \frac{\rho \boldsymbol{u} \cdot \boldsymbol{n}^{CNST}}{m} A_{xz}, \tag{3.4.6}$$

where ρ and $\boldsymbol{u}$ are the local density and velocity of the CFD fluid at the $\boldsymbol{X}_i^{CNST} + d_y^c/2\,\hat{\mathbf{j}}$ coordinate ($\hat{\mathbf{j}}$ being a unitary vector in the y direction), $A_{xz} = L_x^{cfd} \times L_z^{cfd}$ and $\boldsymbol{n}^{CNST}$ is the normal of the top boundary plane in the CNST region as depicted in Figure 3.6.

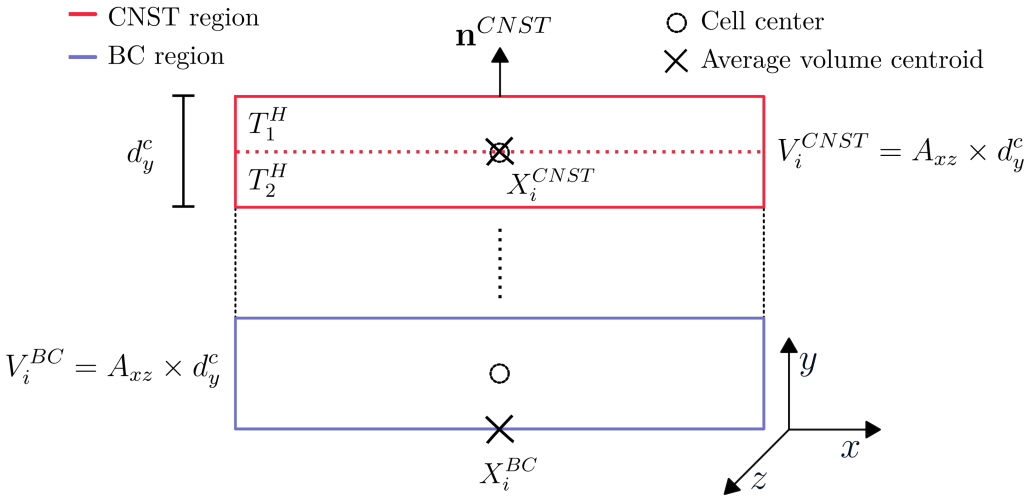


Figure 3.6: Diagram of the overlap region showing averaging points $\boldsymbol{X}_i^{BC}$ in the BC region, coincident with the CFD boundary condition, and $\boldsymbol{X}_i^{CNST}$ collocated with cell centers inside the CNST region. V_i^{BC} and V_i^{CNST} are the volumes of cell i inside each respective region. The top plane of the overlap region (top of the MD domain) is defined by the $\boldsymbol{n}^{CNST}$ normal. Two thermostats (T_1^H, T_2^H) are placed evenly distributed along the y direction in the CNST region. The diagram shows an overlap region with just one cell in the y direction inside both CNST and BC regions for simplicity.

Equation 3.4.7 assumes that all the molecules have the same mass m (single component fluid). At each CFD time step, $s(\boldsymbol{X}_i^{CNST}, t_{cfd})$ particles would have to be dynamically introduced (removed if $s(\boldsymbol{X}_i^{CNST}, t_{cfd}) < 0$) per unit time in the CNST region.

Under incompressibility conditions and if the variation of local density due to temperature changes is small, then $s(\boldsymbol{X}_i^{CNST}, t_{cfd}) \approx 0$ and mass coupling can be disregarded. Simulations carried out in this work in Section 5.3.5 and Section 5.4.3 did not implement mass coupling since density changes, under the selected simulation conditions can be neglected.

The equivalent expression for Equation 3.4.5 to achieve mass coupling within the BC region would be:

$$\rho\left(\boldsymbol{X}_i^{BC}, t_{cfd}\right) = \left\langle \frac{1}{V_i^{BC}} \sum_{k \in V_i^{BC}} m_k \right\rangle \left(\boldsymbol{X}_i^{BC}, t_{cfd}\right), \tag{3.4.7}$$

where m_k is the mass of the atom/molecule k within the volume V_i^{BC}.

3.4.6 Momentum Coupling

The coupling of momentum in the BC region can be enforced by an expression akin to Equation 3.4.5 by considering the volume V_i^{BC} with N_i^{BC} molecules inside

$$\boldsymbol{u}^{cfd}\left(\boldsymbol{X}_i^{BC}, t_{cfd}\right) = \langle \bar{\boldsymbol{v}}_{com} \rangle \left(\boldsymbol{X}_i^{BC}, t_{cfd}\right), \tag{3.4.8}$$

$$\bar{\boldsymbol{v}}_{com} = \frac{\sum_{j \in V_i^{BC}} \boldsymbol{v}_{com}^j}{N_i^{BC}}, \tag{3.4.9}$$

$$\boldsymbol{v}_{com}^j = \frac{\sum_k^{N_j} \boldsymbol{v}_k m_k}{M_j}, \tag{3.4.10}$$

where $M_j = \sum_k^{N_j} m_k$ is the mass of molecule j with N_j atoms and $\boldsymbol{v}_{com}^j$ is its center of mass velocity

$$\boldsymbol{R}_{com}^j = \frac{\sum_k^{N_j} m_k \boldsymbol{r}_k}{M_j}, \tag{3.4.11}$$

where $\boldsymbol{v}_k$, $\boldsymbol{r}_k$ and m_k are the velocity, position and mass of particle k within molecule j. Consequently, $\langle \bar{\boldsymbol{v}}_{com} \rangle$ and $\langle \bar{\boldsymbol{R}}_{com} \rangle$ are the respective time-averaged quantities computed using Equation 3.4.4 for the molecules inside the volume V_i^{BC}. Note that the mean center of mass $\langle \bar{\boldsymbol{R}}_{com} \rangle$, is not necessarily equal to $\boldsymbol{X}_i^{BC}$, which is both the coordinate at which the boundary condition is imposed in the CFD and the reference point for averaging in MD. However, for a sufficiently large V_i^{BC}, the difference between $\langle \bar{\boldsymbol{R}}_{com} \rangle$ and $\boldsymbol{X}_i^{BC}$ is minimal. The quantity $\boldsymbol{v}_{com}^j$ represents the

coarse-grained momentum associated to a single molecule, which allows treating poly-atomic molecules as point-particles from the standpoint of momentum coupling.

The continuum momentum flux $\rho \boldsymbol{u}\boldsymbol{u} + \boldsymbol{\sigma}$ applied to the particle ensemble within the CNST region, is computed at the top boundary plane of the MD domain defined by the $\boldsymbol{n}^{CNST}$ normal (see Figure 3.6). The CFD momentum flux is applied to the MD at time each time t_{cfd}, through the expression

$$ms\langle \bar{\boldsymbol{v}} \rangle + \sum_{k \in V_i^{CNST}} \boldsymbol{F}_k^{ext} = [\rho \boldsymbol{u}\boldsymbol{u} + \boldsymbol{\sigma}] \cdot \boldsymbol{n}^{CNST} A_{xz}, \tag{3.4.12}$$

where the $\sum_{k \in V_i^{CNST}} \boldsymbol{F}_k^{ext}$ term is a distributed external force over the atoms inside the volume V_i^{CNST}. By inserting Equation 3.4.6 into 3.4.12, it can be shown the rate of change of momentum due to convection and local stresses are correctly introduced if

$$\langle \boldsymbol{v}'_{com} \rangle \ = \boldsymbol{u}^{cfd}\left(\boldsymbol{X}_i^{CNST}, t_{cfd}\right), \tag{3.4.13}$$

$$\begin{aligned} \left\langle \sum_{k \in V_i^{CNST}} \boldsymbol{F}_k^{ext} \right\rangle &= A_{xz} \boldsymbol{\sigma} \cdot \boldsymbol{n}^{CNST}, \\ &= A_{xz}\left(p\boldsymbol{n}^{CNST} + \boldsymbol{\tau} \cdot \boldsymbol{n}^{CNST}\right), \end{aligned} \tag{3.4.14}$$

where $\boldsymbol{v}'_{com}$ is the velocity of the center of mass for the molecules inserted in the CNST region. The expression for the spatially-dependent distributed force $\boldsymbol{F}_k^{ext}$ employed by Flekkoy *et al.* [114] (see Table 2.5), was designed to ensure a limiting extension to the application of the hydrodynamic pressure p, as well as guaranteeing the existence of a small region where particles can be inserted with minimum risk of overlapping. These two benefits are not relevant under incompressibility conditions (since particle insertion is not necessary) and if a reflective boundary is employed as the termination of the MD domain. In these circumstances, the simpler form employed by Buscalioni *et al.* [112] (see Table 2.5), which does not depend on the particle label k, can be used instead

$$\left\langle \sum_{k \in V_i^{CNST}} \boldsymbol{F}_k^{ext} \right\rangle = \boldsymbol{F}^{ext} N_{V_i^{CNST}}, \tag{3.4.15}$$

$$\boldsymbol{F}^{ext} = -\frac{A_{xz}\boldsymbol{\sigma} \cdot \boldsymbol{n}^{CNST}}{N_{V_i^{CNST}}}, \tag{3.4.16}$$

where $N_{V_i^{CNST}}$ is the number of particles in the volume V_i^{CNST}. This form, as later explained in Section 3.4.7, simplifies the condition to be satisfied for heat dissipation (Equation 3.4.22). Additionally, employing a reflective wall in the y direction, the term $p\boldsymbol{I} \cdot \boldsymbol{n}^{CNST}$ becomes irrelevant, since the change of momentum in Equation 3.2.6 acts as an effective force to contain the particles from leaving the molecular domain. The effective pressure exerted by the reflective boundary condition is not necessarily the same as p but this seems to have very little effect on the outcome of shearing coupled simulations according to the studies carried out in Section 5.3.5. The expression for the distributed force used in this work is only dependent on the continuum shear stress tensor $\boldsymbol{\tau}$

$$\boldsymbol{F}_{\boldsymbol{\tau}}^{ext} = -\frac{A_{xz}\boldsymbol{\tau} \cdot \boldsymbol{n}^{CNST}}{N_{V_i^{CNST}}}. \tag{3.4.17}$$

3.4.7 Energy Coupling

The energy flux $\rho e\boldsymbol{u} + \boldsymbol{\sigma} \cdot \boldsymbol{u} + \boldsymbol{q}$ is composed, from left to right, by the advection, dissipation and conduction terms. Achieving energy balance in the CNST region requires the following equations to be fulfilled [112]

$$\langle \epsilon' \rangle = e, \tag{3.4.18}$$

$$\left\langle \sum_{k \in V_i^{CNST}} \boldsymbol{F}_k^{ext} \cdot \boldsymbol{v}_{com}^k \right\rangle = -A_{xz}\boldsymbol{\sigma} \cdot \boldsymbol{u} \cdot \boldsymbol{n}^{CNST}, \tag{3.4.19}$$

$$\left\langle \mathbf{J}^q \cdot \boldsymbol{n}^{CNST} \right\rangle = A_{xz}\boldsymbol{q} \cdot \boldsymbol{n}^{CNST}, \tag{3.4.20}$$

where $\mathbf{J}^q$ is the molecular heat flux defined in Equation 2.2.30, $\boldsymbol{v}_{com}^k$ is the center of mass velocity of molecule k inside the volume V_i^{CNST} and $\epsilon' = \epsilon'_k + \epsilon'_\Psi$ is the energy of the particles inserted/removed, which has a kinetic ϵ'_k and configurational ϵ'_Ψ contribution. The specific energy $e = u^2/2 + 3T/2 + \Psi$ includes the translational thermal kinetic energy and the configurational (potential) energy [112].

In order to satisfy Equation 3.4.18 (advection) the particles inserted have to obey (a) $\langle \epsilon'_k \rangle = u^2 + 3/2T$, which can be fulfilled by drawing their velocities, at the time of insertion, from a Maxwellian distribution for 3D velocities [2,112]

$$P\left(\boldsymbol{v}'\right) = \frac{1}{\left(2\pi k_B T/m\right)^{3/2}} \exp\left[\frac{-\left(\boldsymbol{v}' - \boldsymbol{u}\right)^2}{2k_B T}\right], \tag{3.4.21}$$

where $\boldsymbol{v}'$ is the velocity of the particle and $\boldsymbol{u}$ the local continuum flow velocity; (b) $\langle \epsilon'_\Psi \rangle = \Psi$ which can be satisfied by inserting the particle with the USHER algorithm [107], which employs such condition to search the energy landscape within the CNST region to find a suitable insertion position. In the current coupling strategy, particles do not leave nor enter the domain (close boundary MD) and fluids simulated are considered incompressible, therefore advection of energy is always satisfied.

In the case of energy dissipation, employing Equation 3.4.15 for the distributed force $\boldsymbol{F}_k^{ext}$, Equation 3.4.19 can be rewritten as

$$\begin{aligned} \boldsymbol{F}^{ext} \cdot \left\langle \sum_{k \in V_i^{CNST}} \boldsymbol{v}_{com}^k \right\rangle &= N_{V_i^{CNST}} \boldsymbol{F}^{ext} \cdot \langle \bar{\boldsymbol{v}}_{com} \rangle \\ &= -A_{xz} \boldsymbol{\sigma} \cdot \boldsymbol{u} \cdot \boldsymbol{n}^{CNST}, \end{aligned} \tag{3.4.22}$$

where $\langle \bar{\boldsymbol{v}}_{com} \rangle \equiv \langle \bar{\boldsymbol{v}}_{com} \rangle \left(\boldsymbol{X}_i^{CNST}, t_{cfd}\right)$. In order to satisfy Equation 3.4.22, it is easy to see that (a) $\boldsymbol{u} = \langle \bar{\boldsymbol{v}}_{com} \rangle$ and (b) $N_{V_i^{CNST}} \boldsymbol{F}^{ext} = -A_{xz} \boldsymbol{\sigma} \cdot \boldsymbol{n}^{CNST}$. Condition (a) implies velocity continuity across MD and CFD domains, which can be achieved if the momentum balance is satisfied as described in Section 3.4.6, in which case (b) is also satisfied.

In order to couple heat conduction between domains in the CNST (Equation 3.4.20) region, a heat current needs to be established from the

CFD into the MD. The phenomenological Fourier's law defines a heat flux linear with the local temperature gradient (first order diffusion process)

$$\boldsymbol{q} = \lambda^{cfd} \nabla T, \tag{3.4.23}$$

where λ^{cfd} is the heat conductivity in the CFD. Since the temperature is always a linear function inside a CFD cell by design (linear cell-to-face interpolation scheme) it is sufficient to create an equivalent linear temperature gradient in the molecular domain to match the continuum heat flux inside the CNST region. For this purpose, two Nosé-Hoover chain thermostats [150] with temperatures T_1^H and T_2^H are placed along the y direction as depicted in Figure 3.6. If $\Delta T = T_2^H - T_1^H \approx 0$ (locally quasi-isothermal) then a single thermostat is employed for simplicity. The temperature field can be computed by solving Equation 3.3.5.

In order to couple the MD and CFD temperature fields within the BC region, boundary conditions to Equation 3.3.5 are provided at each time t_{cfd} as

$$T\left(\boldsymbol{X}_i^{BC}, t_{cfd}\right) = \left\langle \bar{T}_k \right\rangle \ \left(\boldsymbol{X}_i^{BC}, t_{cfd}\right), \tag{3.4.24}$$

$$\bar{T}_k = \frac{1}{3N_i^{BC} K_b} \sum_{k \in V_i^{BC}}^{N_i^{BC}} m_k \left\| \boldsymbol{v}_k - \bar{\boldsymbol{v}}_{com} \right\|^2, \tag{3.4.25}$$

where $\bar{T}_k$ is the local kinetic temperature computed by subtracting the mean molecular flow velocity $\bar{\boldsymbol{v}}_{com}$ to the atomic velocities $\boldsymbol{v}_k$ of the particles inside the volume V_i^{BC}. K_b is the Boltzman constant and N_i^{BC} is the number of atoms in V_i^{BC}.

3.5 Chapter Summary

In this chapter, different mathematical and algorithmic aspects of the hybrid MD-CFD coupling methodology, as implemented in this work, have been presented. This multi-scale approach allows reaching larger length-scales otherwise impractical to simulate at present in full atomistic simulations.

Nevertheless, the MD domain remains the trickiest domain to configure and it is found to be the most critical building block of the scheme due to the spatio-temporal constraints it imposes on coupling. The contents presented in this chapter are listed next.

The details of the implementation of the molecular domain have been given in Section 3.2. The atomistic equations of motion of the fluid, along with the thermostatting strategy employed has been discussed in Section 3.2.1. A description of two different atomistic surface implementations based on a non-bonded LJ potential and a bonded harmonic potential, are given in Section 3.2.2. In addition, the equations of motion and thermostatting strategy were also provided for the atomistic surface. Furthermore, the implementation of periodic and reflective boundaries in MD, is examined in Section 3.2.3. This section also includes details about a parallel-efficient density correction algorithm (FOREV [104]) developed in `LAMMPS`.

On the CFD side, the continuum Newtonian and non-Newtonian momentum and thermal equations, their implementation in `OpenFOAM`, as well the numerical solvers and discretization schemes employed, are presented in Section 3.3.

Finally, Section 3.4 describes the coupling methodology implemented. The overlap region configuration, which is itself composed of the CNST and BC regions, is discussed in Section 3.4.1. Within the overlap, data is exchanged between domains by employing a synchronized process supported by the `CPL_LIBRARY` [111]. Also, the physical constraints arising from molecular and continuum physical, as well as purely numerical, considerations are discussed in Section 3.4.3. A FV coupling scheme has been chosen as a trade-off between accuracy and simplicity. Additionally, Sections 3.4.5, 3.4.6 and 3.4.7 provide respectively, mathematical expressions for coupling mass, momentum and energy fluxes in the CNST region. Averaging expressions used to provide coarse-grained boundary conditions to the CFD domain within the BC region have been presented in Section 3.4.4.

Chapter 4

Software Framework for Coupling

4.1 Introduction

With the advent of high performance computing, coupled multi-scale simulations have the potential to revolutionize science and engineering. The great strength of multi-scale techniques is the potential to sample properties of hierarchical systems by modeling events that take place across a wide range of time and length scales which go beyond the capability of any single method alone [90]. However, multi-scale techniques are yet to permeate into engineering applications. This is no surprise given the complexity of the techniques and the lack of availability of well-established and robust software packages implementing them. These types of techniques are commonly developed *in-house* by academic groups employing outdated software development practices and scarce documentation, which makes them difficult to achieve mainstream uptake. Furthermore, multi-scale simulations' workflows are complex and require multiple steps and acquaintance with

different simulation techniques. A diagram of the multi-scale workflow followed in this work is presented in Figure 4.1. This workflow is specific for MD-CFD domain decomposition coupling of fluids, but can give an idea of the complexities involved in multi-scale techniques.

i) Monolithic	ii) Library	iii) Framework
SediFOAM [151]	CPL_LIBRARY [111,154]	MUSCLE [157]
LIGGHTS/CFD-DEM [125]	MUI [155]	OASIS3 [159]
OpenFOAM/LAMMPS [152]	MaMiCo [153]	MCT [156]
		Precice [158]

Table 4.1: Table taken from [8] summarizing the different software packages available for coupling.

Domain decomposition-based concurrent coupling has seen rapid development, as it allows on-the-fly information exchange and interaction between multiple simulation sub-domains handled by different numerical solvers. To support the development of domain-decomposition techniques, a series of software packages have been recently developed. In recent work [8] we categorize them into three different types *i) monolithic ii*) *libraries* and *iii) frameworks*. Table 4.1 shows existing software packages for coupling under each category. Type *i), monolithic,* consists on pre-packaging two pieces of software as a single application, by compiling and linking them together and, as a result, a single binary is produced. This provides an advantage for the user as it simplifies the installation and allows a combined single interface for both codes. In type *ii), libraries*, coupling is handled through software libraries which facilitate information exchange between solvers. Libraries are lightweight and minimize the impact of coupling, making data exchange transparent through an explicit library call. Lastly, *frameworks* are codes which provide a series of tools and techniques as an environment for general coupling of two or more packages. Many of these frameworks are summarized in reviews [160,161].

In this work the `CPL_LIBRARY` was employed to couple `LAMMPS` and `OpenFOAM` solvers. The simplicity of the interface it provides, the easy of use

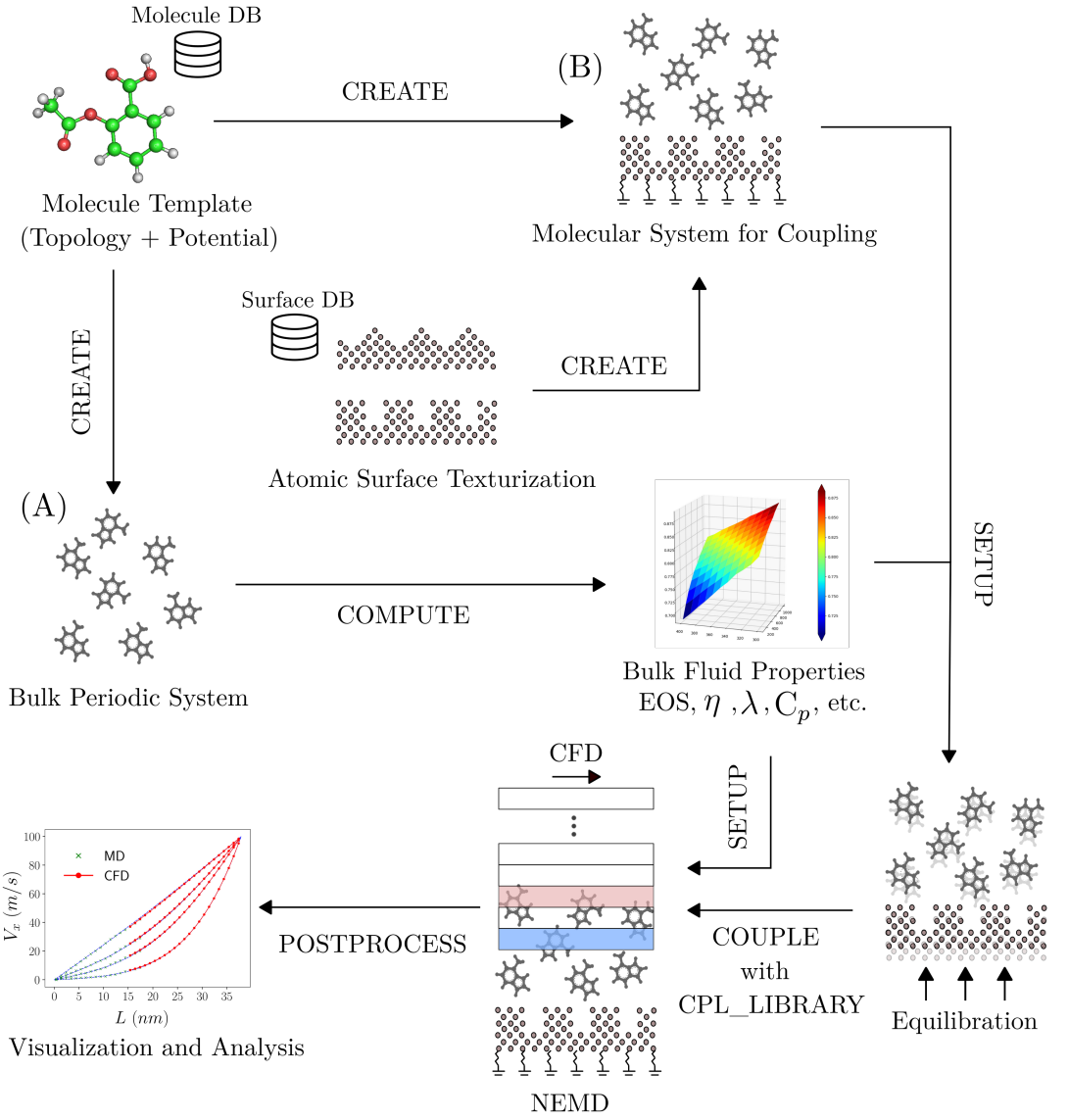


Figure 4.1: Schematic of the workflow to perform coupled MD-CFD simulations by domain decomposition. Two types of molecular systems are created: (A) bulk periodic systems employed to characterize the molecular liquid and (B) a confined system with an atomistic wall. From (A) liquid properties like the Equation of State (EOS) or viscosity (η) are obtained from equilibrium an non-equilibrium molecular dynamics simulations. System (B) is equilibrated and configured to be coupled with the CFD solver employing the `CPL_LIBRARY`. Properties computed from (A) systems are used as inputs for the CFD solver and to estimate reasonable values for coupling parameters (see Section 3.4). The MD and CFD solutions produced for each of the sub-domains, are post-processed and combined to reconstruct the final coupled solution.

and the possibility to be called from C++ and Python code made it the most sensible choice for both implementing the coupling interface and testing.

In this chapter, Section 4.2 presents how the CPL_LIBRARY operates and the programming interface it provides. In particular, Section 4.2.2 explains the object-oriented code infrastructure (CPLSocket) developed around the CPL_LIBRARY. The CPLSocket was employed in the implementation of LAMMPS and OpenFOAM coupled applications used in this work. Details about testing and deployment into HPC clusters are provided in Section 4.4 and the parallel scalability of the framework is analyzed in Section 4.3.

4.2 CPL Coupling Library and CPL Applications

The CPL_LIBRARY was designed as a minimal framework to support the development of multi-scale scientific software applications [8]. It facilitates efficient concurrent communication of two codes parallelized by domain decomposition employing MPI (Message Passing Interface). It also provides infrastructure to enable the user to test, in a systematic manner, each individual software component being coupled. This strategy allows developing each code independently with the CPL_LIBRARY acting as a communication interface between them. Hence, any solver which honors such interface can be coupled to the other end-point without having to re-compile any source code (see Figure 4.3). An alternative to this strategy, where a library acts as a gateway between solvers, development can be done in a *monolithic* fashion. Examples of monolithic codes typically occur in more applied applications, where the focus is on providing complete tools to solve engineering problems. For granular systems, open source examples include direct linking of LAMMPS-OpenFOAM, adaptations like SediFOAM and mixed commercial open-source such as CFDEM (see Table 4.1). The *library* strategy, although more involved, has several notable advantages:

a) *Independent development and testing of each coupled solver is much easier.* This is true because each source code and compiled binary is

independent of the other. This is exploited in this work by writing tests which validate the coupled MD solver and the CFD solver independently. This is achieved by *mocking* the counterpart solver with a `Python` script which implements the `CPL_LIBRARY` interface and sends/receives pre-defined data (see Figure 4.9). In Section 4.3 more details are given about the testing strategy followed.

b) *Version upgrading of a particular solver is less involved.* If a coupled 3rd-party solver is to be upgraded to a new version (e.g `OpenFOAM-3.01`→`OpenFOAM-4.01`), the source code of the `OpenFOAM` coupled application is not entangled with the MD code, facilitating the upgrading process.

c) *Avoid coupling incompatibility due to incompatible library dependencies.* A complicated issue to solve arises if two 3rd-party solvers to be coupled depend directly (or indirectly) on two different incompatible versions of a particular library. This is not uncommon provided many solvers depend on mainstream popular numerical libraries like `FFTW` [162], `SCALAPACK` [163], etc.

d) *Arbitrary parallel domain decomposition on each solver.* Each solver's domain can be partitioned independently without much trouble for the developer or the user. In this case, the `CPL_LIBRARY` works out which MD processes have to communicate with the CFD processes and handle this transparently, once the overlap region has been defined.

The benefits of using the `CPL_LIBRARY,` outweigh the initial extra effort necessary to familiarise with the package and the current work has benefited massively from an early adoption of it. The next section presents a summary of the `CPL_LIBRARY` design and the `API` it provides.

4.2.1 `CPL_LIBRARY` Design and API

The `CPL_LIBRARY` is designed to handle effectively communications between two coupled solver codes which use domain decomposition parallelization using `MPI`. Here we refer only to the CFD-MD coupling case, but the library

can be used with any pair of solvers, if at least one of them discretises its domain by means of a structured grid (solver mesh) [8]. Structured meshes are currently the only type explicitly supported by the `CPL_LIBRARY`, but interpolation schemes could be applied to map unstructured to structured meshes to bypass this limitation [164,165]. Focusing on uniform grids allows making use of `MPI` Cartesian communicators for intra-solver data transfers and `MPI` graphs for the inter-solver communication [9,166]. As `MPI` is often built tailored for a particular supercomputer architecture, using intrinsic `MPI` Cartesian and graph functions can potentially enhance communications for a given platform by exploiting architecture-specific optimizations of the bespoke `MPI` build. Furthermore, two types of communication strategies based on (a) `MPI_Open_Port` and (b) Multiple Program Multiple Data (`MPMD`) execution mode of `MPI`, are supported by the `CPL_LIBRARY`. Strategy (a) is less intrusive as each group of processes from each solver, "lives" inside its own independent `MPI_COMM_WORLD` communicator. This avoids accidental undesired cross-talk between MD and CFD processes. In strategy (b), a single `MPI_COMM_WORLD` communicator is created containing all CFD and MD processes and has to be partitioned and replaced so the intra-solver communications work as expected. This is the strategy employed in this work, since it was found that not every HPC facility supported multi-node execution of `MPI` processes based on `MPI_Open_Port.` Using `MPMD` mode the `CPL` coupled solvers can be executed as:

```
mpirun -n N1 CPLCfdSolver : -n N2 CPLMdSolver,
```

where `N1` and `N2` is the number of processors allocated for the `CPLCfdSolver` and `CPLMdSolver`. Additionally, despite the term "coupling" appears in the library's name, it does not contains any of the coupling algorithms' implementations presented in Section 3.4. These algorithms have to be implemented within the MD and CFD codes (e.g applying momentum constraints to the atoms within the MD).

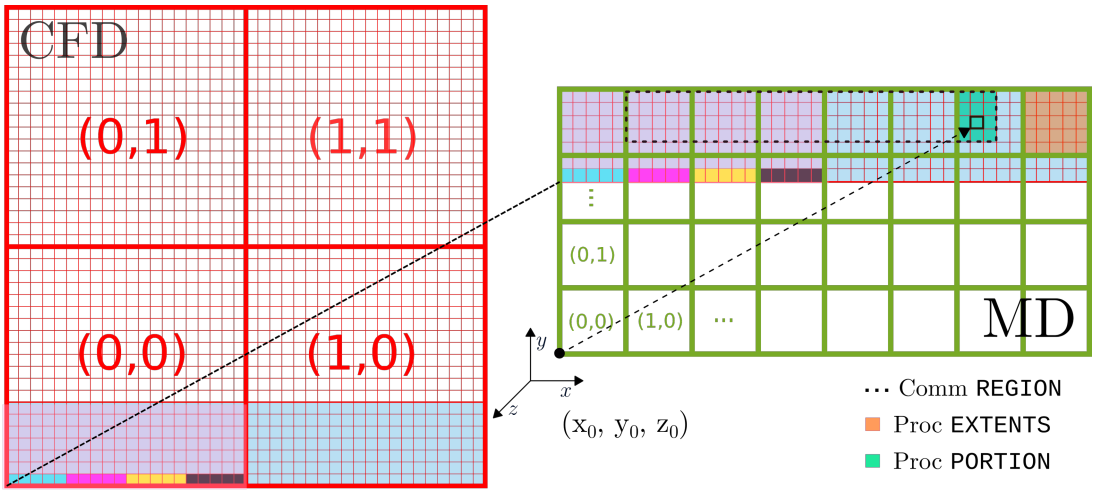


Figure 4.2: Schematic of the topological mapping between CFD and MD processors provided by the `CPL_LIBRARY`. A 2D view is presented and z dimension has been left out. The overlap region between MD and CFD domains is depicted in light blue with a shaded region indicating the overlap between the (0,0) CFD processor and eight MD processors (configuration particular to the example covered in this figure). The thin red grid represents the CFD mesh. This is also present in the MD domain where each cell acts as a coupling cell for taking averages and applying constraints (see Section 3.4.4). Colored cells inside the shaded region show the equivalence of a cell inside the CFD domain and a physical region in the MD domain. In the MD domain the concept of a communication `REGION` is depicted by a dotted rectangle (not present in the CFD for clarity of the diagram). A `REGION` is represented by a sextuplet $[c_{x0}, c_{x1}, c_{y0}, c_{y1}, c_{z0}, c_{z1}]$ which contains the cell range within the overlap region. The processor `EXTENTS` is the cell range of the overlap region contained within it (orange-shaded area). The processor `PORTION`, with respect to a particular `REGION`, is the cell range withing that `REGION` that is inside the processor (green-shaded area). Both quantities are represented by a sextuplet as well. The origin of the coupled domain is located in the lower-left corner.

The `CPL_LIBRARY` works in three phases: (1) the initialization, (2) communication (send/receive operations) and (3) finalisation. Table 4.2 contains a list of the most relevant functions provided by `CPL_LIBRARY`. In the initialization phase, inter-solver communication channels (`MPI` communicators) are created based on physical processor overlapping in the overlap region (see Figure 4.2). Furthermore, each coupled solver specifies the

Function Name	Description
CPL_INIT()	Initialize MPI communications inside the overlap region for MD and CFD processors which are physically overlapping.
CPL_SETUP(…)	Read CPL configuration file and broadcast its contents to all MD and CFD processes. CFD mesh parameters and domain origin coordinates are specified for both domains.
CPL_SEND(DATA_ARRAY, REGION)	Send DATA_ARRAY from a REGION (specified by a cell range). Communication only takes place if the process to execute CPL_SEND intersects the REGION. Only the PORTION (see Figure 4.2) of the REGION contained in the current process is sent.
CPL_RECV(DATA_ARRAY, REGION)	Receive DATA_ARRAY from a REGION (specified by a cell range). Communication only takes place if the process to execute CPL_RECV intersects the REGION. Only the PORTION (see Figure 4.2) of the REGION contained in the current process is received.
CPL_MAP_CELL2COORDS(CELL)	Obtain the COORDS associated to the center of a CELL. Returns FALSE if the cell is not present in the process.
CPL_MAP_COORDS2CELL(COORDS)	Obtain the CELL containing a COORDINATE. Cell boundaries are treated specially. Returns FALSE if the cell is not present in the process.
CPL_FINALIZE()	Finalize communications and clean MPI data structures. Wait for all CFD and MD processes to finish.

Table 4.2: A list of the most relevant functions provided by the CPL_LIBRARY [111]. The concept of REGION and PORTION are presented in Figure 4.2. A CELL is a triplet $[c_x, c_y, c_z]$ indicating the cell index within the overlap region and COORDS is a triplet $[x, y, z]$, representing a 3D coordinate local to the domain of the solver where the function is called.

physical domain origin and dimensions, the simulation total time, time stepping information and their internal *Cartesian processor topology* [9]. For the CFD code (or the code implementing regular discretization), the number of mesh cells in each dimension is provided to the `CPL_LIBRARY`. The remaining coupling-related parameters, including the dimensions and cell number of the overlap, CNST and BC regions (see Figure 3.1) and the time-step ratio between both solvers, are specified in a separate configuration file. Internally, the library performs the mapping between overlapping processors in both solvers as outlined in Figure 4.2 for the MD-CFD case. This mapping reduces spurious communication which can affect the computational efficiency of the coupled application (computed from Equation 4.4.2). The `CPL_INIT()` and `CPL_SETUP()` methods are called during the initialization phase. Once this phase is completed, `CPL_SEND` and `CPL_RECV` can be invoked to send data from a certain region located within the overlap limits. The process calling the function will send/receive the `PORTION` of the region inside it. This is explained in Figure 4.2. The last phase is cleanup of data structures and waiting synchronously for all the CFD and MD processes to finalize their execution. For this purpose the `CPL_FINALIZE` is provided.

4.2.2 CPL Applications. Object-Oriented CPLSocket Interface

The `CPL_LIBRARY` provides a procedural interface inspired by the `MPI` standard. It was designed to keep its `API` lean and straight forward to use across different programming languages with *bindings* available for `Fortran`, `C/C++` and `Python`. However, object oriented programming has proven in the last decade to be the paradigm where professional software development has blossomed. `LAMMPS` and `OpenFOAM` themselves have been developed using an object-oriented approach and implemented in `C++`. Object orientation facilitates software design by abstracting concepts into entities (*objects* or *classes*) which can contain data in the form of fields (known as *attributes)*, and code in the form of procedures (known as *methods*). The data represents the state of the object and the methods the behaviour. Relations between

the different entities can be established, the *inheritance* relation being the most prominent one, in which one object inherits from another its attributes and methods (or a subset of them). To describe object-oriented architectures, a fully standardized modelling language `UML` (Unified Modeling Language) [167] exists, and has been employed in Figure 4.4 and Figure 4.6. A reduced list of `UML` notation is provided in Appendix A for reference.

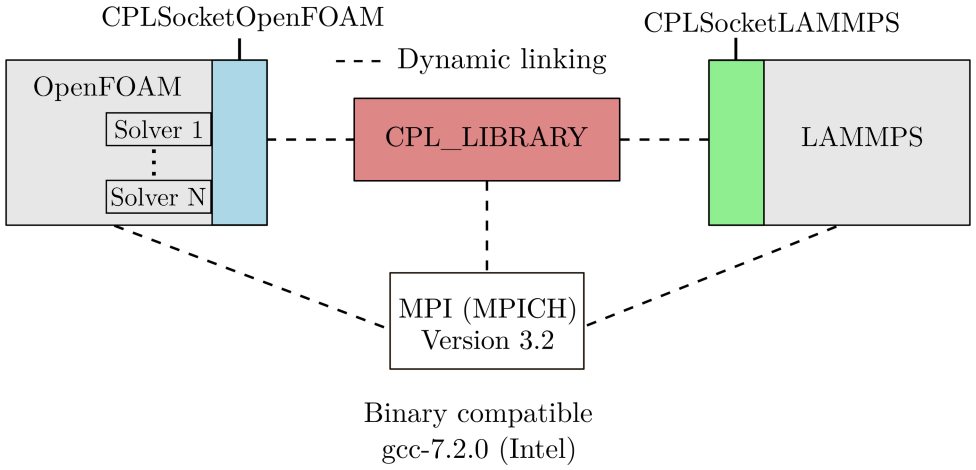


Figure 4.3: Dependency diagram for the `CPL` `OpenFOAM` and `LAMMPS` coupled applications, `MPI` and the `CPL_LIBRARY`. The two specialized classes `CPLSocketOpenFOAM` and `CPLSocketLAMMPS` implement the `CPLSocket` interface (see Figure 4.4). For `OpenFOAM`, multiple solvers can employ the `CPLSocketOpenFOAM` interface to communicate with `LAMMPS`, allowing at run-time to select which solver is the one to be used for a coupled simulation in a *plug-and-play* fashion.

To facilitate the development of the `LAMMPS` and `OpenFOAM` coupled applications, an interface based on *abstract* classes was implemented for `C++`. Table 4.3 lists the most important classes provided by the object-oriented interface and some other relevant 3rd-party classes from `LAMMPS` and `OpenFOAM`. Solvers can communicate through the `CPLSocket` interface which provides a set of methods to be overridden (*virtual* methods) by specialized child classes `CPLSocketOpenFOAM` and `CPLSocketLAMMPS` as explained in Figure 4.4. The `CPLSocket` defines methods `initComms`() and `init`(), to initialize the `CPL_LIBRARY`, provided an implementation is given to the

abstract methods `setTimingInfo()`, `setCartCommInfo()` and `setRealDomainInfo()`. It also defines `allocateBuffers()` to allocate memory for the attributes which hold the buffers `sendBuffer` and `recvBuffer`. Both are implemented as `CPL::ndArray` (see Table 4.3). The `setTimingInfo()` method is used to extract the timing information of the solver, `setCartCommInfo()` to extract the underlying Cartesian communicator topology of a solver and `setRealDomainInfo()`, the physical domain dimensions of the solver. A method `send()`, for sending an `OutgoingFieldPool` and `receive()`, to receive an `IncomingFieldPool` are provided. These pools of fields, are a collection of `IncomingField`/`OutgoingField` objects (see `UML` diagram in Figure 4.6) which represent physical fields being received from or sent to one coupled solver to another. For example, in our CFD-MD coupling scheme, the velocity field is an `IncomingField` for the BC region in the CFD and an `OutgoingField` in the MD. The ancestor class `TransmittingField` provides two important attributes. A `PortionField` which contains the cell ranges of the processor `PORTION` inside a particular communication `REGION` and a `BufferPtr` object, which gives direct access to the portion of memory allocated for the field on the corresponding communication buffer `sendBuffer` or `recvBuffer` (see Figure 4.5). `IncomingField` and `OutgoingField` define `setup()`, `update()` and `unpack()`/`pack()` methods respectively, which have to be implemented individually for each physical field to be transmitted within each solver. The `setup()` method performs allocation and initialization of any resources related with the field prior to starting the coupled simulation. The `unpack()` method is responsible for extracting the field data from the `recvBuffer` for the local processor `PORTION` inside the communication `REGION` and feed the solver with it (e.g set boundary condition values at the appropriate mesh nodes). Conversely, the `pack()` method extracts the field from the solver and fills the `sendBuffer` with the data accordingly. Lastly, the `update()` method is called after the `unpack()` method for an `IncomingField` and before the

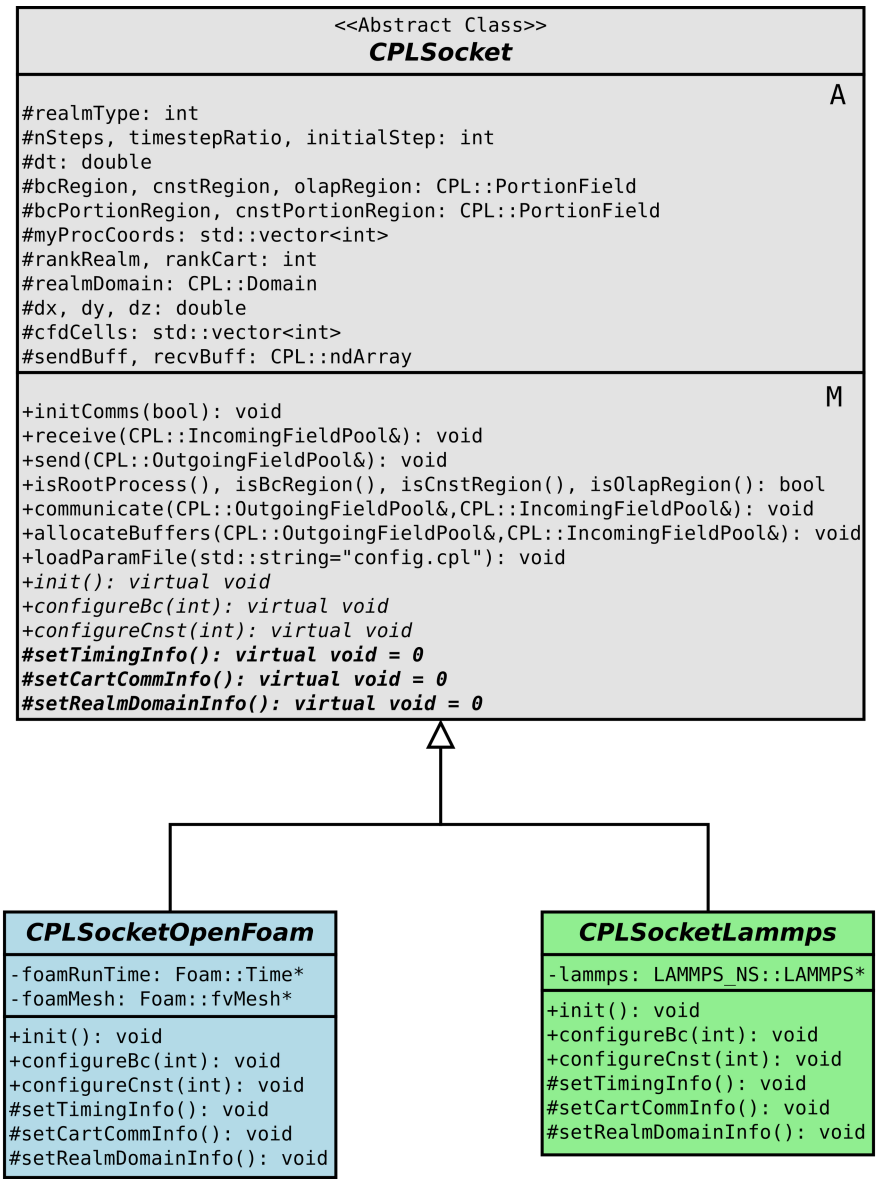


Figure 4.4: Simplified `UML` inheritance diagram for the `C++` `CPLSocket` interface provided by the `CPL_LIBRARY`. The child classes `CPLSocketOpenFoam` and `CPLSocketLammps` hold solver-specific data and must implement the virtual abstract interface of `CPLSocket`. Such interface is comprised of the methods `setTimingInfo()`, `setCartCommInfo()` and `setRealDomainInfo()` where, the timing information of the solver, the underlying Cartesian communicator topology and the physical domain dimensions are, respectively, extracted from the solver and passed to the `CPL_LIBRARY`. Non-abstract virtual methods `init()`, `configureBc()` and `configureCnst()` can be also re-implemented by the child classes if extra specific configuration is required for the initialization of the `CPL` application or the configuration of the BC or CNST regions.

`pack()` method for an `OutgoingField` to do any post-unpack or pre-pack operation respectively (e.g recompute a boundary field). To minimize the network communication time, a single call to `send()/recv()` is required by employing a combined single buffer (see Figure 4.5) for each pool of outgoing and incoming fields. This reduces the number of communication round trips with respect to performing a `send()/recv()` call per field. This is important to minimize network latency impact, which as shown at the end of Section 4.4.2, is the key aspect to improve parallel efficiency.

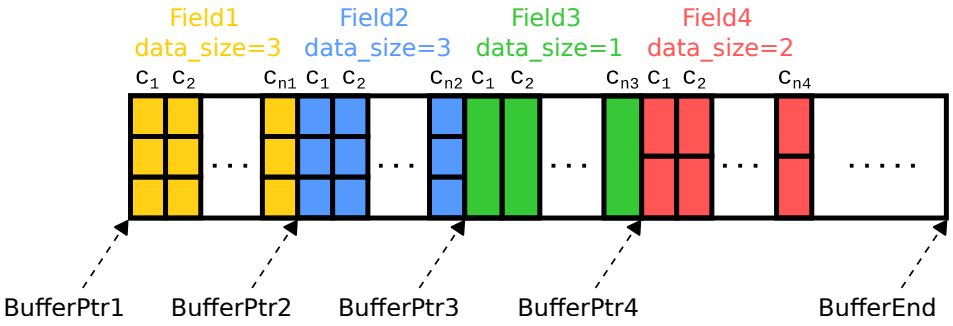


Figure 4.5: Graphical representation of a data buffer allocated by the `CPLSocket` object. The `sendBuff` and `recvBuff` attributes within `CPLSocket` are the sending and receiving buffer respectively. Each `TransmitingField` holds a `BufferPtr` object which contains a pointer to the beginning of its own data inside the send/receive memory buffer. The `TransmitingField`'s `data_size` attribute indicates the dimensionality of the data (e.g `data_size=1` for a scalar and `data_size=3` for a 3D vector). Each vertical rectangle represents, for a particular `Field1, Field2, etc.,` a cell within the range indicated by the `PortionField` region. Cells are labeled in the picture as $c_1, c_2, \ldots, c_{nk}$ for each `Field`k.

In addition, the event diagram in Figure 4.7 can be used to understand the timeline of operations during a coupled simulation. The methods' names provided by the object-oriented interface, are used to describe such operations as a sequence of events. The diagram shows how a single CFD process interacts with multiple MD processes. The frequency of data exchange between them is determined by the time-step ratio (TSR) between the MD and CFD solvers (see Figure 3.5). Synchronization points occur for

Initialization and Finalization stages, and for recv() operations, which make the processes invoking them to block (dashed lines in Figure 4.7). This is necessary to ensure that all communications, for a particular time-step, have been completed before the next communication cycle takes place. On the other hand send() operations are asynchronous and as soon as the process has sent the data, it will continue its execution. This scheme of communications enhances the parallel efficiency as it minimizes the number of blocking operations which are the root cause of idle times.

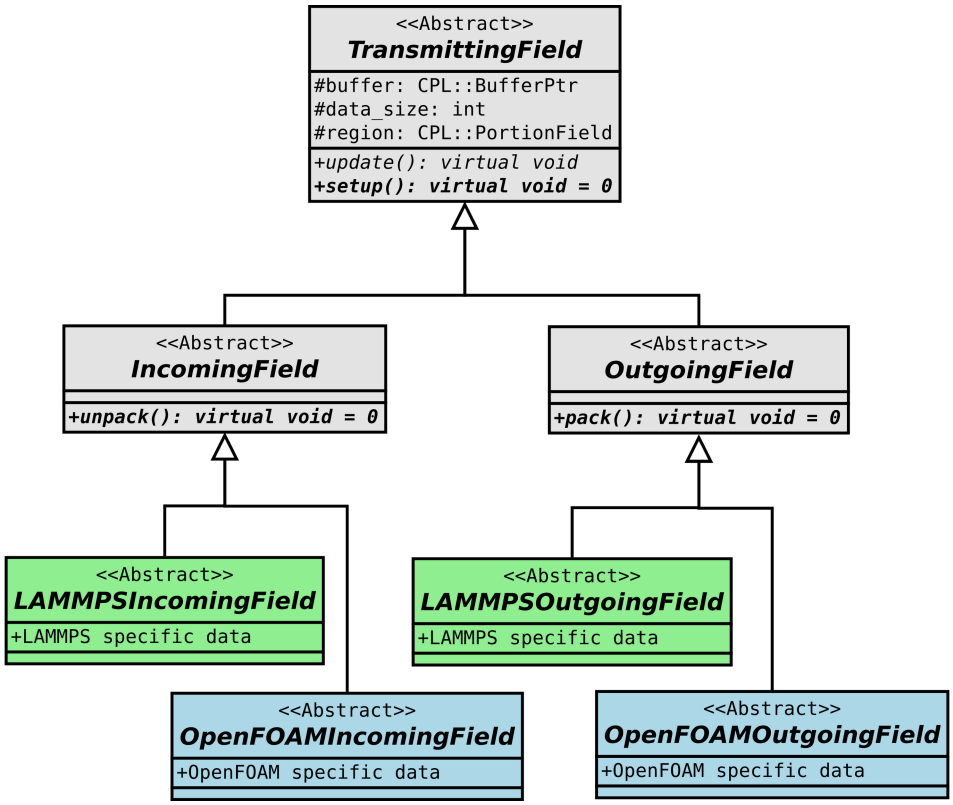


Figure 4.6: Simplified UML diagram for IncomingField and OutogingField classes provided by CPL_LIBRARY to facilitate writing C++ CPL applications. The TransmittingField class provides update() and setup() methods which must be overridden by child classes. Physical fields on each solver to be received/sent (green and blue boxes) can be created by sub-classing IncomminField/OutgoingField and implementing unpack()/pack() methods.

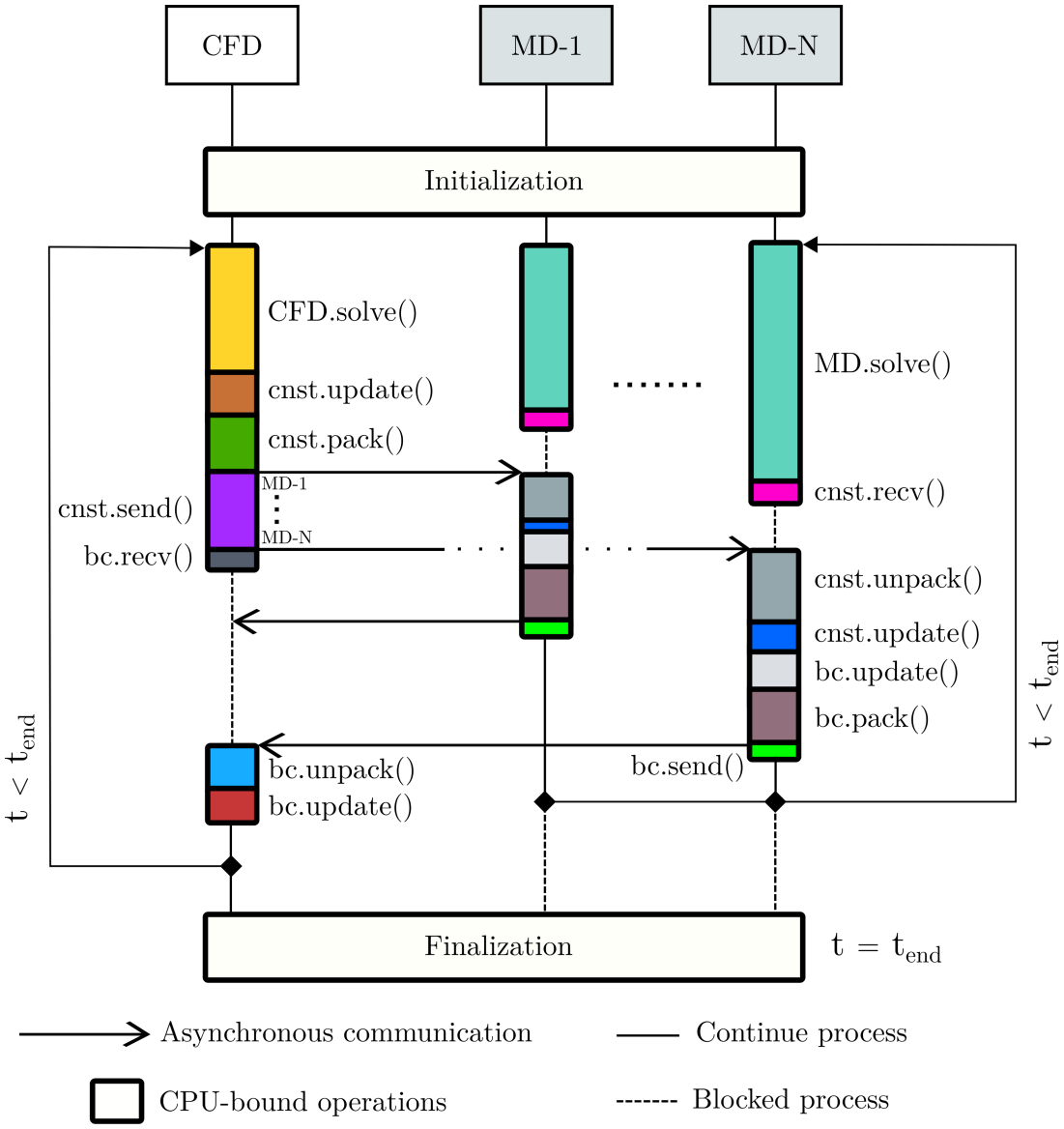


Figure 4.7: Event diagram for a coupled simulation. The `solve()` method indicates internal solver operations for the MD and CFD. This can be CPU or communication-bound depending on the configuration and internal partitioning of each solver. The diagram shows a single CFD process within the overlap region coupled to multiple MD processes. The rest of methods correspond to `CPLSocket` and `IncomingField/OutgoingField` interface. They are preceded by `bc` (boundary condition region) or `cnst` (constrained region) to indicate the region the fields are within. Furthermore, `send()` operations are always asynchronous and `recv()` synchronous (they block). `Initialization` and `Finalization` operations are both synchronous. Boxes with the same color indicate execution of the corresponding code.

Domain::ClassName	Description
(a) CPL C++ classes	
CPL::CPLSocket	Abstract class which provides the interface to communicate between C++ CPL applications.
CPL::IncomingField/ OutgoingField	Class representing a physical field to be received/sent by the CPL::CPLSocket.
CPL::Domain	Class representing a physical rectangular region.
CPL::PortionField	Class representing a physical field (e.g velocity) inside a processor PORTION.
CPL::ndArray	N-dimensional array ($N \leq 4$)
CPL::IncomingFieldPool /OutgoingFieldPool	Pool of CPL::IncomingField/OutgoingField objects being received/sent by the CPL::CPLSocket.
CPL::BufferPtr	Object which provides access to the memory location inside the send/receive buffer allocated for a TransmitingField.
(b) 3rd-party C++ classes	
FOAM::Time	Class holding information about timing in the OpenFOAM solver.
FOAM::fvMesh	Class representing OpenFOAM solver's mesh.
LAMMPS_NS::LAMMPS	Entry-point class to access LAMMPS molecular dynamics engine's internal data.

Table 4.3: List of the most important classes (a) provided by the CPL_LIBRARY and (b) provided by 3rd-party packages OpenFOAM and LAMMPS, to implement the C++ CPL applications developed in this work. The name of the classes include the C++ namespace identifier as a prefix.

4.3 Deployment and Testing of CPL Applications

High performance computing (HPC) clusters are very particular environments, which are in constant evolution to accommodate the growing needs in computational power of the scientific computing community. These computing clusters can be quite heterogeneous in terms of hardware architecture (Intel, ARM, IBM Power, etc.), operating systems, interconnect fabrics (e.g Ethernet, Omni-Path, Infiniband [26]), resource management and task scheduling software (eg. PBS, Slurm, LSF, SGE [169]) or even user security and data policies enforced by the system administrators to make use of the cluster. Such heterogeneity makes building software and executing computing workflows a very cluster-specific endeavor. Incorporating (*deploying*) new software to a cluster is commonly a task carried by HPC sysadmins. The usual practice is to install/build new software on demand and use a hierarchical *module system* like `LMOD` [170] to make it available as a loadable module to the HPC user. The module is then loaded with all the necessary dependencies for its execution. This system works well in general but is far from ideal for a number of reasons, particularly in scientific computing research:

a) *Convenience.* It is very convenient to build software once and be able to deploy and run it in any computing cluster regardless of the computing environment.
b) *Reproducibility.* It is not sufficient to share data and source code in order to make research fully reproducible. Variability in computing environments and software versions can influence results or even fail to compile and/or run (e.g an incompatibility of two versions of particular software can cause undetermined behaviour at runtime).
c) *Software development.* Developing software for HPC environments can become a daunting experience if the development cycle is taken out of the cluster. There is no guarantee that, modifying, rebuilding and testing code successfully in an external machine (e.g a personal

computer or a continuous integration framework like `Travis CI`) will yield the same outcome when the code is executed in the HPC cluster.

In this section the strategy followed to package and deploy the coupled software framework developed, is presented in Section 4.3.1. Also an overview of the comprehensive software testing strategy adopted is given in Section 4.3.2. For additional information about the software packages discussed here refer to the section Software Versions and Repositories in Appendix A.

4.3.1 Deploying CPL Applications and Reproducibility

The importance of reproducibility of results from HPC workflows and the deployment of HPC applications, has recently started to be a concern in the HPC scientific computing community and a number of strategies have become available to tackle them [171,172]. The pivotal idea is to create a self-contained exportable software environment, which can be moved between computing environments. We can identify three different layers of technologies/tools in order to (1) manage software dependencies, (2) isolate studies from the particularities of local computational environments, and (3) virtualize entire studies for complete portability and preservation.

Firstly, managing software dependencies effectively ensures that one can obtain the exact versions of all software libraries a computer program depends on at run-time (at the point the program is loaded into memory to be run). Because most software relies on external libraries and simulation/analysis workflows employ multiple tools, it is necessary to do book-keeping of numerous dependencies. This becomes difficult if an automated system is not used due to the variety of operating systems and local environment configurations that currently exist [173]. `Conda` [174] is a powerful package and environment manager which is operating system agnostic. It supports installation of specific package versions, and allows the creation of ‘‘virtual environments’’, which internally reproduce the file system’s structure and contain most of the libraries necessary to run within them a binary “shipped” by a `Conda` package. This self-contained environment solves the compatibility problem described in Figure 4.3.

Packages are built from the so called "package recipes", which are based on `YAML` configuration files containing all the necessary information to execute the build process and later installation. Alternatives to `Conda` are `Debian-Med` [175] and `linuxbrew` [176] but they are more rudimentary and do not support virtual environments. Even though `Conda` environments can be exported, there are core operating system libraries bound to the operating system's kernel, which cannot be ported from one computing platform to another using virtual environments. An example is the GNU C Library in Linux, which contains the OS *system calls* which allow a *userspace* application to perform *kernelspace* actions [177]. This limitation motivates the use of virtualization technologies which can provide further level of environment isolation.

Operating system level virtualization, also called *container-based* virtualization has gained popularity in recent years [172]. Here, the host's kernel allows the execution of multiple isolated userspace instances of these containers, which share simultaneously the same kernel, but might run a completely different software stack (system libraries, services, applications). In short, containers are run directly on the host operating system's kernel but encapsulate every other aspect of the runtime environment. Containers can therefore share resources with the host without incurring much of the performance penalties of hardware-level virtualization encountered in virtual machines (`VMs`). Thus, they provide a performant virtualization strategy with a level of isolation that is superior to `Conda` virtual environments, hence improving reproducibility of results and further minimizing library dependency mismatch on deployment. Containers can be regarded as light-weight virtualization. The most widespread implementation of containers is `Docker` [178], which is a very mature project, but has some limitations within HPC environments [172]. Firstly, HPC are multi-user environments with security policies difficult to implement in Docker, since for all practical purposes it gives superuser privileges inside the container. In addition, Docker architecture might also cause performance degradation of high performance applications based on `MPI` [179]. An HPC-oriented implementation of containers technology is `Singularity` [180] which

facilitates both user and system administration usage of containers in computing clusters by making easy network configuration, user permissions or data processing [180]. Furthermore, there is evidence that `Singularity` can achieve near-native performance for `MPI`-based applications [172,181]. Despite the multiple benefits of containerization, this technology was not employed in producing this work, since it was not yet supported by the HPC clusters we had access to. The deployment strategy has only focused on `Conda`, because despite being likely support for containers on HPC will be provided in the near future, we can easily build Docker and Singularity containers from `Conda` recipes [173].

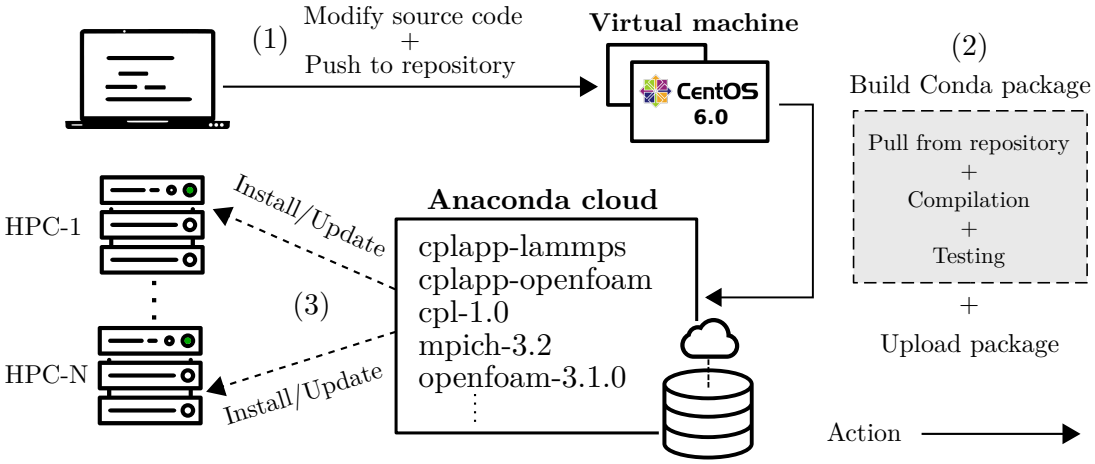


Figure 4.8: Development workflow followed in this work from (1) code modification and (2) `Conda` package building, to (3) deploying of the new packages necessary to perform MC-CFD coupled simulations in multiple HPC environments.

The workflow followed to deploy coupled MD-CFD applications into clusters is summarized in Figure 4.8 and it is comprised of three steps. First, a change is made to the source code of one of the coupled applications (`cplapp-openfoam` and `cplapp-lammps` packages listed in Figure 4.8) and pushed to the project repository. Then, a virtual machine with `Linux CentOS` is used to build a `Conda` package for the application. As mentioned before, `Conda` virtual environments, as opposed to containers, do not completely guarantee complete environment isolation. Hence, the package to build will

always have some system-level dependencies specific to the virtual machine's environment. Therefore, our best hope is then to build the package using a system environment which would guarantee best compatibility with most of the computing clusters. Linux `CentOS6.0` was selected as the OS for package building for two reasons: (a) the CentOS distribution is a non-commercial bifurcation of Red Hat Enterprise(RHEL) [182], by far the most popular Linux distribution in HPC clusters; (b) version 6.0 provides good forward compatibility for the `GNU LIBC` library and its dependencies. Lastly, the process of building a package consists of retrieving the last source code from the project repository, compiling it and then running a predefined suit of tests. Section 4.3.2 gives an overview of the testing performed in this stage. Only if all the test pass, the package is deemed good and then uploaded to `Anaconda` distribution cloud, where it can be fetched for installation on any cluster with HTTP traffic available.

4.3.2 Testing CPL Applications

Automated software testing has become an integral part of modern software development [183]. Adding new source code to a existing code base needs to verify it does not break any existing working functionality. Manual verification is error-prone and not scalable as the code grows in complexity and size. This is particularly relevant in collaborative software development as many scientific computing research projects do. For this purpose there exist language-specific frameworks like `Googletest` [184] for `C++`, `pytest` [185] for `Python`, `Junit` [186] for `Java`, etc. The idea behind automated software testing relies on triggering the execution of a suite of software tests at some point during the development cycle before new code is integrated into the code base. Passing all tests is a necessary condition for the code to be included. This process can be automated by using continuous integration (CI) frameworks like `TravisCI` or `Jenkins`. This testing methodology has been employed in this work in order to develop both the `CPL_LIBRARY` and `CPL` coupled applications.

The `CPL_LIBRARY` has a comprehensive set of unit tests itself. These include library initialization and topological setup tests; communication tests

for the supported range of processor topologies, overlap and grid setup options and memory leak checking using `Valgrind` [187]. Many of these tests are there to ensure safeguards against unreasonable configurations work properly, otherwise causing runtime errors which are difficult and time-consuming to fix for the user of the library. For specific details on the `CPL_LIBRARY` testing refer to Smith *et al.* [8]. More interestingly, the `CPL_LIBRARY` provides code infrastructure to implement a particularly useful testing strategy for coupled applications. This strategy is based on the concept of `CPL_Mocks`, where each coupled application (e.g `LAMMPS` MD engine) can be tested independently by *mocking* the counterpart coupled application (e.g `OpenFOAM` CFD solver) communicating with it. These mocks are implemented in `Python` scripts using the `pytest` package to exercise the test suite. This particular testing strategy is only possible thanks to the modularity provided by the coupling library, which allows development of each solver as an independent application and would have been impossible to implement in a monolithic approach (see Section 4.2). This strategy is represented in Figure 4.9 for the `CPLAPP-LAMMPS` application and it is described next. A similar testing procedure is used as well for the `CPLAPP-OPENFOAM` coupled application, but it is not presented here for brevity.

The testing procedure is represented as two independent set of states (numerated red and blue boxes in Figure 4.9) and the actions to transition between them are depicted as solid line arrows. The color of the boxes identifies a particular data flow path, through the BC and CNST regions. These are implemented as independent tests even though appear jointly in the same diagram. Let's first examine the BC testing procedure. Firstly, the coupled test infrastructure is initialized (0→1), then (1→2) a spatially-dependent physical field $F(t_i, x, y, z) \equiv F_i$ (velocity, stress, temperature, etc.) is generated for the current time step t_i and dumped to a file for later use. The generation of F_i is performed in `CPLAPP-LAMMPS` from the time-evolution of a predefined initial field F_0 (tests uses the LJ fluid model). Next (2→3), F_i is packed into the communication buffer (`OutgoingField.pack()` in Figure 4.6) and sent by using the `CPL_Socket` (see Figure 4.4) interface which `CPLAPP-LAMMPS` implements, and it is received by the `CPL_Mock` script. The

received field is labeled as G_i. Now, the field G'_i is reconstructed (this field could be scattered over multiple `CPL_Mock` processes) and dumped to a file (3→4). At this point, having F_i and G'_i saved, they are compared for each time step point-wise up to a certain tolerance making the condition for the test to pass at each time-step $F_i = G'_i$. Conversely, the CNST testing procedure follows similarly for the initialization step of the testing infrastructure (0→1) and the generation of the field F_i (1→2). However F_i is, in this case, generated as a random 3D array with the same number of cells in the three dimensions as the coupling grid within the CNST region specified as part of the initialization of the `CPL_LIBRARY`. Next (2→3), F_i is sent by using directly `CPL_send` (`Python` interface) and received as G_i on the other end by the `CPLAPP-LAMMPS`. The received field now is unpacked (`IncomingField.unpack()` in Figure 4.6) into internal `LAMMPS` data-structures (3→4) and applied the necessary updates (4→5) to the internal state (`IncomingField.update()` in Figure 4.6). Lastly, the field G'_i is dumped to file by extracting it from the internal data-structures of `CPLAPP-LAMMPS` application. Note $G_i = G'_i$ only if the application of the coupling algorithm inside `CPLAPP-LAMMPS` has succeeded (e.g stress imposed over atoms in the CNST region is consistent with the stress received $G_i = S_{xy,i}^{CNST}$). Similarly, the condition for the test to pass in this case is again $\forall t_i$, $F_i = G'_i$. As mentioned earlier, a similar testing procedure for both the CNST and BC regions has been developed for the `CPLAPP-OPENFOAM` but the description presented above is enough to illustrate the testing strategy followed.

These tests run when the `Conda` package, for each coupled application, is built in the `CentOS6.0 VM` as part of the workflow presented in Figure 4.8. In principle, if all the tests pass within the `Conda` virtual environment at the building stage, this gives confidence that the application will work as expected when it is deployed to a new computing environment. Nonetheless, the tests themselves are packaged as well with `Conda` and can be deployed and run locally in the new environment after installation, in order to increase the confidence or to diagnose environment-specific issues.

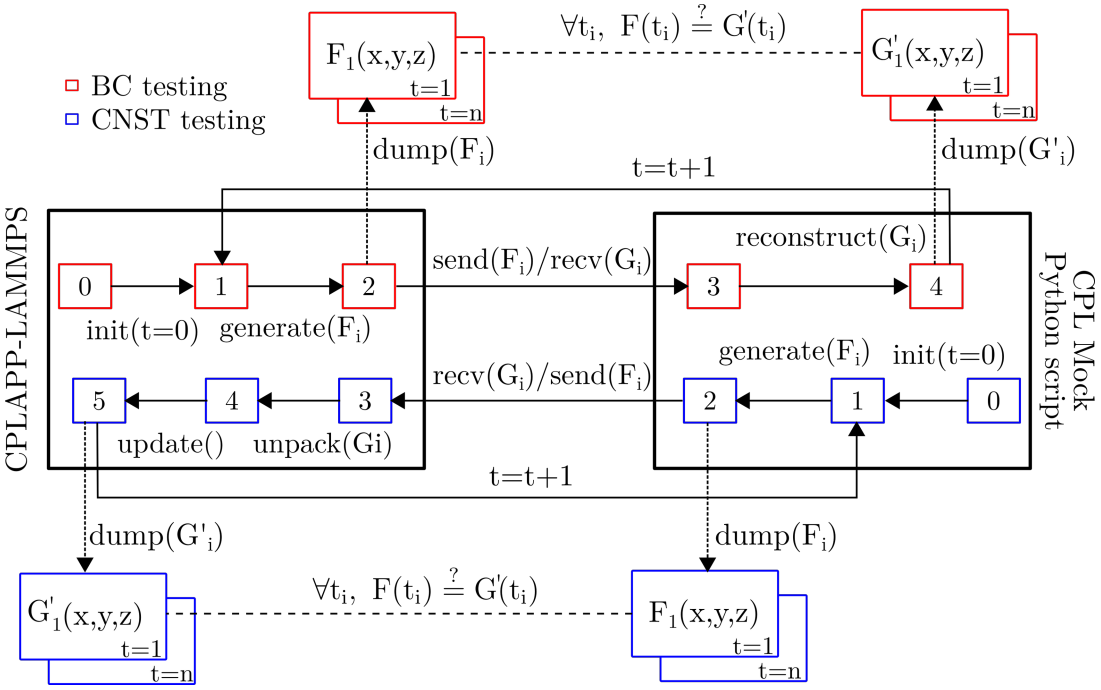


Figure 4.9: Diagram for the testing procedure for the `CPLAPP-LAMMPS` coupled application. The numerated boxes indicate different states in the procedure and the arrows the actions to transition between states. The actions do either take place in the `CPLAPP-LAMMPS` application or in the `CPL_Mock` `Python` script mocking the CFD solver. Red and blue colors indicate testing of the boundary condition region (BC) and constrained region (CNST). The fields G_i and F_i are, respectively, the received and transmitted fields at time step t_i. The field G'_i is the field dumped from the internal solver state after `unpack()` and `update()` has been executed (see Figure 4.7). The condition for a test to pass is G'_i and F_i to be equal for all times t_i. Tests are performed for a variety of different setups.

4.4 Framework Parallel Scalability

Scaling analysis of algorithms and software designed to run in parallel/distributed architectures is of paramount importance. There is very little point in devising an ingenious method to harness the computational power of high performance computing (HPC), if its implementation fails to scale reasonably as the use of computing processing units is increased. Therefore, it is important to design empirical *scalability tests* [188] which

provide a meaningful measure of performance. Scalability, within the HPC context, can be analyzed in terms of *strong scaling* and *weak scaling*. Strong scaling measures how the computing time varies with the number of processors, when problem size remains fixed. Conversely, how computing time varies if the problem size per processor is fixed (e.g fixed number of atoms per processor in MD), is known as weak scaling (see Figure 4.10). It is customary to introduce the concept of *speedup* of a parallel algorithm which is defined as

$$S(N,p) = \frac{T_s(N)}{T_p(N;p)} \equiv \frac{T_s}{T_p}, \tag{4.4.1}$$

where, N is a parameter defining the size of the problem, p is the number of processors, $T_s(N)$ is the time of the best known sequential algorithm and $T_p(N;p)$ is the elapsed time from the moment a parallel computation starts to the moment the last processor execution ends. Similarly, we can define the *parallel speedup* S^p when $T_s = T_{p=1}$, where $T_{p=1}$ is the time taken for the parallel algorithm using one processor. Strong scaling may be then used to determine the optimal number of processing units to be utilized and the maximum speedup achievable. On the other hand, weak scaling can be used to make predictions about how an implementation of a parallel algorithm, on a specific architecture, will perform for a large number of processors knowing how it performs for a few of them [189]. *Efficiency* is another measure which is commonly reported in scalability analysis and it defined as

$$E = \frac{S}{p} = \frac{T_s}{pT_p}. \tag{4.4.2}$$

Similarly to the parallel speedup, for $T_s = T_{p=1}$ we can define the *parallel efficiency* E^p, which we use in this work as a metric for studying strong and weak scaling as

$$E^p_{strong}\,(\%) = \frac{T_{p=1}}{pT_p} \times 100, \tag{4.4.3}$$

$$E^p_{weak}\,(\%) = \frac{T_{p=1}}{T_p(pN_0;p)} \times 100, \tag{4.4.4}$$

where E^p_{strong} is the *strong parallel efficiency*, E^p_{weak} is the *weak parallel efficiency* and $T_p(N_0 p; p)$ have the same meaning as T_p but considering explicitly that the problem size scales proportionally (from a reference problem size N_0) with the number of processors. Choosing $T_s = T_{p=1}$ makes sense in this work, since we are only interested on how a particular parallel algorithm scales in a distributed computing cluster and we are not concerned with a sequential implementation.

	Processor	**Interconnect**	**Bandwidth** (Mb/s)	**Latency** (μs)	**CPUs per node**
BP-Houston (this work)	Intel Xeon 2.60GHz	Custom (Ethernet)	1000	7-10	24
Cray XT5 (`LAMMPS`)	AMD Opteron 2.0GHz	Cray Cstar	1100	7	N/A

Table 4.4: Network and `CPU` specs for the computing cluster used in this work and one selected from `LAMMPS`' official benchmark [190] (N/A = no data available).

The framework presented in this work has been designed to couple MD and CFD codes implementing parallelisation through domain decomposition using `MPI`. The scalability of the coupling software cannot be tested in a standalone fashion, but has to be benchmarked against the most computationally demanding code, since is the one governing the simulation performance. In our particular case, `LAMMPS` present itself as the bottleneck since we are simulating millions of atoms in MD with respect to a couple dozens of CFD cells. Therefore, in order to measure the scalability of coupled simulations (see Section 4.4.2), the size of the OpenFOAM domain N_0^{OF} was kept constant and the size of the LAMMPS domain was scaled from a reference value N_0^{LMP}, hence we can write $T_p \equiv T_p(pN_0^{LMP}, N_0^{OF}; p)$.

ID	Software	Version	Compiler	MPI
LAMMPS-CONDA	LAMMPS-stable	11Aug2017	GCC 7.2.0	MPICH 3.2
LAMMPS-NATIVE	LAMMPS-stable	16Mar18	GCC 4.8.5	Intel-MPI 2018
OpenFOAM	OpenFOAM	3.0.1	GCC 7.2.0	MPICH 3.2

Table 4.5: List of software details used in this section for the scalability tests. The field "ID" is included as a shorthand to reference the software.

Scalability analysis is not, in general, transferable from cluster-to-cluster architecture. Bandwidth and latency of the network, CPU model and memory hierarchy or the level of congestion in the internal network could significantly alter the analysis [191,192]. Therefore, despite the existence of extensive benchmarks for LAMMPS [190], a scalability study of this software was carried out on the HPC cluster labeled as "BP-Houston" in Table 4.4, This is the cluster in which the majority of the coupled simulations have been run. This first batch of studies, provided scalability data against which we have compared the coupled framework (Figure 4.10). Every MD simulation performed in this section was done using the same setup utilized in the official LAMMPS benchmark [190]: Lennard-Jones liquid at reduced density $\rho = 0.8442$, LJ parameters $\sigma = \epsilon = 1.0$, force cutoff $r_{cutoff} = 2.5\sigma$ yielding an average of 55 neighbors/atom (within force cutoff), neighbor skin 0.3 and NVE velocity Verlet integration (in LAMMPS jargon). Information about versions and compiling information of the software used in this section can be found in Table 4.5.

4.4.1 LAMMPS scalability

A set of weak scaling tests for LAMMPS were conducted using two different builds of the package (see Table 4.5). Each of the simulations were run for 100 time-steps scaling from 1 to 4096 processors and keeping the molecular system density constant. The system size of 32K atoms per processor was selected in order to be consistent with LAMMPS official benchmarks [190]. The aim was firstly to compare the performance of the version compiled in the

external virtual machine using `Conda` (see Section 4.3) and linked to `MPICH`, against another one built specifically for the cluster architecture using `Intel MPI`. From Figure 4.10, it can be seen that for single node runs (processors < 24) `LAMMPS-NATIVE` is mildly faster than `LAMMPS-CONDA`, probably due to a better selection of compiler flags at build time for the cluster architecture. Nevertheless, for multi-node runs, `LAMMPS-CONDA` surprisingly exhibits better parallel efficiency and lower wall-clock time. This supports, in terms of computational efficiency arguments, the strategy for building and deploying the coupled framework presented in Section 4.3. On the other hand, both `LAMMPS-CONDA` and `LAMMPS-NATIVE` are always faster than Cray XT5 (see Table 4.4), although their parallel efficiency is significantly lower for multi-node execution. This is a reasonable outcome, since the BP-Houston cluster has a superior CPU. However, it is likely the custom Cray Cstar interconnect in Cray XT5 cluster provides more efficient network communications, which are a key aspect in multi-node runs and therefore impact parallel efficiency.

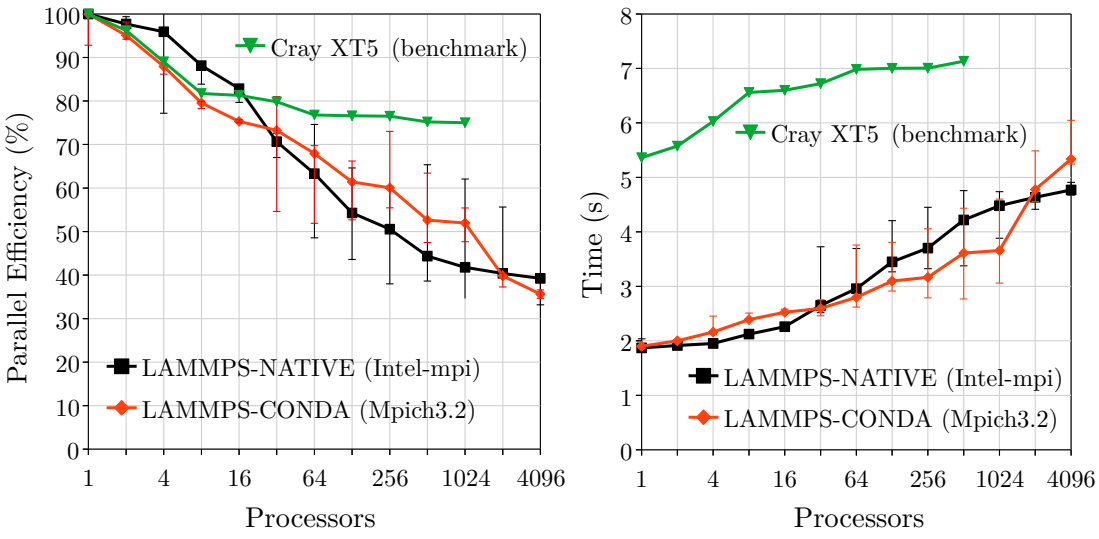


Figure 4.10: (a) Parallel efficiency and (b) total simulation time comparison between `LAMMPS-CONDA`, `LAMMPS-NATIVE` and one of the Cray XT5 `LAMMPS` benchmark for a Lennard-Jones fluid. The range of processors goes from 1 to 4096 in powers of 2. Similar performance is exhibited by `LAMMPS-CONDA` and `LAMMPS-NATIVE` but they compare poorly against Cray XT5 benchmark in terms of parallel efficiency.

To eliminate the possibility of the system size being too small to exhibit good weak scaling (low CPU load), the tests were repeated using `LAMMPS-CONDA` for five different reference system sizes N_0 (number of atoms per processor). As Shown in Figure 4.11, increasing the size of the system only enhances parallel efficiency from 512 processors onward, but not quite significantly. For 4096 processors there is only about 10% enhancement in efficiency with $N_0 = 32K$ particles per processor with respect to the one with $N_0 = 256K$ particles. The scalability of `LAMMPS` in the cluster, in absolute terms, was not a major concern for this work, but only how the coupled software compared against it. Therefore, no further investigations on how to improve the scalability of `LAMMPS` in the cluster were carried out.

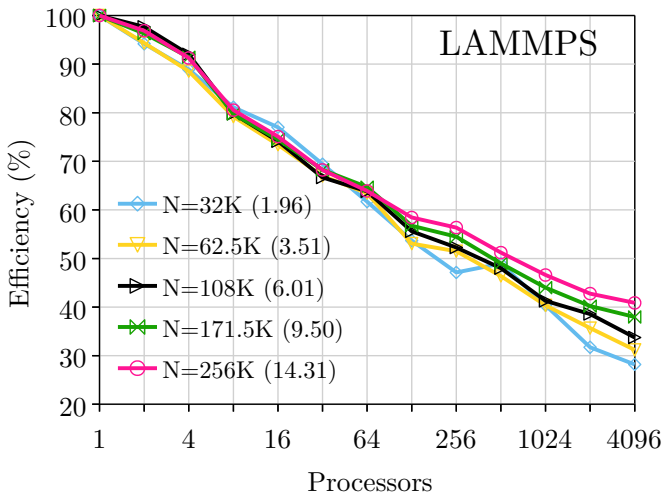


Figure 4.11: Weak scaling of `LAMMPS-CONDA` for a range system sizes N (number of atoms per processor). An increase of 8-fold in the problem size yields about 10% of improvement in parallel efficiency for 4096 processors. The total simulation time in seconds is in brackets for each case.

4.4.2 Coupled OpenFOAM-LAMMPS scalability

In the best-case scenario, the scalability of the coupled software would match the one exhibited by the most computationally intensive code. However, the overhead introduced by communication and the CPU time spent in coupling routines, could certainly deteriorate the parallel efficiency. Three parameters,

a priori, can be identified to affect directly the amount of *time per processor* spent in coupling-related operations (see Figure 4.7) and therefore potentially impact efficiency. Firstly, (i) the number of cells per processor K and (ii) the MD-CFD time-step ratio TSR, both can affect the CPU time and communication bandwidth; and (iii) the number of atoms per processor within the overlap region N_0 can affect CPU time.

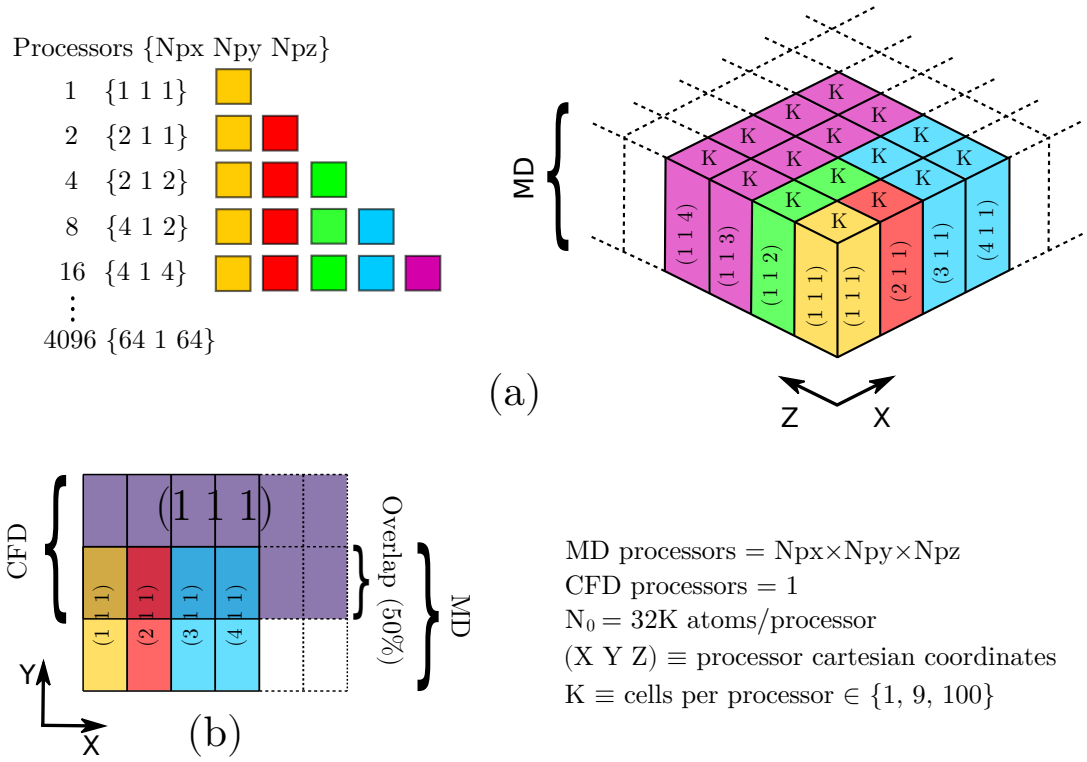


Figure 4.12: A graphical summary of the scalability tests performed in this section. (a) Shows how processors are allocated as the total number of processors scales up in the X-Z plane from 1 to 4096. For example, for 8 processors, the ones colored by yellow, red, green and blue represent the decomposition followed. Each processor in the MD contains a fixed number of $N_0 = 32K$ atoms and a fixed density of cells per processor (K) inside the overlap region. In (b) it is depicted how the setup is fixed in the y direction. The CFD and MD domains have the same size and the overlap occupies 50% of each domain. There is only one CFD processor $(1, 1, 1)$ allocated. Half of each MD processor lies inside the overlap region which only contains one cell in the y direction. The boundary (BC) and constrained (CNST) regions occupy the entire overlap.

A set of tests were designed to study the weak scaling of the coupled software. The geometrical and topological parameters were fixed in the y direction. The number of cells were set to, $N_{cy} = 2$ for `OpenFOAM` and $N_{cy}^{olap} = 1$ for the overlap region. The boundary condition (BC) and constrained (CNST) regions were coincident and occupied the whole overlap region, which extended 50% into `LAMMPS` and `OpenFOAM` domain Figure 4.12.b). Only one processor was allocated for `LAMMPS` in the y direction, yielding an average of 16K atoms per processor laying inside the overlap region. The number of processors assigned to `OpenFOAM` was kept constant and equal to 1. The system was scaled in the x and z dimensions as,

$$\begin{aligned} N_{cx} &= \sqrt{K} N_{px}, \\ N_{cz} &= \sqrt{K} N_{pz}, \\ K &\in \{1, 9, 100\}, \\ N &= N_p \, N_0, \\ N_p = N_{px} \times N_{py} \times N_{pz} &\in \{1, 2, 4, ..., 2048, 4096\}, \end{aligned} \tag{4.4.5}$$

where N_{cx}, N_{cz} and N_{px}, N_{pz} are, for the x and z direction, the number of cells in `OpenFOAM` and the number of processors assigned to `LAMMPS`; K is the number of cells per MD processor; N_p is the total number of MD processors allocated; N_0 and N are, respectively, the number of atoms per processor and total number of atoms in the MD domain. For each value of K, a parametric study was performed for three different time-step ratios $TSR = \{1, 50, 100\}$. Results of the scalability tests are reported for $K = 1$ in Figure 4.13 and for $K = 100$ in Figure 4.14. Data for $K = 9$ is not included as it yielded qualitatively the same results as $K = 100$. The simulation time reported in Figure 4.13 and Figure 4.14 is normalized by the time-step ratio $t = t_{total}/TSR$, where the total simulation time was set to $t_{total} = \{100, 5000, 10000\}\ \tau$, respectively for the three TSR reported above. This fixes to 100 the number of coupling data exchanges taking place between the MD and CFD domains.

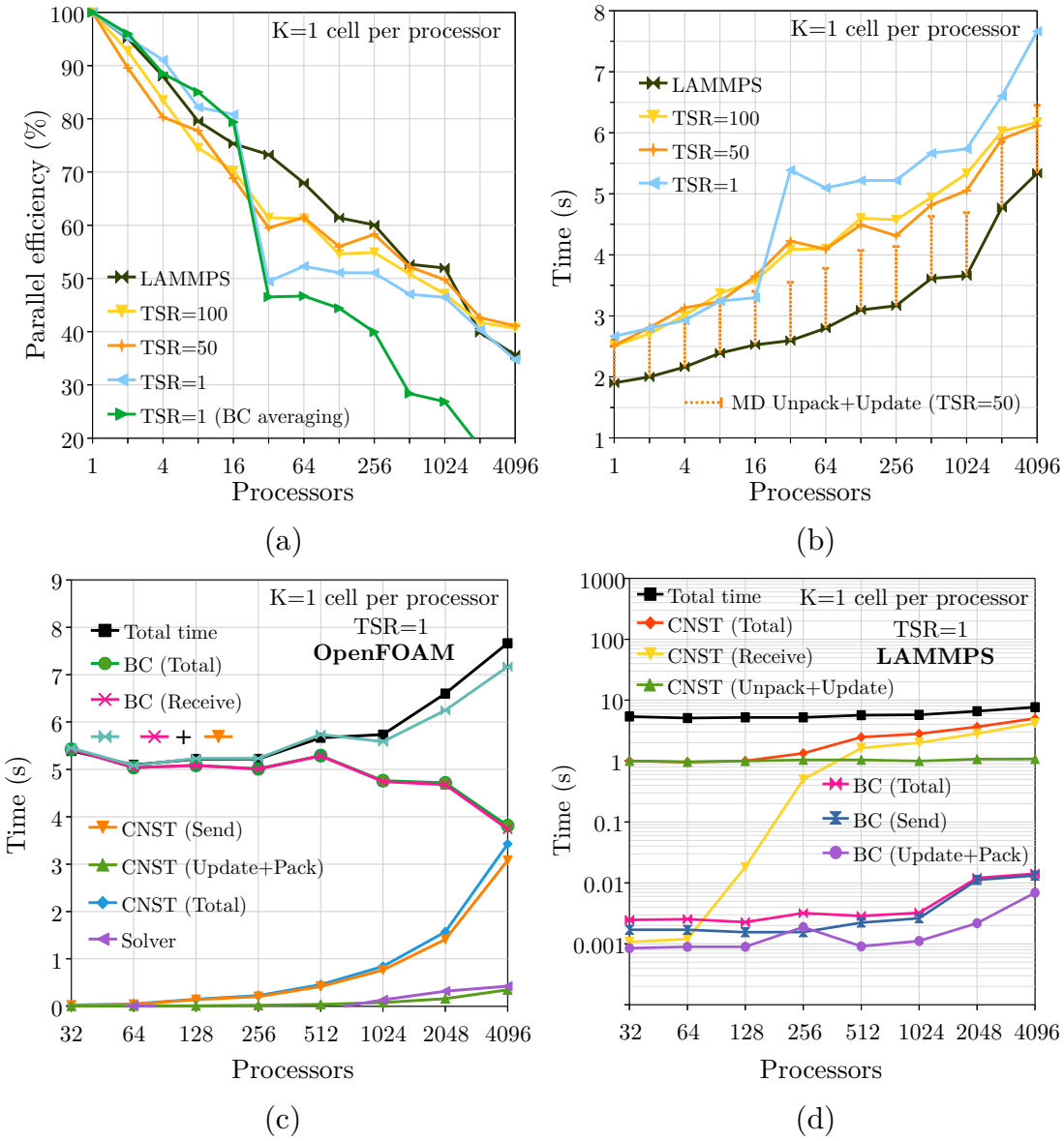


Figure 4.13: The (a) parallel efficiency and (b) normalized total simulation time for `LAMMPS-CONDA` is presented for a range of time-step ratios (TSR) and $K = 1$ cells per processor. The green line in (a) includes an inter-processor averaging routine used to compute BC data, which produces a drop in the efficiency. The dotted orange line in (b) represents the `unpack()` and `update()` operations in the CNST region for $TSR = 50$. It can be seen that those operations account for most of the overhead introduced in coupling. The breakup of times spent in coupling in the CNST and BC regions are plotted in (c) for `OpenFOAM` and in (d) for `LAMMPS`.

The first batch of coupled weak scaling studies for $K = 1$, exhibits significant agreement with respect to `LAMMPS` in terms of efficiency. A

maximum discrepancy of ~5% was found for $TSR = \{50, 100\}$ and ~15% for $TSR = 1$ (Figure 4.13.a). Looking at the normalized simulation time in Figure 4.13.b, it can be seen there exists an overhead due to coupling. This overhead, for $TSR = \{50, 100\}$, is mostly related to the extra time spent in `LAMMPS` data unpacking and updating operations within the CNST region. This time stays fairly constant (green lines in Figure 4.13.d and Figure 4.14.d) for the setup chosen regardless of K, TSR and N_p, since it scales exclusively with the number of atoms per MD processor within the overlap region ($N_0^{olap} = 16K$ on average in this case). The CNST unpacking/updating time is also independent of N_p. Processors outside the CNST region, would not suffer from coupling overhead but the simulation time will be always bound by the slowest process. Nevertheless, CPU time spent in `LAMMPS` internal routines for the setup chosen, is spent in pair-forces calculation and communication. Normally, there will be other *on-the-fly* calculations taking place in `LAMMPS` (inter-process data averaging, thermostats and other constraints, etc.) which would likely reduce the relative extra cost of coupling. The poorer performance exhibited for $TSR = 1$ in Figure 4.13.a, is a consequence of the increase in the time spent in the sending routine within the CNST region in `OpenFOAM`, which amounts almost to the total time spent in operations in that region (orange and blue lines in Figure 4.13.c). The sum of the sending time in the CNST region combined with the time in receiving data in BC region (`OpenFOAM` blocked waiting for `LAMMPS`) add up to the total simulation time in Figure 4.13.c. The total time spent in BC-related routines in `LAMMPS` (Figure 4.13.d) is negligible on the other hand.

Furthermore, a second set of studies was performed for $K = 1$. They included an inter-processor averaging routine for the BC region in `LAMMPS`. This type of averaging is usually necessary to compute boundary conditions across multiple `LAMMPS` processors. However, it only seemed to degrade performance for $TSR = 1$ (green line Figure 4.13.a). Results for $TSR = \{50, 100\}$ are not plotted in this case. Since results of coupled simulations presented in other sections of this work always employ a

$TSR > 50$, employing averaging routine do not pose a threat to the performance of the simulations carried out in Section 5.3.5 and Section 5.4.3.

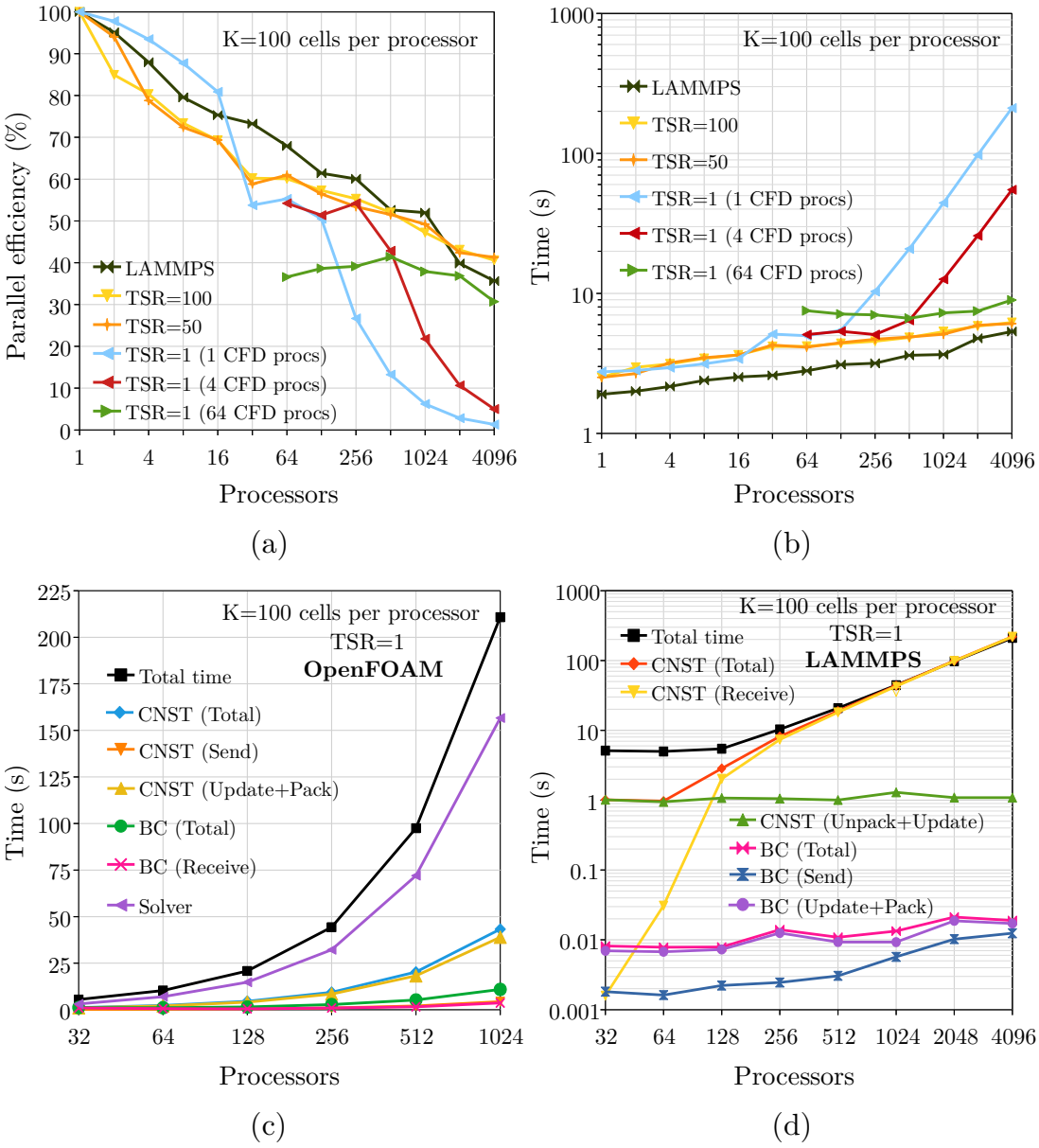


Figure 4.14: The (a) parallel efficiency and (b) normalized total simulation time for `LAMMPS-CONDA` is presented for a range of time-step ratios (TSR) and $K = 100$ cells per processor. For $TSR = 1$, allocating different number of CFD processors improve efficiency and lower simulation time for different ranges of MD processors and load balancing would be required. The breakup of times spent in coupling in the CNST and BC regions are plotted in (c) for `OpenFOAM` and in (d) for `LAMMPS`.

The second batch of studies fixed $K = 100$ and yielded very close results to $K = 1$ for $TSR = \{50, 100\}$. Differences for $TSR = 1$ only showed up as the number of processes was scaled up, as shown in Figure 4.14.a. Figure 4.14.b shows the simulation time increases dramatically for $N_p > 256$. This is due to an increase on the amount of time spent inside the `OpenFOAM` solver and the updating/packing routines in the CNST region (Figure 4.14.c). At $N_p = 256$ `LAMMPS` is no longer the bottleneck and load-balancing between `OpenFOAM` and `LAMMPS` is needed to achieve best performance. When this scenario arises, a careful selection of the number of CFD processors has to be made. It can be seen from Figure 4.14.a that for different numbers of processors allocated to `OpenFOAM` (1, 4 and 64), the improvement of the efficiency for each of them, occurs just for a certain range of allocated `LAMMPS` processors. This scenario (multiple MD communicating with multiple CFD) is handled efficiently by the `CPL_LIBRARY` [8] using smart mapping between MD and CFD processors. This is done by avoiding spurious communication from non-overlapping processors as explained in Section 4.2.1. Lastly, it is worth noting the jump in parallel efficiency when the number of processors is increased from 16 to 32 in Figure 4.13.a and Figure 4.14.a. The reason is each node in the cluster have 24 processing units (Table 4.4), hence with 32 processors multi-node communication starts to take place, which causes a degradation of the efficiency.

Furthermore, a simple analysis can be done to determine if the bandwidth of the cluster network could have impacted the parallel efficiency due to coupling-related `MPI` communications. The bandwidth used by the coupled software can be estimated (excluding the overhead in network packages due to networking protocols headers and trailers [193]) as:

$$Bw = \frac{8 \times sizeof(sendtype) \times N_p^{olap} \times K}{10^6 \times t_{total}/TSR} \text{ (Mb/s)}, \qquad (4.4.6)$$

where Bw is the bandwidth in megabits per second, $sizeof(sendtype)$ is the size in bytes of the data type being sent, N_p^{olap} is the number of processors in the overlap region for a certain domain (MD or CFD), K is the number of cells per processor, t_{total} is the total simulation time in seconds, and TSR is

the time-step ratio. The maximum bandwidth used, in the performed scalability tests, was for the case with $TSR = 100$ and $K = 100$,

$$Bw = \frac{8 \times 8 \times 4096 \times 100}{10^6 \times 6.2} = 4.228 \text{ (Mb/s)},$$

which can be perfectly handled by the 1000 Mb/s network available. Hence, we can identify network latency as the limiting communication factor in our scalability studies.

The scalability of a fairly comprehensive range of different scenarios for CFD-MD coupled simulations were studied in this section. There are a number of significant conclusions we can draw from the results:

a) The parallel efficiency for $TSR > 50$ (most likely also holds for lower numbers) closely follows the parallel efficiency of `LAMMPS`.
b) There exists an overhead due to coupling which is proportional to the number of atoms per processor inside the CNST region (N_0^{CNST}). The overhead is almost constant regardless of the number of MD processors, TSR and K.
c) For $TSR = 1$ (most demanding coupling scenario) an increase in the time spent in sending data within the CNST drops efficiency and increases the total simulation time. This has been identified as a latency-related issue, since the maximum bandwidth for coupled communication is in the order of ~10 Mb/s (maximum) and a network capable of handling up to 1000 Mb/s was used.
d) For $TSR = 100$ and $K = 100$, `OpenFOAM` becomes a bottleneck due to the large number of cells. In this situation, a load balancing analysis is needed to allocate the appropriate number of `OpenFOAM` processors for a given number of processors in `LAMMPS` to increase efficiency and lower simulation times.

These conclusions are, in general, only valid if the assumption that the CFD is much less computationally expensive than the MD holds and only a single CFD processor is sufficient to do the solving efficiently.

4.5 Chapter Summary

It has become increasingly evident the importance of writing scientific software in academia following good software development practices [194]. The epitome of this growing concern, has been the appearance of the role of the *research software engineer* who assists with software development activities in scientific computing research [194]. Research software should be well-tested, comprehensive to extend, maintain and use. In addition, in the case of applications written to exploit parallel and distributed architectures, a low entry barrier for deploying or building software within HPC environments is also desirable. Ultimately, this enhances the quality and reproducibility of scientific results and speedups knowledge transfer between research peers or project stakeholders.

In this work, a conscious effort was made to follow some of these good practices. This chapter has collated the software development and computational aspects of the coupled framework developed. In Section 4.2 the rationale about using the `CPL_LIBRARY` to couple two 3rd-party solvers was presented. By employing a library to communicate both solvers, the independent development and testing of each solver was made possible. This would not have been accomplished, should a monolithic approach had been used. Section 4.2.2 was devoted to analyze the architecture of the `C++` interface which was later employed to couple `LAMMPS` and `OpenFOAM` solvers by implementing the `CPLSocket` and `TransmittingField` object-oriented interface. By employing this code infrastructure, the process of coupling both solvers was eased and made readily extensible to accommodate for future developments like adding support to couple additional physical fields (e.g temperature). The testing strategy for each independent coupled solver (`LAMMPS` and `OpenFOAM`) has been reviewed in Section 4.3. Integration tests for each solver were integrated in the building process of `Conda` packages [174] to verify correctness of the coupling procedure when the code base changes and packages need to be re-built. The advantages and disadvantages of using `Conda` environments as a vehicle for deploying coupled applications into computing clusters has been outlined. This deployment strategy allowed

us to compile once and deploy to different HPC facilities with compatible hardware architecture. Also it ensures that the environment (set of versioned libraries) was also consistent across multiple installations in different machines which is important towards ensuring good reproducibility of results. In addition, containerization was considered as a plausible option for the near future, when support in HPC facilities becomes more widespread. Lastly, Section 4.4 was devoted to study the parallel efficiency of the coupled implementation of `LAMMPS` and `OpenFOAM` solvers. Good scalability was found for values of coupling parameters similar to the ones employed for the simulations performed in this work in Sections 5.3.5 and 5.4.3. A constant overhead was observed in `LAMMPS` proportional to the number of atoms in the CNST region, consequence of the application of the momentum flux coupling force. Furthermore, scenarios where the parallel efficiency drops significantly and the ratio between the CFD and MD processors needs to be carefully balanced were presented.

Chapter 5

Coupled Simulations: Setup and Results

5.1 Introduction

This chapter aims to validate and put into use the coupling methodology presented in Chapter 3 and contains the core set of results of this thesis. The implementation details (software design, testing and deployment strategy, parallel scalability) of the software framework employed in this work, have been already discussed in Chapter 4. The simulation setup procedure and results obtained in this section, are presented separately for each type of fluid model studied: Lennard-Jones (mono-atomic) and molecular (benzene and squalane). All simulations have been conducted using `LAMMPS` [35] as the molecular dynamics engine and the `OpenFOAM` [72] package providing CFD solvers. The software versions utilized for both are listed in Table 4.5.

Firstly, the properties of liquids relevant for the coupling methodology are presented in Section 5.2. Equilibrium and non-equilibrium properties computed from MD simulations are discussed for the two fluid models.

Section 5.3 contains the results for coupled simulations of Lennard-Jones (LJ) fluids under different shear conditions. These simulations were used to evaluate the numerical correctness of the the coupling algorithms against analytic expressions for three different parallel flow types: Couette, Poiseuille and Stokes-Rayleigh flows. See Section 2.4.2 for a description of each of them. Section 5.4 is devoted to describing the process of building, equilibrating and running NEMD shear-driven coupled simulations of realistic molecular fluids (benzene and squalane). The systems simulated are a coupled version of fully atomistic systems which can be found in the specialized computational nanotribology literature [4,14,130,135]. Lastly, Section 5.5 discusses the appearance of a noise amplification effect in the velocity field within the overlap region. This effect has not been reported before in the literature and seems to be a consequence of a positive feedback mechanism due to the dynamics inside the overlap region characteristic of the two-way coupling methodology.

5.2 Properties of Liquids from Molecular Dynamics

Physical properties computed from MD simulations are directly related to the mathematical model which defines the atom-atom interactions, also known as *force-field,* and the set of parameter values chosen for it (e.g energy ϵ and length σ parameters in the LJ potential). Conversely, in CFD, such properties appear as equation's coefficients which are inputs to the continuum model (e.g viscosity in Equation 3.3.4). Therefore, as discussed in Section 3.4, measuring the physical properties of the MD fluid (*characterizing* the fluid), is the first step prior to running a coupled simulation in order to have consistency between the atomistic and continuum model. In order to do so, it is customary that the atomistic system is in a well-defined liquid state to ensure we can use the set of continuum equations presented in Section 3.3. One way of verifying the liquid state, is by computing the radial distribution function (RDF) [2] defined as:

$$g(r) = \frac{\delta n_r}{4\pi r^2 \rho \, \delta r}, \tag{5.2.1}$$

where δn_r is a function that counts the number of particles within a shell of thickness δr and ρ is the local density within the shell. Fluids possess a very characteristic RDF compared to gas and solid phases as it can be appreciated in the inset of Figure 5.1. Furthermore, for incompressible fluids, the kinematic viscosity ($\nu = \eta/\rho$) and thermal diffusivity ($\alpha = \lambda/C_p\rho$), where the value of ρ is given by equation of state of the fluid, need to be computed to be passed as transport coefficients in the CFD solver. These coefficients can be computed from equilibrium or non-equilibrium techniques [195].

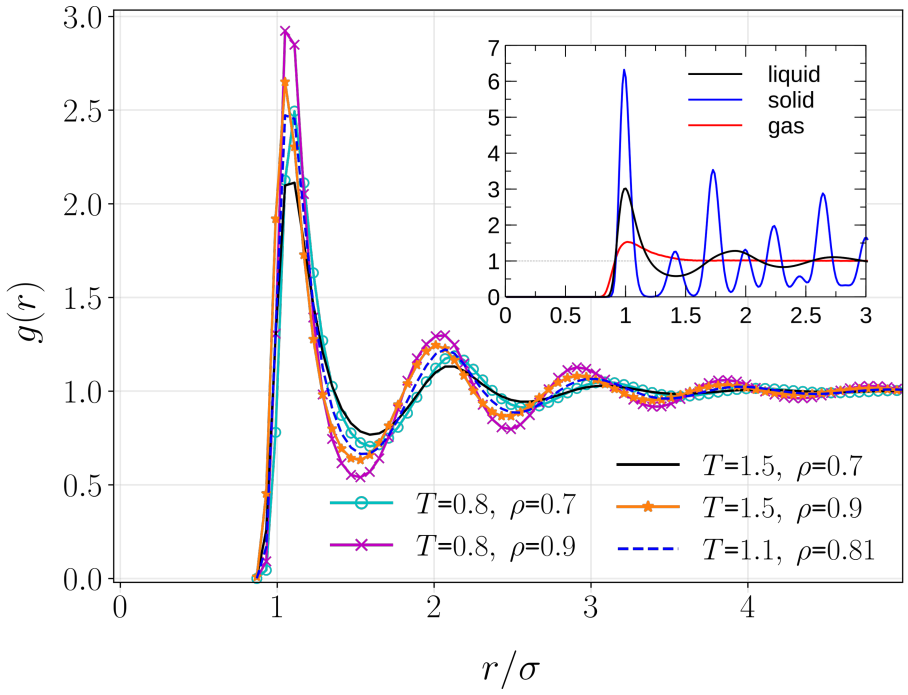


Figure 5.1: Radial distribution function (RDF) computed following Equation 5.2.1 for a LJ fluid with potential parameters from Table 5.6 at five different thermodynamic points. RDF functions obtained by averaging the results from $N_R = 50$ independent runs taking $N_s = 1000$ samples per run. The inset shows RDFs for solid ($T = 50K$), liquid ($T = 80K$) and gaseous ($T = 300K$) argon as an example of characteristic RDFs for different states of matter.

For our purposes, an evident advantage of NEMD methods is the possibility to calculate properties which depend on the local rate of deformation in the fluid (e.g shear-rate dependency of viscosity in non-Newtonian fluids [126]), as opposed to equilibrium methods which are limited to the zero shear-rate limit. The most popular equilibrium method to compute transport coefficients is grounded in the linear transport theory which leads to (Equations 2.2.27, 2.2.28 and 2.2.29) the Green-Kubo formulas [29]. On the other hand, the Müller-Plathe method [67] and simulations based on the SLLOD equations of motion (see Section 2.3.4) are popular techniques to obtain transport coefficients from non-equilibrium shear simulations. The latter is used in this section to study bulk periodic systems under constant shear. Fluid properties computed for Lennard-Jones fluids can be found in Section 5.2.1 and for benzene and squalane molecules in Section 5.2.2.

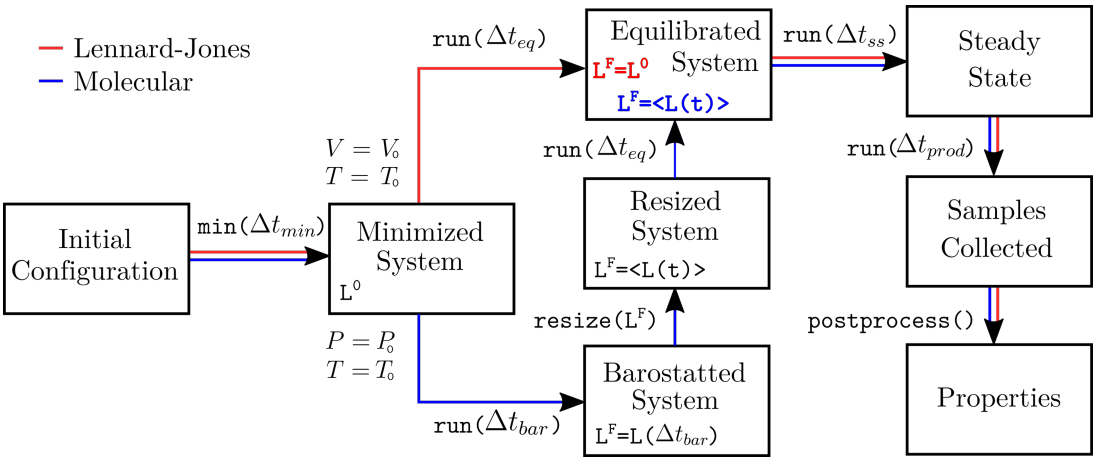


Figure 5.2: Schematic of the two workflows followed to compute properties from MD simulations. Red and blue lines indicate the route followed, respectively, for Lennard-Jones and molecular fluid studies. The boxes indicate workflow stages and arrows are transitions between stages, which are associated with an action. The actions appearing in the schematic are energy minimization of the system (`min`), time-evolution of the system over a time Δt (`run(`Δt`)`), resize of the simulation box (`resize`) and post-processing of data (`postprocess`). The different times involved in each action are minimization time Δt_{min}, NVT equilibration time Δt_{eq}, NPT equilibration time Δt_{bar}, time to steady-state Δt_{ss} and production time Δt_{prod}. The symbols L_0 and L_F are respectively the initial domain length after minimization and the final domain length at a particular workflow stage.

5.2.1 Lennard-Jones Fluids

A Lennard-Jones model with potential parameters $\sigma = 1.0$, $\epsilon = 1.0$, $r_{cut} = 2.5$ was employed in the simulations conducted in this work. The cutoff radius value selected is pervasively used in the literature as it allows the correct prediction of bulk viscosity [195]. Five different thermodynamic points were studied where the system is in a well-defined liquid state [196], which can be appreciated by the characteristic RDF shape in Figure 5.1. The initial configuration of the atomistic system is initialized from a FCC lattice with the lattice parameter a selected to yield the desired density ρ. An energy minimization algorithm is used to relax the system prior to equilibration. The equilibration of the system is performed by applying a Langevin [142] thermostatting force to the fluid atoms. The atomistic equations of motion are integrated using a velocity Verlet algorithm described in Section 2.3.3 with a time-step $\Delta t_{md} = 0.005\tau$, which demonstrates correct conservation of energy in the NVE ensemble [2] and periodic boundary conditions in the three dimensions are imposed [35]. Simulations were performed in `LAMMPS`, which supports parallelization by domain decomposition. Both equilibrium and non-equilibrium simulations were precluded by a set of "preamble" simulations in order to find appropriate efficient sampling parameters (total number of samples and sampling period) as well as appropriate integration cutoff times in order to compute transport coefficients from Green-Kubo formulas. More sophisticated *adaptive methods* exist, where the sampling is adjusted on-the-fly [197] but they have not implemented here.

Equilibrium simulations

The simulations presented in this section have been carried out in the NVT (canonical) ensemble although the NPT could have been equally used [2], since equilibrium properties obtained in both ensembles should be equivalent [198]. A Nosé-Hoover chain thermostat [150] was used to keep the temperature constant during *production* (this is the stage where statistics are collected after the system is equilibrated). This thermostat is known to

correctly sample the NVT ensemble and does not destroy momentum conservation, a common problem of stochastic thermostats which can lead to erroneous transport coefficients [2]. Every atomistic system contained $N_p = 1000$ atoms and the system equilibration time was $\Delta t_{eq} = 2500\tau$. The equilibrium auto-correlation functions for a quantity q were numerically computed by using the formula from [2]

$$\langle C^q(t_i) \rangle = \frac{1}{N_R N_o} \sum_{k=0}^{N_o-1} \sum_{n=1}^{N_R} q^n \left(k\Delta t_s\right) q^n \left(k\Delta t_s + t_i\right), \tag{5.2.2}$$

where t_i is the discretised time, N_o is the number of time origins, N_R is the number of independent realizations used to smooth $C^q(t_i)$ and Δt_s is the sampling interval.

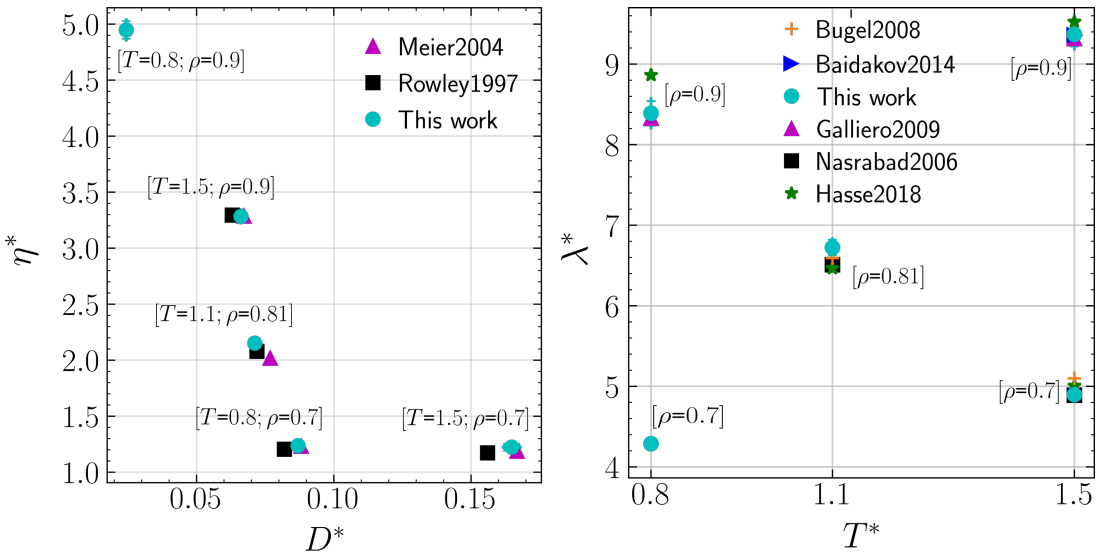
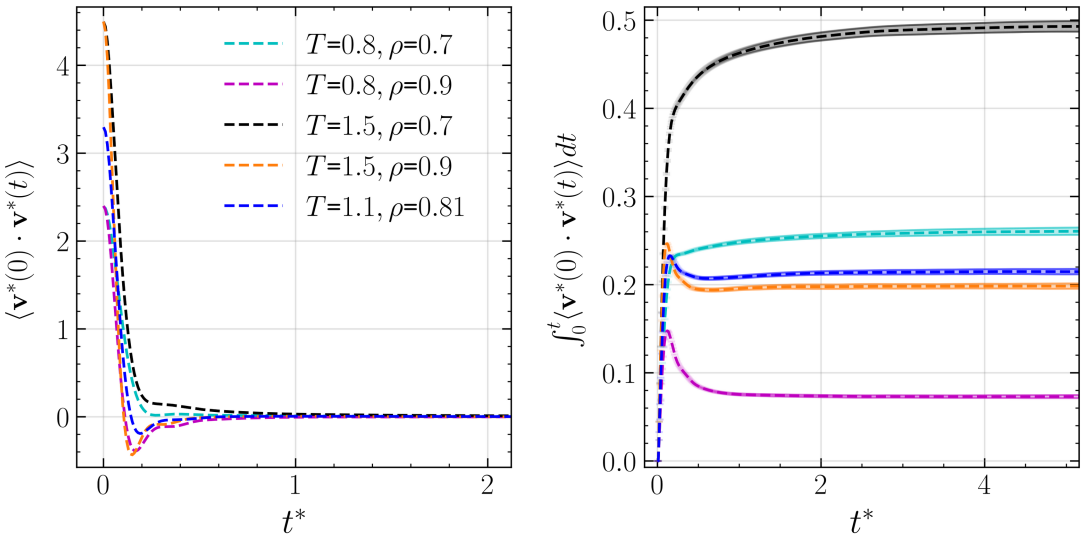


Figure 5.3: Comparison of values obtained in the present work and found in the literature for (a) the mass diffusivity D^* and dynamic viscosity η^*; (b) thermal conductivity λ^* at five different thermodynamic points. All values are in reduced units. Green-Kubo formulation from Section 2.2.5 was employed to compute the coefficients. Values from the literature are Meier2004 [244,245], Rowley1997 [240], Galliero2009 [246], Nasrabad2006 [247], Hasse2018 [241], Bugel2008 [243] and Baidakov2014 [242].

The trapezoidal rule [88] was used for the numerical integration of the auto-correlation function appearing in the Green-Kubo formulas. The Newtonian

viscosity η^*, diffusion coefficient D^* and the thermal conductivity λ^* were computed for $N_o = 1000$, $N_R = 50$ and $0 < t_i < 8\tau$ in Equation 5.2.2. An interval $\Delta t_s = 0.01\tau$ was used to sample particle velocities and $\Delta t_s = 0.025\tau$ to sample the stress S^*_{xy} and heat flux $\mathbf{J}^q$. The computed values for the three transport coefficients are plotted in Figure 5.3 compared with values extracted from the literature. Good agreement was found with data from other authors.



(a) (b)

Figure 5.4: (a) Auto-correlation of particle's velocity as a function of time and (b) the integral of the auto-correlation function at five different thermodynamic points. The shaded area is the uncertainty characterized as the SEM error computed from Equation 2.2.13.

The auto-correlation times for quantity q can be computed from

$$\tau_q \equiv \int_0^{+\infty} t \langle C^q(t) \rangle \, dt, \tag{5.2.3}$$

over a range of temperatures and densities for q being particle velocities and shear stress. Results are presented in Figure 5.5. These times can be compared with the cutoff value where the auto-correlation function becomes effectively zero. For instance, the particle velocity auto-correlation function plotted in Figure 5.4.a shows that self-decorrelation takes place at times between 0.4τ and 1τ, which is a value significantly above the values of τ^*_v in

Figure 5.5.b. This is because the actual definition of the correlation time in Equation 5.2.3 is mathematically meaningful for exponentially-shaped auto-correlation functions, which are commonly found in gases [2]. Hence, the auto-correlation time τ_v^* value used to define the coupling constraint C1.1 in Table 3.7, has been taken by inspection of the auto-correlation function plot Figure 5.4.a and not by the value obtained from Equation 5.2.3.

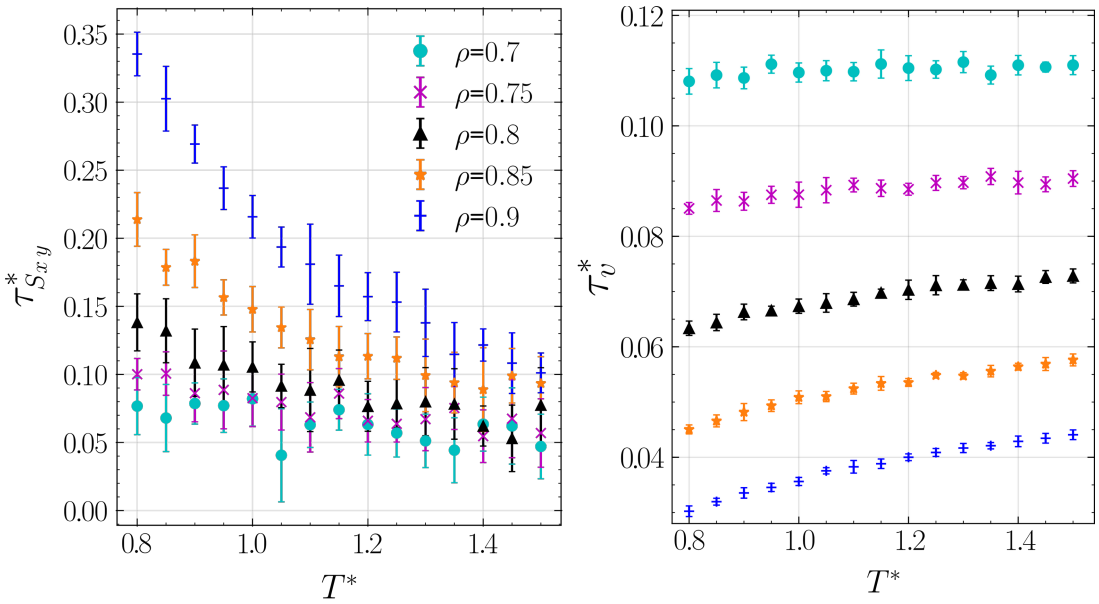
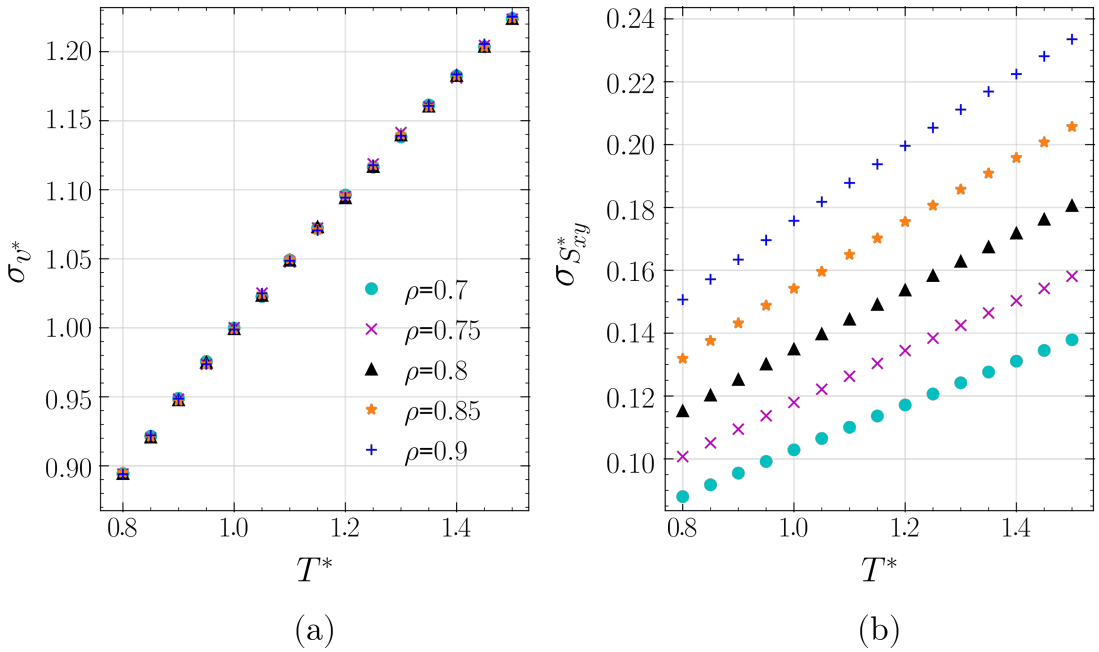


Figure 5.5: (a) Shear stress auto-correlation time $\tau^*_{S_{xy}}$ and (b) particle velocity auto-correlation time τ_v^* as a function of temperature T* at five different densities. Auto-correlation times are computed following Equation 5.2.3 for stress and particle velocity. Error bars represent the uncertainty characterized as the SEM error computed from Equation 2.2.13.

Another quantity of interest, plotted in Figure 5.6, is the noise in particle velocities and shear stresses due to thermal fluctuations. The noise is characterized as the standard deviation of the Gaussian distribution followed by these quantities in equilibrium [2]. With an expression characterizing the noise, the signal-to-noise (SNR) can be defined for a particular thermodynamic point. The SNR constraint C2.1 in Table 3.7 establishes an estimate of the amount of sampling needed to recover the velocity field with

a certain degree of accuracy (see Section 3.4.3). According to Figure 5.6.a the noise magnitude in the particle velocities is insensitive to density and only varies with temperature, contrary to stress in Figure 5.6.b, where the noise increases with the increase of density and temperature.



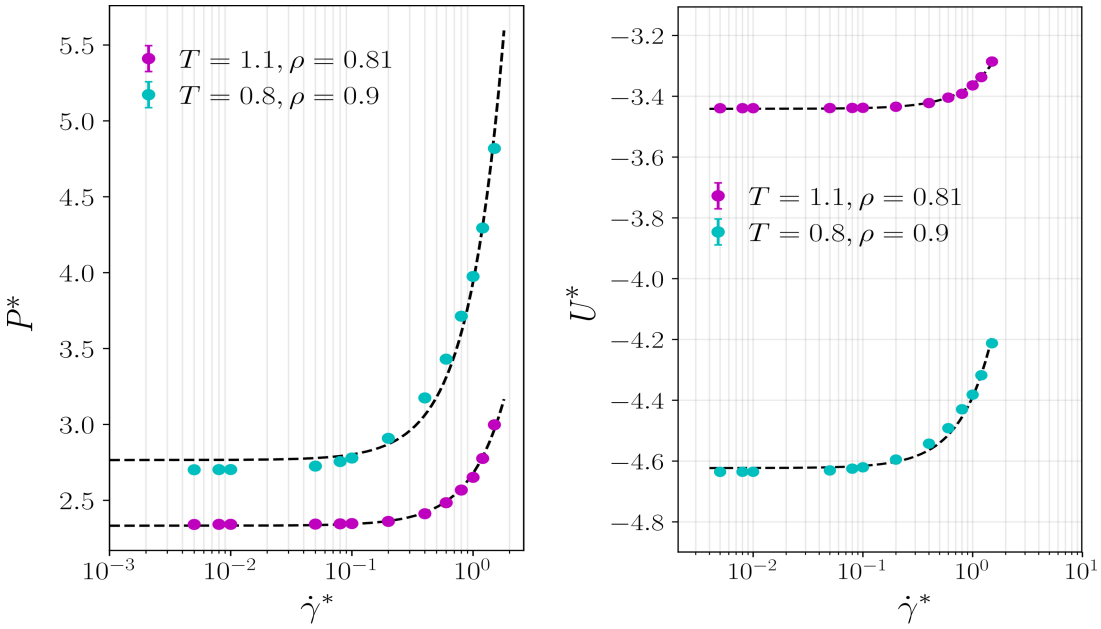
(a) (b)

Figure 5.6: Noise in (a) particle velocities σ_{v^*} and (b) shear stress $\sigma_{S^*_{xy}}$ characterized as the standard deviation of the quantity computed from Equation 2.2.13. Their probability distributions are expected to follow a Gaussian function in equilibrium.

Non-equilibrium Simulations

Coupling between MD and CFD domains is always performed in non-equilibrium conditions. So far, only equilibrium fluid properties and physical quantities have been considered. In order to understand the behaviour of LJ fluids out of equilibrium, simulations were conducted where the system was driven to a constant shear pseudo-steady-state. The SLLOD equations of motion [68] has been used in combination with Lees-Edwards boundary conditions [69] and coupled to a Nosé-Hoover chain thermostat [150] in order to evolve in time atomistic positions and velocities. This set of equations is equivalent to the Newtonian ones under shear conditions [199]. The time step

employed to integrate the equations was shear-rate dependent with $\Delta t_{md} = 0.005\tau$ for $\dot{\gamma}^* < 1.0$ and $\Delta t_{md} = 0.001\tau$ for $\dot{\gamma}^* \geqslant 1.0$ and the thermostatting damping constant $\xi = 100\Delta t_{md}$ were used. In this setup, the simulation box is deformed at constant strain rate $\dot{\gamma}^*$ modifying the box in the x dimension following $L_x = L_x^0 (1 + \dot{\gamma}\Delta t_{md})$ and atomic velocities are remapped when atoms cross periodic boundaries, acquiring a velocity that matches the box change so that their motion naturally tracks the box changes without explicit remapping of their coordinates. More details about this algorithm and thermostatting in this particular setup can be found in Section 2.3.4. The systems studied had $N_p = 2000$ atoms and were equilibrated for $\Delta t_{eq} = 2500\tau$. In order to reach steady-state, the system was run for $\Delta t_{ss} = 10000\tau$ and a production time of $\Delta t_{prod} = 10000\tau$ was used for statistics collection.



(a) (b)

Figure 5.7: (a) Thermodynamic pressure P^* (Equation 2.2.25) and (b) internal energy U^*(Equation 2.3.9) computed at different shear rates $\dot{\gamma}^*$ during the steady-state of a NEMD shearing simulation. Results averaged over $N_R = 10$ independent runs.

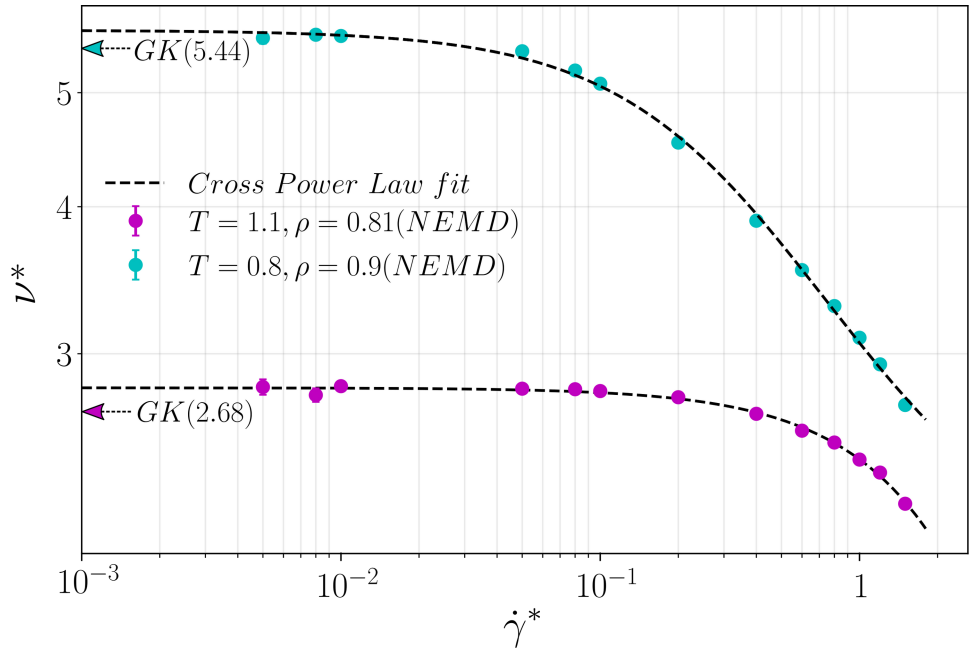


Figure 5.8: Kinematic viscosity $\nu^* = \eta^*/\rho^*$as a function of the shear rate $\dot{\gamma}^*$ computed from NEMD simulations as described in Section 2.3.4 for two different thermodynamic points. Results averaged over $N_R = 10$ independent runs. The zero shear-rate limit Green-Kubo viscosity value is indicated on the vertical axis with an arrow. The data-points have been fitted to a cross-power law from Equation 5.2.4. Error bars are within the symbol size.

Simulations where performed at two different thermodynamic points for shear rates in the range $0.001 \leqslant \dot{\gamma}^* \leqslant 1.5$. Figure 5.7 shows a change in the thermodynamic pressure P^* and the total energy U^* of the system with the shear rate. This is an effect associated to the existence of non-zero normal stress differences which are related to non-linear effects and are the result of the anisotropic fluid structure under shear conditions [200]. A large number of complex fluids such as polymers, colloidal suspensions, micelles, etc. exhibit non-zero normal stresses, effect which manifests itself in other thermodynamic quantities which are a function of P^* and U^*. Also, the kinematic viscosity was computed as $\nu^*(\dot{\gamma}) = \eta^*(\dot{\gamma})/\rho^*$ where $\eta^*(\dot{\gamma})$ is obtained from Equation 2.3.21 where the shear rate $\dot{\gamma}^*$, is computed from the slope of a linear fit to the steady-state velocity profile and the shear stress S^*_{xy} is sampled every $\Delta t_s = 0.25\tau$. The relative error in the shear rate

attained with respect to the target was always below 0.01% for any of the data-points computed.

From Figure 5.8 we can see non-Newtonian effects (in this case *shear thinning*) occurs at extremely high shear rates $\dot{\gamma}^* \in [0.01, 0.1]$ which correspond in real physical units to $\dot{\gamma} \in \left[5 \times 10^9, 5 \times 10^{10}\right] s^{-1}$ (see Table 2.1 for conversion factors). As the shear rate increases, the viscosity decreases following an inverse cross-power power-law which can be used to fit the data-points

$$\nu = \nu_\infty + \frac{\nu_0 - \nu_\infty}{1 + (m\dot{\gamma}^n)}, \tag{5.2.4}$$

where ν_0 is the lower bound (Newtonian) viscosity, ν_∞ is the upper bound viscosity, m is a time constant and n is a positive fitting exponent. The Green-Kubo value appears to be slightly below the ν_0 value from the fit at the two thermodynamic points selected, which is a reasonable discrepancy due to the disparity in the nature of the two methods. The fitting parameter values for $\nu(\dot{\gamma})$ can be found in Table 5.1. The thermal conductivity was not computed employing NEMD simulations since it is known to be highly insensitive to changes in the shear rate [49].

(ρ^*, T^*)	ν_0	ν_∞	m	n
(0.9, 0.8)	2.806523058	-1.625042772	0.129143729	1.1753300771
(0.81, 1.1)	5.649778536	1.9646617936	2.183361148	1.0951611619

Table 5.1: Fitting values for parameters of the cross-power law viscosity model from Equation 5.2.4 for two different Lennard-Jones thermodynamic points.

5.2.2 Molecular Fluids: Benzene and Squalane

In this section, the equilibrium and non-equilibrium equation of state (EOS) and the non-Newtonian viscosity have been computed for benzene and squalane. The same two molecules (see Table 5.2) have been later used to perform coupled simulations reported in Section 5.4. The benzene molecule

has been selected due to being a relatively simple poly-atomic molecule with a high degree of symmetry. On the other hand, squalane is a branched hydrocarbon which is commonly used in experimental [201–203] and computational [5,16,18,19,204] research of lubricant base oils, for its physical and chemical similarities with them. The all-atom potentials OPLS(AA) [56] and L-OPLS(AA) [63] were used, respectively, to model inter-atomic interactions for benzene and squalane. The L-OPLS potential has been specifically tuned for long-chain hydrocarbons and it is known to reproduce with high accuracy the rheology of such molecules [52]. The force field parameter values used in this work can be found in Table 5.3. Refer to Section 2.3.2 for a brief review of other interesting force-fields which are suitable to model organic molecules.

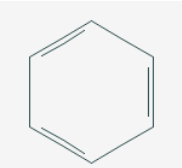
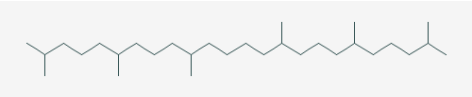
	Benzene	**Squalane**
2D representation		
Formula	C_6H_6	$C_{30}H_{62}$
Mol. Weight	78.11 g/mol	422.8133 g/mol
Family	Aromatic hydrocarbon	Linear,branched hydrocarbon
Force-field	OPLS(AA) [56]	L-OPLS(AA) [63]

Table 5.2: General molecule information for benzene and squalane.

The results reported in this section were computed using the molecular workflow in Figure 5.2, which is described later in this section. Note this workflow involves a barostatting stage (NPT simulation) in order to equilibrate the system. In the case of real molecules it is more interesting to design computational studies at a particular temperature and pressure, which are commonly the control variables in real experiments. The parameter study carried was defined by a set of four *extreme* thermodynamic points contained in the the P-T plane: (300 K, 100 MPa), (300 K, 1000

MPa), (400 K, 100 MPa) and (400 K, 1000 MPa). Those points define the square area sampled. A set of preamble simulations were carried at the four extreme thermodynamic points in order to find (a) the number of molecules N_m (system size) for which the EOS and viscosity converge and (b) efficient values for Δt_{eq}, Δt_{bar}, Δt_{ss}, Δt_{prod} and sampling interval Δt_s (see Table 5.4 for a definition of each time). We have assumed here that the system size and timing parameters selected from these preamble simulations will be valid for other thermodynamic points contained by the squared area they define in the P-T plane. The results of the preamble simulations are not reported here but just the parameter values selected from them for benzene and squalane, which can be found in Table 5.4. It was checked by visual inspection of the RDFs and snapshots of the atomic system that a well-defined liquid state was attained at the extreme thermodynamic points considered.

		Benzene (OPLS)			**Squalane (L-OPLS)**		
Interaction potential	**Type**	**Parameters**			**Parameters**		
Lennard-Jones (Equation 2.3.7)		$\epsilon(\frac{Kcal}{mol})$	σ(Å)		$\epsilon(\frac{Kcal}{mol})$	σ(Å)	
	C	0.07	3.55		0.066	3.5	
	H	0.03	2.42		0.03	2.5	
Bond stretching (Equation 2.3.10)		$K_b(\frac{Kcal}{mol.Å^2})$	l_0(Å)		$K_b(\frac{Kcal}{mol.Å^2})$	l_0(Å)	
	C-C	469.0	1.4		268.0	1.529	
	C-H	367.0	1.08		340.0	1.09	
Angle bending (Equation 2.3.11)		$K_a(\frac{Kcal}{mol.deg^2})$	$\theta_0(deg)$		$K_a(\frac{Kcal}{mol.deg^2})$	$\theta_0(deg)$	
	C-C-C	63.0	120.0		58.35	112.7	
	C-C-H	35.0	120.0		37.5	110.7	
	H-C-H	-	-		33.0	107.8	
Torsions (Equation 2.3.12)		$A_1(\frac{Kcal}{mol})$	A_2	A_3	$A_1(\frac{Kcal}{mol})$	A_2	A_3
	C-C-C-C	0.0	7.25	0.0	0.6447	-0.2143	0.1782
	C-C-C-H	0.0	7.25	0.0	0.0	0.0	0.3
	H-C-C-H	0.0	7.25	0.0	0.0	0.0	0.3

Table 5.3: OPLS and L-OPLS force-field parameters for benzene and squalane used in this work.

The parameter study carried out comprised a set of thermodynamic points defined by the Cartesian product $P^{\Omega} \times T^{\Omega}$ where $P^{\Omega} = \{100, 500, 1000\}$ MPa and $T^{\Omega} = \{300, 320, 340, 360, 380, 400\}$ K, which the molecular systems where equilibrated to. The `Packmol` [205] software was employed to generate random initial configurations and `Moltemplate` [206] to generate `LAMMPS` input files (see Appendix A for software versions) with N_m initial number of molecules and box size L_0. The molecular workflow from Figure 5.2 is exercised as follows. Firstly, the system energy is minimized prior to equilibration. The first run after minimization takes place in the NPT ensemble by employing a Nosé-Hoover chain barostatting and thermostatting algorithm [150] with Verlet time-integration closely following the implementation in Tuckerman *et al,* [207], This run brings the system to an equilibrium constant pressure and temperature thermodynamic state with a fluctuating box size $L(t)$ and mean box size $\langle L(t) \rangle$. Next, the simulation box is resized to $L_F = \langle L(t) \rangle$. Particles which are left outside the simulation box after box re-scaling are remapped following the same procedure as if they had crossed the periodic boundary during a run [2]. After the box is re-scaled, a NVT run of Δt_{eq} follows, in order to achieve a well-equilibrated state. At this point, the final particles' positions and velocities are saved to be used as initial configurations for NEMD shear simulations.

Parameter	**Benzene**	**Squalane**
# molecules (N_m)	500	100
Box initial size (L_0)	50.3878 Å	51.7393 Å
Sampling period (Δt_s)	10 fs	10 fs
Time step (Δt_{md})	2 fs	1.5 fs
Barostatting time (Δt_{bar})	1 ns	3 ns
NVT-Equilibration time (Δt_{eq})	0.5 ns	1 ns
Steady state time (Δt_{ss})	$\dot{\gamma}_t^{-1}$	$\dot{\gamma}_t^{-1}$
Production time (Δt_{prod})	$10 \times \dot{\gamma}_t^{-}1$	$10 \times \dot{\gamma}_t^{-1}$

Table 5.4: System size and timing parameters used in the parameter study where the EOS and viscosity η reported in Figure 5.9 and Figure 5.10 have been computed. The target shear rate is $\dot{\gamma}_t$.

For each individual equilibrated system, seven independent NEMD simulations at seven different shear rates were carried out. In order to bring the molecular system to a constant shear steady-state, the SLLOD equations of motion were used to describe atomistic dynamics coupled to a Nosé-Hoover chain thermostat [150]. The steady-state is achieved at times of the order of $\dot{\gamma}_t^{-1}$ [18], where $\dot{\gamma}_t$ is the target shear rate to attain in the steady-state. Once the steady-state is reached, the viscosity η is computed from Equation 2.3.21 by sampling the shear stress S_{xy} and computing the slope of the linear velocity field across the simulation box every $\Delta t_s = 10$ fs for a total time $\Delta t_{prod} = 10 \times \dot{\gamma}_t^{-1}$, in order to get good statistics. More details about the procedure to drive the system to a constant shear steady-state using SLLOD and how viscosity is computed, can be found in Section 2.3.4.

The scaling of the simulation time with $\dot{\gamma}_t^{-1}$, heavily limits the shear-rates that can be studied with this method to values $\dot{\gamma}_t > 10^6$ [18,52]. We have ensured that error in the shear-rate achieved in the steady-state, with respect to the target shear-rate, was always $< 1\%$. The viscosity and shear stress computed for benzene at $P = 1000$ MPa and squalane at $P = 100$ MPa are reported in Figure 5.9 and Figure 5.10 respectively. Viscosity results for each molecule at the other two sampled pressures in the parameter study, are not reported here since they were not used for coupled simulations reported in Section 5.4. In order to use the viscosity computed from MD as the viscosity model for the coupled non-Newtonian CFD solvers the viscosity data-points were fitted to a continuous shear-rate and temperature dependent expression. A modified log-cross-power law function modified to include temperature-dependence was found to produce the best fitting results. The fitting function employed had the form

$$\eta\left(T,\dot{\gamma}\right) = A(T)\left[\eta_\infty + \frac{\eta_0 - \eta_\infty}{1 + \left(\left(b_1 T\right)^k log_{10}\left(1+\dot{\gamma}\right)\right)^n}\right], \qquad (5.2.5)$$

$$A(T) = e^{a_1(1/T - 1/T_{\min}) + a_2(1/T - 1/T_{\min})^2},$$

where $T_{min} = 300$ K is the minimum of the temperature across all data-points and the rest of the parameters are fitting constants. This expression

was purely developed on numerical grounds (fitting error and convergence) and not derived from phenomenological or physical models which are commonly just suitable for the low shear rate regime [208–210]. The values obtained here for L-OPLS squalane viscosity agree very well with values computed from MD simulations from Jadhao *et al.* [18], which uses the AIREBO potential [211] and covers a wide range of temperature, pressures and shear-rates. Furthermore, squalane viscosity in the zero shear rate limit (η_0) can be compared against values from an experimentally fitted expression from Comuñas *et al.* [203] in Figure 5.10.a. Remarkable agreement was found between both, provided low shear rate viscosity (η_0 fitting parameter) was obtained as an extrapolation of data-points obtained from sampling large shear-rates ($\dot{\gamma} \geqslant 10^8\ s^{-1}$). This is solid evidence not only of the quality of the L-OPLS potential to model squalane's rheology, but also of the accuracy of the simulation procedure and fitting method employed here. Additionally, it would have been possible to compute the Newtonian viscosity from an equilibrium method like Green-Kubo (see Section 2.2.5). However, even though EMD and NEMD methods to compute viscosity might be comparable in terms of computational time [212], the Green-Kubo method is subjected to a large amount of variability (the larger the molecule, the larger the variability), because the plateau region of the stress auto-correlation integral is difficult to identify unambiguously [213]. Nevertheless, it is quite improbable that using the extra data point generated from Green-Kubo at $\dot{\gamma} = 0$, would have significantly impacted the fitted model (Equation 5.2.5), hence improving the accuracy of MD-CFD coupled simulations, which are performed in the high shear rate regime.

	P(MPa)	a_1(K)	a_2(K)	η_0(Pa.s)	η_∞(Pa.s)	$b_1(K^{-1})$	k	n
Squalane	100	4663	836397	0.1755	-1.53e-3	0.1020	-0.6223	17.104
Benzene	1000	3003	585579	0.0521	5.27e-4	0.6172	-0.4296	23.372

Table 5.5: Fitting values for parameters of the temperature-dependent cross-power law viscosity model from Equation 5.2.5 for benzene and squalane.

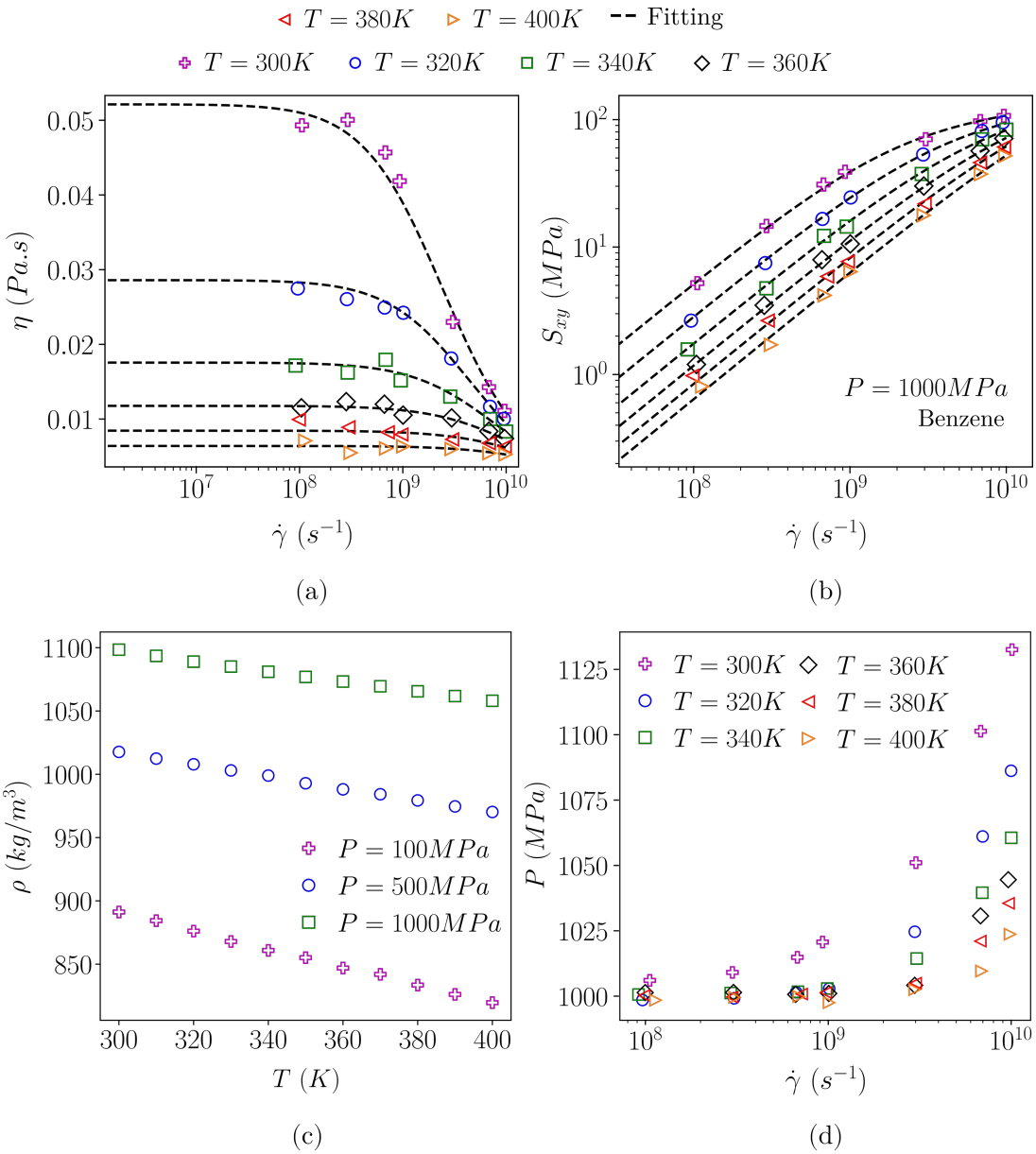


Figure 5.9: Temperature and shear-rate dependent (a) viscosity η and (b) shear stress $S_{xy} = \dot{\gamma}\eta$ for benzene at $P = 1000$ MPa. Data points computed from NEMD molecular dynamics simulations and fitted to a temperature dependent cross-power law following Equation 5.2.5. Fitting parameters can be found in Table 5.5. Plots (c) and (d) present, respectively, benzene's equilibrium equation of state (EOS) and the pressure dependence on shear rate for multiple thermodynamic points.

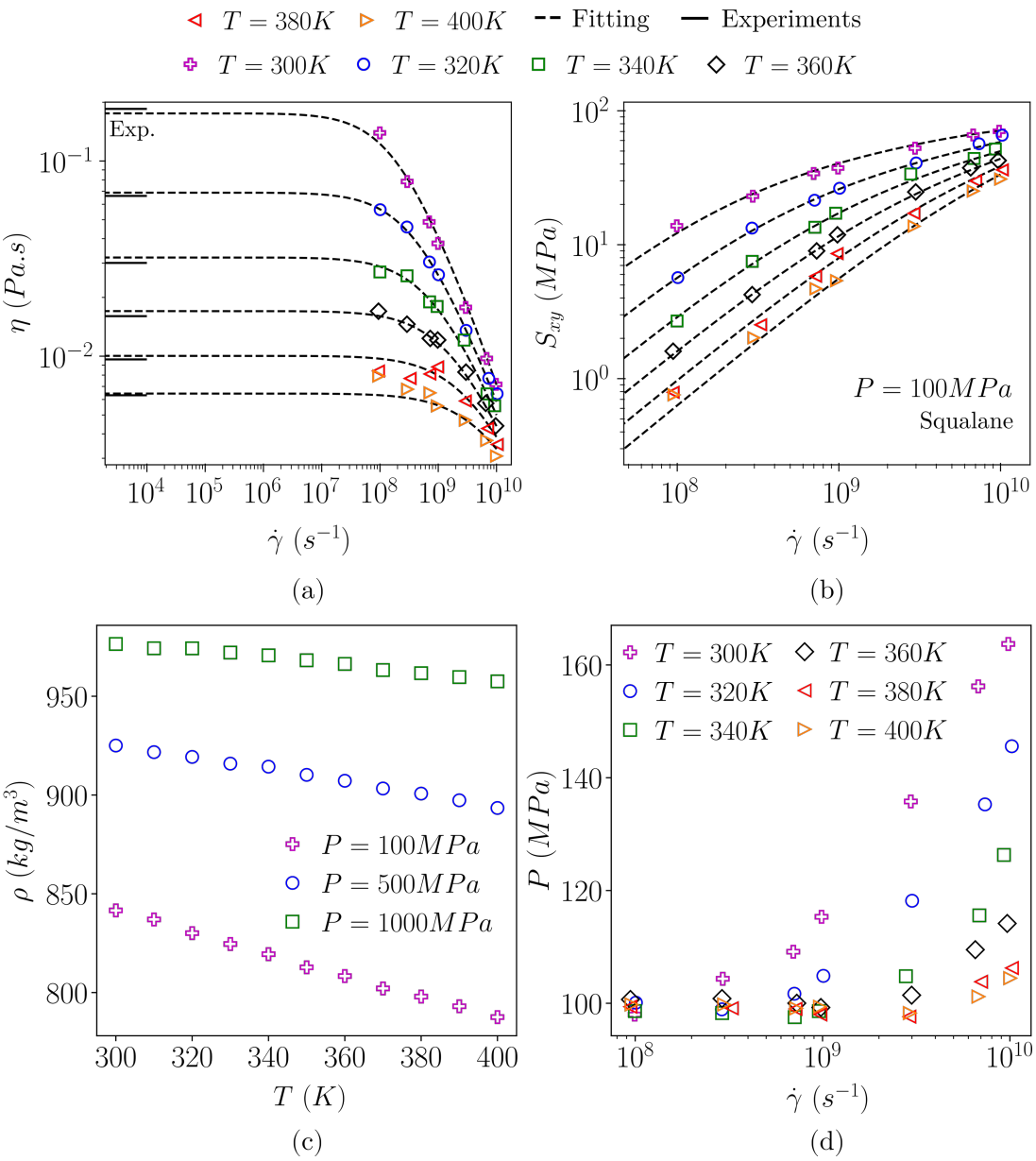


Figure 5.10: Temperature and shear-rate dependent (a) viscosity η and (b) shear stress $S_{xy} = \dot{\gamma}\eta$ for squalane at $P = 100$ MPa. Data points computed from NEMD molecular dynamics simulations and fitted to a temperature dependent cross-power law following Equation 5.2.5. Fitting parameters can be found in Table 5.5. Experimental values for the Newtonian viscosity were computed from an empirically fitted expression from Comuñas *et al.* [203]. Plots (c) and (d) present, respectively, squalane's equilibrium equation of state (EOS) and the pressure dependence on shear rate for multiple thermodynamic points.

The thermal properties C_p (constant pressure heat capacity) and λ (thermal conductivity) appearing in the thermal equation (Equation 2.4.18), for benzene and squalane, have not been computed here but references to tabulated experimental data are given. For benzene, experimental values of λ the can be found in Assael *et al.* [214] for a wide range of temperatures and pressures, and in [215] for C_p. For squalane, both λ and C_p can be found in [216–218]. In general the variation of these two quantities over the range of temperatures of our interest (300 K–400 K) are mild at high pressure.

Lastly, the equation of state and the thermodynamic pressure P under non-equilibrium shear conditions were also computed. A quasi-linear dependence with temperature is observed in the EOS for the range of temperatures explored in both molecules in Figure 5.9.c and Figure 5.10.c.

5.3 Coupled Lennard-Jones Simulations

The coupling mathematical framework presented in Chapter 3 and its software implementation in Chapter 4, are validated in this section for the Lennard-Jones (LJ) fluid model. The simplicity of the LJ fluid model and the existence of extensive scientific literature on CFD-MD coupling based on them, make this type of fluids the ideal candidate for a test model. At the beginning of Section 5.4, a list of the extra difficulties molecular fluids entail over LJ fluids, to implement coupling, is provided and discussed. Despite the convenience of LJ fluids, they posses certain intrinsic characteristics which limit their use in some relevant scenarios. An example is the impracticality of testing the coupling strategy within the non-Newtonian regimen. Extremely high shear-rates are required to reach the non-Newtonian region of their $\dot{\gamma}$-η curve (Figure 5.9). Furthermore, the shear thinning effects are extremely weak when they occur. Sampling the high shear-rate region of the $\dot{\gamma}$-η curve in NEMD, and therefore in MD-CFD coupled simulations, results in a tremendous amount of heat generated within the fluid from the action of shear stresses. In this scenario, the time step has to be dramatically reduced in order to be able to keep the SLLOD algorithm stable [219]. The author has not been able to produce a successful coupled simulation using the LJ

fluid model in the non-Newtonian regime, due to the development of numerical instabilities in the MD solver when integrating the atomistic equations of motion. For this reason, the results presented in this section, are from systems where the LJ fluid exhibits Newtonian behaviour. Coupled simulations where non-Newtonian effects are significant and a non-Newtonian CFD solver was utilized, can be found in Section 5.4.3 for squalane and benzene at high pressure. The rest of Section 5.3 is structured as follows. In Section 5.3.1 it is describe how LJ systems are built and equilibrated. In order to determine a set of suitable parameters for the density correction algorithm, FOREV [104], a parametric study is presented in Section 5.3.2. Furthermore, the accuracy of five different setups collated in Table 5.8, has been quantitatively compared by employing an error metric (Equation 5.3.16) in Section 5.3.4. Conclusions drawn from that study are, in principle, equally applicable beyond simple LJ fluids. Finally, Section 5.3.5 presents the results for transient coupled simulations for Couette flow (wall-driven), Poiseuille flow (pressure-driven) and Rayleigh-Stokes flow (wall-driven). All simulations are compared against time-dependent analytical expressions presented in Section 2.4.2, in order to validate the correctness of the coupling framework. Furthermore, the notation for quantities in reduced units used in Section 5.2.1 ("$*$" super-index), has been dropped for clarity in this section.

5.3.1 Building and Equilibrating the MD Domain

In order to perform a NEMD coupled simulation, the MD domain needs to be initialized to an equilibrated state, where the properties of the fluid are well-defined. The building and equilibration procedure presented here are vastly different from the ones applied to molecular systems, which are described in detail in Section 5.4.1 and Section 5.4.2.

Atom Type	σ	ϵ	r_c	m	ϵ^{fs}	σ^{fs}
fluid (f)	1.0	1.0	2.5	1.0	2.0	1.0
surface (s)	1.0	4.0	2.2	1.0		

Table 5.6: Lennard-Jones potential parameters for fluid and surface atoms in reduced units. See Section 2.3.1 for a description.

In order to build the MD domain, a FCC atomistic surface is positioned perpendicular to the y direction at the bottom of the MD simulation box. A space of size $L_{ch} + 2\sigma$ (σ is the length-scale parameter of the LJ potential) is left above the surface (see Figure 5.11), where L_{ch} is the effective channel length and 2σ accounts for the gaps left near the reflective boundary at the top of the domain and above the atomistic surface, which facilitates the initial random insertion of fluid particles. The first bottom layer of surface atoms (highlighted in green in Figure 5.11) is fixed using a harmonic tethering potential (Equation 3.2.2) with a $K^{fix} = 100\,\epsilon$. The volume defined by $L_{ch} \times A_{xz}$ is then filled by introducing N_0 particles randomly placed, in order to achieve the target density ρ^T following the expression

$$N_0 = \left\lfloor \frac{\rho^T A_{xz} L_{ch}}{m} \right\rceil, \tag{5.3.1}$$

where m is the fluid particle mass, A_{xz} is the cross-section area defined by the area of the atomistic surface (Table 5.9) and the $\lfloor \cdot \rceil$ operator rounds to the nearest integer. The initial velocities for the particles are drawn from a Maxwellian distribution following Equation 3.4.21 with mean velocity $\boldsymbol{u} = 0$. After the insertion procedure, an energy minimization algorithm is applied to the system in order to separate particles which might have been introduced too close. This could cause numerical instabilities during the integration of the atomistic equations of motion, due to high gradients in the interaction energies between particles. The time-step employed for the velocity Verlet integration algorithm was $\Delta t_{md} = 1\tau$. In order to thermalize the system, a Langevin thermostat [142] was applied during the simulation time in the three spatial directions with a relaxation time of 0.5τ,

However, introducing N_0 particles following Equation 5.3.1, does not strictly guarantee a density ρ^T in the bulk of the fluid (far from the reflective and atomistic wall). This is due to the existence of density inhomogeneities close to the non-periodic boundaries as explained in Section 3.2.3. In order to ensure a density value closer to ρ^T, the following strategy was implemented. A set of ten independent MD simulations with the same setup described above, were carried out with different initial number of fluid particles

$$N_0 = \left\lfloor \frac{\left(\rho^T + \Delta\rho\right) A_{xz} L_f}{m} \right\rceil, \tag{5.3.2}$$

where $\Delta\rho \in \{0.001, 0.002, ..., 0.01\}$ is the excess of density with respect to ρ^T.

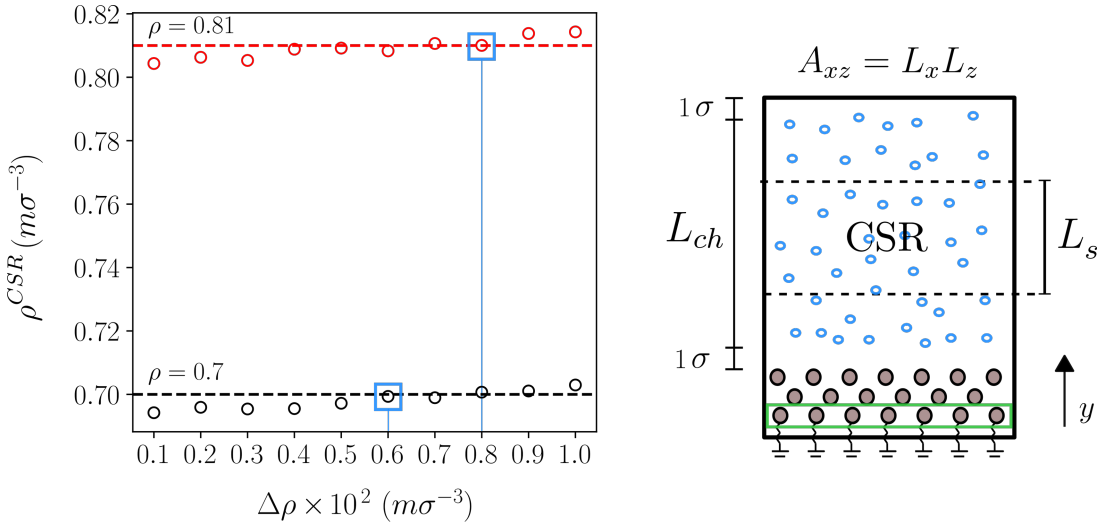


Figure 5.11: (a) Values of ρ^{CSR} obtained from ten independent simulations for different $\Delta\rho$ at thermodynamic points $P1$ and $P2$ from Table 5.7. In blue boxes are highlighted the closest values of ρ^{CSR} to ρ^T (dashed lines), linked to their respective $\Delta\rho$. (b) Schematic of the MD domain with the control slab region (CSR) defined to compute ρ^{CSR}. L_{ch} is the effective channel length and L_s is the width of the CSR. The green rectangle encloses the first bottom layer of surface atoms which are tethered to their lattice sites.

After equilibrating the system for $\Delta t_{eq} = 250000\tau$, the average density ρ^{CSR} is computed within the CSR (control slab region) by taking 10000 samples equispaced in time by a sampling period $\Delta t_s = 0.05\tau$. The aim, is to find the value of $\Delta\rho$ which fulfills the minimization condition $\min_{\Delta\rho} \left|\rho^T - \rho^{CSR}\left(\Delta\rho\right)\right|$. Figure 5.11.a presents the values of ρ^{CSR} computed for thermodynamic points P_1 and P_2 listed in Table 5.7, for different values of $\Delta\rho$. These values are valid for a particular simulation random seed (employed to generate initial particle positions and velocities) and for the specific set of potential parameters for liquid and surface particles listed in

Table 5.6. For instance, changing the surface-fluid interaction ϵ^{sf} will, most likely, shift the $\Delta\rho$ value fulfilling the minimization condition.

Thermodynamic point	T	ρ	η_{gk}^{md}	λ	τ_v	$\Delta\rho$
P_1	1.1	0.81	2.14	6.5	0.8	0.008
P_2	1.1	0.7	1.21	4.6	0.8	0.006

Table 5.7: Relevant quantities of the LJ fluid (reduced units) at two different thermodynamic points. T is the temperature, ρ is the density, η_{gk}^{md} is the dynamic viscosity (from Green-Kubo formulation), λ is the thermal conductivity, τ_v is the auto-correlation time of particle velocities (selected from Figure 5.4) and $\Delta\rho$ is the density excess from Equation 5.3.2.

5.3.2 Parameter Selection for Density Correction Algorithm

As seen in Section 3.2.3, non-periodic boundaries induce a inhomogeneous equilibrium density profile close to them. In our case the non-periodicity is present due to the existence of a reflective boundary where the reflection takes place in the y direction, as defined in Equation 3.2.7. The effect is a consequence of an unbalanced atomistic force along the non-periodic direction. If the region close to the boundary is divided into bins, we can write

$$f_y\left(y_{bin}\right) \equiv \langle F_{y;bin,k} \rangle \neq 0, \tag{5.3.3}$$

where y_{bin} is the bin's center y coordinate (see Section 3.2.3 for a definition of $\langle F_{y;bin,k} \rangle$). Figure 5.12 shows an example of non-constant density ρ and force f_y profiles near the reflective boundary which can potentially have a detrimental effect on coupling algorithms since the CNST region is located within the volume affected by these inhomogeneities. To address this issue, the FOREV [104] density correction algorithm based on exponential smoothing (Equation 3.2.9) was implemented. The algorithm is applied over a distance L_c, which is the length over which inhomogeneities are significant

and two parameters need to be specified as described in Section 3.2.3: the smoothing factor α and the number of bins along the y direction N_b.

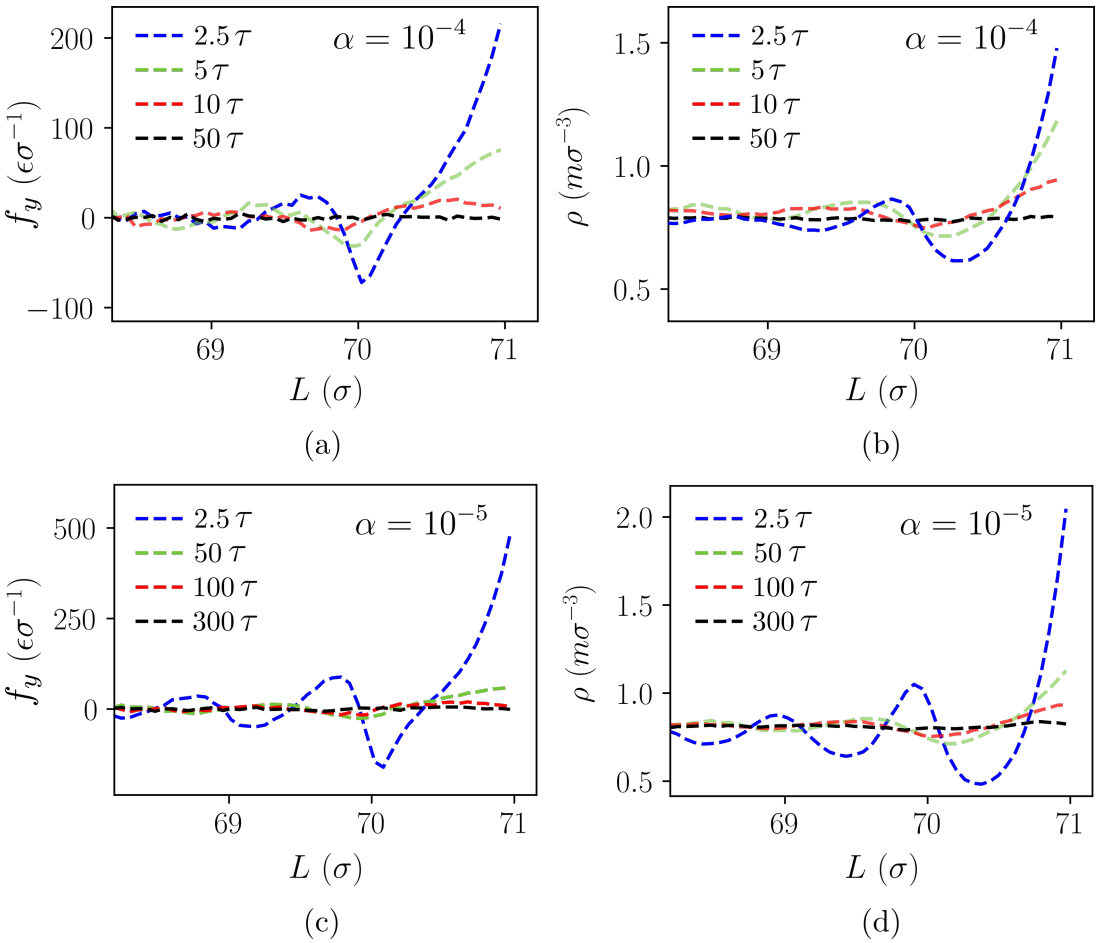


Figure 5.12: Time-evolution of density ρ and net atomistic force in the y direction f_y (Equation 5.3.3) profiles close to the reflective boundary when the density correction algorithm FOREV is applied. Plots (a) and (b) are generated for an equilibrium MD simulation using an smoothing factor $\alpha = 10^{-4}$ and (c) and (d) for $\alpha = 10^{-5}$.

A parametric study to select an appropriate value for the smoothing factor $\alpha \in \{10^{-4}, 10^{-5}\}$ and the number of bins $N_b \in \{50, 100, 150\}$ was carried for a LJ fluid at $\rho = 0.81$ and $T = 1.1$, with potential parameters from Table 5.6, MD geometry from Table 5.11 and surface topology from Table 5.9. Setup and equilibration of the domain were conducted as described in Section 5.3.1. The correction algorithm was applied over a

distance from the top of the domain of $L_c = 3\sigma$, which is larger than the potential cut-off radius $r_c = 2.5\sigma$ [104]. Figure 5.12 shows the profiles of density, ρ and the total net atomic force in the y direction f_y, for $n_b = 50$. The profiles in Figure 5.12 were generated by averaging over a time period $\Delta t_{avg} = 1\tau$ following Equation 5.3.12 and employing a sampling period $\Delta t_s = 0.05\tau$. Both quantities' profiles were found to be independent of higher values of n_b and they have not been plotted. We can observe in Figure 5.12 how as time evolves and $f_y \to 0$, the density field $\rho \to 0.81$ and the convergence of the correction algorithm is achieved after a time period of roughly 50τ for $\alpha = 10^{-4}$ and 300τ for $\alpha = 10^{-5}$. The smallest the value of α the larger is the characteristic time of the exponential moving average for Equation 3.2.9. We want to pick a value that is large enough to respond to field changes during transient, but small enough to smooth out the density inhomogeneities without affecting particle dynamics. The value $\alpha = 10^{-4}$ was selected for subsequent coupled simulations since the time taken to reach convergence lies below $\Delta t_R = 100\tau$ (see Equation 5.3.5), which has been used as an averaging interval to compute the mean-flow fields from coupled simulations in Section 5.3.5.

5.3.3 Sampling Physical Fields

Due to the noisy nature of physical quantities sampled in MD, averaging procedures are necessary to obtain sufficiently smooth fields. This section is not exclusively relevant to LJ fluids and describes how physical fields are computed as well in molecular fluids within Section 5.4. The coupled physical fields of interest are $F \in \{\rho, V_x, S_{xy}, T\}$, being ρ the mass density, V_x the streaming velocity in the x direction, the shear stress S_{xy} on the X-Z plane and the temperature T. The coupled solution for a field F is composed of the partial solutions obtained from the sub-domains Ω_{md} and Ω_{cfd} as

$$F = \begin{cases} \left\langle F^{md} \right\rangle (y_i, t_i), & y_i \in \Omega_{md} \\ \left\langle F^{cfd} \right\rangle (y_i, t_i), & y_i \in \Omega_{cfd} \end{cases}, \tag{5.3.4}$$

where y_i and t_i are the discrete spatial and time variables and the $\langle \cdot \rangle$ operator is defined for each domain $d \in \{\Omega_{md}, \Omega_{cfd}\}$ as

$$\left\langle F^d \right\rangle (y_i, t_i) = \frac{1}{2\Delta t_R N_R} \int_{t_i - \Delta t_R}^{t_i + \Delta t_R} \left(\sum_k^{N_R} F_k^d(y_i, t) \right) dt, \tag{5.3.5}$$

where $2\Delta t_R = k\Delta t_{cfd}$ (k is an integer) is the averaging period to compute the *mean-flow* field value at time $t_i > \Delta t_R$ and N_R is the number of independent coupled simulations performed for different initial atomistic configurations. Note that the $\langle \cdot \rangle$ operator is applied to compute mean flow fields both in MD and CFD domains to recover the mean value of the field. The fluctuations in the boundary conditions provided by the MD in the BC region propagate into the CFD, resulting in noise being added to the fields computed in that domain (see Section 3.4.4). The expressions employed to compute each field F within a sampling bin i can be written as follows. For the flow velocity V_x

$$V_x^{md} (y_i, t_i) = \frac{1}{N_i \Delta t_{avg}} \int_{t_{i-1}}^{t_{i-1} + \Delta t_{avg}} \sum_{k \in V_i}^{N_i} v_{x,k}^{com} dt, \tag{5.3.6}$$

$$V_x^{cfd} (y_i, t_i) = u_x^{cfd}, \tag{5.3.7}$$

where N_i is the number of atoms/molecules inside the sampling bin volume $V_i = d_y^c \times A_{xz}$ and $v_{x,k}^{com}$ is the x component of the center of mass velocity of molecule/atom k as defined in Equation 3.4.10. Note the time-average is centered around the mid-point time $t_{i-1} + \Delta t_{avg}/2$ instead of t_i. This is due to the nature of the synchronized time coupling scheme detailed in Section 3.4.2. Similarly, the mean shear stress is computed from

$$S_{xy}^{md} (y_i, t_i) = \frac{1}{\Delta t_{avg}} \int_{t_{i-1}}^{t_{i-1} + \Delta t_{avg}} P_{xy} (y_i, t) \quad dt, \tag{5.3.8}$$

$$S_{xy}^{cfd} (y_i, t_i) = \tau_{xy} (y_i, t_i), \tag{5.3.9}$$

where P_{xy} is the xy component of the instantaneous atomistic pressure tensor given by Equation 2.2.26 and τ_{xy} is the continuum shear stress from Equation 3.3.1. The temperature field is computed as

$$T^{md} = \frac{1}{3K_b N_i \Delta t_{avg}} \int_{t_{i-1}}^{t_{i-1}+\Delta t_{avg}} \sum_{k \in V_i}^{N_i} m_k \left\| \bar{\boldsymbol{v}}_k \right\|^2 dt, \tag{5.3.10}$$

$$T^{cfd}\left(y_i, t_i\right) = T, \tag{5.3.11}$$

where K_b is the Boltzman constant, N_i is the number of atoms and $\bar{\boldsymbol{v}}_k\left(y_i, t_i\right) = \boldsymbol{v}_k\left(y_i, t_i\right) - \boldsymbol{v}_i^{com}\left(y_i, t_i\right)$ is the peculiar velocity of particle k within sampling bin i. Lastly the density field is computed as

$$\rho^{md}\left(y_i, t_i\right) = \frac{1}{V_i \Delta t_{avg}} \int_{t_{i-1}}^{t_{i-1}+\Delta t_{avg}} \sum_{k \in V_i}^{N_i} m_k \, dt, \tag{5.3.12}$$

$$\rho^{cfd}\left(y_i, t_i\right) = \rho, \tag{5.3.13}$$

where m_k is the mass of the atom or molecule k.

5.3.4 Setup Benchmarking

This section is devoted to study the impact of different thermostatting strategies and the application of the FOREV algorithm from Section 3.2.3, on the the accuracy of the coupled scheme. Coupled simulations for five different setups, listed in Table 5.8, were benchmarked. A Langevin thermostat [142] in the y direction was employed for TH=CNST cases with the definition of the local kinetic temperature being

$$\bar{T}_k^{CNST} = \frac{1}{3N^{CNST}K_b} \sum_{k \in V^{CNST}}^{N^{CNST}} m_k \left\| \boldsymbol{v}_k - \bar{\boldsymbol{v}}_{com}^{CNST} \right\|^2, \tag{5.3.14}$$

and for TH=fluid cases computed as

$$\bar{T}_k^{f} = \frac{1}{N^f K_b} \sum_{k \in V^f}^{N^f} m_k \left\| \boldsymbol{v}_y^k \right\|^2, \tag{5.3.15}$$

where the symbols are defined similarly to the ones in Equation 3.4.25. The atomistic surface was thermostatted as described in Section 3.2.2. The FOREV [104] method for density correction presented in Section 3.2.3 was employed.

Setup	Thermostatting (TH)	Density Correction (DC)
1	fluid	yes
2	CNST	yes
3	no	yes
4	CNST	no
5	no	no

Table 5.8: List of setups benchmarked. The thermostatting strategies evaluated were TH=*fluid* (thermostatting all fluid atoms), TH=*CNST* (thermostatting just atoms in the CNST region) and TH=*no* (with no thermostat applied). The values DC=*yes* and DC=*no* indicate whether the density correction algorithm was applied or not.

The benchmarked scenario chosen was a Couette flow for thermodynamic point P_1 in Table 5.7 and geometry setup, boundary conditions and coupling parameters from Table 5.10 and Table 5.11. The density correction algorithm was applied after equilibration with parameters $\alpha = 10^{-4}$, $L_c = 3\sigma$, and $n_b = 50$ over a period of 100τ prior to inducing shearing. In order to compare quantitatively the accuracy of each setup in Table 5.8, an error metric L^2 was defined for a physical fluid quantity q in the domains as

$$L_D^2(N_R) = \sum_{t_i} \sum_{y_i \in D} \left| \langle q_D^c(y_i, t_i; N_R) \rangle - q^a(y_i, t_i) \right|^2, \tag{5.3.16}$$

and in the overlap region as

$$L_{Olap}^2(N_R) = \sum_{t_i} \sum_{y_i \in Olap} \left\langle \left| q_{cfd}^c(y_i, t_i; N_R) - q_{md}^c(y_i, t_i; N_R) \right|^2 \right\rangle, \tag{5.3.17}$$

where y_i is the discrete spatial coordinate, t_i is the discrete time, q_D^c is the value of the quantity in the coupled simulation, q^a is the analytical solution for the quantity for a Couette flow (see Section 2.4.2) and $\langle . \rangle$ is an operator which averages over N_R independent realizations of the coupled simulation. The quantities q considered for the benchmark within the MD and CFD domains were the fluid velocity in the x direction V_x, computed from

Equations 5.3.6 and 5.3.7, and the shear stress S_{xy}, computed from Equation 5.3.8 and 5.3.9.

N^s	a	ρ^s	$lattice$	N_{cy}	N_{cx}	N_{cz}
4800	1.58740	1.0	FCC	3	20	20

Table 5.9: Atomistic wall topology parameters. The wall contains N^s atoms arranged on a FCC (face-centered cubic) lattice with lattice parameter a and a mass density ρ^s. The limiting volume enclosing the surface is determined by $aN_{cx} \times aN_{cy} \times aN_{cz}$.

From the results reported in Figure 5.13 it can be seen that setup number 4 (TH=CNST, DC=no) is consistently the configuration with the smallest error in the MD, CFD and Overlap region for both V_x and S_{xy}. Not thermostatting the fluid causes an increase in the error. This is likely a consequence of a local temperature rise in the CNST region from applying (a) the momentum flux coupling force $\boldsymbol{F}_{\tau}^{ext}$ (Equation 3.4.17) and (b) the correction force F_y^c from the density correction algorithm. The worst performing setup is setup number 3 (TH=no, DC=yes), which consistently exhibit the highest error attending to results in Figure 5.13. The FOREV algorithm was designed to improve the behaviour of the fluid near non-periodic reflective boundaries in CFD-MD coupled simulations. However, a deleterious effect in the coupled scheme employed in this work (FV coupling [7]) can be observed. A plausible explanation of this behaviour could be due to the following two points:

a) The coupling scheme employed in this work consists on imposing a momentum flux within the CNST region, which will diffuse across the MD domain. The momentum balance is then always satisfied regardless of the local fluid structure. In other coupled schemes like (VV or VF) the velocities of the atoms are constrained based on the local density and viscosity of the fluid to ensure the local momentum balance [102,220,221]. Therefore, it is likely these types of coupling

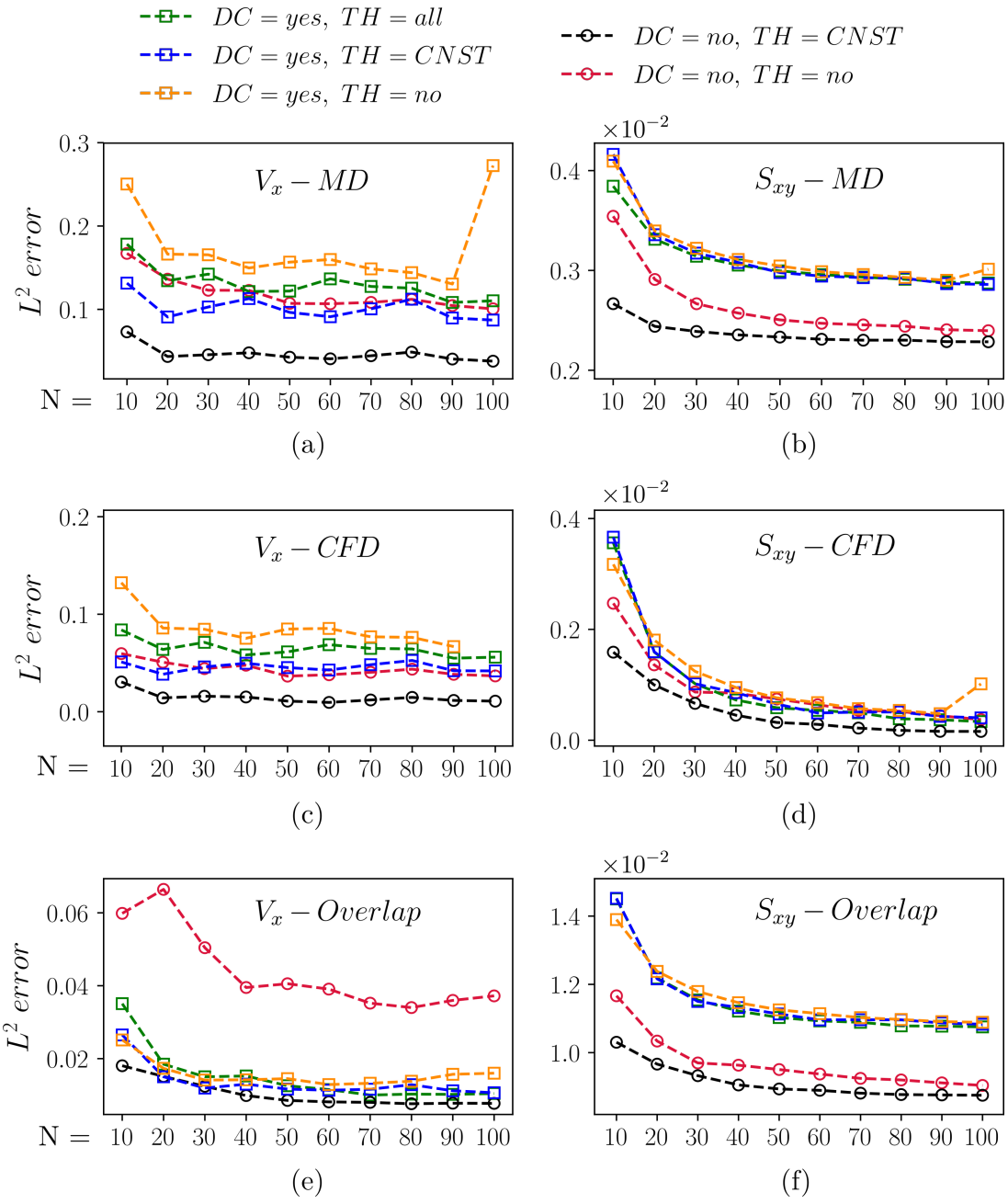


Figure 5.13: The L^2 (Equations 5.3.16 and 5.3.17) error metric is presented for different number of independent coupled simulations realizations (N) employed to compute V_x and S_{xy}. Results are plotted in (a) and (b) for the MD domain, (c) and (d) for the CFD domain and (e) and (f) for the overlap region. Five different setups (Table 5.8) are considered, with density correction (DC) being turned on/off and three different thermostatting strategies (TH) applied.

schemes are the ones to benefit from the FOREV algorithm as in Wang *et al.* [222].

b) The size of the region near to the reflective wall affected by density inhomogeneities is $\approx 2\sigma$ (see Figure 5.12), which is similar to the size of the coupling cell $d_y^c = 2\sigma$, where a single constant value of stress is applied.

Furthermore, considering just the setups with DC=yes, we can observe for TH=CNST, the V_x error is lower than for TH=*fluid* in both the CFD and MD domain. Since the shear heating is almost negligible (1% temperature rise in the steady-state from Figure 5.15.a), thermostatting every atom in the fluid is likely to interfere with the particle dynamics, shifting the local MD viscosity of the fluid from the value provided to the CFD solver. Hence, Setup 4 is, attending to the error metric L^2, the one consistently achieving the most accurate results, and is therefore the one employed in simulations presented in Section 5.3.5 and Section 5.4.3.

5.3.5 Numerical Validation of Coupled Flows

The purpose of this section is to validate the coupling strategy implementation described in Chapter 3 and the software framework developed presented in Chapter 4. Three different parallel flows discussed in Section 2.4.2 were studied by MD-CFD domain decomposition coupling (see Figure 3.1): Couette, Poiseuille and Rayleigh-Stokes. The mean-flow coupled solution for the physical fields V_x, S_{xy}, T and ρ are reported here plotted against their analytical expressions for a no-slip scenario.

The mean-flow fields were computed following Equation 5.3.4, with a time-average period of $\Delta t_R = 100\tau$. The MD fields were sampled at an interval $\Delta t_s = 0.05\tau$ and averaged over a $\Delta t_{avg} = \Delta t_{cfd}$ period and $N_R = 10$ realizations of the flow as described in Section 3.4.2 and following Equations 5.3.6 for V_x, 5.3.8 for S_{xy}, 5.3.10 for $T$and 5.3.12 for ρ. A mono-atomic Lennard-Jones MD fluid model was employed with interactions defined by the potential parameters given in Table 5.6. For each flow type, the boundary conditions applied to each domain (MD and CFD) and timing

parameters values are given in Table 5.10. Analogously, geometrical and mesh parameter values can be found in Table 5.11. The specifics about the coupling methodology followed in the CNST and BC regions are detailed in Section 3.4 and the building and equilibration of the coupled systems were carried out as described in Section 5.3.1. Wall surface and topology parameters are found respectively in Table 5.7 and Table 5.9. Setup 4 from Table 5.8 was used. This setup was found to be the one leading to the smaller error as discussed in Section 5.3.4. It is important to clarify that the CFD domain lengths L_{cfd} used in the studies in this section, are not big enough to obtain a significant computational advantage over simulating the whole domain using MD. The purpose of the studies presented here was to test the validity of the implementation and not to take advantage of the coupling methodology. Coupled simulations with CFD domains an order of magnitude larger than the MD can be found in Section 5.4.3.

Boundary Conditions							
Flow type		u_x^{Top}	u_x^{surf}	T_{Top}	T_{surf}	P_{Right}	P_{Left}
1	Couette	1.0	0	1.1	1.1	0	0
2	Poiseuille	0	0	1.1	1.1	0	0.01
3	Rayleigh-Stokes	$\sin(2\pi \times 3 \cdot 10^{-4} t)$	0	1.1	1.1	0	0
Timing							
Flow type		TSR	Δt_{cfd}	Δt_{avg}	Δt_{md}	t_{eq}	t_{nemd}
All		50	0.250	0.250	0.005	1250	5000

Table 5.10: Boundary conditions and timing parameters employed in coupled simulations for Couette, Poiseuille and Rayleigh-Stokes flow study cases. All values are in reduced units (see Table 2.1). Boundary conditions are labeled as *Top*, *Right* and *Left* according to the boundary planes in the CFD (see Figure 3.2, Tables 3.3 and 3.5). The u_x^{surf} and T_{surf} symbols, are the atomistic surface's center of mass velocity and the thermostat target temperature applied to the surface. The TSR is the time-step ratio between the CFD (Δt_{cfd}) and MD (Δt_{md}) time steps and Δt_{avg} is the averaging time to generate a field sample. Δt_{eq} and Δt_{nemd} stand for the equilibration and NEMD simulation time.

CFD and Coupling Mesh								
Flow type		d_y^c	N_{cx}	N_{cy}	N_{cz}	N_{cy}^{olap}	N_{cy}^{CNST}	N_{cy}^{BC}
1	Couette	2.0	1	50	1	15	1	1
2	Poiseuille	2.0	1	30	1	10	1	1
3	Rayleigh-Stokes	2.0	1	20	1	10	1	1
Geometry								
Flow type		L_{ch}	L_{cfd}	L_{md}	L_{olap}	A_{xz}	L_{surf}	L
1	Couette	71	100	80	30	31.748^2	9	150
2	Poiseuille	71	60	80	20	31.748^2	9	120
3	Rayleigh-Stokes	71	40	80	20	31.748^2	9	100

Table 5.11: CFD and coupling mesh and geometry parameters employed in coupled simulations for Couette, Poiseuille and Rayleigh-Stokes flow study cases. All values are in reduced units (see Table 2.1 for conversions). The N_{cx} , N_{cy}, and N_{cz} symbols are the number of cells of the CFD mesh in each direction and , N_{cy}^{CNST}, N_{cy}^{BC} and N_{cy}^{olap} are the number of cells along the y direction in the CNST, BC and overlap region as shown in Figure 3.3. The mesh cell-size for the CFD and coupling mesh is d_y^c and the area of the MD and CFD domain perpendicular to the y direction is A_{xz}. The L_{md}, L_{cfd}, L_{ch} , L_{olap} and L_{surf} lengths are defined along the y direction as depicted in Figure 3.1. The effective length occupied by the fluid is $L = L_{ch} + L_{cfd} - L_{olap}$.

The first study case considered was Couette flow. It has become the *de facto* flow type to simulate in the specialized MD-CFD coupling literature for benchmarking any new coupling methodology [7]. The simplicity of the flow and the setup similarity with full MD shear-driven simulations designed to study slip at the atomistic level (e.g for nanotribology applications [79,223]), make it a good initial test case. In this setup, a velocity boundary condition $u_x^{Top} = 1.0$ is applied at the top of the CFD domain. Two different thermodynamic points $P_1 \equiv (T = 1.1, \rho = 0.81)$ and $P_2 \equiv (T = 1.1, \rho = 0.7)$ were studied to provide stronger evidence the coupling strategy works regardless fluid's thermodynamic conditions. Looking at Figure 5.15 we can

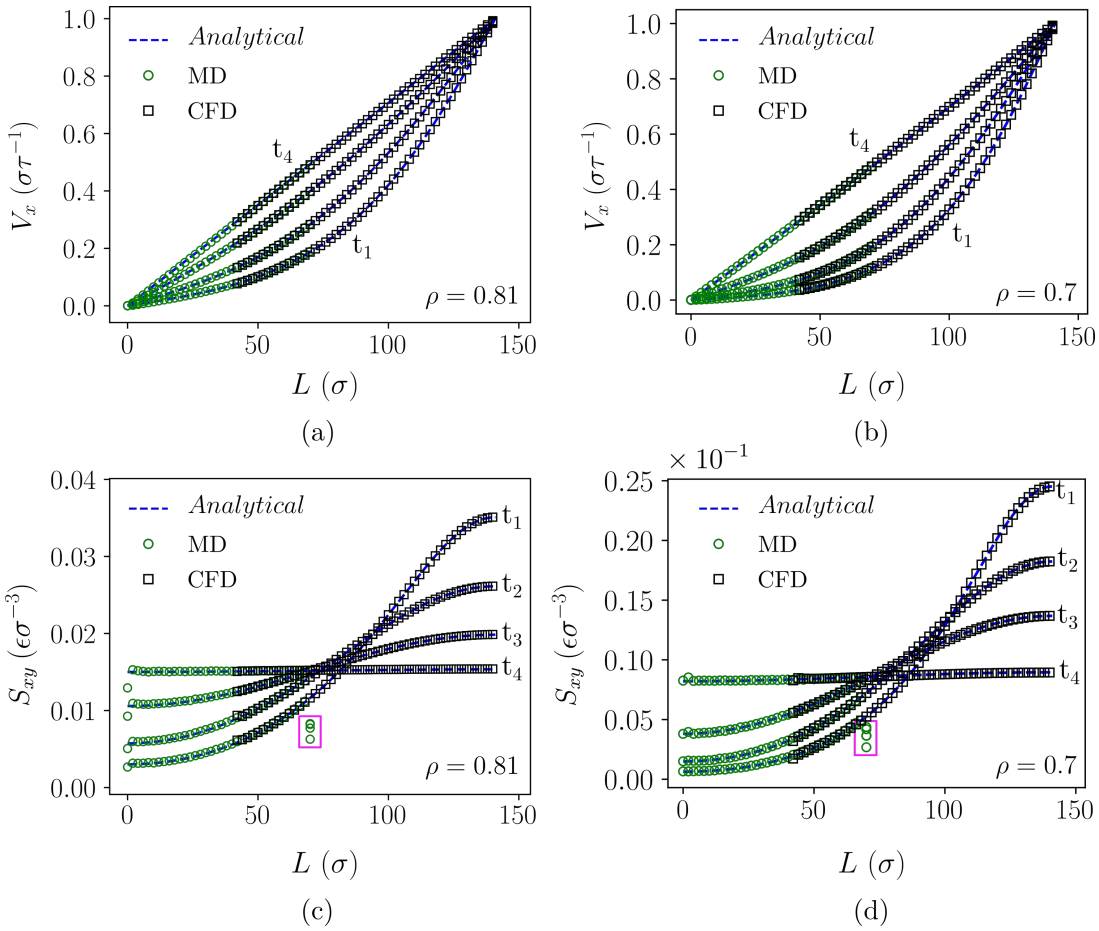


Figure 5.14: Time-evolution of the mean-flow velocity V_x and shear stress S_{xy} in a Couette flow (simulation parameters in Table Table 5.10) at temperature $T = 1.1$ and densities $\rho = 0.81$ and $\rho = 0.7$ (reduced units). The coupled solution is divided into the MD (green circles) and CFD (black squares) solutions which overlap. The analytical expression for the flow is plotted as a blue dashed line and has been generated from Equation 2.4.21. The solutions are given for times $t_i \in \{600, 900, 1500, 4000\}\,\tau$ for $\rho = 0.81$ and $t_i \in \{600, 900, 1500, 4500\}\,\tau$ for $\rho = 0.7$. An averaging period of $\Delta t_R = 1000\tau$ and $N_R = 10$ independent instances of the flow has been used for denoising following Equation 5.3.5. Pink boxes surround the shear stress data-points in the MD domain closest to the top reflective boundary.

see that the steady-state temperature field only experiments a mild maximum increase in the MD of $\approx 1\%$ with respect to the initial temperature. This increase has a negligible impact on liquid properties, hence the simulations can be considered quasi-isothermal. The liquid

properties of the LJ fluid at thermodynamic points P_1 and P_2 can be found in Table 5.7. Since the shear-rates explored during the time-evolution of the flow fall within the Newtonian region of the $\dot{\gamma}$-ν curve (Figure 5.8), a Newtonian CFD solver was used with a single viscosity coefficient computed from equilibrium Green-Kubo simulations from Section 5.2.1. A total of $N_1^f = 59075$ fluid atoms were simulated for thermodynamic point P_1 and $N_2^f = 50987$ for P_2. The MD domain employed a $5 \times 1 \times 4$ processor partitioning scheme for parallelization in `LAMMPS` and $1 \times 1 \times 1$ in `OpenFOAM` (single processor). Furthermore, spatial and temporal coupling constraints listed in Table 3.7 were make sure to be fulfilled. From Figure 5.14 we can observe that the agreement between the analytical solution and the coupled solution for V_x and S_{xy} are within the plot's symbols size. Furthermore, there is also excellent agreement between the MD and CFD solutions in the overlap region, except for the S_{xy} values close to the top MD boundary which are enclosed in a pink box. Those values are always below the CFD solution since they do not include the shear stress contribution from momentum flux coupling force in the CNST region, which is collocated with the sampling bin for the data point.

The second test case was a coupled planar Poiseuille flow. The setup employed was very similar to the Couette case described above, except for the boundary conditions applied (Table 5.10). This type of flow would be similar to the one found in an very long pipe where a pressure difference $\Delta p = P_{Right} - P_{Left}$ exists between the far ends, generating a constant force per unit length $F_x^{\Delta p} = \Delta p / L_x$ (assuming constant gradient) in the fluid, L_x being the longitudinal length of the pipe section considered along the x direction. Hence, in order to simulate the pressure difference in the MD domain a constant force was added to the atomic equations of motion as

$$F_{x,i}^{\Delta p} = \frac{m_i \Delta p}{L_x \left(\rho - \Delta\rho\right)}, \tag{5.3.18}$$

where m_i is the atomic mass, $L_x = \sqrt{A_{xz}}$ and ρ, $\Delta\rho$ are the density and density offset from Table 5.7. The simulation was performed at thermodynamic point P_1 with a total number of fluid atoms $N_1^f = 59075$ and

the MD domain employing a $1 \times 5 \times 4$ processor partitioning scheme for parallelization in `LAMMPS` and $1 \times 1 \times 1$ in `OpenFOAM`. The quasi-isothermal condition was also enforced in this case and excellent agreement was found between coupled and analytical solutions for velocity and shear stress fields, as it can be seen in Figure 5.16.

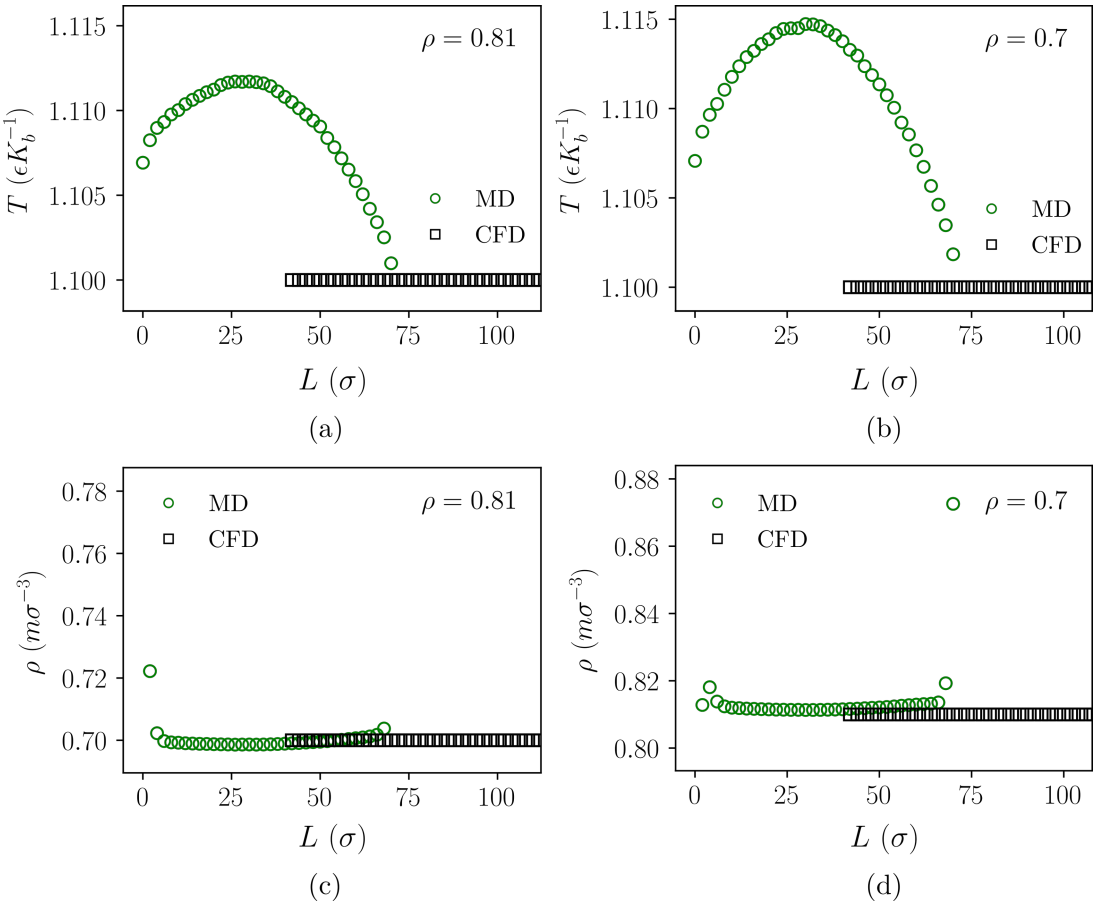


Figure 5.15: Steady-state temperature (a), (b) and density (c), (d) in a Couette flow (simulation parameters in Table 5.10) at temperature $T = 1.1$ and densities $\rho = 0.81$ and $\rho = 0.7$ (all in reduced units). The coupled solution is divided into the MD (green circles) and CFD (black squares) solutions which overlap.

Lastly, a Rayleigh-Stokes flow was studied to evaluate the coupling implementation with time-dependent boundary conditions. In this setup, the atomistic wall remains steady and an oscillating boundary condition $u_x^{Top} = \sin(2\pi f t)$ with $f = 3 \cdot 10^{-4}$ is applied on the CFD. The rest of the

setup was very similar to the Couette case study. From Figure 5.17 it is clear that the agreement between MD, CFD and analytical solutions, for V_x and S_{xy} fields, is again excellent. This signifies the coupling procedure reacts correctly during the transient to a time-evolving velocity boundary condition, which is important in the context of capturing time-dependent slip behaviour.

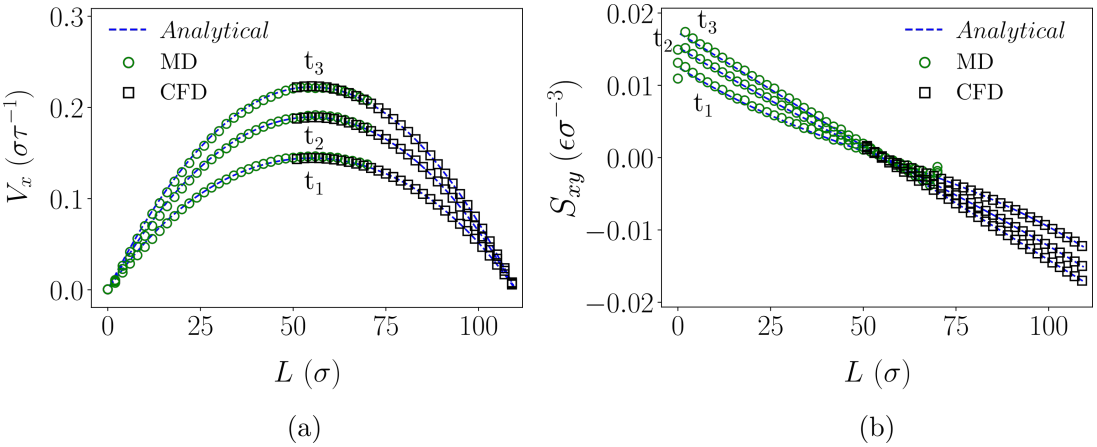
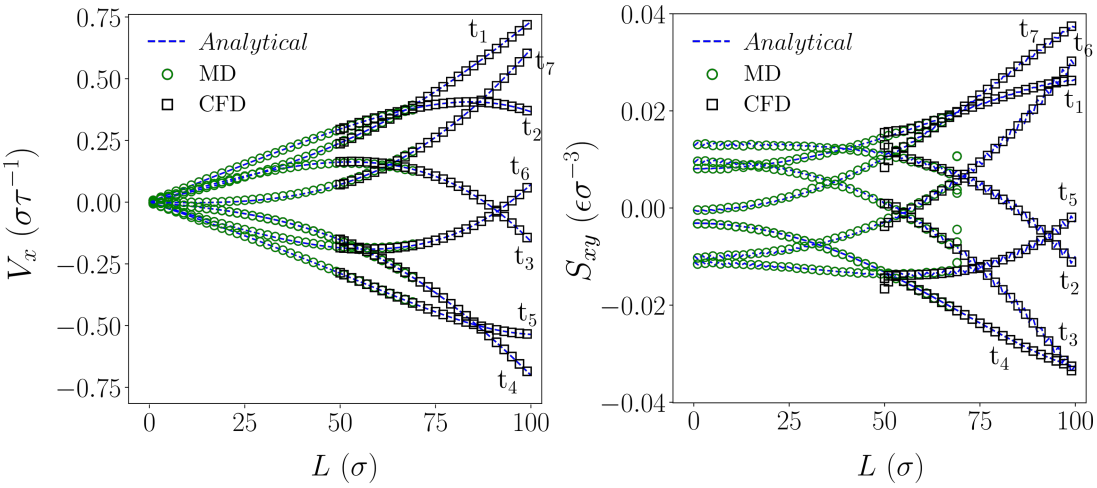


Figure 5.16: Time-evolution of the mean-flow velocity V_x and shear stress S_{xy} in a Poiseuille flow (simulation parameters in Table 5.10) at temperature $T = 1.1$ and density $\rho = 0.81$ (in reduced units). The coupled solution is divided into the MD (green circles) and CFD (black squares) solutions which overlap. The analytical expression for the flow is plotted as a blue dashed line and has been generated from Equation 2.4.26. The solutions are given for times $t_i \in \{600, 1000, 4000\}\,\tau$. An averaging period of $\Delta t_R = 1000\tau$ and $N_R = 10$ independent instances of the flow has been used for denoising following Equation 5.3.5.

In this section the correctness of the coupling strategy and software implementation developed for LJ quasi-isothermal Newtonian fluids has been verified. Three different types of flows has been studied for shear-driven and pressure driven scenarios. It has been proven that (i) both velocity V_x and stress S_{xy} fields are consistent between MD and CFD domains within the overlap region and (ii) that the full transient coupled solution consistently agrees with the respective flow analytical expression. The results obtained

pave the ground to study molecular fluids in Section 5.4, which present higher complexity and their own particular challenges.



(a) (b)

Figure 5.17: Time-evolution of the mean-flow velocity V_x and shear stress S_{xy} in a Rayleigh-Stokes flow (simulation parameters in Table 5.10) at temperature $T = 1.1$ and density $\rho = 0.81$ (in reduced units). The coupled solution is divided into the MD (green circles) and CFD (black squares) solutions which overlap. The analytical expression for the flow is plotted as a blue dashed line and has been generated from Equation 2.4.32. The solutions are given for times $t_i \in \{900, 1500, 1900, 2500, 3000, 3500\}\,\tau$. An averaging period of $\Delta t_R = 1000\tau$ and $N_R = 10$ independent instances of the flow has been used for denoising following Equation 5.3.5.

5.4 Coupled Molecular Simulations

The simplicity of Lennard-Jones (LJ) fluids makes them ideal candidates to test new methodologies and design studies which can produce insights transferable to more complex systems [4,128,135,224]. Much of the methodology and infrastructure developed for CFD-MD coupling in this work has been, on a first approach, built using LJ particles and then adapted to work with more complex molecules using high quality force-fields to model their interactions [52]. There are several complexities that arise from working

with molecular fluids which can be both technical or operational. However, ultimately most of them lead directly to an increase of the computational time needed to simulate those systems. Some of the most significant difficulties, pervasive to molecular simulations and in particular to coupling, are:

a) Selection of a suitable *force-field* depending on the nature of the simulations. In this work we have used OPLS [56] and its variant for long chain hydrocarbons L-OPLS [63] in its all-atom version which produce realistic results for thermodynamic and hydrodynamic properties at high pressure (see results for squalane in Section 5.2.2).
b) *Noise* due to the presence of molecular high frequency modes [225]. Bonds of light atoms like hydrogen and the contributions to the viral stress of other force-field energy terms (not present in LJ fluids) can increase dramatically the signal to noise ratio (SNR) while computing thermodynamic and hydrodynamic quantities. Noise appear to be particularly harmful for the two-way coupling strategy used in this work and the matter is discussed in Section 5.5. To reduce noise, higher number of statistical samples and bigger overlap regions are needed resulting in an increase of the computational simulation time.
c) *Equilibration* criteria and equilibration time. The particular molecular topology and the thermodynamic point at which the equilibration is performed, can affect significantly the equilibration time in molecular system due to the large configuration space to be sampled [226,227].
d) Building initial configurations and mapping force-fields to atom types in order to produce input scripts for molecular dynamics solvers (e.g `LAMMPS`), can be time-consuming.
e) The need for parallel techniques and high performance computing in order to simulate molecular systems at length- and time-scales of interest. Hence, any multi-scale framework developed to study these systems has to be designed to run in parallel/distributed computing environments. In Section 4.4.2 a series of parallel scalability studies of the present framework can be found.

In general, the process of performing a coupled CFD-MD simulation can be divided into three consecutive independent steps: (1) building the system, (2) equilibrating the system and (3) running a NEMD coupled simulation. Operationally, this separation of steps is less important for LJ fluids, since building and equilibration are usually trivial, and a single script can efficiently pipeline the work. For the molecular case, each of the step can take significant time on its own and it could be very inefficient if they are not decoupled one from each other. For instance, an initial system configuration can be used to perform several equilibrations at different thermodynamic points. Similarly, an equilibrated system can be leveraged to run several NEMD simulations (e.g different wall velocities and/or pressure gradients applied as in Section 5.4.3). Sections 5.4.1 and 5.4.2 explain the process of building and equilibrating the molecular systems employed in coupling in Section 5.4.3, where NEMD coupled simulations results are presented for benzene and squalane.

5.4.1 Building Coupled Molecular Systems

As outlined in the previous section, building molecular systems requires significantly more effort than building a system containing mono-atomic fluids. This is due to existence of molecular topology and the need to assign force-field parameters to groups of atoms inside a molecule (bonds, angles and dihedrals in OPLS [56]). For this purpose, two specialized software packages were used in this work. `Packmol` [205] was employed to generate initial configurations for the molecules and `Moltemplate` [206] to generate `LAMMPS` input files containing topology and force-field information of the system (see Figure 5.18).

A systematic approach was developed and tested to build every molecular system used for MD-CFD coupling, which allowed the automation of several steps of the process. This effort is aligned with trying to lower the cost of entry for new researchers to use the multi-scale coupling methodology developed in this work. The strategy followed was independent of any force-

field, molecular topology or system size, and it has proven to work robustly for the molecules tested (*benzene* and *squalane*) at different thermodynamic points.

In Figure 5.18 a schematic of the process of generating the set of input files for `LAMMPS` necessary for the equilibration of the molecular system is depicted. We can distinguish three regions along the y direction: the bulk of the fluid, the adsorbate layer and the surface. Firstly, a slab of a crystalline structure is generated using the ASE (Atomistic Simulation Environment) software package [228]. A height-map mask is then applied to the slab to sculpt the texture pattern of interest on the surface (posts, sinusoidal, correlated/fractal roughness, etc.). Next, the interaction potential between the surface atoms is specified and bonds between groups of atoms are defined by atom type using a cut-off distance criteria. A `Moltemplate` template file (`surface.lt`) is generated for the surface. In this work, a mono-atomic Cu surface with FCC lattice structure has been employed, but other structures can be easily built following the same procedure. The potential parameters used for the surface can be found in Table 5.12, which is based on harmonic interactions (Equation 3.2.3). Following the creation of the surface, a combined system (`system.lt`) is defined where the domain is divided into the bulk fluid sub-domain, the adsorbates sub-domain and the surface sub-domain. All sub-domains have the same physical dimensions in the x and z direction (both periodic), which are determined by the wall (surface) dimensions along those axes. A buffer distance between sub-domains and domain edges of $d_b = 2$ Å is enforced, which is the tolerance distance specified for `Packmol` minimization algorithm. This ensures overlap is avoided between atoms in different sub-domains and also that molecular atoms are sufficiently separated from each other for the `LAMMPS` minimization algorithm to work efficiently.

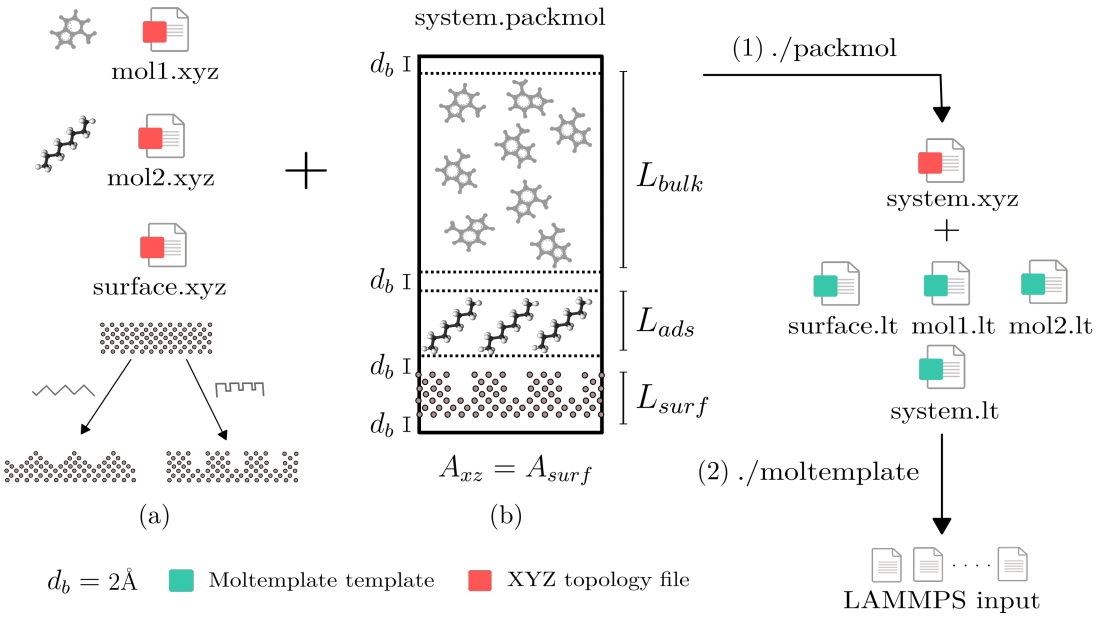


Figure 5.18: Schematic of the process to create the necessary input files for `LAMMPS` in order to perform coupled simulations. In the first build stage (1) a XYZ file `system.xyz` is generated with the initial random topology of the system using `Packmol`. The input files provided to `Packmol` are (a) the XYZ topology files of the molecules and the surface and (b) the file with the system topology `system.packmol`. The system topology is comprised of three stacked regions (fluid, adsorbates and surface), each separated in the y direction by a distance d_b. The system area in the X-Z plane A_{xz} is given by the area of the atomistic wall defined in the `surface.xyz` file. In the second build stage (2) `Moltemplate` combines the topology in the `system.xyz` file with `Moltemplate`'s template files (*.lt files), which provide force-field parameters for molecules and the surface. The output of `Moltemplate` are all the necessary input files for `LAMMPS`. At this point the system is ready to be equilibrated.

The initial length, along y direction, of the bulk-fluid (L_{bulk}) and adsorbates (L_{ads}) sub-domains is computed as

$$L_{bulk/ads} = (1+f)\frac{N_1M_1 + N_2M_2 + ... + N_nM_n}{A_{xz}(X_1\rho_1 + X_2\rho_2 + ... + X_n\rho_n)}, \tag{5.4.1}$$

where N_i, X_i, M_i and ρ_i are the number of molecules, mass fraction, total mass and target density for a type of molecule i respectively; A_{xz} is the area of the system in the X-Z plane and the factor $f \in [0,1]$ has been included to

facilitate `Packmol` achieving a valid initial configuration. In high density fluids ($\approx 10^3$ kg/m^3) molecules are considerably packed and a value of $f \approx 0.2$-0.4 was found to be necessary to help `Packmol` optimization algorithm find a valid initial configuration. Using the full system topology (`system.xyz`) generated by `Packmol`, and the `Moltemplate` templates (.lt files) for each of the molecules and the surface, the input files for `LAMMPS` are created.

The last step prior to the equilibration of the system is to minimize its energy, which is done by using one of `LAMMPS` built-in minimization routines based on a conjugate gradient algorithm. Since the system is non-periodic in the y direction, atoms may fall out of the box during minimization. To avoid this issue, two virtual Lennard-Jones virtual walls (no physical atoms are added) are placed at the top and bottom of the domain before the minimization procedure is applied. The interaction energy between fluid atoms and the virtual wall is modeled using the LJ potential with parameters $\sigma = 2.0$Å, $r_c = 2.0$Å and $\epsilon = 0.1$ kcal/mol.

After the energy of the system is minimized, the LJ virtual walls are replaced by reflective walls in the y direction which modify the atoms' positions and velocities according to Equation 3.2.6 and Equation 3.2.7. This configuration remains the same for both equilibration and NEMD coupled simulations.

5.4.2 Equilibration of Coupled Molecular Systems

An equilibrated configuration of the molecular system is required to start a coupled simulation, so the liquid is in a well-defined thermodynamic state. There are two strategies (see Figure 5.19) for bringing a non-periodic system bounded by a wall close to an equilibrated state: (a) density control techniques (*mass-statting*) within the bulk region or (b) applying an external pressure (*barostatting*) to the system through the surface.

The natural target pair of thermodynamic quantities for equilibration are temperature and pressure, therefore, the mass-statting algorithm requires the equation of state of the liquid to be known. Two techniques were implemented in this work: (a) damped barostatting [229] and (b) mass-

statting by wall displacement. To compare both techniques, benzene (OPLS) and squalane (L-OPLS) were used as an example of a simple small molecule and a larger and more complex hydrocarbon [203,204]. Differences in the final state of the system were found between the two methods which are presented at the end of this section.

N_s	σ_{sf} (Å)	K_s^1 $\frac{\text{Kcal}}{\text{molÅ}^2}$	$R_{in}^1/R_{out}^1/r_{ij}^0$ (Å)	K_s^2 $\frac{\text{Kcal}}{\text{molÅ}^2}$	$R_{in}^2/R_{out}^2/r_{ij}^0$ (Å)
50000	$\sqrt{3\sigma_f^j}$	200	0/3/2.55265	200	3.5/4/3.61

Table 5.12: Parameters for a 180.5×18.05×180.5 Å^3 FCC(100) atomistic Cu surface. N_s is the number of atoms in the surface, σ_{sf} is the surface-fluid length-scale LJ parameter following Lorentz-Berthelot rules (see Equation 2.3.9). Harmonic potential parameters $K_s^n, R_{in}^n, R_{out}^n$ and r_{ij}^0 are defined for each of the two shells of atoms $n = 1, 2$ following Equation 3.2.3.

The simulation setup for testing mass-statting and barostatting techniques for both benzene and squalane was the same except for the domain length L_0**.** The domain dimensions in the X-Z plane were fixed to 180.5×180.5 Å^2 and an initial domain length of $L_0 = 140.15$ Å was employed for benzene and $L_0 = 310.15$ Å for squalane. Periodic boundary conditions where applied in the x and z directions and the domain was bounded by reflective walls in the y direction following Equation 3.2.6 and 3.2.7. A flat copper surface with the [100] lattice vector perpendicular to the surface plane was employed. The surface contained $N_s = 50000$ atoms interacting through a harmonic potential (Equation 3.2.3). Potential parameters for the surface and surface-fluid interactions can be found in Table 5.12. Lennard-Jones interactions were disabled between wall atoms and initial configurations of fluid molecules were generated as described in Section 5.4.1.

A Langevin stochastic thermostatting force [142] was applied to all atoms (fluid and surface) in x and z directions (the center of mass velocity of the system in the y direction is non-zero during equilibration due to the wall compression). The first layer of wall atoms were tethered (see Figure 3.1) to the lattice sites using a harmonic potential with $K = 1000$ Kcal/Å^2. The

equations of motion of the atoms were integrated using a velocity Verlet algorithm (see Section 2.3.3) with time-step $\Delta t_{md} = 1.0$ fs. Furthermore, a load-balancing algorithm, based on the number of particles per processor, was applied in the y direction to increase the parallel efficiency of `LAMMPS` equilibration. Since the bottom of the simulation box becomes empty as the system undergoes compression, a re-partition of the domain is desirable to assign more processors to high density regions.

The equilibration process is done in three stages represented in Figure 5.19. Firstly in stage (1), a post-minimization run of 0.2 ns precedes barostatting and mass-statting runs. In this stage the surface's first layer of atoms is tethered to the lattice sites using a harmonic potential as written in Equation 3.2.2. This stage ensures departing from a well-behaved initial system configuration (not necessarily fully equilibrated). Stage (2) comprises the actual barostatting or mass-statting algorithm and both are explained below in detail. The surface is not tethered to the lattice sites during this stage. Lastly, in stage (3) the simulation box is re-scaled in the y direction to eliminate the empty space underneath the surface, leaving just a gap of 2 Å between the surface and the bottom of the box domain. The surface is tethered back as it was during stage (1). At this point the system is prepared to be coupled with the CFD solver and perform NEMD shear simulations.

Barostat Equilibration

A damped barostat algorithm has been implemented where a target pressure P_T was applied to the first layer of surface atoms in addition to a damping force. The damping force helps the to dampen the fluctuations of the system, specially for low pressure simulations. The expression for the force added to the equations of motion of each individual atom within the wall's bottom atomic layer is

$$F_i^{bar} = \frac{\left(1 + \gamma v_{y_{com}}^s\right) P_T A_{xz}}{N_s^1}, \tag{5.4.2}$$

where F_i^{bar} is the barostatting force applied to atom i inside the first layer L_s^1 of the surface, $\gamma \in (0, 1)$ is a damping constant, $v_{y_{com}}^s$ is the y component

of the center of mass velocity of the surface, N_s^1 is the number of atoms in the first layer of atoms of the surface, P_T is the target pressure and A_{xz} is the surface area parallel to the X-Z plane.

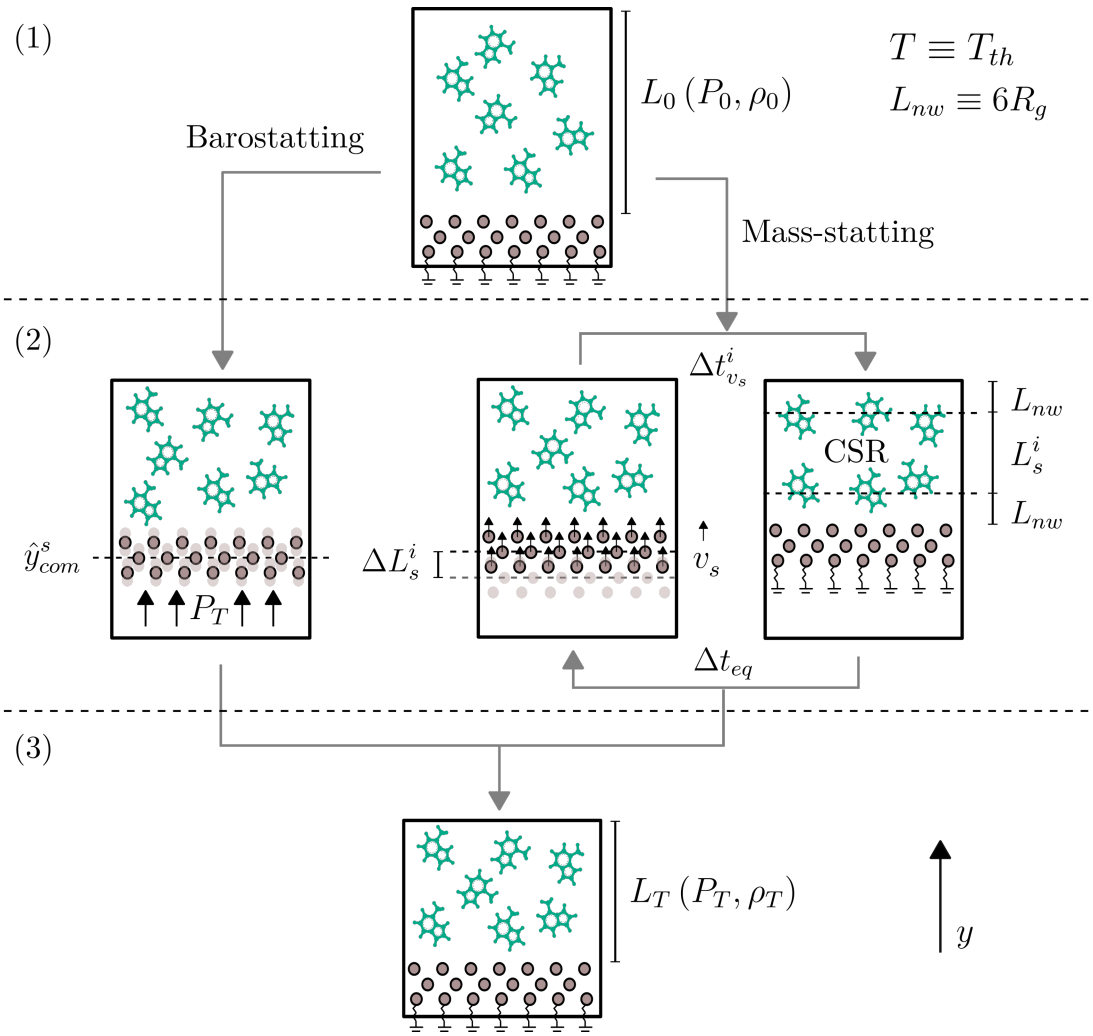


Figure 5.19: Flow diagram of the equilibration of a molecular system with a bounding atomistic wall at temperature T_{th} using either *barostatting* or *mass-statting*. The process can be divided in three stages being (1) and (3) common between both techniques. During stage (1) the surface atoms are tethered at their initial lattice positions and the system is equilibrated in that configuration; at stage (2) the barostatting/mass-statting algorithms are executed to drive the system close to the thermodynamic point (P_T, ρ_T). Lastly, in stage (3), the system is resized in the y direction to eliminate the empty space at the bottom of the domain and the atomistic surface is tethered back. Also, a final equilibration run is performed to leave the system ready to run NEMD simulations.

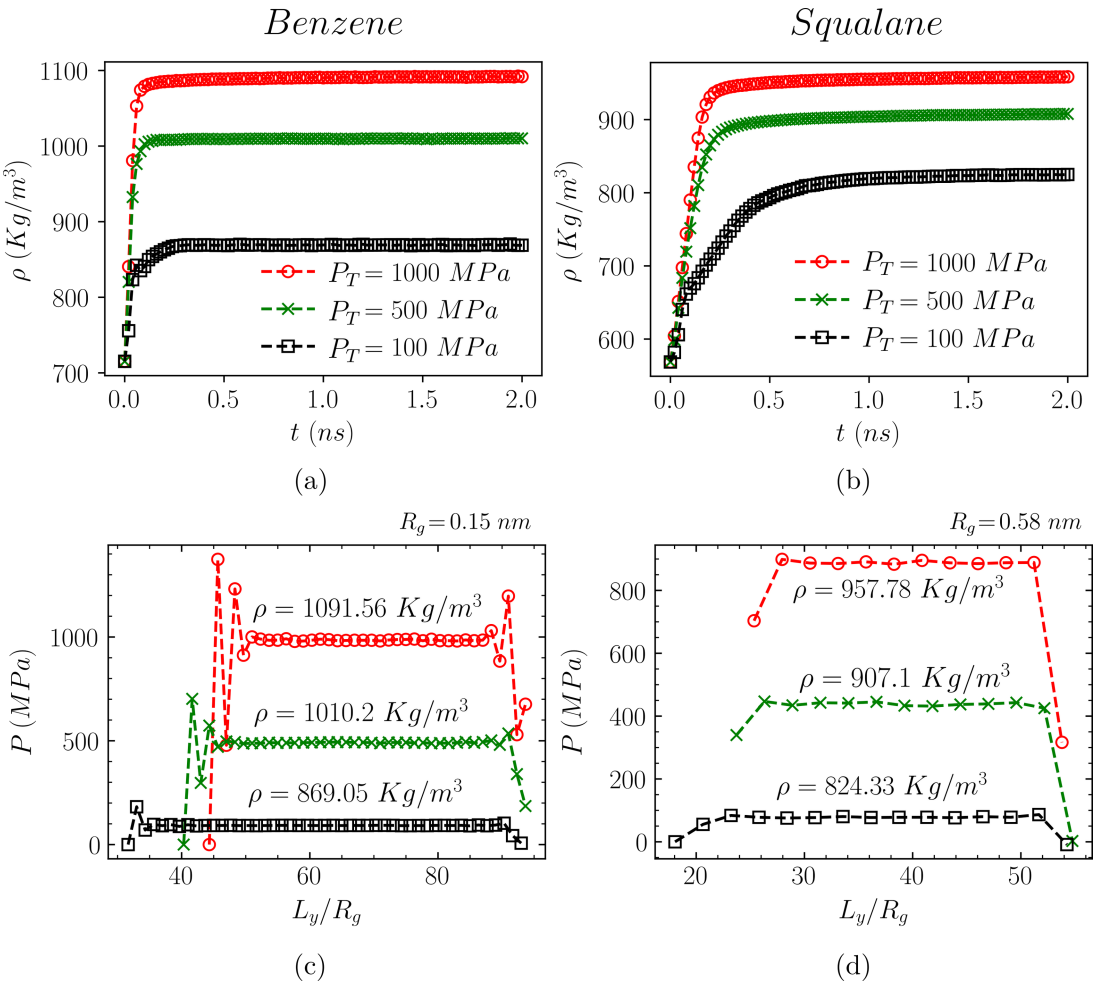


Figure 5.20: Plots (a) and (b) show the time-evolution of the average fluid density for benzene and squalane at $T = 300$ K using the barostatting technique at three different target pressures $P_T \in \{100, 500, 1000\}$ MPa. The time for reaching a constant density for both molecules, regardless of the pressure applied, is between 0.25-1.25 ns. The pressure profiles across the channel in the y direction for both molecules are plotted in (c) and (d). The bulk fluid region (flat pressure) starts around 5-7 R_g from top and bottom walls.

Three simulations where performed for benzene and squalane at $P_T \in \{100, 500, 1000\}$ MPa. The barostatting force was applied for 2 ns with a damping constant $\gamma = 0.01$ s/m. This allowed enough time for the system to reach a steady-state density as shown in Figure 5.20.a and 5.20.b. The mean value of the surface center of mass y coordinate $\hat{y}^s_{com}$, was computed

from $t \in [1.5, 2.0]$ ns period and the position of the wall was corrected with a post-barostatting run of $10^4 \times \Delta t_{md}$ time by freezing the surface atoms (no forces applied to them) and adding to every surface atom a constant velocity

$$v_y^{s,i} = \frac{y_{com}^s\,(t = t_f) - \hat{y}_{com}^s}{10^4 \times \Delta t_{md}}, \tag{5.4.3}$$

where $v_y^{s,i}$ is the velocity applied in the y direction to atom i within the surface, y_{com}^s is the instantaneous y component of the center of mass of the surface at $t_f = 2$ ns and Δt_{md} the time-step. After the position correction was applied, the surface is tethered back and a production run was carried out to compute the density profile across the domain presented in Figure 5.20.c and 5.20.d. A bin size of 0.2 nm and 1.5 nm was employed for benzene and squalane respectively and a total of $N_s = 50000$ samples were taken over 0.5 ns.

By examining the density profiles we can observe that for both benzene and squalane, at any of the pressures studied, the bulk behaviour of the fluid starts around 5-7 R_g (radius of gyration) distances from the atomistic surface and the reflective wall (regions with no ripples). This is an important number to know for two reasons: (a) to place the mass-stat control slab region (CSR) inside the bulk of the fluid and (b) to design the overlap region for coupled simulations such the BC region lays sufficiently far from the reflective wall where the bulk fluid behaviour is found.

Mass-stat Equilibration

Contrary to barostatting, the mass-statting algorithm implemented in this work is based on an iterative procedure (see Figure 5.19). The implementation is based on displacing, every time step i, the bottom atomistic surface a distance $\left|\Delta L_s^i\right|$ in the y direction. The system departs from an initial low density point $\rho_0 \equiv \rho(P_0, T_{th})$, to reach a target higher density $\rho_T \equiv \rho(P_T, T_{th})$ where P_0 and P_T are the initial and target equilibrium pressure and T_{th} is the global thermostat temperature. The density is computed in the *control slab region* (CSR), which is located in the

middle of the channel separated by a constant distance L_{nw} (length of near-wall regions in Figure 5.19) from both the top reflective wall and the bottom

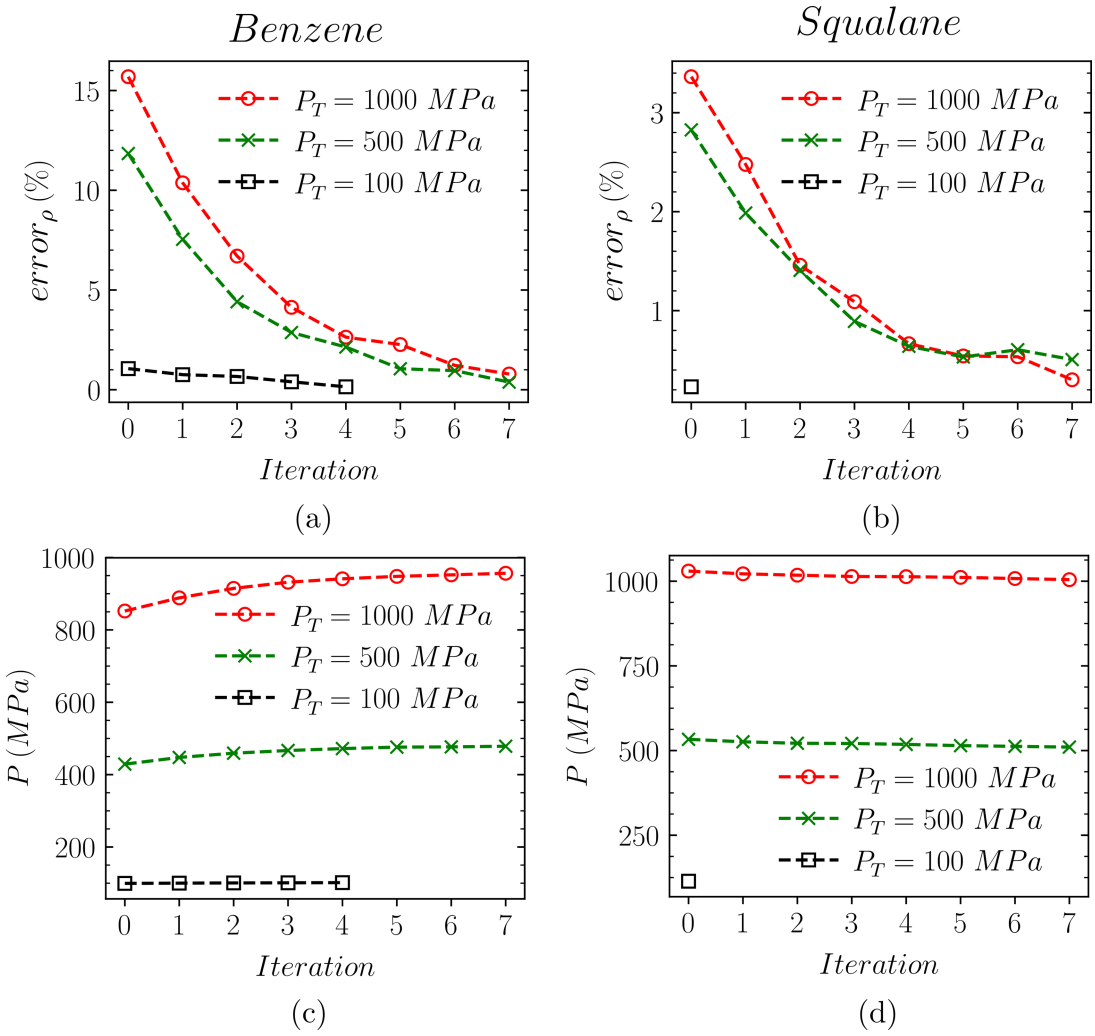


Figure 5.21: Percentage error for density inside the CSR for (a) benzene and (b) squalane at three different target pressures $P_T \in \{100, 500, 1000\}$ MPa at $T = 300$ K. The target density ρ_T for each corresponding pressure was computed from the respective EOS (Figure 5.9.a and Figure 5.10.a). The maximum number of iterations was $i_{max} = 8$ and the error tolerance allowed was $tol_\rho(\%) = 0.1$, which was only achieved for $P_T = 100$ MPa in both molecules. Plots (c) and (d) show the evolution of the pressure in the CSR as the iterations increase. Only one data point appear in (d) for squalane at $P_T = 100$ MPa since just a single iteration was needed to achieve the target density.

atomistic surface. The CSR must enclose the region of fluid which exhibit bulk behaviour. By inspecting Figure 5.20.c and Figure 5.20.d, a value of $L_{nw} = 6R_g$ ($R_g \equiv$ radius of gyration of the molecule) was found to be sufficiently large regardless of the system pressure or molecule. The surface displacement value ΔL_s^i at step i to attain $\rho_s^{i+1} \equiv \rho_T$ at step $i+1$ can be derived by considering

$$\begin{aligned} \Delta L^i &= L_s^{i+1} - L_s^i, \\ M_s^{i+1} &= \rho_T L_s^{i+1} A_{xz}, \\ M_s^i &= \rho_s^i L_s^i A_{xz}, \end{aligned} \tag{5.4.4}$$

where L_s^i, M_s^i and ρ_s^i are the length in the y direction, the mass and the density of the CSR at time step i respectively. Then, subtracting M_s^i to M_s^{i+1} we finally obtain

$$\begin{aligned} M_s^{i+1} - M_s^i &= \rho_T L_s^{i+1} A_{xz} - \rho_s^i L_s^i A_{xz}, \\ \implies \Delta L_s^i &= \frac{\rho_s^i - \rho_T}{\rho_T} L_s^i + \frac{M_s^{i+1} - M_s^i}{\rho_T A_{xz}}. \end{aligned} \tag{5.4.5}$$

In principle, if all the quantities in Equation 5.4.5 were known, just a single step would be needed to reach ρ_T. However, M_s^{i+1} is not trivial to estimate, and a sensible approximation is needed in order to compute ΔL_s^i as it is explained next.

Firstly, the total mass of the fluid at step i is $M_T^i = M_s^i + M_{nw}^i$, where M_{nw}^i is the near-wall mass computed from the fluid molecules outside the CSR. The density in those regions is always non-uniform (see Figure 5.12) and changes in a non-trivial way as ρ_s^i changes (depends on surface-fluid interactions, surface structure, etc.). We know the change of mass in the slab region between two time-steps is $M_s^{i+1} - M_s^i = M_{nw}^i - M_{nw}^{i+1}$, since $M_T^i = M_T^{i+1}$ and L_{nw} being constant. As the system is compressed, the mass inside the near-wall regions increases at expense of the mass inside the control slab. Since $L_{nw} \ll L_s^i$, it is expected that this transfer of mass would be comparatively small, making the second term in Equation 5.4.5 much less significant than the first, which accounts for the relative change of density

inside the CSR. Hence, we can approximate ΔL_s^i by dropping the second term in Equation 5.4.5

$$\Delta L_s^i \approx \frac{\rho_s^i - \rho_T}{\rho_T} L_s^i. \tag{5.4.6}$$

This updating rule, as shown in Figure 5.21.a and 5.21.b, makes the relative error for density inside the CSR

$$error_\rho = \left| \frac{\rho_s^i - \rho_T}{\rho_T} \right|, \tag{5.4.7}$$

to drop steadily as the number of iterations increases. An error tolerance $error_\rho^{tol} = 0.001$ (0.1% error in density) was used in all mass-statting simulations for both benzene and squalane. The relative error for the pressure $error_P$ plotted in Figure 5.22 was defined equivalently as

$$error_P = \left| \frac{P_s^i - P_T}{P_T} \right|. \tag{5.4.8}$$

A detailed description of the mass-stat algorithm can be found in Table 5.13.

Choosing an Equilibration Method

It is important to make clear that molecular systems resulting from the two equilibration methods described above, are designed to be used in a coupled NEMD simulation. The most important aspect at the end of the equilibration process is to obtain a small $error_\rho$. The smaller the error, the higher the agreement between pre-computed liquid properties passed to the CFD solver (see coupling workflow in Figure 3.4) and the actual intrinsic properties of the equilibrated MD fluid.

A comparison of $error_P$ and $error_\rho$ for barostatting and mass-statting at $T = 300K$ (benzene and squalane) was performed and reported in Figure 5.22. Numerical values for the final pressure P_F and density ρ_F at the three thermodynamic points studied are listed in Table 5.14. By examining Figure 5.22.a and 5.22.b it is clear that barostatting performs, in general, worse than mass-statting, even though pressure is the method's target variable

Mass-stat Algorithm

input: $\rho_T \equiv target\ density$
$tol_\rho \equiv density\ error\ tolerance$
$i_{max} \equiv max\ iterations$
$v_s := 100\ m/s$
$steps_{prod} := 300000,\ steps_{post} := 50000$
$dt_{md} := 1\ fs$

`/* Initialization */`

1 $CSR_{top} := y_{max} - 6R_g$;
2 $i := 0$;
3 $error_\rho^{i=0} := tol_\rho$;
4 $\Delta L_s^{i=0} := L_0 - \frac{M_T}{A_{xz}\rho_T}$;
5 **while** $(i < i_{max})\ and\ \left(error_\rho^i > tol_\rho\right)$ **do**
`/* Adjust surface position run */`
6 $steps_{adj} := round\left(\Delta L_s^i / \left(v_s \times dt_{md}\right)\right)$;
7 **run** ($\Delta t_{v_s}^i$ =steps$_{adj} \times dt_{md}$) {
8 $v_y^{k,surf} := v_s$;
9 };
`/* Equilibration run */`
10 tether_surface($x_y^{k,surf}$);
11 **run** (Δt_{post} =steps$_{post} \times dt_{md}$);
12 $CSR_{bot}^i := max(x_y^{k,surf})$;
`/* Production run */`
13 untether_surface($x_y^{k,surf}$);
14 **run** (Δt_{prod} =steps$_{prod} \times dt_{md}$) {
15 $P_s^i, \rho_s^i :=$ average_quantities(CSR_{bot}, CSR_{top});
16 $\rho_s^i :=$ average_density(CSR_{bot}, CSR_{top});
17 write(P_s^i, ρ_s^i);
18 };
`/* Update variables */`
19 $\Delta L_s^i := \frac{\rho_s^i - \rho_T}{\rho_T}\left(CSR_{top} - CSR_{bot}\right)$;
20 $error_\rho^i := \left|\frac{\rho_s^i - \rho_T}{\rho_T}\right|$;
21 $i := i + 1$;
22 **end while**

Table 5.13: The novel iterative mass-statting algorithm. The inputs are the target density ρ_T, the density error tolerance tol_ρ, and the maximum number of iterations allowed i_{max}. The control slab region (CSR) is defined to be $6R_g$ far from the top reflective boundary based on results reported in Figure 5.20.

		Barostatting		Mass-statting	
P_T (MPa)	$\rho_T \equiv \rho_{EOS}$ (kg/m^3)	P_F (MPa)	ρ_F (kg/m^3)	P_F (MPa)	ρ_F (kg/m^3)
Benzene					
100	890.51	91.89	869.05	101.54	890.36
500	1017.04	490.84	1010.20	478.47	1016.66
1000	1097.57	984.73	1091.56	956.50	1096.79
Squalane					
100	840.79	77.92	824.33	114.19	840.56
500	924.97	438.65	907.1	510.22	923.46
1000	975.14	888.12	957.78	1004.55	974.84

Table 5.14: Final pressure P_F and density ρ_F obtained at $T = 300$ K for three different target pressures $P_T \in \{100, 500, 1000\}$ MPa. Results are presented for mass-statting and barostatting. The target density ρ_T is defined as $\rho_T = \rho_{EOS}(P_T, T = 300 \text{ K})$.

A particular poor accuracy was found for squalane, with errors above 10% for the three pressures studied. A possible explanation could lie in the method of computing the pressure inside the bulk of the fluid. Since the virial pressure (Equation 2.2.25) assumes an infinite homogeneous system in the three dimensions, it would not be unreasonable to think that the violation of these assumptions (non-periodicity of the domain in the y direction) could lead to errors. On the other hand, by looking at $error_\rho$ in Figure 5.22.c and 5.22.d it is clear that the mass-statting algorithm always drives the system close to the EOS density. Table 5.14 shows that ρ_F is always underestimated by barostatting for both benzene and squalane. Small variations in density can lead to big changes in pressure in high density fluids as they behave as they are highly incompressible [230]. Hence, density seems a better candidate as a equilibration quantity to target since (a) its computation is unequivocal and (b) a viable iterative approach exists which allows to precisely control the error. Implementing an iterative barostatting algorithm would have been possible, but density is a much less noisy

quantity to calculate than pressure and it is always well-defined inside the CSR.

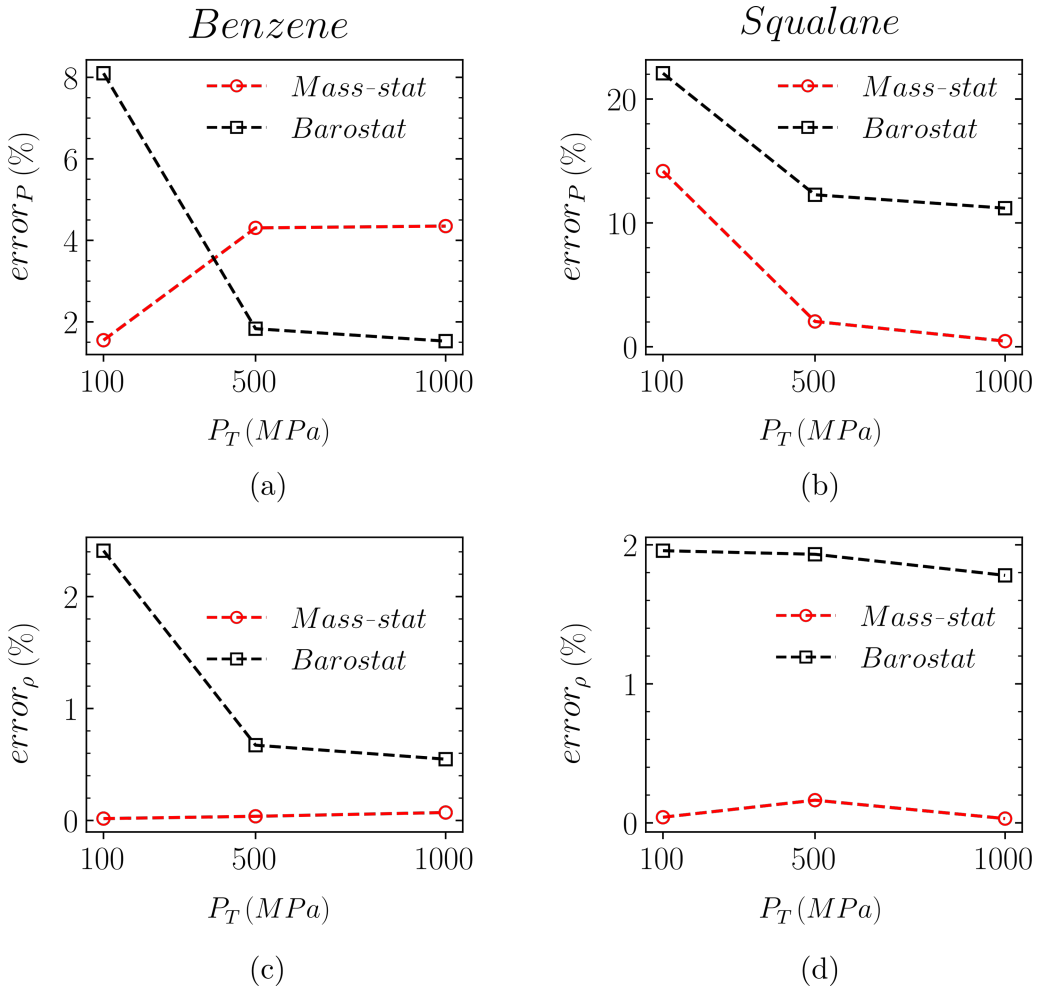


Figure 5.22: Percentage error of pressure, $error_P$, and percentage error of density, $error_\rho$, for benzene and squalane obtained by mass-statting and barostatting at $T = 300$ K. The error was calculated for the final accumulated mean value of the pressure and density computed inside the CSR. Refer to Figure 5.21.c and Figure 5.21.d for mass-statting and Figure 5.20.c and Figure 5.20.d for barostatting.

A clear advantage of barostatting over the mass-statting implementation presented in this section, is the shorter time it takes to drive the system to the final target thermodynamic point. From Figure 5.20, it can be seen it takes less than 2 ns to equilibrate the system int the worst-case scenario (big molecule and low pressure). Mass-tatting, on the other hand,

employs up to 1 ns per iteration. However, as previously stated, our concern is to lower $error_\rho$. A single equilibrated system is used to perform multiple coupled NEMD simulations making the time spent in equilibration negligible w.r.t the coupled simulation time. Therefore, coupled NEMD simulations carried in Section 5.4.3 have been all equilibrated using the mass-statting strategy.

5.4.3 NEMD Coupled Simulations

In Section 5.3.5 we showed results from shear-driven MD-CFD coupled simulations employing a mono-atomic Lennard-Jones fluid model. In this section we go a step further and successful coupled simulations of more complex fluid models are presented: OPLS benzene, and L-OPLS squalane with force-field parameters listed in Table 5.3.

Case	P (MPa)	**Molecule**	**CFD solver**	u_x^{Top} (m/s)	ε_{sf} (Kcal)	Δt_{nemd} (ns)	**Slip**
1	100	benzene	Newtonian	100	$\sqrt{0.5\varepsilon_f^j}$	2.0	no
2	1000	benzene	non-Newtonian	50	$\sqrt{4\varepsilon_f^j}$	0.7	no
3	1000	benzene	non-Newtonian	50	$\sqrt{0.5\varepsilon_f^j}$	0.5	yes
4	100	squalane	non-Newtonian	50	$\sqrt{4\varepsilon_f^j}$	1.2	no
5	100	squalane	non-Newtonian	50	$\sqrt{0.5\varepsilon_f^j}$	1.2	yes

Table 5.15: List of coupled NEMD simulation cases for benzene and squalane. The expression for the LJ interaction energy ε_{sf} between surface and liquid atoms are given, where ε_f^j correspond to the atom type j of a molecule in the liquid (see Table 5.3 for a list of values). The slip column indicates the appearance of velocity slip in the simulation and u_x^{Top} is the top velocity boundary condition applied.

At the time of writing this thesis, there are, to the extend of the author's knowledge, very few scientific publications reporting results on coupled MD-CFD simulations with poly-atomic molecular fluids [123,231]. Nevertheless, none of those explore the high pressure regime (hundreds of

MPa) which are common in tribological applications. As a common rule for fluids, the higher the pressure, the higher the viscosity of the fluid and internal shear stresses [18]. We have found this regime to favor the appearance of a deleterious effect in the coupling strategy due to the amplification of noise in the overlap region, which is likely to happen as well in any scheme based on an HSI [100]. This effect has not been reported before and is explained in Section 5.5 with detail. Additionally, in this section the non-Newtonian regime is explored in coupling, and molecular fluids which exhibit shear-thinning are modeled in the CFD by using Equation 3.3.3 and Equation 3.3.4.

Five MD-CFD coupling Couette-like simulations (see Table 5.15), driven by a constant velocity BC u_x^{Top} in the CFD are presented here. Simulation conditions were quasi-isothermal and high-pressure (100 MPa and 1000 MPa). The coupled simulations start from a system equilibrated using mass-statting as described in Section 5.4.2. The initial density in the middle of the channel can be found in Table 5.14 for both molecules. The surface-fluid interaction parameter ε_{sf} (following Lorentz-Berthelot rule in Equation 2.3.9) was selected to produce either velocity slip or no-slip at the atomistic surface. The NEMD simulation time Δt_{nemd} (time over which the system is sheared) was selected large enough for the simulations to achieve steady-state. Setup number 4 from Table 5.8 was used in every case for the thermostatting and density correction strategy. The momentum equation for the Newtonian (case 1) and non-Newtonian cases (cases 2-5) was solved in the CFD domain as described in Table 3.1. Common boundary conditions values for the five simulation cases are $u_x^{surf} = 0$, $T_{surf} = T_{Top} = 300$ K and $P_{Left} = P_{Right} = 0$, which are labeled as Top, $Right$ and $Left$ according to the boundary planes in the CFD (see Figure 3.2, Table 3.3 and 3.5). The time-step ratio employed in all the cases was $TSR = 100$ and the MD and CFD time steps sizes were, respectively $\Delta t_{cfd} = 100$ fs and $\Delta t_{md} = 1$ fs. The physical quantities reported were averaged over a time period $\Delta t_{avg} = \Delta t_{cfd}$ as described in Section 5.3.3, with a sampling period of $\Delta t_s = 10$ fs. The coupling cell sizes d_y^c has been selected to be significantly bigger than the

radius of gyration of the molecule ($R_g \approx 0.6$ nm for squalane [232] and $R_g \approx 0.1$ nm for benzene).

CFD and Coupling Mesh							
Case	d_y^c	N_{cx}	N_{cy}	N_{cz}	N_{cy}^{olap}	N_{cy}^{CNST}	N_{cy}^{BC}
1	1	1	23	1	3	1	1
2	2	1	52	1	2	1	1
3	2	1	52	1	2	1	1
4	2	1	25	1	3	1	1
5	2	1	25	1	3	1	1
Geometry (in nm)							
Case	L_{ch}	L_{cfd}	L_{md}	L_{olap}	$A_{xz}(\text{nm}^2)$	L_{surf}	L
1	16.9672	23	19.8272	3	18.05^2	2.86	36.9672
2	6.29822	104	9.15822	4	18.05^2	2.86	106.29822
3	6.29822	104	9.15822	4	18.05^2	2.86	106.29822
4	18.5561	50	21.2861	6	18.05^2	2.73	62.5561
5	18.5561	50	21.2861	6	18.05^2	2.73	62.5561

Table 5.16: CFD and coupling mesh and geometry parameters employed in for coupled simulation cases in Table 5.15. The N_{cx}, N_{cy}, and N_{cz} symbols are the number of cells of the CFD mesh in each direction and , N_{cy}^{CNST}, N_{cy}^{BC} and N_{cy}^{olap} are the number of cells along the y direction in the CNST, BC and overlap regions. The mesh cell-size for the coupling mesh is d_y^c and the area of the MD and CFD domain perpendicular to the y direction is A_{xz}. The L_{md}, L_{cfd}, L_{ch}, L_{olap} and L_{surf} lengths are defined along the y direction as depicted in Figure 3.1. The effective length occupied by the fluid is $L = L_{ch} + L_{cfd} - L_{olap}$.

The atomistic lower surface employed was a 18.05×18.05 nm^2 Cu wall with multiple harmonic interactions as described in Equation 3.2.3 and parameters from Table 5.12. The rest of case-specific geometrical and mesh parameters are listed in Table 5.16 and a $25 \times 1 \times 24$ processor partitioning was used for the MD domain, whether just a single processor was used for

the CFD. The shear stress has not been plotted for any of the studies, since it is unfeasible to be computed during transient (highly noisy). However, the velocity field is sufficient alone to ascertain if the coupled solution is correct in a FV scheme, which will consistently evolve in time if the stress in the CNST region is imposed correctly. This is an advantage of coupling by fluxes in the CNST region as opposed to constraining the velocity field. Although the velocity field can be forced, by a constraining algorithm, to be consistent between MD and CFD domains, the stress field might not be (e.g due to a mismatch between the viscosity in the MD and CFD [233]).

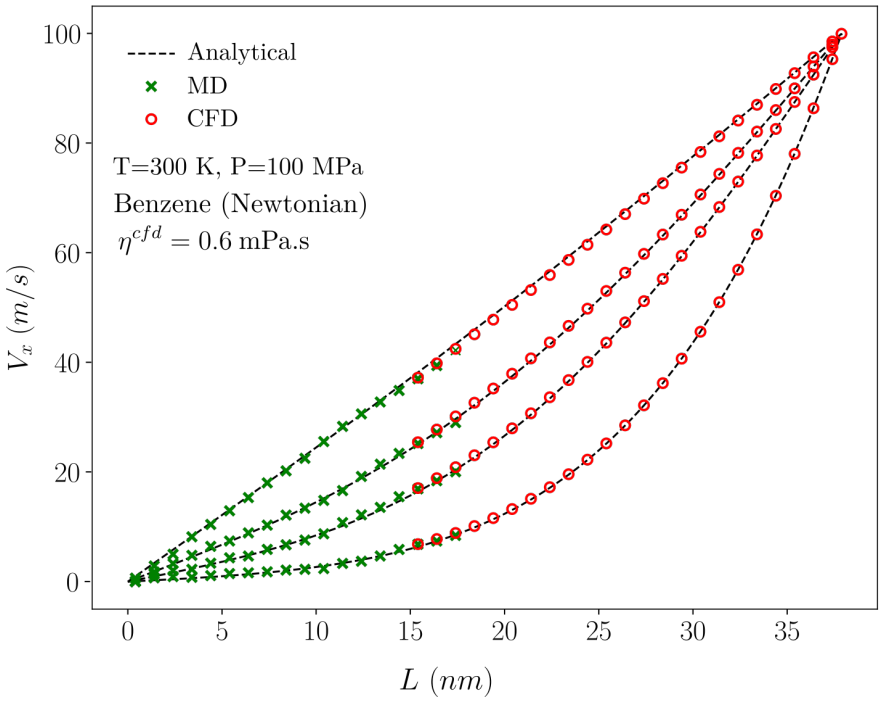
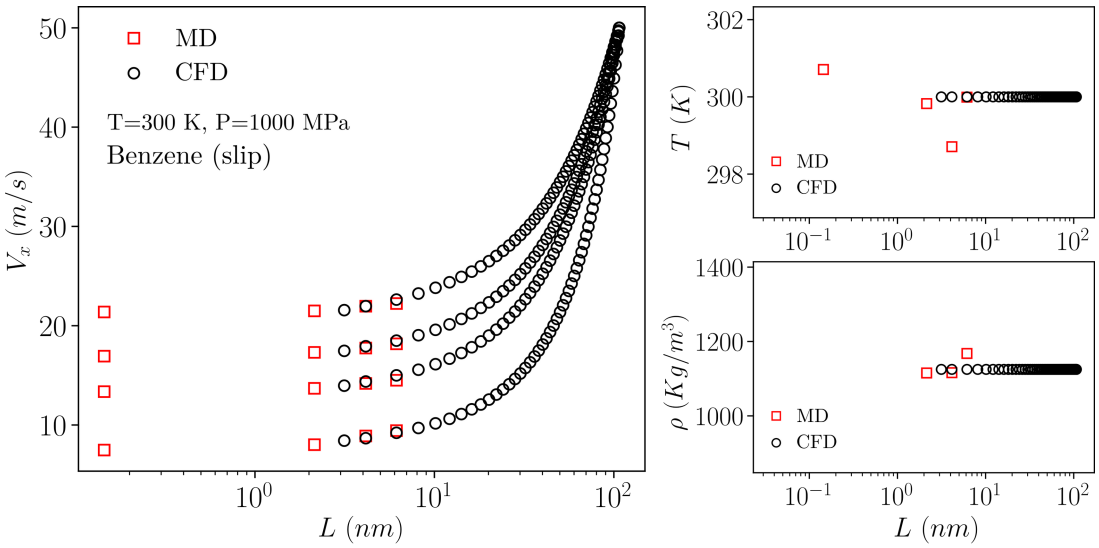
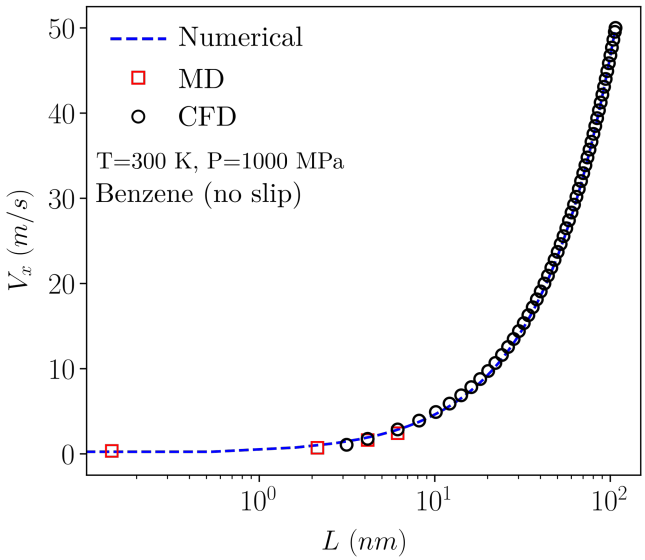


Figure 5.23: Time-evolution of velocity field for a CFD-MD coupled simulation of 40691 molecules of OPLS benzene. Initial configuration equilibrated at $T = 300$ K and $P = 100$ MPa with a density $\rho = 0.89$ kg/m^3 . The NEMD viscosity (see Section 5.2.2) of benzene measured at $\dot{\gamma} = 10^{10}$ s^{-1} at the given conditions is $\eta = 0.6$ mPa.s. Shearing velocity at the top wall $v = 100$ m/s with a domain length $L = 40$ nm. Fluid–surface interactions are strong enough to avoid slip. A shear rate of $\dot{\gamma} = 2.5 \times 10^{10}$ s^{-1} is reached at the steady-state with a linear velocity profile. An averaging period of $\Delta t_R = 0.1$ ns and $N_R = 2$ independent instances of the field has been used to compute the mean flow value following Equation 5.3.5.



(a)



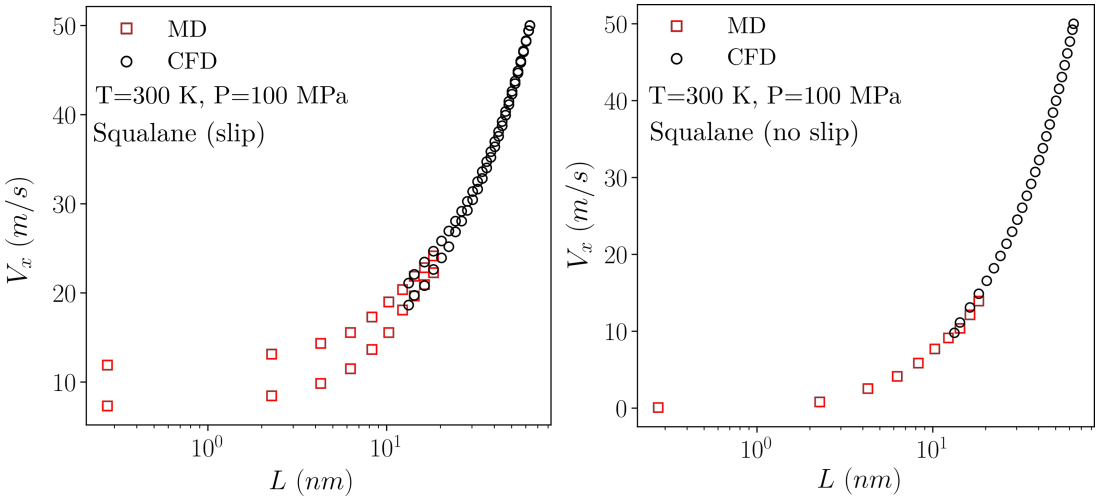
(b)

Figure 5.24: Results for two parallel shear flows (case 2 and 3 in Table 5.15) obtained using CFD-MD coupling for benzene at $T = 300$ K and $P = 1000$ MPa. Black circles are the time-averaged CFD solution and red squares the time-averaged MD solution computed by using Equation 5.3.5 with $\Delta t_R = 0.05$ ns and $N_R = 2$ independent instances of the flow. (a) Time-evolution of the velocity field and steady-state density and temperature profiles with presence of slip near the atomistic surface and (b) steady-state velocity profile with no slip close to the atomistic surface.

The first attempt to couple a molecular fluid was conducted for benzene at $T = 300$ K and $P = 100$ MPa (case 1). At this thermodynamic point benzene behaves as a Newtonian fluid with viscosity $\eta = 0.6$ mPa.s and no-slip was enforced through a high ε_{sf} interaction parameter between benzene molecules and the surface. It can be checked that the Couette quasi-isothermal condition (Equation 2.4.25) is fulfilled for the simulation setup with $T_w = 300$ K, $U_0 = 100$ m/s $\lambda \approx 0.15$ WK^{-1}m^{-1} [214], obtaining $0.017 \ll 1$. The velocity profile solution is plotted in Figure 5.23. A fantastic match is observed throughout the transient compared with Equation 2.4.21.

Simulation cases 2 and 3 from Table 5.15, are the first two simulation cases to exploit the benefits of coupling by using benzene at $T = 300$ K and $P = 1000$ MPa. The CFD domain is roughly ten times bigger than the MD domain with the full coupled domain reaching the 0.1 μm length-scale. The molecule posses mild non-Newtonian behaviour at that thermodynamic point and a non-Newtonian solver has been used in the CFD domain with a shear-rate dependent viscosity model (Equation 5.2.5) and fitting parameters from Table 5.5. The Couette quasi-isothermal condition cannot be strictly applied due to the viscosity dependence on shear rate. However, it can give an upper bound by taking the maximum viscosity value over the range of shear rates $\eta_0 \approx 0.05$ Pa.s, which using Equation 2.4.25 gives $0.26 \ll 1$, for $T_w = 300$ K, $U_0 = 50$ m/s and $\lambda \approx 0.2$ WK^{-1}m^{-1} [214]. This is confirmed by looking at the computed coupled temperature profile in Figure 5.24.a, which is practically constant across the channel. The resulting coupled steady-state velocity profile (no slip) in Figure 5.24.b is plotted against the CFD numerical solution which is obtained solving Equation 3.3.4 in the full domain. A logarithmic scale has been used for the horizontal axis for better visualization, due to the disparity on sizes between both domains. A good agreement is found in the steady-state. The transient has not been plotted due to the insufficient sampling to get a smooth enough solution. Conversely, Figure 5.24.b presents the results of the first simulation with slip at the surface. The time-dependent slip evolution is captured by the coupling algorithm, which seems to react correctly to the change of velocity near the

atomistic surface and keep the velocity profile in the CFD and MD domains consistent within the overlap region.



(a) (b)

Figure 5.25: Results for two parallel shear flows (case 4 and 5 in Table 5.15) obtained using CFD-MD coupling for squalane at $T = 300$ K and $P = 100$ MPa. Black circles are the time-averaged CFD solution and red squares the time-averaged MD solution computed by using Equation 5.3.5 with $\Delta t_R = 0.2$ ns and $N_R = 2$ independent instances of the flow. (a) Time-evolution of the velocity field (case 5) with presence of slip near the atomistic surface and (b) steady-state velocity profile with no slip close to the atomistic surface (case 4).

Two equivalent simulations to cases 2 and 3 were performed for squalane, a more complex molecule with branches and folding configurations, at $T = 300$ K and $P = 100$ MPa. Squalane exhibit a greater viscosity than benzene and non-Newtonian effects are more acute (see Figure 5.10). The length of the overlap region $L_{olap} = 6$ nm, coupling cell size $d_y^c = 2$ nm and the $TSR = 100$ was found suitable to diminish the noise amplification effect described in Section 5.5 which can become significant due to the high viscosity of the squalane at the simulated thermodynamic point. The quasi-isothermal condition is respected using Equation 2.4.25 (as an upper boundary) which gives $0.94 \ll 1$ for $\eta_0 \approx 0.18$ Pa.s, $T_w = 300$ K, $U_0 = 50$ m/s and $\lambda \approx 0.2$ WK^{-1}m^{-1} [217]. Again, good agreement is found

between MD and CFD velocity profiles within the overlap region for case 4 (no slip, Figure 5.25.b) and case 5 (with slip, Figure 5.25.a).

5.5 Overlap Region Dynamics

Let us look more closely at how the overlap region operates in a two-way coupling scheme (see Figure 5.26 for a schematic). Assume for simplicity that the MD-CFD coupled system is at rest at an initial time $t = 0$ with zero velocity boundary conditions applied. In general, the instantaneous value of the velocity $V_x^{BC}(t)$ computed within the BC region in the MD and the value of the shear stress $S_{xy}^{CNST}(t)$ computed within the CNST region in the CFD can be written as

$$\begin{aligned} V_x^{BC}(t) &= \bar{V}_x^{BC}(t) + \delta V_x^{BC}(t), \\ S_{xy}^{CNST}(t) &= \tau_{xy}(t), \end{aligned} \tag{5.5.1}$$

where $\bar{V}_x^{BC}(t)$ is the mean velocity value, $\delta V_x^{BC}(t)$ is the velocity fluctuating component (due to thermal noise) and $\tau_{xy} = \boldsymbol{\tau} \cdot \boldsymbol{n}^{CNST}$ is the CFD shear stress from which the distributed force per atom (Equation 3.4.17) is computed to satisfy momentum coupling in the CNST region. The instantaneous value of the velocity $V_x^{BC}(t)$ can be computed from Equation 3.4.9. In order to exercise coupling at discrete times $t = t_N \equiv N\Delta t_{cfd}$, where $\Delta t_{cfd} = TSR \times \Delta t_{md}$ and $N \in \{1, 2, ...\}$ is the time-step number, both $V_x^{BC}(t_N)$ and $S_{xy}^{CNST}(t_N)$ are computed as

$$\begin{aligned} V_x^{BC}(t_N) &\equiv \frac{1}{\Delta t_{cfd}} \int_{t_N - \Delta t_{cfd}}^{t_N} \left[\bar{V}_x^{BC}(t) + \delta V_x^{BC}(t)\right] dt, \\ S_{xy}^{CNST}(t_N) &\equiv \tau_{xy}(t_N), \end{aligned} \tag{5.5.2}$$

where the velocity computed is an average over Δt_{cfd} in the MD domain. Over the first time period $t \in [0, \Delta t_{cfd}]$, since the fluid is at rest, $S_{xy}^{CNST}(t) = 0$ and $V_x^{BC}(t) = \delta V_x^{BC}(t)$ since $\bar{V}_x^{BC}(t) = 0$. As V_x^{BC} is applied as the CFD bottom boundary condition, momentum will diffuse a length L_O^* (Figure 5.26) in the BC→CNST direction over a time-scale τ_{olap}^{cfd} before

reaching the CNST region. Conversely, S_{xy}^{CNST} will be applied at the top of the MD domain producing a variation of momentum inside the CNST region. The momentum will diffuse in the CNST→BC direction over a τ_{olap}^{md} time-scale before reaching the BC region. This process can be regarded as a close loop where V_x^{BC} is affected by S_{xy}^{CNST} and vice versa (Figure 5.29), which is characteristic of a feedback system [234]. We can then express these two quantities as a function of each other as

$$\begin{aligned} S_{xy}^{CNST}(t_N) &= \mathcal{T}_{BC\rightarrow CNST}\left(\Omega^{BC}(N)\right) , \\ V_x^{BC}(t_N) &= \mathcal{T}_{CNST\rightarrow BC}\left(\Omega^{CNST}(N)\right) , \end{aligned} \tag{5.5.3}$$

where $\Omega^{BC}(N)$ and $\Omega^{CNST}(N)$ are, respectively, a set of $V_x^{BC}(t_k)$ and $S_{xy}^{CNST}(t_k)$ values computed at time-steps $k \in \{1, ..., N\}$. $\mathcal{T}_{BC\rightarrow CNST}$ and $\mathcal{T}_{CNST\rightarrow BC}$ are non-trivial functions that map $\Omega^{BC}(N)$ to $S_{xy}^{CNST}(t_N)$ and $\Omega^{CNST}(N)$ to $V_x^{BC}(t_N)$.

Both $\Omega^{BC}(N)$ and $\Omega^{CNST}(N)$ can be constructed given the diffusion time-scales in the overlap region for the CFD domain τ_{olap}^{cfd} and the MD domain τ_{olap}^{md} as

$$\begin{aligned} &\Omega^{BC}(N) = \\ &\quad \left\{ V_x^{BC}(t_k = k\Delta t_{cfd}) \; \forall k \in 1, ..., N \mid (N-k)\Delta t_{cfd} \geqslant \tau_{olap}^{cfd} \right\} , \\ &\Omega^{CNST}(N) = \\ &\quad \left\{ S_{xy}^{CNST}(t_k = k\Delta t_{cfd}) \; \forall k \in 1, ..., N \mid (N-k)\Delta t_{cfd} \geqslant \tau_{olap}^{md} \right\} . \end{aligned} \tag{5.5.4}$$

Equation 5.3.3 shows that $V_x^{BC}(t_k)$ and $S_{xy}^{CNST}(t_k)$ are only selected if the time between step k and N is longer than the diffusion time-scale for each respective domain. Since $\Delta t_{cfd} = TSR \times \Delta t_{md}$, $\tau_{olap}^{md} \approx L_O^{*2}/\nu^{md}$ and $\tau_{olap}^{cfd} \approx L_O^{*2}/\nu^{cfd}$, the fundamental simulation quantities that determine the behaviour of the overlap region are ν^{cfd}, ν^{md} , TSR and L_O^*. This process is graphically described in Figure 5.26.

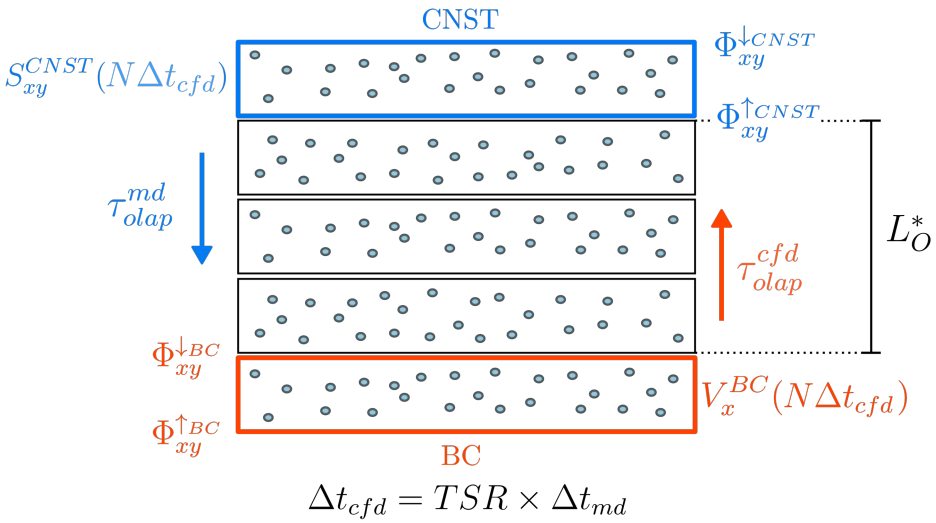


Figure 5.26: Schematic of momentum transport in the overlap region. Both V_x^{BC} and S_{xy}^{CNST} are computed every $N\Delta t^N = TSR \times \Delta t_{md}$ from an initial time t_0. BC and CNST regions are separated by a length L_O^*. The viscosity of the fluid (could be different for the MD and CFD domain by construction). In the CFD domain, momentum diffuses from BC to CNST over a time scale τ_{olap}^{md}. Likewise, in the MD domain momentum diffuses from CNST to BC over a time-scale τ_{olap}^{cfd}. Symbols $\Phi^{\downarrow}$ and $\Phi^{\uparrow}$ are, respectively, the momentum flux coming from the upper and lower cell.

The mutual interdependence between V_x^{BC} and S_{xy}^{CNST} exhibit properties of a positive feedback system [234]. Under certain conditions, it was found that the perturbation introduced to the CFD through the boundary condition imposed by the MD at $t = \Delta t_{cfd}$ (first time at which the boundary condition from the MD is applied), would grow over time until V_x^{BC} and S_{xy}^{CNST} stabilize to a nearly single-frequency oscillatory steady-state with constant phase and amplitude. This is shown in e.g Figure 5.27 and discussed more in detail below. This behaviour was observed for both benzene and squalane molecular fluids, under certain conditions, but never in LJ fluids. The reason seems to lay in the higher viscosity inherent to non-monoatomic fluids, which means faster momentum diffusion and higher shear stresses in the fluid. This effect is examined next through a set of parametric studies for fluid benzene.

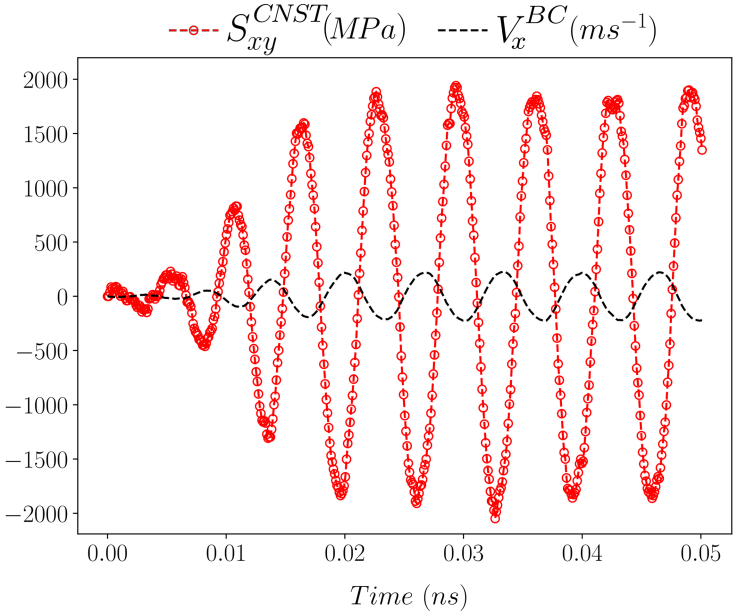


Figure 5.27: Positive feedback effect for V_x^{BC} and S_{xy}^{CNST}. A small perturbation on the CFD velocity boundary condition (due to thermal noise within the MD), grows over time until a oscillatory steady-state is reached. Amplitude, frequency and phase of this state are constant, with an observed phase difference $\Delta\phi$ between velocity and stress in the range $\frac{\pi}{2} < \Delta\phi < \pi$ for the range of simulations listed in Table 5.17.

A series of parametric simulations were carried out for liquid benzene to study the positive feedback effect in the overlap region. The values of ν^{cfd}, ν^{md} (varied by changing pressure in the MD), TSR and L_O^* employed are listed in Table 5.17. An isothermal Newtonian CFD solver was used to allow fixing ν^{cfd} to a single value. The value for ν^{md} was taken as the Newtonian viscosity (zero shear rate) of the fluid at pressures $P = 100$ MPa and $P = 1000$ MPa. However, significant shear thinning behaviour of molecular benzene is observed at $P = 1000$ MPa (see Figure 5.9), so a single viscosity value is not as meaningful as for the $P = 100$ MPa case, in which non-Newtonian effects are negligible. The simulation setup was the same as the one used for coupled simulations in Section 5.4.3, but with boundary conditions $V_{x,wall}^{md} = V_x^{cfd}(y = L_{cfd}) = 0$ (no-flow conditions). The total length of the coupled CFD-MD domain $L_{cfd} + L_{md} = 100$ nm was kept constant for different values of L_O^* studied. All simulations ran for a total of

0.5 ns and the number of samples to compute $V_x^{BC}(t_N)$ were $N_s = 10$, regardless of the TSR value.

From the values of $\nu^{cfd} \in \{9e^{-7}, 9e^{-6}, 9e^{-5}\}$ m^2/s studied, only $9e^{-5}$ seems to trigger positive feedback (Figure 5.28), and it does it for any of the combinations of the parameter's values studied (cases 1 to 8 in Table 5.17). For $\nu^{cfd} = 9e^{-6}$ and $9e^{-7}$ m^2/s (cases 9 to 12 in Table 5.17), there seems to be no evidence of the effect by visual inspection of Figure 5.29.a. Cases 9 to 12, exhibit velocities with a regular noisy behaviour (amplitude of noise is similar across all the four cases) and no evident amplitude growth. However, a deeper analysis reveals that the apparent lack of positive feedback in these cases is just related to a poor signal to noise ratio. By comparing the spectra of cases 1 to 8 (Figure 5.30) with the ones of cases 9 to 12 (Figure 5.31), we see the same well-defined peaks in the frequency region of 0 to 200 ns^{-1}, which indicates the presence of the feedback effect. Since in cases 9 to 12 the noise amplitude is of the same order of magnitude as the velocity modes introduced by the positive feedback, we can disregard its presence.

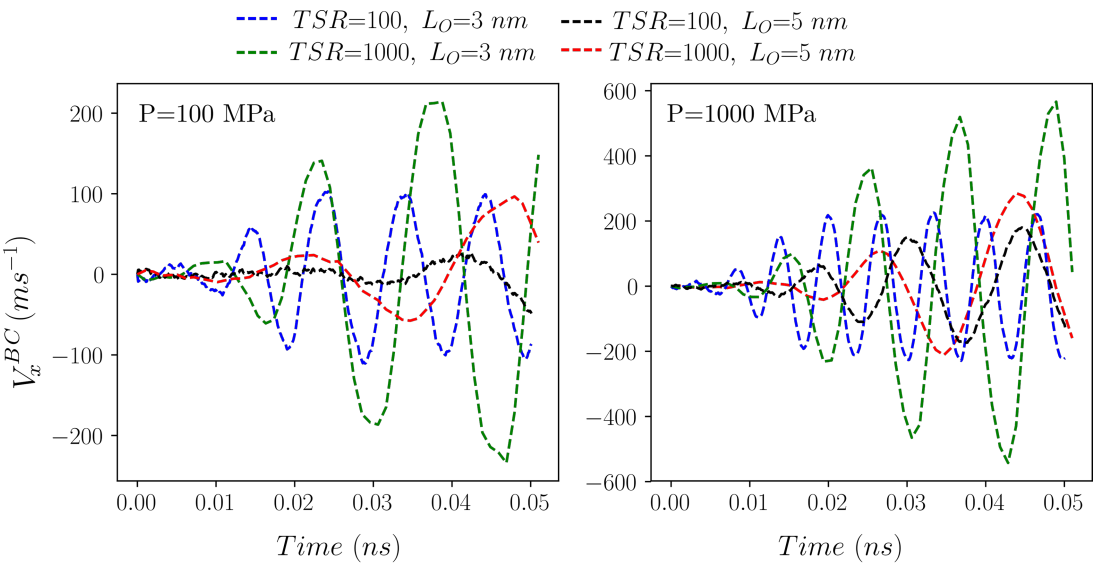


Figure 5.28: Velocity in the BC region V_x^{BC} being amplified by a positive feedback effect for (a) $P = 100$ MPa ($\nu_{md} = 6.5e^{-7}$ m^2/s) and (b) $P = 1000$ MPa ($\nu_{md} = 3e^{-5}$ m^2/s), at different simulation conditions. Plots correspond to cases 1 to 8 listed in Table 5.17 with $\nu_{cfd} = 9e^{-5}$ m^2/s.

Case	P (MPa)	TSR (Δt_{cfd}) (Δt_{md}=1fs)	L_O (L_O^*) (nm)	ν_{md} (m^2/s)	τ_{olap}^{md} (ns)	τ_{olap}^{cfd} (ns)	τ_{osc} (ns)	$\frac{\tau_{osc}(L_O^*=3)}{\tau_{osc}(L_O^*=1)}$
$\nu_{cfd} = 9e^{-5}$ m^2/s								
1	100	100 (1e-4)	3 (1)	6.5e-7	1.5e-3	1e-5	0.010	2.0
2	100	100 (1e-4)	5 (3)	6.5e-7	1.5e-2	1e-4	0.020	
3	100	1000 (1e-3)	3 (1)	6.5e-7	1.5e-3	1e-5	0.014	1.71
4	100	1000 (1e-3)	5 (3)	6.5e-7	1.5e-2	1e-4	0.024	
5	1000	100 (1e-4)	3 (1)	3e-5	3e-5	1e-5	0.007	1.86
6	1000	100 (1e-4)	5 (3)	3e-5	2.5e-4	1e-4	0.013	
7	1000	1000 (1e-3)	3 (1)	3e-5	3e-5	1e-5	0.010	1.7
8	1000	1000 (1e-3)	5 (3)	3e-5	2.5e-4	1e-4	0.017	
$\nu_{cfd} = 9e^{-6}$ m^2/s								
9	100	100 (1e-4)	3 (1)	6.5e-7	1.5e-3	1e-4	NA	NA
10	1000	100 (1e-4)	3 (1)	3e-5	3e-5	1e-4	NA	NA
$\nu_{cfd} = 9e^{-7}$ m^2/s								
11	100	100 (1e-4)	3 (1)	6.5e-7	1.5e-3	1e-3	NA	NA
12	1000	100 (1e-4)	3 (1)	3e-5	3e-5	1e-3	NA	NA

Table 5.17: Parametric simulations to study positive feedback in the overlap region for benzene using a Newtonian CFD solver. P is the pressure, TSR is the MD-CFD time-step ratio defined as $TSR = \Delta t_{md}/\Delta t_{cfd}$, L_O is the overlap region length, L_O^* is the distance between the BC and CNST region as given in Figure 5.26, ν_{md} and ν_{cfd} are the Newtonian MD and CFD kinematic viscosity, τ_{olap}^{md} and τ_{olap}^{cfd} are the momentum diffusion time-scales in the overlap region for the MD and CFD domains, τ_{osc} is the oscillation period for V_x^{BC} computed from the spectrum in Figure 5.30.a and 5.30b. The last column is the ratio between the oscillation period τ_{osc} for $L_O = 3$ and $L_O = 5$. τ_{osc} = NA, are for cases which did not exhibit oscillatory behaviour in the time domain.

If the the positive feedback effect is strong enough, it can hamper the coupling methodology causing coupled simulations to become unstable or a nonphysical raise of the kinetic temperature. It is therefore of great interest

to know what is the impact of ν^{cfd}, ν^{md} , TSR and L_O^* on the amplitude and period of the oscillations caused by positive feedback so countermeasures can be applied to mitigate it.

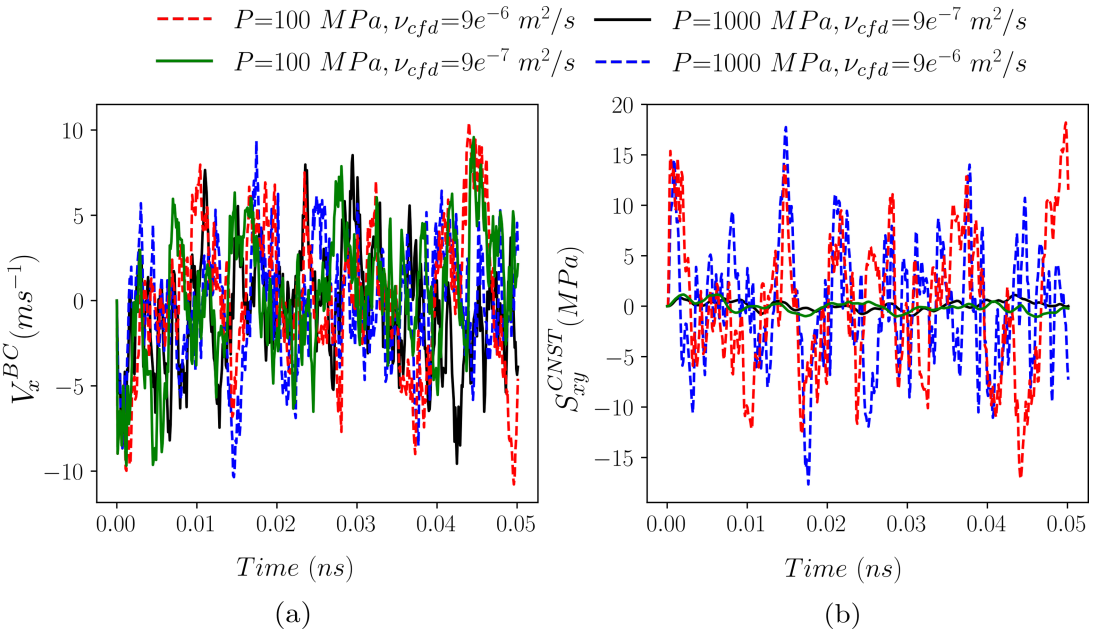


Figure 5.29: Velocity at the BC region V_x^{BC} and stress at the CNST region S_{xy}^{CNST} without any apparent positive feedback effect for different ν_{cfd} and P values (cases 9 to 12 in Table 5.17). The magnitude of V_x^{BC} is on the same scale in every case.

A number of observations can be made by analyzing Figure 5.28 and Table 5.17:

1. Oscillation periods τ_{osc} are greater than diffusion times τ_{olap}^{md} and τ_{olap}^{cfd}, which means diffusion of momentum is involved in between oscillation cycles.
2. The larger the intrinsic viscosity of the molecular fluid ν^{md}, (i) the greater the amplitude and (ii) the shorter the period of oscillations τ_{osc}. This can be understood because (i) the larger the viscosity, the larger the stress "kick" (momentum flux) applied in the CNST and (ii) momentum diffuses faster, so the feedback effect causing oscillations in the velocity field takes place faster too.

3. The larger the TSR, the greater is the amplitude developed but the oscillation period τ_{osc} does not seems to be affected significantly.
4. The bigger the overlap region L_O^*, the larger the period of oscillations τ_{osc}. The effect on the amplitude is not conclusive from the data. Fixed the rest of parameters, the ratio $\frac{\tau_{osc}(L_O^*=3)}{\tau_{osc}(L_O^*=1)} \approx 2$ stays fairly constant, which means L_O^* is the key parameter regarding the oscillation period.

These observations are based purely on the data observed in Figure 5.28, which it would have been more meaningful had all the cases reached the steady-state amplitude. This was not possible since there were no more resources available to re-run the parametric study to obtain data at larger times.

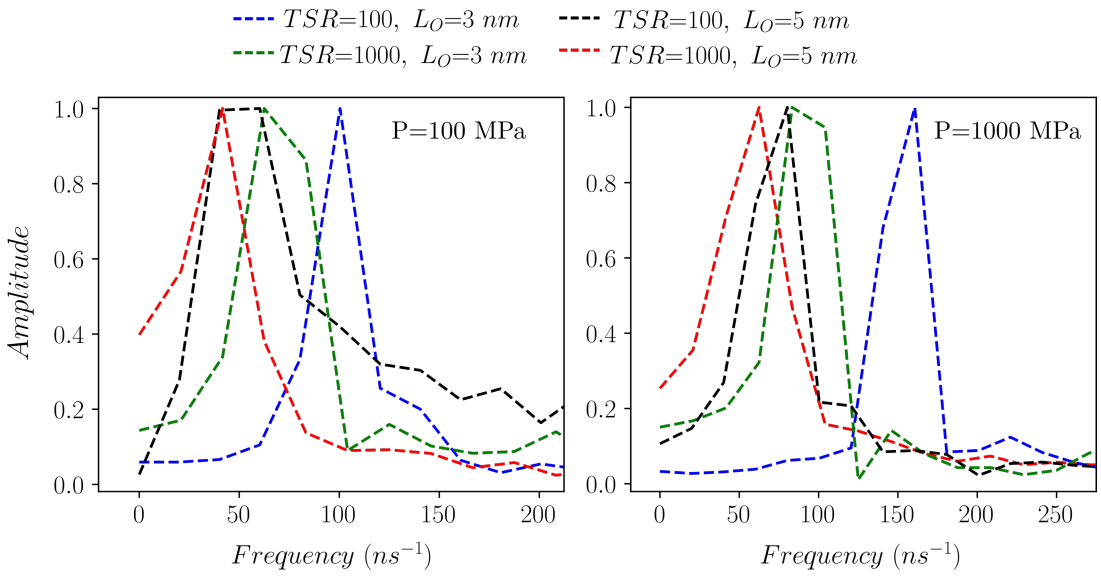


Figure 5.30: Low frequency region of the normalized spectrum of V_x^{BC} at different simulation conditions for (a) $P = 100$ MPa and (b) $P = 1000$ MPa. It can be seen that the power of the signal is mostly concentrated in the 0 to 200 ns^{-1} range of frequencies with well-differentiated peaks at discrete frequencies. Plots correspond to cases 1 to 8 listed in Table 5.17.

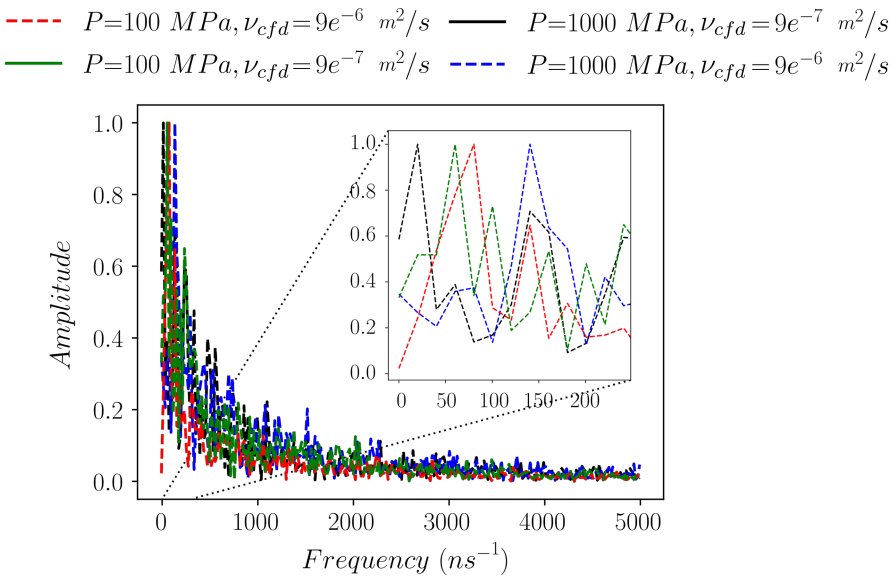


Figure 5.31: Normalized spectrum of the $V_x^{BC}(t)$ signal for cases 9 to 12 listed in Table 5.17. The inset shows the low frequency region of the spectrum where four visible peaks indicate the presence of a positive feedback mode for each case.

In addition to the parameter study presented employing a Newtonian CFD solver, two simulations were conducted using a non-Newtonian solver for benzene at $P = 1000$ MPa under no-flow conditions. The system setup was the same as the ones used in coupled simulations within Section 5.4.3. The geometrical parameters (see Figure 3.1 for a description) were $L_{md} = 10$ nm, $L_{cfd} = 130$ nm, $L_{surf} = 3$ nm and $N_{cy}^{\text{olap}} = 3$. and an overlap region of $L_{olap} = L_O = 3.0$ nm ($TSR = 100$ case) and $L_O = 6.0$ nm ($TSR = 50$ case) using a cell-size of $d_y^c = 1.0$ nm and $d_y^c = 2.0$ nm respectively. Those two cases are statistically equivalent in terms of number of samples N_s for computing V_x^{BC}, since $N_s \propto TSR \times d_y^c$. The viscosity model of the CFD solver was taken from Equation 5.2.5 for $T = 300$ K with fitting parameters from Table 5.5. By looking at Figure 5.32.a it is clear that the positive feedback effect, which is present in the the $TSR = 100$ case, has been mitigated in the $TSR = 50$ case. The spectrum for $TSR = 50$ in Figure 5.32.b shows a similar structure to the ones presented in Figure 5.30 and Figure 5.31, with the energy concentrated in the low frequency region. Contrary to cases 9 to 12, we can see that the spectrum for $TSR = 100$ in

Figure 5.32.b shows sharp peaks in the high frequency region. These are likely to correspond with normal modes of benzene, which are experimentally measured to be in the frequency range of 10^3-10^4 ns^{-1} [235]. Therefore, the major contributions to V_x^{BC} are not coming from low frequency modes, meaning the positive feedback effect has been reduced by decreasing TSR.

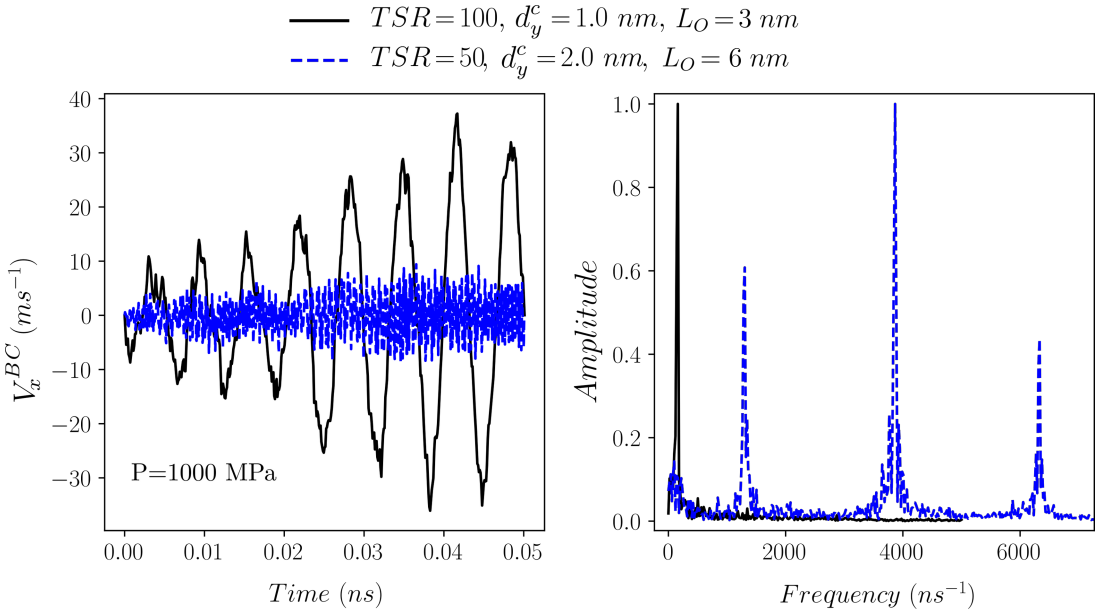


Figure 5.32: Velocity in the BC region V_x^{BC} from a coupled simulation of benzene at $P = 1000$ MPa employing a non-Newtonian CFD solver. For $TSR = 100$ and a coupling cell of $d_y^c = 1.0$ nm a positive feedback is observed, whether for $TSR = 50$ and a coupling cell of $d_y^c = 2.0$ nm the effect seems to disappear. (b) Spectrum for $TSR = 100$ case shows a narrow, high intensity peak at low frequency (< 200 ns^{-1}). For $TSR = 50$, the spectrum shows three peaks in the high frequency region (benzene's normal modes) and the large peak at low frequency has disappeared.

To summarize our findings, as general rule, by making TSR smaller and d_y^c and L_O larger, the oscillatory effect can be mitigated. However, by doing so, the performance of the coupled scheme will be unavoidably degraded. In spite of that, the most critical parameter has proven to be ν_{cfd}. It is important to note ν_{cfd} was a free parameter to generate study cases listed in Table 5.17, but in properly setup coupled simulations $\nu_{cfd} \approx \nu_{md}$, in order to

satisfy momentum conservation. There is evidence that the positive feedback effect is always present in the current coupling strategy even for low ν_{cfd} as discussed above. Figure 5.31 shows, in the frequency domain, the presence of peeks related to oscillatory modes for cases with $\nu^{cfd} \in \left\{9e^{-7}, 9e^{-6}\right\}$ m^2/s, which are not visible in the time domain in Figure 5.29. Hence, this effect will only becomes significant when the amplitude of the oscillations are greater to the thermal noise when computing $V_x^{BC}(t_N)$ using Equation 5.5.2. This will happen for a critical value of ν_{cfd}, as it occurred in cases 1-8 for $\nu_{cfd} = 9e^{-5}$ m^2/s. The thermal noise is, for practical purposes, almost exclusively a function of temperature as shown in Figure 5.6. Therefore, as the pressure increases, the noise amplitude will remain the same but the viscosity ν_{md} will increase [18], causing this effect to be more accute. To my knowledge, this effect has not been reported anywhere in the specialized literature, most likely because simulation cases tend to be at pressures where viscosities are not high enough for this effect to become relevant. Finally even though the effect can be mitigated, it is not clear that for sufficiently high viscosities, it can be overcome, making the coupling strategy presented here, and possibly other two-way coupling strategies based on an HSI [100], nonviable under such conditions.

5.6 Chapter Summary

This chapter contains the set of results which validates the coupling methodology described in Chapter 3 and the software framework presented in Chapter 4. Also, the limitations of the current implementation and the MD-CFD two-way coupling methodology in general, have been discussed based on the data obtained.

In order to perform a coupled simulation, MD liquid properties need to be known in advance (see Section 5.2). Firstly, in Section 5.2.1 different properties and transport coefficients of a Lennard-Jones fluid were computed at several thermodynamic points and compared against values found in the literature. These properties were later used in Section 5.3.5 to implement coupling. The simplicity of LJ fluids makes them a good first test case to

ensure the correctness of the coupled framework built. However, the consequence of such simplicity, is that non-Newtonian effects found are very small, with shear thinning appearing only at extremely high shear rates and having very limited impact on viscosity as seen in Figure 5.8. Therefore, this fluid model was not viable to study such effects on coupling and two more complex fluids were considered: molecular benzene and squalane, which properties are reported in Section 5.2.2. The OPLS and L-OPLS all atom potentials were used, respectively, to model the inter-atomic interactions of the two molecules. The viscosity and equation of state, over a range of temperatures and shear rates, were computed for benzene at $P = 1000$ MPa and reported in Figure 5.9 and at $P = 100$ MPa for squalane in Figure 5.10. Furthermore, a shear rate and temperature dependent viscosity fitting expression (Equation 5.2.5) was devised and used to provide accurate fitting to the viscosity data generated through NEMD simulations. Furthermore, very good agreement with experimental and simulation data in the literature was found, with surprisingly accurate extrapolation to the low shear rate regimen (NEMD data generated at $\dot{\gamma} > 10^7$ s^{-1}).

The first batch of coupled simulations was reported in Section 5.3 for a Lennard-Jones fluid. The details on building and equilibration of the molecular systems coupled to a CFD solver are given in Section 5.3.1. More interestingly, Section 5.3.4 contains the benchmark of different simulation setups corresponding to different thermostatting strategies and the application of a density correction algorithm (FOREV [104]) near the top reflective wall. It was found that the best performing setup was Number 4 from Table 5.8. In this setup, the thermostatting force is just applied over the y direction within the CNST region and no density correction algorithm is used. This was the setup used in simulations reported in Section 5.3.5 and Section 5.4.3. The numerical validation of three different Lennard-Jones coupled flows Couette (Figure 5.14), Poiseuille (Figure 5.16) and Rayleigh-Stokes (Figure 5.17) can be found in Section 5.3.5. The time-evolution of the coupled velocity and shear stress fields for each flow, were compared against analytical expressions presented in Section 2.4.2. Very good agreement

between coupled and analytical solutions and between MD and CFD solutions in the overlap region was found.

In the second batch of coupled simulations, molecular benzene and squalane were used as fluid model and results are presented in Section 5.4. Guidelines for systematically building molecular systems for `LAMMPS`, to be used in coupling, can be found in Section 5.4.1. In addition, Section 5.4.2 contains the equilibration process description for the MD domain, using barostatting and mass-statting by the incremental displacement of the bottom atomistic wall. Looking at Figure 5.20, Figure 5.21 and Figure 5.22, it can be seen that the barostatting procedure does converge faster but is less accurate than mass-statting (described in Table 5.13) on driving the bulk fluid pressure to the target pressure P_T. Hence, mass-statting has been the algorithm of choice to perform the equilibration of molecular systems used for coupling.

For the start up Couette flow described in Section 2.4.2, five different study cases listed in Table 5.15 were simulated employing liquid benzene and squalane at $T = 300$ K. Results are presented in Section 5.4.3. For benzene, a Newtonian ($P = 100$ MPa) coupled simulation (Case 1) was compared against the analytical expression for the no-slip case, with excellent agreement found between the coupled and analytical solution for the velocity field. Two non-Newtonian ($P = 1000$ MPa) simulations (Cases 2 and 3) were also conducted, respectively, for the slip and no-slip case, showing also correct coupling in Figure 5.24. These pair of simulations have been the first to exploit the length-scale advantage of the multi-scale method implemented in this work, by simulating flow channels of lengths $L_{ch} > 100$ nm for a total surface area of 18.05×18.05 nm^2, which we believe are the largest simulations of their kind to date. Similarly, two large non-Newtonian simulations ($P = 100$ MPa) were conducted for squalane with the appearance of slip (Case 5) and no-slip (Case 4). Squalane is an industrially relevant molecule, significantly more complex than benzene and with stronger non-Newtonian behaviour and higher viscosity as shown Figure 5.10. Coupled velocity profiles plotted in Figure 5.25 also show very good agreement between the MD and CFD domain in the overlap region. It is worth noticing

that coupling seems to react correctly to the time-evolution of slip observed for both benzene in Case 3 (Figure 5.24.a) and squalane in Case 5 (Figure 5.25.a).

Finally, a description of the dynamics of the overlap region for the coupling strategy implemented was given in Section 5.5. In addition, results of coupled simulations for Newtonian (benzene) and non-Newtonian (squalane) in no-flow conditions (fluid at rest), have evidenced the appearance of a positive feedback effect in the velocity field which can be potentially harmful to the coupling strategy for certain conditions. It was observed that introducing a small perturbation in the CFD velocity boundary condition computed from the MD domain (V_x^{BC} from Equation 5.5.2), the velocity and therefore the stress fields, evolve to a well-defined oscillatory steady-state which is system-dependent. This can be seen in Figure 5.27. This effect seems to be always present in any coupled simulation, but it is exacerbated with the increase of pressure, which increases the fluid's viscosity and only become harmful when the amplitude of the oscillations are sufficiently large to affect the time-averaged samples generated for V_x^{BC}. In addition, it was found that the oscillatory effect can be potentially mitigated by making TSR smaller and d_y^c and L_O larger. However, for sufficiently high viscosities, it is unclear if the current coupling methodology can overcome this effect without losing its practicality.

Chapter 6

Conclusions and Future Work

6.1 Context

Physical processes at small length and time scales, determine the macroscopic properties of a physical system. In this context, atomistic simulations have emerged as a valuable tool to study scales too small to resolve with current experimental technology. The field of computational nanotribology has bloomed in the last two decades with the aim of understanding fundamental friction and lubrication mechanisms driven by microscopic phenomena. Extensive literature can be found on the study of tribological behaviour of confined molecules between two surfaces using molecular dynamics simulations. In particular, the effect of shear on boundary and hydrodynamic lubrication regimes is an active field of research as discussed in Section 2.1. Different friction-related quantities like the friction coefficient, slip at the surface-liquid boundary and viscous dissipation can be directly measured from these type of simulations. Although atomistic

simulations are a valuable tool, their cost is prohibitive for length scales larger than a few dozens of nanometers and time-scales of the order of nanoseconds. In this work this problem has been tackled by applying a domain decomposition multiscale technique which allows the study of larger length-scales with less computational cost. A software framework implementing MD-CFD coupling has been developed by employing modern software development techniques in order to facilitate the simulation of such systems.

6.2 Overview of this Work

In this work, a multi-scale methodology based on domain decomposition, coupling MD and CFD methods, has been developed. This methodology has been implemented as a software framework allowing the study of slip and friction effects in fluids under shear (high-pressure, high-viscosity conditions). Modern software development techniques have been applied where applicable to increase its usability, extensibility and robustness.

6.2.1 Computational Aspects and Software Development

A significant amount of code was written to support the research carried out in this thesis. The bulk of the software development effort has been on building a framework (a set of libraries, scripts, automation tools, etc.) to facilitate the simulation workflow depicted in Figure 4.1. The most prominent features implemented are discussed in Chapter 4, however, no discussion about the software packages for simulation post-processing and generation of parameter studies, has been included (in the interest of keeping the chapter concise).

In order to implement coupling between the MD and CFD solvers, the `CPL_LIBRARY` has played a key role, enabling concurrent exchange of data between both domains using efficient `MPI` communication channels. Some notable benefits of using the library are: (a) Independent development and testing of each coupled solver is easier, making the upgrade of a particular

solver less involved; (b) Potential incompatibilities are avoided between two solvers which depend on different versions of a particular library; (c) Arbitrary parallel domain decomposition of each solver is allowed. Each solver can independently decompose its domain, whereas in a monolithic approach, one decomposition has to fit both solvers.

The most significant software-related work carried out during this thesis can be summarised as:

- Several contributions to the open-source `CPL_LIBRARY` project were made: (a) redesign of the library interface in order to improve its usability; (b) development of a Python interface for the library (library *bindings*), which has been leveraged to write a testing framework using the software testing strategy of *mocking*. This has been branded as the `CPL_Mock` strategy as described Smith *et al.* [8], in which a Python script "mocks" one of the coupled solvers for testing purposes. (c) Development of a suite of automated tests for the library considering multiple variations of parameters (overlap region size, number of processors, cell size, etc.).

- Development of a coupled MD solver by modifying the `LAMMPS` [35] software package and two coupled CFD solvers by adapting `OpenFOAM's` Newtonian (`icoFoam`) and non-Newtonian (`nonNewtonianIcoFoam`) solvers. In addition, a new shear-rate and temperature dependent viscosity model implementing Equation 5.2.5 was developed for `OpenFOAM`. A third coupled non-Newtonian solver for the temperature equation (see Equation 3.3.5) was also implemented. However, time constraints have not allowed any coupled simulations, which are left to future work. A testing strategy to verify the physical fields being coupled has been developed using a batch of tests written using the `CPL_Mock` functionality provided by the `CPL_LIBRARY`. In addition, a deployment system for the coupled applications was designed based on the `Conda` package manager. This allows the applications to be built as packages that are compiled once within a particular environment (set of library dependencies) and then

can be installed in compatible systems (same hardware architecture e.g Intel) reproducing the same environment they were built in. This procedure enhances both the reproducibility of simulation results and the ease of use of the software framework, which was used to run coupled simulations in multiple HPC clusters without having to build each piece of software multiple times.

- Several parallel scalability studies were carried out to verify the efficiency of the software framework. Firstly, the parallel efficiency of the standalone `LAMMPS` software was studied in an HPC cluster with no significant degradation observed with respect to the bespoke version of `LAMMPS` available in the cluster. This demonstrates, to some extent, that the build and deployment strategy based on `Conda`, does not affect, its performance when running in parallel. Additionally, the parallel efficiency of the coupled implementation of `LAMMPS` and `OpenFOAM` solvers was tested in multiple configurations. Good scalability was found overall for the set of coupling parameters employed in the full coupled simulations carried out in Sections 5.3.5 and 5.4.3. A constant overhead was observed in `LAMMPS` proportional to the number of atoms in the CNST region due to the application of the momentum flux from the CFD at every time-step. It is worth clarifying that, generally, the outcome of scalability studies will depend upon the coupling technique used, the solver implementation, the computing environment and the hardware they run on. However, through the comprehensive batch of studies ran, no red flags where found, which provides confidence about the quality of the implementation.

6.2.2 Coupling Methodology

In order to enable the study of tribologically relevant molecular fluids at larger length-scales, a concurrent two-way multiscale methodology was applied in non-equilibrium shear conditions. To achieve this, the domain decomposition method was used with a MD subdomain modelling the

surface-fluid interface and nearby region, and a CFD solver modelling the bulk of the fluid. The coupling between solvers was performed by means of an overlap region in which the hydrodynamic physical fields in both domains are enforced to be consistent through the constrained region (CNST) and boundary condition region (BC), see Figure 3.1. The MD domain was terminated direction by a pair of reflective boundaries at the top and bottom (in the y direction). This strategy was originally used for fluids by O'Connell & Thomson [100] which popularized the concept of Hybrid Solution Interface (HSI), to denote the overlapping region between domains. The coupled scheme developed uses a strategy where boundary conditions are imposed in terms of state variable averages from the MD to the CFD and fluxes from the CFD to the MD within the CNST region (FV coupling [7]). This strategy has proven to work well in light of the quality of the coupling results obtained for LJ fluids in Section 5.3.5 and molecular fluids in Section 5.4.3.

Four simulation setups where benchmarked, using a mean squared error metric as written in Equations 5.3.16 and 5.3.17. The case of study used for the benchmark was a Couette flow using the Lennard-Jones fluid model. The error metric is computed from the squared distance between a the coupled solution obtained for a physical field (e.g velocity) and the corresponding time-dependent analytical expression. The benchmark considered: (a) different thermostatting strategies found in the NEMD literature and (b) the application or not of the FOREV density correction algorithm near the top reflective wall. The FOREV algorithm, was developed by Issa & Poesio [104] to address the issue of unphysical density oscillations near a non-periodic boundary and was implemented in this work within `LAMMPS` considering parallel scalability. It was found that the most performant setup (see Table 5.8) was to apply thermostatting over the y direction (perpendicular to the flow) just within the CNST region in order to remove the excess of heat generated by the coupling distributed force and not applying the FOREV algorithm.

This setup was used to validate the coupling framework using LJ fluids, before performing coupling with more complex molecular fluids. Three different parallel flows with different boundary conditions were simulated.

The coupled solutions of a Couette, Poiseuille and a Rayleigh-Stokes flow, were plotted against their time-dependent analytical expressions to assess the correctness of the methodology implementation. Excellent agreement was found in the three cases both during transient and in the steady-state.

In this work, benzene and squalane molecules have been used to study coupling of molecular fluids. The OPLS and L-OPLS potentials were used respectively to model their interactions for their quality in reproducing thermodynamic and rheological properties [52]. Some guidance has been provided in Section 5.4.1 on how to effectively build molecular fluids in a systematic way. Furthermore two different equilibration strategies where examined: *barostatting* and *mass-statting*. It was found that the barostatting procedure converges faster but is less accurate than mass-statting on driving the bulk fluid pressure to the target pressure P_T. Since accuracy was more important than the extra cost, mass-statting was the method of equilibration employed for coupled molecular systems.

Additionally, a potential artifact was discovered in coupling strategy implemented, not reported before in the literature. It was observed that, even at rest, introducing a small perturbation (due to the thermal noise) in the CFD velocity boundary condition computed from the MD, causes the appearance of a spurious oscillatory flow in the overlap region. This effect seems to be always present in any coupled simulation, but it gets amplified with the increase of pressure, which increases the fluid's viscosity (a common scenario in tribological systems). If the amplitude of oscillations is sufficiently high, they can hinder the applicability of the coupling methodology. The oscillatory effect can be potentially mitigated by making the time-step ratio TSR smaller and the coupling cell size d_y^c and overlap region size L_O larger, which in turn will drop the efficiency of the coupled simulation. However, for sufficiently high viscosities, this might compromise the viability of the proposed method. Exploration of this aspect has been left for future work.

6.2.3 Physical Simulations

The Lennard-Jones fluid was used first to test the coupling methodology implemented and the correct operation of the software and its parallel scaling on a computing cluster. Successful simulations performed with the LJ fluid model set the groundwork for more physically sophisticated simulations using real molecules. Benzene was selected for its simplicity and small size as a first candidate and squalane as molecule which has tribological relevance for lubrication studies of base oils. Two types of physical simulations with real molecules were conducted in this thesis:

- To compute bulk physical properties for benzene and squalane. In order to run coupled simulations we need to provide the CFD with consistent physical properties matching the ones on the MD fluid. Bulk NEMD simulations were performed to compute the equation of state (EOS) and the shear-rate and temperature dependent viscosity of both molecules. The expression in Equation 5.2.5 was found to provide a good fit for both the viscosity data generated from NEMD simulations in the high shear-rate regime, and experimental data in the low shear-rate region. This expression was used as the viscosity model for the non-Newtonian CFD solver in coupled simulations.

- Simulations leveraging the coupling methodology by domain decomposition to study fluid channels under shear at large length-scales (in the context of molecular simulations). Large-scale systems containing fluid benzene and squalane at high pressures (100-1000 MPa), were sheared under the action of a sliding wall. Successful results were obtained in the non-Newtonian regimen for flow channels of $L_{ch} > 100$ nm with benzene and $L_{ch} > 50$ nm with squalane, for a total surface area of 18.05×18.05 nm^2. Cases exhibiting slip and no-slip at the atomistic surface-liquid interface for both molecules were obtained. These simulations were the first to take advantage of the domain decomposition method in order to reach

those lenght-scales and are the largest of their kind in terms of flow channel length, compared to what is reported in the literature.

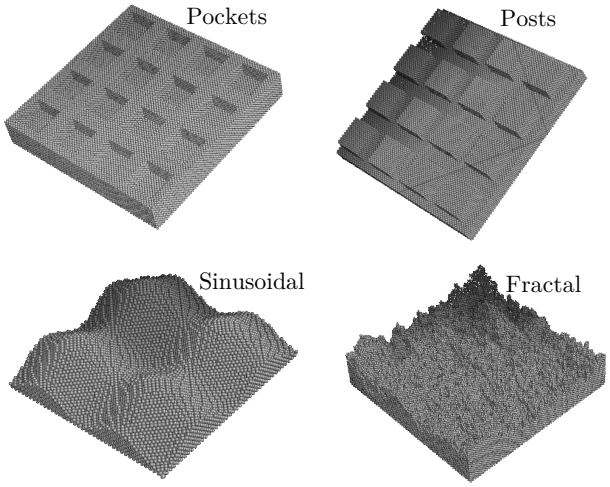


Figure 6.1: Atomistic surfaces generated with different textures. A block of a crystalline material is created and sculpted by a mask created programmatically. The surfaces are built to be used within the coupled framework for the surface potential supported.

6.3 Future Work

There are many possible extensions that can follow on from the work carried out in this thesis:

- Perform coupled simulations in non-isothermal conditions. The framework developed contains a non-Newtonian solver ready to be used but time has not allowed to design simulations to test it. Viscosities for benzene and squalane has been computed in this work for a moderate range of temperatures and fitted to an expression which is ready to be used with a CFD solver solving the thermal equation (see Equation 3.3.5).
- Study if state variable coupling algorithms, applied within constrained region (see Table 2.5), suffer from the oscillatory effect described in

Section 5.5. This will discriminate if the effect is pervasive to other two-way coupling strategies or just a particularity of the one implemented in this thesis.

- Find a strategy to circumvent the impact of the harmful oscillatory effect for high viscosity fluids. This effect can render the current coupling methodology unusable for practical implementation in tribological conditions, since at sufficiently high viscosities, the oscillations can interfere with the correct development of the physical flow.

- Use the framework developed to study different fluid-solid interfaces at length scales not possible before. In this thesis only flat surfaces have been used and slip has been induced by reducing the fluid-surface energy interactions. It would be interesting to modify the surface by introducing different types of atomistic roughness or patterning. The software framework developed includes tools for sculpting atomistic surfaces surfaces such as the one shown in Figure 6.1.

- Support the deployment of the software framework by using containers. At the time the framework was developed, the support for Docker and Singularity containers in HPC clusters was minimal. Including all `Conda` packages created in a container would provide better environment isolation and reproducibility of results.

- Comparing simulations with experimental data for systems under shear. The possibility of simulating liquid lubricant films of hundreds of nanometres to micrometres, would permit the comparison of time-evolving velocity fields obtained from numerical simulations, with the ones obtained from thin-film experiments such as fluorescence lifetime-based techniques [236,237]. Furthermore, lubricant layers of this size are technologically relevant since they can be found in lubricated engine parts [238] and in biological systems [239].

- Even though, multiple steps have been streamlined in the workflow displayed in Figure 4.1, there is still room for automation at different stages. For example, automatic selection of efficient coupling parameters to minimize the size of the system needed for a particular study and smart selection of coupling parameters (cell size, time step ratio, overlap region size, etc.).

Bibliography

[1] Paggi M. *Modeling and Simulation of Tribological Problems in Technology*. Vol. 593. 2020. 95-130 p.

[2] Allen MP, Tildesley DJ. *Computer simulation of liquids*. Second Edition. Oxford University Press; 2017.

[3] Lei Y, Leng Y. *Stick-slip friction and energy dissipation in boundary lubrication*. Phys Rev Lett. 2011;107(14):1–5.

[4] Ewen JP, Gattinoni C, Morgan N, Spikes HA, Dini D. *Nonequilibrium Molecular Dynamics Simulations of Organic Friction Modifiers Adsorbed on Iron Oxide Surfaces*. Langmuir. 2016;32(18):4450–63.

[5] Porras-Vazquez A, Martinie L, Vergne P, Fillot N. *Independence between friction and velocity distribution in fluids subjected to severe shearing and confinement*. Phys Chem Chem Phys. 2018;20(43):27280–93.

[6] Ferziger JH, Peric M. *Computational Methods for Fluid Dynamics*. Springer Berlin Heidelberg; 2001.

[7] Mohamed KM, Mohamad AA. *A review of the development of hybrid atomistic-continuum methods for dense fluids*. Microfluid Nanofluidics. 2010;8(3):283–302.

[8] Smith ER, Trevelyan DJ, Ramos-Fernandez E, Sufian A, O'Sullivan C, Dini D. *CPL library — A minimal framework for coupled particle and continuum simulation*. Comput Phys Commun. 2020;250:107068.

[9] Gropp W, Thakur R, Lusk E. *Using MPI-2: advanced features of the message passing interface*. MIT press; 1999.

[10] Todd BD, Daivis PJ. *Nonequilibrium Molecular Dynamics: Theory, Algorithms and Applications*. Cambridge University Press; 2017.

[11] Evans DJ, Morriss G. *Statistical Mechanics of Nonequilibrium Liquids*. 2nd ed. Cambridge: Cambridge University Press; 2008.

[12] Hoover WG, Ashurst WT. *Nonequilibrium Molecular Dynamics*. Theoretical Chemistry. Elsevier; 1975. p. 1–51.

[13] Bitsanis I, Magda JJ, Tirrell M, Davis HT. *Molecular dynamics of flow in micropores*. J Chem Phys. 1987;87(3):1733–50.

[14] Jabbarzadeh A, Atkinson J, Tanner R. *Effect of the wall roughness on slip and rheological properties of hexadecane in molecular dynamics simulation of Couette shear flow between two sinusoidal walls*. Phys Rev E. 2000;61(1):690–9.

[15] Priezjev NV., Troian SM. *Influence of periodic wall roughness on the slip behaviour at liquid/solid interfaces: molecular-scale simulations versus continuum predictions*. J Fluid Mech. 2006;554(1):25.

[16] Doig M, Warrens CP, Camp PJ. *Structure and Friction of Stearic Acid and Oleic Acid Films Adsorbed on Iron Oxide Surfaces in Squalane.* Langmuir. 2014;30(1):186–95.

[17] Gubbins KE, Liu Y-C, Moore JD, Palmer JC. *The role of molecular modeling in confined systems: impact and prospects.* Phys Chem Chem Phys. 2011;13(1):58–85.

[18] Jadhao V, Robbins MO. *Probing large viscosities in glass-formers with nonequilibrium simulations.* Proc Natl Acad Sci. 2017;114(30):201705978.

[19] Bair S, Mccabe C, Cummings PT. *Calculation of viscous EHL traction for squalane using molecular simulation and rheometry.* 2002;13(4):251–4.

[20] Spikes H, Jie Z. *History, origins and prediction of elastohydrodynamic friction.* Tribol Lett. 2014;56(1):1–25.

[21] Taylor RI, de Kraker BR. *Shear rates in engines and implications for lubricant design.* Proceedings of the Institution of Mechanical Engineers, Part J: Journal of Engineering Tribology. SAGE Publications; 2017;231(9): 1106–1116.

[22] Landau LD, Lifshitz EM. *Statistical Physics: Volume 5.* Elsevier Science; 2013.

[23] Beale PD. *Statistical Mechanics.* Elsevier Science; 2011.

[24] Hansen JS, Dyre JC, Daivis P, Todd BD, Bruus H. *Continuum Nanofluidics.* Langmuir. 2015;31(49):13275–89.

[25] Irving JH, Kirkwood JG. *The Statistical Mechanical Theory of Transport Processes. IV. The Equations of Hydrodynamics.* J Chem Phys. 1950;18(6):817–29.

[26] Batchelor GK. *An Introduction to Fluid Dynamics*. Cambridge University Press; 2000.

[27] De Miguel E, Jackson G. *The nature of the calculation of the pressure in molecular simulations of continuous models from volume perturbations*. J Chem Phys. 2006;125(16):1–11.

[28] Subramaniyan AK, Sun CT. *Continuum interpretation of virial stress in molecular simulations*. Int J Solids Struct. 2008;45(14–15):4340–6.

[29] Kubo R. *Statistical-mechanical theory of irreversible processes. I. general theory and simple applications to magnetic and conduction problems*. J Phys Soc Japan. 1957;12(6):570–86.

[30] Surblys D, Matsubara H, Kikugawa G, Ohara T. *Application of atomic stress to compute heat flux via molecular dynamics for systems with many-body interactions*. Phys Rev E. 2019;99(5):51301.

[31] Alder BJ, Wainwright TE. *Studies in Molecular Dynamics. I. General Method*. J Chem Phys. 1959 Aug 1;31(2):459–66.

[32] Rahman A. *Correlations in the Motion of Atoms in Liquid Argon*. Phys Rev. 1964 Oct 19;136(2A):A405–11.

[33] Travis KP, Daivis PJ, Evans DJ. *Computer simulation algorithms for molecules undergoing planar Couette flow: A nonequilibrium molecular dynamics study*. J Chem Phys. 1995;103(3):1109–18.

[34] Goldstein H, Poole CP, Safko JL. *Classical Mechanics*. Addison Wesley; 2002.

[35] Plimpton S. *Fast Parallel Algorithms for Short – Range Molecular Dynamics*. J Comput Phys. 1995;117(June 1994):1–19.

[36] Liu H, Silva CM, Macedo EA. *Unified approach to the self-diffusion coefficients of dense fluids over wide ranges of temperature and*

pressure - Hard-sphere, square-well, Lennard-Jones and real substances. Chem Eng Sci. 1998;53(13):2403–22.

[37] Saager B, Fischer J. *Predictive power of effective intermolecular pair potentials: MD simulation results for methane up to 1000 MPa.* Fluid Phase Equilib. 1990;57(1–2):35–46.

[38] Zabaloy MS, Machado JMV, Macedo EA. *A study of Lennard-Jones equivalent analytical relationships for modeling viscosities.* Int J Thermophys. 2001;22(3):829–58.

[39] Silva CM, Liu H, Macedo EA. *Models for self-diffusion coefficients of dense fluids, including hydrogen-bonding substances.* Chem Eng Sci. 1998;53(13):2423–9.

[40] Jones JE, Chapman S. *On the determination of molecular fields. —II. From the equation of state of a gas.* Proc R Soc London Ser A, Contain Pap a Math Phys Character. 1924 Oct 1;106(738):463–77.

[41] Hirschfelder JO, Curtiss CF, Bird RB, of Wisconsin-Milwaukee U, of Wisconsin. Theoretical Chemistry Laboratory U, of Wisconsin. Naval Research Laboratory U. *The Molecular Theory of Gases and Liquids.* New York: John Wiley & Sons; 1954.

[42] Maitland GC, Maitland GD, C MG, Rigby M, Smith EB, Wakeham WA, et al. *Intermolecular Forces: Their Origin and Determination.* Oxford: Clarendon Press; 1987.

[43] Lim TC. *The Relationship between Lennard-Jones (12-6) and Morse Potential Functions.* Zeitschrift fur Naturforsch - Sect A J Phys Sci. 2003;58(11):615–7.

[44] Tersoff J. *New empirical approach for the structure and energy of covalent systems.* Phys Rev B. 1988 Apr 15;37(12):6991–7000.

[45] Dunikov DO, Malyshenko SP, Zhakhovskii V V. *Corresponding states law and molecular dynamics simulations of the Lennard-Jones fluid.* J Chem Phys. 2001;115(14):6623–31.

[46] Tchipev N, Seckler S, Heinen M, Vrabec J, Gratl F, Horsch M, et al. *TweTriS: Twenty trillion-atom simulation.* Int J High Perform Comput Appl. 2019 Jan 6;33(5):838–54.

[47] Stephan S, Liu J, Langenbach K, Chapman WG, Hasse H. *Vapor−Liquid Interface of the Lennard-Jones Truncated and Shifted Fluid: Comparison of Molecular Simulation, Density Gradient Theory, and Density Functional Theory.* J Phys Chem C. 2018 Nov 1;122(43):24705–15.

[48] Borgelt P, Hoheisel C, Stell G. *Exact molecular dynamics and kinetic theory results for thermal transport coefficients of the Lennard-Jones argon fluid in a wide region of states.* Phys Rev A. 1990 Jul 1;42(2):789–94.

[49] Lautenschlaeger MP, Horsch M, Hasse H. *Simultaneous determination of thermal conductivity and shear viscosity using two-gradient non-equilibrium molecular dynamics simulations.* Mol Phys. 2019;117(2):189–99.

[50] Jaeger F, Matar OK, Müller EA. *Bulk viscosity of molecular fluids.* J Chem Phys. 2018;148(17).

[51] Lautenschlaeger MP, Hasse H. *Transport properties of the Lennard-Jones truncated and shifted fluid from non-equilibrium molecular dynamics simulations.* Fluid Phase Equilib. 2019;482:38–47.

[52] Ewen JP, Gattinoni C, Thakkar FM, Morgan N, Spikes HA, Dini D. *A Comparison of Classical Force-Fields for Molecular Dynamics Simulations of Lubricants.* 2016;1–17.

[53] Cornell WD, Kollman P a. *A Second Generation Force Field for the Simulation of Proteins, Nucleic Acids, and Organic Molecules*. J Am Chem Soc. 1995;117(19):5179–97.

[54] Allinger NL, Yuh YH, Lii JH. *Molecular mechanics. The MM3 force field for hydrocarbons. 1*. J Am Chem Soc. 1989;111(23):8551–66.

[55] MacKerell AD, Bashford D, Bellott M, Dunbrack RL, Evanseck JD, Field MJ, et al. *All-Atom Empirical Potential for Molecular Modeling and Dynamics Studies of Proteins*. J Phys Chem B. 1998 Apr 1;102(18):3586–616.

[56] Price MLP, Ostrovsky D, Jorgensen WL. *Gas-phase and liquid-state properties of esters, nitriles, and nitro compounds with the OPLS-AA force field*. J Comput Chem. 2001;22(13):1340–52.

[57] Ponder JW, *Force Fields for Protein Simulations*. In: Protein Simulations. Academic Press; 2003. p. 27–85.

[58] van Duin ACT, Dasgupta S, Lorant F, A. GW. *ReaxFF: A Reactive Force Field for Hydrocarbonds*. J Phys Chem A. 2001;105(41):9396–409.

[59] Feller SE, Pastor RW, Rojnuckarin A, Bogusz S, Brooks BR. *Effect of Electrostatic Force Truncation on Interfacial and Transport Properties of Water*. J Phys Chem. 1996 Jan 1;100(42):17011–20.

[60] Thompson AP, Plimpton SJ, Mattson W. *General formulation of pressure and stress tensor for arbitrary many-body interaction potentials under periodic boundary conditions*. J Chem Phys. 2009;131(15).

[61] Martin MG, Siepmann JI. *Novel Configurational-Bias Monte Carlo Method for Branched Molecules. Transferable Potentials for Phase*

Equilibria. 2. United-Atom Description of Branched Alkanes. J Phys Chem B. 1999 May 1;103(21):4508–17.

[62] Allen W, Rowley RL. *Predicting the viscosity of alkanes using nonequilibrium molecular dynamics: Evaluation of intermolecular potential models.* J Chem Phys. 1997;106(24):10273.

[63] Siu SWI, Pluhackova K, Bo RA. *Optimization of the OPLS-AA Force Field for Long Hydrocarbons.* 2012;

[64] Hairer E, Lubich C, Wanner G. *Geometric numerical integration illustrated by the Störmer–Verlet method.* Acta Numer. 2003/07/29. 2003;12:399–450.

[65] Priezjev N V. *Rate-dependent slip boundary conditions for simple fluids.* Phys Rev E - Stat Nonlinear, Soft Matter Phys. 2007;75(5):1–7.

[66] Bair S, Mccabe C, Cummings PT. *Comparison of Nonequilibrium Molecular Dynamics with Experimental Measurements in the Nonlinear Shear-Thinning Regime.* 2002;88(5):2–5.

[67] Müller-Plathe F. *Reversing the perturbation in nonequilibrium molecular dynamics: an easy way to calculate the shear viscosity of fluids.* Phys Rev E. 1999;59(5):4894–8.

[68] Evans DJ, Morriss GP. *Nonlinear-response theory for steady planar Couette flow.* Phys Rev A. 1984;30(3):1528–30.

[69] Lees AW, Edwards SF. *The computer study of transport processes under extreme conditions.* J Phys C Solid State Phys. 1972;5(15):1921–8.

[70] Arya G, Maginn EJ, Chang HC. *Efficient viscosity estimation from molecular dynamics simulation via momentum impulse relaxation.* J Chem Phys. 2000;113(6):2079–87.

[71] Daivis PJ, Todd BD. *A simple, direct derivation and proof of the validity of the SLLOD equations of motion for generalized homogeneous flows*. J Chem Phys. 2006;124(19).

[72] *OpenFOAM (official website)* [Internet]. Available from: http://www.openfoam.org.

[73] Holzmann T. *Mathematics, Numerics, Derivations and OpenFOAM®*. 2019.

[74] Jasak H. *Error Analysis and Estimation for the Finite Volume Method with Applications to Fluid Flows*. Direct. 1996;M(June):394.

[75] Ramírez-González P V., Escobar-Barrios VA. *Viscosity and normal stress forces of Lennard-Jones chains using reverse non-equilibrium molecular dynamics*. Mol Phys. 2017;115(23):2970–7.

[76] Acheson DJ, Acheson FDJ. *Elementary Fluid Dynamics*. Clarendon Press; 1990. (Comparative Pathobiology - Studies in the Postmodern Theory of Education).

[77] Gattinoni C, Heyes DM, Lorenz CD, Dini D. *Traction and nonequilibrium phase behavior of confined sheared liquids at high pressure*. Phys Rev E - Stat Nonlinear, Soft Matter Phys. 2013;88(5):1–10.

[78] Zhang L, Balasundaram R, Gehrke SH, Jiang S. *Nonequilibrium molecular dynamics simulations of confined fluids in contact with the bulk*. J Chem Phys. 2001;114(15):6869–77.

[79] Savio D, Fillot N, Vergne P, Zaccheddu M. *A model for wall slip prediction of confined n-alkanes: Effect of wall-fluid interaction versus fluid resistance*. Tribol Lett. 2012;46(1):11–22.

[80] Hansen JS, Todd BD, Daivis PJ. *Prediction of fluid velocity slip at solid surfaces*. Phys Rev E - Stat Nonlinear, Soft Matter Phys. 2011;84(1):1–8.

[81] Jabbarzadeh A, Atkinson JD, Tanner RI. *Wall slip in the molecular dynamics simulation of thin films of hexadecane*. J Chem Phys. 1999;110(5):2612.

[82] Lauga E, Brenner MP, Stone HA. *Microfluidics: The no-slip boundary condition*. Springer, Berlin, Heidelberg; 2005.

[83] Pai S. *Viscous Flow Theory: Laminar flow*. D. Van Nostrand Company, Inc.; 1956.

[84] Çengel YA, Cimbala JM. *Fluid Mechanics: Fundamentals and Applications, 4e in SI Units*. McGraw-Hill Education; 2019.

[85] Kuneš J. *3 - Fluid Mechanics*. In: Kuneš J, editor. Dimensionless Physical Quantities in Science and Engineering. Oxford: Elsevier; 2012. p. 53–133.

[86] Sigalotti LDG, Klapp J, Sira E, Meleán Y, Hasmy A. *SPH simulations of time-dependent Poiseuille flow at low Reynolds numbers*. J Comput Phys. 2003;191(2):622–38.

[87] Faraji F, Rajabpour A, Kowsary F. *Temperature profile for nanoscale Poiseuille flow: a multiscale study*. J Mech Sci Technol. 2016;30(2):803–7.

[88] Hamming RW. *Numerical Methods for Scientists and Engineers*. Dover Publications, Inc. 1986.

[89] Vakis AI, Yastrebov VA, Scheibert J, Nicola L, Dini D, Minfray C, et al. *Modeling and simulation in tribology across scales : An overview*

Tribology International Modeling and simulation in tribology across scales : An overview. Tribol Int. 2018;125(February):169–99.

[90] Weinan E, Li X, Vanden-Eijnden E. *Some Recent Progress in Multiscale Modeling.* In: Attinger S, Koumoutsakos P, editors. Multiscale Modelling and Simulation. Berlin, Heidelberg: Springer Berlin Heidelberg; 2004. p. 3–21.

[91] Holland DM, Lockerby DA, Borg MK, Nicholls WD, Reese JM. *Molecular dynamics pre-simulations for nanoscale computational fluid dynamics.* Microfluid Nanofluidics. 2015;18(3):461–74.

[92] Yasuda S, Yamamoto R. *A model for hybrid simulations of molecular dynamics and computational.* Physics of Fluids. 2008;20(11):113101

[93] Ren W, Weinan E. *Heterogeneous multiscale method for the modeling of complex fluids and micro-fluidics.* J Comput Phys. 2005;204(1):1–26.

[94] Hoover WG, Hoover CG, Petravic J. *Simulation of two- and three-dimensional dense-fluid shear flows via nonequilibrium molecular dynamics: Comparison of time-and-space-averaged stresses from homogeneous Doll's and Sllod shear algorithms with those from boundary-driven shear.* Phys Rev E. 2008 Oct 2;78(4):46701.

[95] Asproulis N, Kalweit M, Drikakis D. *Advances in Engineering Software A hybrid molecular continuum method using point wise coupling.* Adv Eng Softw. 2010;

[96] Borg MK, Lockerby DA, Reese JM. *Fluid simulations with atomistic resolution : a hybrid multiscale method with field-wise coupling.* J Comput Phys. 2013;255:149–65.

[97] Bian X, Praprotnik M. *Domain Decomposition Methods for Multiscale Modeling.* Handb Mater Model. 2018;1–21.

[98] Vu VH, Trouette B, To QD, Chénier E. *Hybrid atomistic-continuum multiscale method for fluid flow with density variation in microchannels*. Int J Numer Methods Heat Fluid Flow. 2018;28(1).

[99] De Fabritiis G, Delgado-Buscalioni R, Coveney P V. *Multiscale modeling of liquids with molecular specificity*. Phys Rev Lett. 2006;97(13):1–4.

[100] O'Connell ST, Thompson PA. *Molecular dynamics-continuum hybrid computations: A tool for studying complex fluid flows*. Phys Rev E. 1995;52(6):5792–5.

[101] Curtin WA, Miller RE. *Atomistic/continuum coupling in computational materials science*. Model Simul Mater Sci Eng. 2003;11(3):R33–68.

[102] Nie XB, Chen SY, E WN, Robbins MO. *A continuum and molecular dynamics hybrid method for micro- and nano-fluid flow*. J Fluid Mech. 2004;500(2004):55–64.

[103] Werder T, Walther JH, Koumoutsakos P. *Hybrid atomistic-continuum method for the simulation of dense fluid flows*. J Comput Phys. 2005;205(1):373–90.

[104] Issa KM, Poesio P. *Algorithm to enforce uniform density in liquid atomistic subdomains with specular boundaries*. Phys Rev E - Stat Nonlinear, Soft Matter Phys. 2014;89(4).

[105] Hadjiconstantinou NG. *Hybrid Atomistic-Continuum Formulations and the Moving Contact-Line Problem*. J Comput Phys. 1999;154(2):245–65.

[106] Delgado-Buscalioni R. *Tools for Multiscale Simulation of Liquids Using Open Molecular Dynamics BT - Numerical Analysis of Multiscale*

Computations. In: Engquist B, Runborg O, Tsai Y-HR, editors. Berlin, Heidelberg: Springer Berlin Heidelberg; 2012. p. 145–66.

[107] Delgado-Buscalioni R, Coveney P V. *USHER: An algorithm for particle insertion in dense fluids*. J Chem Phys. 2003;119(2):978–87.

[108] Praprotnik M, Delle Site L, Kremer K. *Adaptive resolution molecular-dynamics simulation: Changing the degrees of freedom on the fly*. J Chem Phys. 2005;123(22).

[109] Kotsalis EM, Walther JH, Koumoutsakos P. *Control of density fluctuations in atomistic-continuum simulations of dense liquids*. Phys Rev E. 2007 Jul 30;76(1):16709.

[110] Hadjiconstantinou NG, Garcia AL, Bazant MZ, He G. *Statistical error in particle simulations of hydrodynamic phenomena*. J Comput Phys. 2003;187(1):274–97.

[111] Smith E. *CPL Library (official website)* [Internet]. Available from: http://www.cpl-library.org.

[112] Delgado-Buscalioni R, Coveney P V. *Continuum-particle hybrid coupling for mass, momentum, and energy transfers in unsteady fluid flow.* Phys Rev E Stat Nonlin Soft Matter Phys. 2003;67(4 Pt 2):046704.

[113] Smith ER, Heyes DM, Dini D, Zaki TA. *A localized momentum constraint for non-equilibrium molecular dynamics simulations*. J Chem Phys. 2015;142(7):74110.

[114] Flekkøy EG, Wagner G, Feder J. *Hybrid model for combined particle and continuum dynamics*. Europhys Lett. 2000;52(3):271–6.

[115] Hadjiconstantinou NG, Patera AT. *Heterogeneous Atomistic-Continuum Representations for Dense Fluid Systems.* Int J Mod Phys C. 1997 Aug 1;08(04):967–76.

[116] Scott G, Richardson P. *The application of computational fluid dynamics in the food industry.* Trends Food Sci Technol. 1997;8(4):119–24.

[117] Spalart PR, Venkatakrishnan V. *On the role and challenges of CFD in the aerospace industry.* Aeronaut J. 2016;120(1223):209–232.

[118] Dhaubhadel MN. *Review: CFD Applications in the Automotive Industry.* J Fluids Eng. 1996;118(4):647–53.

[119] Sochi T. *Slip at fluid-solid interface.* Polymer Reviews. 2011.

[120] Praprotnik M, Site LD, Kremer K. *Multiscale simulation of soft matter: from scale bridging to adaptive resolution.* Annu Rev Phys Chem. 2008;59:545–71.

[121] Elber R. *Perspective: Computer simulations of long time dynamics.* J Chem Phys. 2016;144(6):1–12.

[122] Plimpton S. *Computational limits of classical molecular dynamics simulations.* Comput Mater Sci. 1995;4(4):361–4.

[123] Yasuda S, Yamamoto R. *Synchronized molecular-dynamics simulation via macroscopic heat and momentum transfer: An application to polymer lubrication.* Phys Rev X. 2014;4(4):1–10.

[124] Flekkøy EG, Coveney P V. *Fluctuations and continuity in particle-continuum hybrid simulations of unsteady flows based on flux-exchange.* 2005;69(March):959–65.

[125] Cosden IA, Lukes JR. *A hybrid atomistic-continuum model for fluid flow using LAMMPS and OpenFOAM*. Comput Phys Commun. 2013;184(8):1958–65.

[126] Cui ST, Gupta SA, Cummings PT, Cochran HD. *Molecular dynamics simulations of the rheology of normal decane, hexadecane, and tetracosane*. J Chem Phys. 1996;105(3):1214–20.

[127] Ciccotti G, Ryckaert JP. *Molecular dynamics simulation of rigid molecules*. Comput Phys Reports. 1986;4(6):346–92.

[128] Ewen JP, Gattinoni C, Zhang J, Heyes DM, Spikes HA, Dini D. *On the effect of confined fluid molecular structure on nonequilibrium phase behaviour and friction*. Phys Chem Chem Phys. 2017;19:17883–94.

[129] Sendner C, Horinek D, Bocquet L, Netz RR. *Interfacial water at hydrophobic and hydrophilic surfaces: Slip, viscosity, and diffusion*. Langmuir. 2009;25(18):10768–81.

[130] Doig M, Camp PJ. *The structures of hexadecylamine films adsorbed on iron-oxide surfaces in dodecane and hexadecane*. Phys Chem Chem Phys Phys Chem Chem Phys. 2015;5248(17):5248–55.

[131] Yeh I-C, Berkowitz ML. *Ewald summation for systems with slab geometry*. J Chem Phys. 1999;111(7):3155.

[132] Yong X, Zhang LT. *Thermostats and thermostat strategies for molecular dynamics simulations of nanofluidics*. J Chem Phys. 2013;138(8).

[133] Bernardi S, Todd BD, Searles DJ. *Thermostating highly confined fluids*. J Chem Phys. 2010;132(24).

[134] Berro H, Fillot N, Vergne P, Tokumasu T, Ohara T, Kikugawa G. *Energy dissipation in non-isothermal molecular dynamics simulations of confined liquids under shear.* J Chem Phys. 2011;135(13).

[135] Berro H, Fillot N, Vergne P. *Molecular dynamics simulation of surface energy and ZDDP effects on friction in nano-scale lubricated contacts.* Tribol Int. 2010;43(10):1811–22.

[136] Vadgama BN, Jackson RL, Harris DK. *Molecular scale analysis of dry sliding copper asperities.* Appl Nanosci. 2015;5(4):469–80.

[137] Sun K, Fang L, Yan Z, Sun J. *Atomistic scale tribological behaviors in nano-grained and single crystal copper systems.* Wear. 2013;303(1–2):191–201.

[138] Wen M, Xu X-J, Fukuyama S, Yokogawa K. *Embedded-atom-method functions for the body-centered-cubic iron and hydrogen.* J Mater Res. 2001;16(12):3496–502.

[139] Berro H. *A Molecular Dynamics Approach to Nano-scale Lubrication.* PhD Thesis. L'Institut National des Sciences Appliquées de Lyon;2010;

[140] Mao Y, Zhang Y. *Molecular dynamics simulation on rapid boiling of water on a hot copper plate.* Appl Therm Eng. 2014;62(2):607–12.

[141] Dolbow J, Gosz M. *Effect of out-of-plane properties of a polyimide film on the stress fields in microelectronic structures.* Mech Mater. 1996;23(4):311–21.

[142] Schneider T, Stoll E. *Molecular-dynamics study of a three-dimensional one-component model for distortive phase transitions.* Phys Rev B. 1978;17(3):1302–22.

[143] Delle Site L, Praprotnik M. *Molecular systems with open boundaries: Theory and simulation.* Phys Rep. 2017;693:1–56.

[144] Allen MP. *Atomic and molecular representations of molecular hydrodynamic variables*. Mol Phys. 1984 Jun 20;52(3):705–16.

[145] Hyndman R, Koehler AB, Ord JK, Snyder RD. *Forecasting with Exponential Smoothing: The State Space Approach*. Springer Berlin Heidelberg; 2008. (Springer Series in Statistics).

[146] Liu J, Chen S, Nie X, Robbins MO. *A continuum-atomistic simulation of heat transfer in micro- and nano-flows*. J Comput Phys. 2007;227(1):279–91.

[147] Ren W. *Analytical and numerical study of coupled atomistic-continuum methods for fluids*. J Comput Phys. 2007;227(2):1353–71.

[148] Rafael B. *Hybrid molecular-continuum fluid dynamics*. Philos Trans A Math Phys Eng Sci. 2004;362(1821):1639–54.

[149] Borg MK, Lockerby DA, Reese JM. *The FADE mass-stat: A technique for inserting or deleting particles in molecular dynamics simulations*. J Chem Phys. 2014;140(7).

[150] Shinoda W, Shiga M, Mikami M. *Rapid estimation of elastic constants by molecular dynamics simulation under constant stress*. Phys Rev B - Condens Matter Mater Phys. 2004;69(13).

[151] Kloss C, Goniva C, Hager A, Amberger S, Pirker S. *Models, algorithms and validation for opensource DEM and CFD–DEM*. Prog Comput Fluid Dyn an Int J. 2012;12(2–3):140–52.

[152] Sun R, Xiao H. *SediFoam: A general-purpose, open-source CFD-DEM solver for particle-laden flow with emphasis on sediment transport*. Comput Geosci. 2016;89:207–19.

[153] Neumann P, Flohr H, Arora R, Jarmatz P, Tchipev N, Bungartz HJ. *MaMiCo: Software design for parallel molecular-continuum flow simulations*. Comput Phys Commun. 2016;200:324–35.

[154] Smith E, Trevelyan D, Zaki T. *Scalable coupling of Molecular Dynamics (MD) and Direct Numerical Simulation (DNS) of Multi-scale Flows—Part 2*. 2013;(Md):1–7.

[155] Tang Y, Kudo S, Bian X, Li Z, Karniadakis GE. *Multiscale Universal Interface: A Concurrent Framework for Coupling Heterogeneous Solvers*. 2015;1–24.

[156] Ben Belgacem M, Chopard B. *MUSCLE-HPC: A new high performance API to couple multiscale parallel applications*. Futur Gener Comput Syst. 2017;67:72–82.

[157] Bungartz HJ, Lindner F, Gatzhammer B, Mehl M, Scheufele K, Shukaev A, et al. *preCICE – A fully parallel library for multi-physics surface coupling*. Comput Fluids. 2016;141:250–8.

[158] Larson J, Jacob R, Ong E. *The Model Coupling Toolkit: A New Fortran90 Toolkit for Building Multiphysics Parallel Coupled Models*. Int J High Perform Comput Appl. 2005;19(3):277–92.

[159] Craig A, Valcke S, Coquart L. *Development and performance of a new version of the OASIS coupler, OASIS3-MCT_3.0*. Geosci Model Dev. 2017;10(9):3297–308.

[160] Groen D, Coveney P V, Zasada SJ. *Taxonomy of Multiscale Computing Communities*. In: e-Science Workshops, IEEE International Conference on. Los Alamitos, CA, USA: IEEE Computer Society; 2011. p. 120–7.

[161] Borgdorff J, Falcone JL, Lorenz E, Bona-Casas C, Chopard B, Hoekstra AG. *Foundations of distributed multiscale computing: Formalization, specification, and analysis*. J Parallel Distrib Comput. 2013;73(4):465–83.

[162] Frigo M, Johnson S. *The Design and Implementation of FFTW3*. Proc IEEE. 2005;93(2):216–31.

[163] Blackford LS, Choi J, Cleary A, D'Azevedo E, Demmel J, Dhillon I, et al. *ScaLAPACK Users' Guide*. Philadelphia, PA: Society for Industrial and Applied Mathematics; 1997.

[164] Slattery SR. *Mesh-free data transfer algorithms for partitioned multiphysics problems: Conservation, accuracy, and parallelism*. J Comput Phys. 2016;

[165] Kedward L, Allen CB, Rendall TCS. *Efficient and exact mesh deformation using multiscale RBF interpolation*. J Comput Phys. 2017;

[166] Gropp W, Gropp WD, Lusk E, Skjellum A. *Using MPI: portable parallel programming with the message-passing interface*. Vol. 1. MIT press; 1999.

[167] Rumbaugh J, Jacobson I, Booch G. *Unified Modeling Language Reference Manual, The (2nd Edition)*. Pearson Higher Education; 2004.

[168] Allalen M, Gmbh FJ. *Best Practice Guide - Modern Interconnects*. 2019;1–21.

[169] Reuther A, Byun C, Arcand W, Bestor D, Bergeron B, Hubbell M, et al. *Scalable system scheduling for HPC and big data*. J Parallel Distrib Comput. 2018;111:76–92.

[170] McLay R, Schulz KW, Barth WL, Minyard T. *Best Practices for the Deployment and Management of Production HPC Clusters*. In: State of the Practice Reports. New York, NY, USA: Association for Computing Machinery; 2011. (SC '11).

[171] Beaulieu-Jones BK, Greene CS. *Reproducibility of computational workflows is automated using continuous analysis*. Nat Biotechnol. 2017;35(4):342–6.

[172] Younge AJ, Pedretti K, Grant RE, Brightwell R. *A Tale of Two Systems: Using Containers to Deploy HPC Applications on Supercomputers and Clouds*. Proc Int Conf Cloud Comput Technol Sci CloudCom. 2017;2017–Decem:74–81.

[173] Grüning B, Chilton J, Köster J, Dale R, Soranzo N, van den Beek M, et al. *Practical Computational Reproducibility in the Life Sciences*. Cell Syst. 2018;6(6):631–5.

[174] *Anaconda Software Distribution. Version 2-2.4.0* [Internet]. Available from: https://anaconda.com.

[175] Möller S, Krabbenhöft HN, Tille A, Paleino D, Williams A, Wolstencroft K, et al. *Community-driven computational biology with Debian Linux*. BMC Bioinformatics. 2010;11(SUPPL. 12).

[176] *Homebrew (official website)* [Internet]. Available from: https://brew.sh/

[177] *GNU C Library (official website)* [Internet]. Available from: https://www.gnu.org/software/libc/.

[178] Merkel D. *Docker: lightweight linux containers for consistent development and deployment*. Linux J. 2014;2014(239):2.

[179] Ruiz C, Jeanvoine E, Nussbaum L. *Performance Evaluation of Containers for HPC*. In: Hunold S, Costan A, Giménez D, Iosup A, Ricci L, Gómez Requena ME, et al., editors. Euro-Par 2015: Parallel Processing Workshops. Cham: Springer International Publishing; 2015. p. 813–24.

[180] Kurtzer GM, Sochat V, Bauer MW. *Singularity: Scientific containers for mobility of compute*. PLoS One. 2017;12(5):1–20.

[181] Zhang J, Lu X, Panda DK. *Is singularity-based container technology ready for running MPI applications on HPC clouds? ∗*. UCC 2017 - Proc the10th Int Conf Util Cloud Comput. 2017;151–60.

[182] *Red Hat Enterprise Linux (official website)* [Internet]. Available from: https://www.redhat.com/en/technologies/linux-platforms/enterprise-linux.

[183] C. Martin R. *Clean Code: A Handbook of Agile Software Craftsmanship*. Pearson; 2008.

[184] *Googletest (official website)* [Internet]. Available from: https://github.com/google/googletest.

[185] *Pytest (official website)* [Internet]. Available from: https://pytest.org.

[186] *Junit (official website)* [Internet]. Available from: https://junit.org.

[187] Nethercote N, Seward J. *Valgrind: A Framework for Heavyweight Dynamic Binary Instrumentation*. SIGPLAN Not. 2007 Jun;42(6):89–100.

[188] B KM, Oldfield R. *Proceedings of the International Supercomputing Conference 2015*. Int Supercomput Conf. 2015;9137:530.

[189] Kumar V, Gupta A. *Analyzing Scalability of Parallel Algorithms and Architectures*. Vol. 22, Journal of Parallel and Distributed Computing. 1994. p. 379–91.

[190] *LAMMPS scaling benchmark* [Internet]. Available from: https://lammps.sandia.gov/bench.html.

[191] Kutzner C, Van Der Spoel D, Fechner M, Lindahl E, Schmitt UW, De Groot BL, et al. *Speeding up parallel GROMACS on high-latency networks*. J Comput Chem. 2007;28(12):2075–84.

[192] Gropp WD, Kaushik DK, Keyes DE, Smith BF. *Latency, bandwidth, and concurrent issue limitations in high-performance CFD*. Comput Fluid Solid Mech. 2001;1(630):839–42.

[193] Laser T. *TCP/IP - The Internet Protocol Stack*. Santa Cruz, CA, USA: GRIN Verlag; 2013.

[194] Switters J, Osimo D. *European Commission. Directorate General for Research and Innovation. Recognising the importance of software in research: Research Software Engineers (RSEs), a UK example*. [Online] LU: Publications Office; 2019. Available from: doi:10.2777/787013

[195] Hoheisel C. *Bulk viscosity of model fluids. A comparison of equilibrium and nonequilibrium molecular dynamics results*. J Chem Phys. 1986;86(4):2328–33.

[196] Ingebrigtsen TS, Schrøder TB, Dyre JC. *What Is a Simple Liquid?* Phys Rev X. 2012 Mar;2(1):11011.

[197] Jones RE, Mandadapu KK. *Adaptive Green-Kubo estimates of transport coefficients from molecular dynamics based on robust error analysis*. J Chem Phys. 2012;136(15).

[198] Delhommelle J, Evans DJ. *Comparison of thermostatting mechanisms in NVT and NPT simulations of decane under shear*. J Chem Phys. 2001;115(1):43–9.

[199] Hoover WG, Hoover CG, Petravic J. *Simulation of two- and three-dimensional dense-fluid shear flows via nonequilibrium molecular dynamics: Comparison of time-and-space-averaged stresses from homogeneous Doll's and Sllod shear algorithms with those from boundary-driven shear*. Phys Rev E - Stat Nonlinear, Soft Matter Phys. 2008;78(4):1–14.

[200] Barnes HA, Hutton JF, Walters K, editors. *Chapter 4 - Normal Stresses*. In: An Introduction to Rheology. Elsevier; 1989. p. 55–73. (Rheology Series; vol. 3).

[201] Pensado AS, Comuñas MJP, Lugo L, Fernández J. *High-Pressure Characterization of Dynamic Viscosity and Derived Properties for Squalane and Two Pentaerythritol Ester Lubricants: Pentaerythritol Tetra-2-ethylhexanoate and Pentaerythritol Tetranonanoate*. Ind Eng Chem Res. 2006;45(7):2394–404.

[202] Cafolla C, Foster W, Voïtchovsky K. *Lubricated friction around nanodefects*. Sci Adv. 2020;6(14):1–9.

[203] Comuñas MJP, Paredes X, Gaciño FM, Fernández J, Bazile JP, Boned C, et al. *Viscosity measurements for squalane at high pressures to 350 MPa from T = (293.15 to 363.15) K*. J Chem Thermodyn. 2014;69:201–8.

[204] Bair S. *Reference liquids for quantitative elastohydrodynamics: Selection and rheological characterization*. Tribol Lett. 2006;22(2):197–206.

[205] Martínez L, Andrade R, Birgin E, Martínez JM. *PACKMOL: A package for building initial configurations for molecular dynamics simulations.* J Comput Chem. 2009;30:2157–64.

[206] Jewett AI, Zhuang Z, Shea J-E. *Moltemplate a Coarse-Grained Model Assembly Tool.* Biophys J. 2013 Jan;104(2):169a.

[207] Tuckerman ME, Alejandre J, López-Rendón R, Jochim AL, Martyna GJ. *A Liouville-operator derived measure-preserving integrator for molecular dynamics simulations in the isothermal-isobaric ensemble.* J Phys A Math Gen. 2006 Apr;39(19):5629–51.

[208] Bair S, Flores-torres S. *The Viscosity of Polyalphaolefins Mixtures at High Pressure and Stress.* 2019;141(February):1–7.

[209] Shin S, Cho YI. *Temperature effect on the non-Newtonian viscosity of an aqueous polyacrylamide solution.* Int Commun Heat Mass Transf. 1993;20(6):831–44.

[210] Bair S, Clare M. *Chapter 6 Models for the temperature and pressure dependence of low-shear viscosity.* Tribol Interface Eng Ser. 2007;54:101–32.

[211] O'Connor TC, Andzelm J, Robbins MO. *AIREBO-M: A reactive model for hydrocarbons at extreme pressures.* J Chem Phys. 2015;142(2).

[212] Mondello M, Grest GS. *Viscosity calculations of n-alkanes by equilibrium molecular dynamics.* J Chem Phys. 1997;106(22):9327–36.

[213] Zhang Y, Otani A, Maginn EJ. *Reliable Viscosity Calculation from Equilibrium Molecular Dynamics Simulations: A Time Decomposition Method.* J Chem Theory Comput. 2015;11(8):3537–46.

[214] Assael MJ, Mihailidou EK, Huber ML, Perkins RA. *Reference Correlation of the Thermal Conductivity of Benzene from the Triple Point to 725 K up to 500 MPa*. J Phys Chem Ref Data. 2012;41(4).

[215] Czarnota I. *Heat capacity of benzene at high pressures*. J Chem Thermodyn. 1991;23(1):25–30.

[216] Björling M, Habchi W, Bair S, Larsson R, Marklund P. *Towards the true prediction of EHL friction*. Tribol Int. 2013;66:19–26.

[217] Bair SS, Andersson O, Qureshi FS, Schirru MM. *New EHL Modeling Data for the Reference Liquids Squalane and Squalane Plus Polyisoprene*. Tribol Trans. 2018;61(2):247–55.

[218] Korotkovskii VI, Lebedev A V., Ryshkova OS, Bolotnikov MF, Shevchenko YE, Neruchev YA. *Thermophysical properties of liquid squalane C30H62 within the temperature range of 298.15-413.15 K at atmospheric pressure*. High Temp. 2012;50(4):471–4.

[219] Heyes DM. *Shear thinning and thickening of the Lennard-Jones liquid. A molecular dynamics study*. J Chem Soc Faraday Trans 2 Mol Chem Phys. 1986;82(9):1365–83.

[220] O'Connell ST, Thompson PA. *Molecular dynamics–continuum hybrid computations: A tool for studying complex fluid flows*. Physical Review E. American Physical Society (APS);1995;52(6):5792–5.

[221] Smith E, Trevelyan D, Zaki T. *Scalable coupling of Molecular Dynamics (MD) and Direct Numerical Simulation (DNS) of Multi-scale Flows Part 2*. In 2013.

[222] Qian W, Xinhai X, Chao L, Hongyu J, Xiaoguang R, Xuejun Y. *A Study on Coupling Strategies of Hybrid Atomistic-Continuum Method Based on State Variable Coupling*. Adv Mater Sci Eng. 2017;2017:1–21.

[223] Berro H, Fillot N, Vergne P. *Hybrid diffusion: An efficient method for kinetic temperature calculation in molecular dynamics simulations of confined lubricant films.* Tribol Lett. 2010;37(1):1–13.

[224] Minfray C, Le Mogne T, Martin J-M, Onodera T, Nara S, Takahashi S, et al. *Experimental and Molecular Dynamics Simulations of Tribochemical Reactions with ZDDP: Zinc Phosphate–Iron Oxide Reaction.* Tribol Trans. 2008;51(5):589–601.

[225] Masoori M, Greenfield ML. *Reducing noise in computed correlation functions using techniques from signal processing.* Mol Simul. 2017;43(18):1485–95.

[226] Walton EB, VanVliet KJ. *Equilibration of experimentally determined protein structures for molecular dynamics simulation.* Phys Rev E - Stat Nonlinear, Soft Matter Phys. 2006;74(6):1–8.

[227] Stella L, Melchionna S. *Equilibration and sampling in molecular dynamics simulations of biomolecules.* J Chem Phys. 1998;109(23):10115–7.

[228] Larsen AH, Mortensen JJ, Blomqvist J, Castelli IE, Christensen R, Dułak M, et al. *The atomic simulation environment—a Python library for working with atoms.* J Phys Condens Matter. 2017;29(27):273002.

[229] Gattinoni C, Maćkowiak S, Heyes DM, Brańka AC, Dini D. *Boundary-controlled barostats for slab geometries in molecular dynamics simulations.* Phys Rev E - Stat Nonlinear, Soft Matter Phys. 2014;90(4):1–13.

[230] Khasanshin TS, Samuilov VS, Shchemelev AP. *Determination of the thermodynamic properties of liquid n-hexadecane from the measurements of the velocity of sound.* J Eng Phys Thermophys. 2009;82(1):149–56.

[231] Sablić J, Praprotnik M, Delgado-Buscalioni R. *Open boundary molecular dynamics of sheared star-polymer melts*. Soft Matter. 2016;

[232] Liu P, Lu J, Yu H, Ren N, Lockwood FE, Wang QJ. *Lubricant shear thinning behavior correlated with variation of radius of gyration via molecular dynamics simulations*. J Chem Phys. 2017;147(8).

[233] Smith E. *On the coupling of molecular dynamics to continuum computational fluid dynamics*. PhD Thesis. Imperial College London; 2013.

[234] Grami A. *Chapter 3 - Signals, Systems, and Spectral Analysis*. In: Grami A, editor. Introduction to Digital Communications. Boston: Academic Press; 2016. p. 41–150.

[235] Hu MX, Xu T, Momen R, Azizi A, Kirk SR, Jenkins S. *The normal modes of vibration of benzene from the trajectories of stress tensor eigenvector projection space*. Chem Phys Lett. 2017;677:156–61.

[236] Galmiche B, Ponjavic A, Wong JSS. *Flow measurements of a polyphenyl ether oil in an elastohydrodynamic contact*. J Phys Condens Matter. 2016;28(13):134005.

[237] Ponjavic A, Dench J, Morgan N, Wong JSS. In situ *viscosity measurement of confined liquids*. RSC Adv. 2015;5(121):99585–93.

[238] Tung SC, McMillan ML. *Automotive tribology overview of current advances and challenges for the future*. Tribol Int. 2004;37(7):517–36.

[239] Myant C, Underwood R, Fan J, Cann PM. *Lubrication of metal-on-metal hip joints: The effect of protein content and load on film formation and wear*. J Mech Behav Biomed Mater. 2012;6:30–40.

[240] Rowley RL, Painter MM. *Diffusion and viscosity equations of state for a Lennard-Jones fluid obtained from molecular dynamics simulations.* Int J Thermophys. 1997;18(5):1109–21.

[241] Lautenschlaeger MP, Hasse H. *Thermal and caloric properties of fluids from non-equilibrium molecular dynamics simulations using the two-gradient method.* J Chem Phys. 2018;149(24).

[242] Baidakov VG, Protsenko SP. *Metastable Lennard-Jones fluids. II. Thermal conductivity.* J Chem Phys. 2014;140(21).

[243] Bugel M, Galliero G. *Thermal conductivity of the Lennard-Jones fluid: An empirical correlation.* Chem Phys. 2008;352(1–3):249–57.

[244] Meier K, Laesecke A, Kabelac S. *Transport coefficients of the Lennard-Jones model fluid. I. Viscosity.* J Chem Phys. 2004;121(8):3671–87.

[245] Meier K, Laesecke A, Kabelac S. *Transport coefficients of the Lennard-Jones model fluid. II Self-diffusion.* J Chem Phys. 2004;121(19):9526–35.

[246] Galliero G, Boned C. *Thermal conductivity of the Lennard-Jones chain fluid model.* Phys Rev E. 2009 Dec;80(6):61202.

[247] Eskandari Nasrabad A, Laghaei R, Eu BC. *Molecular theory of thermal conductivity of the Lennard-Jones fluid.* J Chem Phys. 2006;124(8).

Appendix A

Unified Modeling Language

UML Element	Description
`A, M`	A for attributes section and M for methods section.
	Inheritance relation from a CHILD class to a PARENT class.
`virtual`	Virtual methods can be overridden by child classes but have implementation in .
`=0`	Virtual abstract method does not have implementation in PARENT class and are to be implemented in child classes.
`<<Abstract Class>>`	Cannot be instantiated (allocated in memory).
`#name`	Protected attribute/method.
`+name`	Public attribute/method.

Table 1: Description of the tags and relations used in the simplified UML diagrams in Chapter 4.

Software Versions and Repositories

This appendix contains a compilation of a list of: (a) software packages developed as part of the coupling framework presented here (see Table 2) and (b) 3rd party software which has contributed to this work (Table 3).

Name	Version	URL
CPLAPP-LAMMPS	11Aug2017	https://github.com/Crompulence/CPL_APP_LAMMPS-DEV/tree/edu159
CPLAPP-OPENFOAM	3.0.1	https://github.com/Crompulence/CPL_APP_OPENFOAM-3.0.1/tree/edu159
CPLUTILS	-	https://github.com/edu159/cplutils
CPL_LIBRARY	2.0.0	https://github.com/Crompulence/cpl-library/tree/edu159

Table 2: Most relevant software packages developed in this work, their version and a download URL to the official repository with the source code.

In order to create new custom-built `Conda` packages, as done for `CPLAPP-LAMMPS` and `CPLAPP-OPENFOAM` coupled applications, a set of configuration scripts are necessary called *recipes* (see documentation in https://www.anaconda.org for further details). The recipes to build the packages, required to perform the simulations presented in this manuscript, can be found in the repository:

https://github.com/Crompulence/cpl_conda_builds/tree/edu159.

All custom-built packages (containing pre-compiled binaries) created in the `CentOS6.0` virtual machine (see Section 4.3.1 for details), can be directly downloaded from the following Anaconda *channel* (online service to host custom-built packages):

https://anaconda.org/edu159/,

or, more conveniently, they can also be installed directly into a `Conda` *environment* along with all their dependencies with a single `Conda` command:

```
conda install -c edu159 package_name
```

Name	Version	URL
LAMMPS	11Aug2017	https://download.lammps.org/tars/lammps-11Aug2017.tar.gz
OpenFOAM	3.0.1	https://sourceforge.net/projects/foam/files/foam/3.0.1/OpenFOAM-3.0.1.tgz
Moltemplate	2.8.1	https://www.moltemplate.org/
Packmol	17.221	https://github.com/mcubeg/packmol/archive/17.221.tar.gz
Conda	4.7.10	https://repo.continuum.io/miniconda/Miniconda2-4.7.10-Linux-x86_64.sh
MPICH	3.2	http://www.mpich.org/static/downloads/3.2/mpich-3.2.tar.gz
CentOS(*)	6.0	http://vault.centos.org/6.0/

Table 3: Relevant software packages used in this work, their version and a download URL to the official repository containing the source code and/or installation scripts. (*)`CentOS` is the operating system installed in the virtual machine which was used to create custom-built `Conda` packages.