Molecular Modelling of Meso- and Nanoscale Dynamics

Marco Kalweit

Submitted for the Degree of Ph.D.

Department of Aerospace Sciences
Cranfield University
Cranfield, UK

2008

Cranfield University

School of Engineering

PhD

Academic Year: 2007-2008

Marco Kalweit

Molecular Modelling of Meso- and Nanoscale Dynamics

Supervisor: Dimitris Drikakis

February 15, 2008

© Cranfield University, 2008.
All rights reserved. No part of this publication may be reproduced
without the written permission of the copyright holder.

Abstract

Molecular modelling of meso- and nanoscale dynamics is concerned with length and time scales that are in the transition zone from molecular to continuum models. Molecular simulation methods, in particular molecular dynamics (MD), only allow the simulation of relatively small nanoscale systems. Continuum methods, such as computational fluid dynamics (CFD), are applicable at macroscopic scales but cease to be valid for nanoscales. This thesis is focused on hybrid MD-CFD methods with geometrical decomposition that seek to bridge the gap between molecular and continuum modelling.

The hybrid solution interface (HSI) establishes the coupling between the molecular and the continuum domain. In this work, different realisation approaches for the HSI, flux and state coupling, are discussed and compared. A detailed investigation on MD flux boundary conditions, the most crucial part of a flux based HSI, is carried out. Different schemes for the imposition of mass, momentum and energy fluxes through convective and viscous transport are presented: direct and indirect flux imposition for convective fluxes; the imposition of momentum fluxes through reflective walls, external forces and the reverse velocity scheme; and imposition of energy fluxes through external forces and an energy transfer scheme. Different combinations of these schemes are compared for standard flow situations.

The momentum and energy transfer by an external force creates a relaxation zone at the MD boundary. The characteristics of this zone is investigated in detail and a theoretical model for the density profile has been derived. The reverse velocity scheme has been created as part of this work to avoid the problems encountered when using the external force for the momentum transfer. It is shown that indirect convective flux imposition in conjunction with the reverse velocity scheme gives the best results for the standard flow situations. The scheme is also tested for liquid flow past Carbon nanotubes.

Acknowledgements

Looking back at the end of this PhD and walking the path again in my mind, I meet many people that have helped me on this sometimes quite steep and stony way. Without them the last years would have been a much less educational, interesting and enjoyable part of my life.

Firstly, I would like to thank my supervisor, Prof. Drikakis, for providing me the opportunity to do this PhD, for his constant effort to ensure good working conditions and for his advices in finding the best direction in research. I also appreciate the many fruitful discussion with the members of the FMACS group, in particulary, Marco Hahn, Sanjey Patel, Nicholas Epiphaniou, Ben Thornber and Evgeniy Shapiro.

Doing a PhD in numerical simulations means that one spends incredible long hours with his computer in the office. I would like to thank the people in office F51 for making this time much more pleasurable: Jason Bennet, Steve Carnduff, Ian Cowling, Simon Croucher, Marco Hahn, Sunil Mistry, John Murphy, Sanjey Patel (for his dictionary services) and Ben Thornber (for all the paper balls that kept me awake).

Finally, I would like to give a special thank to my family, my mother, father, grandma and brother, for their continuous support throughout the past years. I am very grateful for all that you have done for me and I would like to dedicate this work to you.

Marco Kalweit

Contents

List of Figures

Nomenclature

Units

The units which are used in the nomenclature have the following meanings:

M	mass
L	length
T	time
Θ	temperature
F	force
E	energy

Latin letters

Symbol	Unit	Quantity
a	$\mathrm{L/T}$	speed of sound
$\boldsymbol{a}$	$\mathrm{L/T^2}$	acceleration
a^i	$\mathrm{LT^{-1}}$	speed of sound of an ideal gas
a_l	L	lattice parameter
A		area; potential coefficient
Ac		acoustic number
B		potential coefficient
$\mathcal{A}$		general macroscopic variable
$c_{P(P)}$	$\mathrm{E/\Theta}$	isobaric heat capacity per particle
$c_{P(V)}$	$\mathrm{E/L^3\Theta}$	isobaric heat capacity per volume
$c_{V(P)}$	$\mathrm{E/\Theta}$	isochoric heat capacity per particle
$c_{V(V)}$	$\mathrm{E/L^3\Theta}$	isochoric heat capacity per volume
C_0, C_1		constant factors
C		potential coefficient
$\boldsymbol{d}$	L	atomic distance to a grid point
d_{bi}	L	distance of atom i to the surface of the boundary
d_{ij}	L	distance to which atom i and atoms j are constraint

D		potential coefficient, distance
E_p	E	potential energy
E_k	E	kinetic energy
E_t	E	total energy
E_p^{ext}	E	potential energy due to an external force
E_{trans}	E	transferred energy
e	E/L^3	energy density
e_e	E/L^3	external energy density
e_i	E/L^3	internal energy density
e_p	E/L^3	potential energy density
e_k	E/L^3	kinetic energy density
$e_{i,k}$	E/L^3	internal kinetic energy density
$e_{e,k}$	E/L^3	external kinetic energy density
f		force; correction factor
$\boldsymbol{f}_i$	F	force on atom i
$\boldsymbol{F}$	F	force
$F_{\alpha\beta}$		flux, where α is the flux quantity and the β way of transport
$F_{\alpha\beta}^{int}$		integrated flux, where α is the flux quantity and the β way of transport
$F_{\mathcal{A}}^{S}$	$[\mathcal{A}]/\mathrm{TL}^2$	surface flux of quantity $\mathcal{A}$
$F_{\mathcal{A}}^{V}$	$[\mathcal{A}]\mathrm{L}/\mathrm{T}$	volume flux of quantity $\mathcal{A}$
$F_{\langle\alpha\rangle_t}$		relative fluctuation of quantity α
$\boldsymbol{G}_i$	F	extra force to satisfy constraints
i		grid node index in the x dimension; atomic index; multi purpose index
j		grid node index in the y dimension; atomic index
k		grid node index in the z dimension,
k_B	E/Θ	Boltzmann's constant
k^{bend}		coefficient for bond bending potential
k^{sextic}		coefficient for bond bending potential
k^{xi}		coefficient for improper dihedral potential
$\mathcal{K}$	E	kinetic energy
$\boldsymbol{l}$	L	width of a grid cell
$\boldsymbol{l}$	ML^2T	angular momentum
l_α	L	width of a grid cell in dimension α
δl_b	L	width of the relaxation zone
δl_{ci}		half length of the confidence interval
$\mathcal{L}$		Lagrangian
$\boldsymbol{L}$	L	size of the computational box
$\boldsymbol{L}_C$	L	size of a subcell of the computational box
m	M	mass
m_i	M	mass of atom i
M		number of time steps for which the time averaging is taken
$\boldsymbol{M}$		number of cells in each dimension for cell subdivision
M_{sys}	M	mass of the entire molecular system
Ma		mach number
n	$1/\mathrm{L}^3$	number density

n_0	$^1/_{L^3}$	reference number density
$\boldsymbol{n}_b$	L	surface normal of the boundary
N		number, number of atoms
N_c		number of atoms in a grid cell
N_C		number of cells for cell subdivision
N_m		number of measurements
$N^{\alpha}_{m,min}$		minimum number of measurement of quantity α
N_τ		number of time steps
$N_{\tau,min}$		minimum number of time steps to run
$N_{\tau,run}$		number of time steps of a run
$\boldsymbol{p}$	MLT	momentum
p_i	MLT	generalised momentum of atom i
P	$^F/_{L^2}$	pressure
P_{cl}		confidence level
P_s	$^F/_{L^2}$	static pressure
q		generalised coordinate
$\boldsymbol{q}$		vector of generalised coordinates
Q	E	molecular energy flux
r	L	molecular separation
r_{add}	L	additional distance for building the neighbour list
r_c	L	cut-off distance
$\boldsymbol{r}_i$	L	position of atom i
r_{ij}	L	distance between atom i and atom j
r_0	L	reference distance for stretching potentials
$\boldsymbol{r}_{ij}$	L	vector from atom i to atom j
r_{neigh}	L	neighbour list radius
δr	L	distance
δrb_i	L	penetration of the atom i into the boundary force field
s		atomic index in cases where i is used elsewhere
δs	L	displacement for USHER scheme
δS	$^J/_K$	entropy difference
Δs	L	maximum displacement for USHER scheme
Δs_{ovlp}	L	overlapping distance for USHER scheme
t	T	time
δt		simulation time step
δt_m	T	measuring time step
δt_{av}	T	averaging time period
T	Θ	temperature
$\mathbf{T}$	$^M/_{LT^2}$	viscous stress tensor
T_D	Θ	desired temperature
T_e	Θ	temperature at y_e
u_α	$^L/_T$	flow velocity in dimension α
u, v, w	$^L/_T$	velocity components of $\boldsymbol{u}$
$\boldsymbol{u}$	$^L/_T$	flow velocity
u_e	$^L/_T$	flow velocity at y_e

$\boldsymbol{v}$	L/T	velocity
$\boldsymbol{v}_b$	L/T	velocity of the boundary atoms
$\boldsymbol{v}_i$	L/T	velocity of atom i
$v_{i\alpha}$	L/T	velocity of atom i in dimension α
$\boldsymbol{v}_{int}$	L/T	internal velocity
$\boldsymbol{v}_{int}^{sum}$	L/T	summed internal velocity
V	L^3	volme
V_c	L^3	volume of a grid cell
$\mathcal{V}$	E	potential energy
W		weighting function
We		Weber number
$\boldsymbol{x}$	L	position in space, displacement
$\boldsymbol{x}_{i,j,k}$	L	position of grid node i,j,k

Greek letters

Symbol	Unit	Quantity
α		angle, index for dimension x, y or z
β		angle, coefficient for bond bending potential
β_0		reference angle for bond bending potential
ϵ	E	energy coefficient for pair potentials
γ		ratio of the specific heats, surface tension
γ^i		ratio of the specific heats for an ideal gas
γ_V	$\mathrm{M}/\mathrm{LT}^2\Theta$	isothermal
δ_s		time steps between neighbor list updates
λ	$\mathrm{E}/\mathrm{LT}\Theta$	thermal conductivity
μ	M/LT	shear viscosity
$\boldsymbol{\omega}$	$1/\mathrm{T}$	angular velocity
$\boldsymbol{\Pi}$	M/LT^2	stress tensor
ϕ		dihedral angle
ρ	M/L^3	density
σ	L	distance coefficient for pair potentials
σ_k	L^2	kth constraint
σ_n		phase angle for dihedral potential
τ		simulation time step index
τ_c		correlation time in time steps
τ_m		measuring time step index
τ_{mes}		measuring sample rate, i.e. ratio between δt_m and δ_t
τ_f		flux imposing frequency
ξ		improper dihedral angle

ξ_0	reference angle for improper dihidral
ξ_{max}	termination criteria for USHER

Frame Decorators, Superscripts and Subscripts

$\langle \ldots \rangle$	general average
$\langle \ldots \rangle_t$	time average
$(.)^A$	denotes cluster A
$(.)_b$	denotes boundary property
$(.)^B$	denotes cluster B
$(.)^C$	denotes a cell C
$(.)^{ext}$	denotes external frame
$(.)_i$	denotes atom i
$(.)_{i,j,k}$	denotes anything at grid node i,j,k
$(.)_{ida}$	denotes an ideal quantity for removal of atoms
$(.)^{int}$	denotes internal frame; denotes integrated quantity
$(.)^n$	step n in an iterative scheme
$(.)_{pv}$	denotes a possible vector for ZNM scheme
$(.)^n$	step n in an iterative scheme
$(.)_S$	denotes a group of atoms, S
$(.)^{sum}$	denotes a cumulated quantity
$(.)_{sys}$	denotes the entire molecular system

CHAPTER 1

Introduction

Molecular modelling of meso- and nanoscale dynamics is one of the pillars of the expanding fields of mirco- and nanotechnology. Concerned with processes in the range of $0.1\,\mu$m - $100\,\mu$m, microtechnology has its origins in electrical engineering when the development of microscopic transistors and integrated electrical circuits paved the way into the era of digital computers. In the 1980s, microtechnology expanded to integrate mechanical elements, such as sensors or actuators, with electronics into micro-electromechanical systems (MEMS) [124]. This promises to have an impact on nearly every consumer product category through opening the possibility of complete systems-on-a-chip for applications such as biochemical sensors, acceleration detectors for airbags, inkjet printheads or fast switching micro-mirrors for electronic projectors and optical switching in telecommunications. However, the limit of minimisation is not reached at microscale; nanotechnology addresses processes and scientific problems at even small dimensions of 0.1 to 100 nm (1 nm = 10^{-9} m) [56]. With its roots going back to Richard Feynman's ideas in *There's Plenty of Room at the Bottom* [49], nanotechnology emerged as a topic (and technical term) in the 1970's [177] and in the late 1980's [56]. Likewise to microtechnology, the ability to systematically understand, organize and manipulate matter on the nanometer length scale can be considered as a broad definition of nanotechnology. Applications of the subject include: nano-membranes (including biological membranes), nano-crystalline materials; nano-layers with selective optical barriers; dispersions with optoelectronic properties; chemical and bio-detectors; advanced drug delivery systems; protein engineering; new generation of lasers; nano-electromechanical systems on a chip (NEMS); carbon nanotube products; nano-particle reinforced materials; thermal barriers; information recording layers; molecular sieves and filters; high hardness cutting tools and new bottom up fabrication techniques.

Computational modelling and numerical simulations are important for scientific progress and engineering work because they can often provide a more detailed picture of crucial physical processes than laboratory experiments do. At microscale and, in particular, at nanoscale, such computational experiments are indispensable because performing experiments at these dimensions and obtaining quantitative data is extremely difficult and expensive. This is mainly due to lack of suitable analysis tools or the limitations of available tools in measuring variables of interest and in picturing such small systems. Also, the fast time scales at which processes at micro- and nanoscale occur complicate the experimental procedures. Thus, it is difficult for the experimentalist to actually see what happens in a

system of these dimensions, in particular the dynamic behaviour.

For the reasons given above, numerical simulations are crucial to elucidate the physics and dynamics of micro- and nanoscale systems; substitute missing experimental abilities; act as a supplementary investigation tool; serve as a validation source and to give design suggestions or to serve as a virtual test environment for new NEMS and MEMS. Finally, numerical simulations are of importance to engineering science, as well as biology, medicine and pharmacy.

1.1 Two perspectives

Nanoscale is an area where two different approaches to understand matter meet. From the macroscopic (or mesoscopic) point of view, solids, fluids and gases are considered to form a continuum, which can be described by laws such as the Navier-Stokes equations. Space is thought to be filled continuously with matter that at any point, or infinitesimal small volume around the point, is in a local equilibrium. The properties of matter are sufficiently described by macroscopic and thermodynamic state variables such as density, velocity and energy, with changes in space and time being smooth. Approaching smaller length scales, one eventually enters the realm of nanoscale, where the continuum dissolves into discrete particles, i.e. atoms and molecules. The atoms are imagined as mass points that interact with each other through interatomic forces, in particular electrostatic force. The physical properties of matter are determined through the interplay of the atomic positions, velocities and their interactions with each other.

For both perspectives, mathematical models and computational methods have been developed to simulate the behaviour of fluids. Widely used continuum based methods are computational fluid dynamics (CFD) [10] or finite element methods (FEM) [18]. Molecular simulation methods comprise statistical methods, such as direct simulation Monte Carlo (DSMC) [20] or lattice Boltzmann equation (LBE) [29], and the deterministic molecular dynamics (MD) method [2, 43]. Quantum mechanical (QM) methods even include the simulation of the atomic electron cloud [82, 116]. Each of these simulation methods has its limitations and can only be used within specific length and time scales.

Theoretically, all scientific problems, including macroscopic ones, could be solved only using deterministic molecular simulation methods. One would simply compute the motion of all atoms of the system of interest for the respective period of time to obtain the atomic trajectories. Containing the complete information about the system for the computed period of time, the set of trajectories could be used to answer any possible question about the respective system. For instance, any desired macroscopic or thermodynamic variable such as temperature can be calculated, as these quantities represent statistical averages of the detailed molecular data.

Unfortunately, things are not quite that simple and the detailed information has a price. Molecular simulation methods are computationally extremely demanding. On modern

computers, the most accurate quantum mechanical methods will only allow the simulation of few hundred atom at maximum. The classical molecular dynamics method is much more economical and has become a standard tool for simulation at nanoscale. Still, the required computational resources limit its application to systems containing several million atoms. This poses an upper limit in terms of length scales on MD simulation, which is well below microscale, in the range of few nanometers.[1] Larger systems, such as the 320 billion atom simulation by Kadau [103], can only be carried out using massive parallel computers that are exclusively available to few high performance computer centres. Another limitation of MD is that computable time scales are in the range of nanoseconds, far too short for many dynamical problems whose characteristic times are in the range of microseconds or even several seconds.

In contrast, continuum based methods are several orders of magnitude faster and can be used over a broad range of length and time scales, even for atmospheric or oceanic currents. However, as one advances to smaller dimensions, the underlying assumptions of the continuum model cease to be valid. For fluid flow modelled by the Navier-Stokes equations these assumptions are:

The continuum condition: It requires all material properties to be defined at every point in space and to vary continuously and smoothly (Derivatives of all dependent variables exist within reasonable limits) [58, 141]. Material properties comprise kinematic properties (e.g. velocity and acceleration), thermodynamic properties (e.g. pressure and density) and transport quantities (e.g. viscosity and diffusivity). The continuum condition will only be satisfied if there are enough molecules within a sampling point (In reality the sampling point is a small volume fraction). If the considered dimensions, such as the grid spacing, are too small or the fluid is too dilute, the number of molecules in the sampling volume is insufficient and statistical fluctuations become significant. According to Bird [20], a sampling volume needs to contain more than 10,000 molecules to reduce statistical fluctuations below 1 %. Depending on the density, one can give an approximate minimum grid spacing above which the statistical fluctuations are acceptable.

Local thermodynamic equilibrium: The derivation of the Navier-Stokes equations assumes a nearly linear relation of stress to rate-of strain and of heat-flux to temperature-gradient. This only holds if the flow is locally at each point not too far from a thermodynamic equilibrium. To satisfy a thermodynamic equilibrium, the number of encounters (collisions) between molecules within a sample volume during a time period that is small compared to the smallest time scales changes of the flow must be sufficiently high [141].

In general, the continuum condition is mostly satisfied and it is the requirement for thermodynamic equilibrium that breaks down first [58]. For gas flows, the criteria to determine whether a flow can be treated as a continuum is the local Knudsen number $Kn = \lambda/h$, where λ is the molecular mean free path and h a characteristic length scale of the flow, such as

[1] For example, a cube of water at room temperature containing 100 million molecules has a side length of only 144 nm

the channel width [20]. For $Kn < 10^{-3}$ the Navier-Stokes equations are valid with no-slip boundary conditions, for $Kn > 10^{-3}$ slip must be taken into account and non-linear effects invalidate the Navier-Stokes equations for $Kn > 10^{-1}$. There is a transition regime for $10^{-1} < Kn < 10^{1}$ before free molecular flow ($10^{1} < Kn$). Similar effects of the breakdown of the continuum model occur in for the liquid phase. However, the situation is less clear, because of contrary experimental data and the lack of a kinetic theory as in the case of gases [58]. One situation that leads to a breakdown is when the shear rate exceeds approximately twice the molecular interaction frequency [141]. Apart from a general breakdown of the continuum model, there are numerous situations in which the continuum model is valid for the bulk material and only breaks down within certain regions that can not be described adequately by a continuum model. Typical examples comprise: complex flows near interfaces like wetting, drop formation, melting, crystal growth from a fluid phase, moving interfaces between immiscible fluids or biological membranes [41]; phase changes, such as solidification or fouling; crack propagation in solids; flows interacting with nanoscale structures such as carbon nanotubes [193]; chain molecules or biological molecules near interfaces, like polymers that are tethered to a hard wall [17]; singularities such as the moving contact line problem [155].

From the described limitations it is clear that continuum methods are only applicable for macroscopic problems and molecular simulation methods are only of practical use at microscopic length scales. This is sufficient for problems which are either entirely of macroscopic natures or whose dimensions are small enough to be simulated by MD. However, the nature of many scientific problems is intrinsically multiscale. This is when microscopic phenomena on atomic length scales manifest in macroscopic scales. Such problems can not be solved, either by MD, because the required computational effort is simply too large, or by CFD due to the lack of a valid continuum model to describe the important microscopic phenomena. Therefore, such problems are not accessible, either by continuum or by molecular simulations. This is illustrated in Figure 1.1, where the length and time scales covered by the continuum and molecular methods are indicated.

Multiscale methods can be used to bridge the gap in between atomistic and continuum scales. In principle, there are two major directions in multiscale modelling: methods for extending the atomistic length scales are referred to as *spatial* multiscale methods and methods for extending the time length scales as *temporal* multiscale methods. This work is only concerned with spatial multiscale methods.

Spatial multiscale methods exploit the fact that in many situations the atomic description is only required within small parts of the simulation domain while for the majority the continuum model is still valid (examples have been provided above). This provides an opportunity to perform concurrent continuum and molecular simulations for the respective parts of the simulation domain using a coupling scheme as connection between the continuum and molecular method. Calculating the majority of the domain by fast continuum solvers, which are several orders of magnitude faster than molecular simulation methods, can speed up the computation tremendously. For this reason, such methods, called *hybrid molecular-continuum* methods, can enable the simulation of problems that are not accessi-

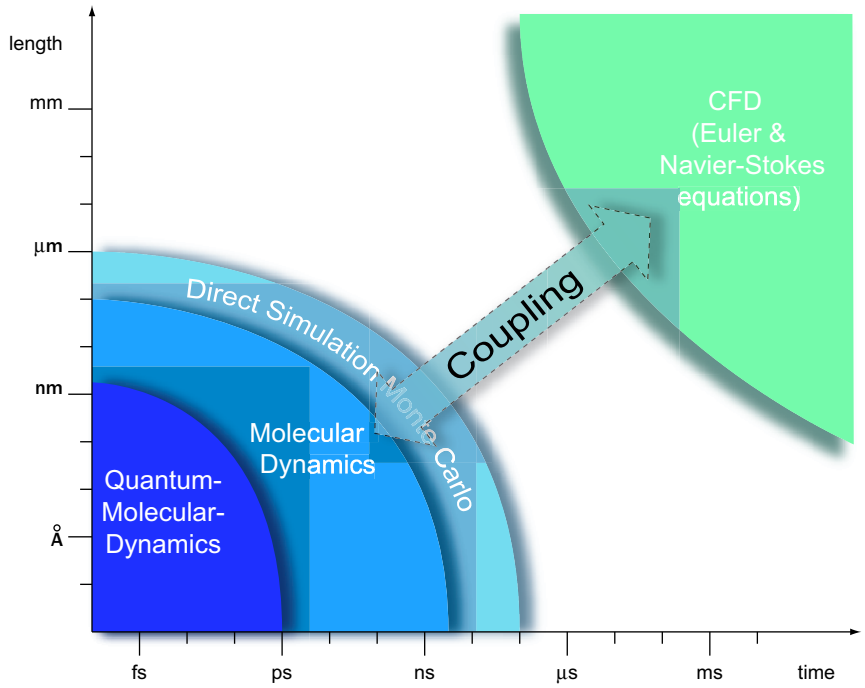


Figure 1.1: Limits of applicability of different computational approaches with respect to length and time scales.

ble either by continuum or by pure molecular simulation methods. Typical examples are the simulation of dynamic friction, which involves melting on the interface between two sliding block of materials [81]; fluid slippage past surfaces [165]; fracture propagation [33]; problems involving phase transitions [188]; fouling at surfaces; wetting; moving contact line between two immiscible liquids [110]; mass transport through membranes; systems with locally high stresses or strong gradients; flows in micro- or nano-channels with specific surface features; flows along surfaces with no ideal non-slip or slip condition or where the liquid molecules interact with wall molecules; simulation of micro- and nanoscale flow control devices (MEMS); transport processes in biological organism; and others. The range of applications indicates that hybrid molecular-continuum methods will have an impact in a number of fields, such as aeronautical engineering (micro-aerial vehicles, pumps and compressors for micro-propulsion), material sciences, micro-engineering (Fuel cells, microjets for ink jet printing, simulation of magnetic disk storage units, MEMS, sensors and actuators, lab-on-a-chips, nanotechnology, NEMS, DNA manipulation and transport, biology (simulation of proteins and cell membranes), medicine, pharmacy (drug delivery systems providing controlled release, biofluidic filters and separators, electroosmotic pumps, integrated microfluidic vents).

1.2 Hybrid molecular-continuum methods

Hybrid molecular-continuum methods have been investigated for different combinations of computational simulation methods. Major research directions are the coupling of

- molecular dynamics with finite element methods (MD-FEM),
- direct simulation Monte Carlo with computational fluid dynamics (DSMC-CFD) and
- molecular dynamics with computational fluid dynamics (MD-CFD).

This work is focused on MD-CFD hybrid methods. Nevertheless, a brief overview over the other two combinations is given below.

MD-FEM

Hybrid MD-FEM methods have been developed for the simulation of solids. Here the coupling between the continuum grid and the molecular domain is established by assigning atoms permanently to grid points. Following the early work of Kohlhoff and Schmauder [109], a number of different methods have been suggested for the construction of the transition region between the FEM mesh and atoms of the MD region. These include the quasicontinuum method, the CLS method, the FEA method, the fully non-local quasicontinuum method and the CADD method Curtin and Miller [34]. A topical review on these methods has been published by Curtin and Miller [34].

Rudd and Broughton [160] reformulated the continuum region by introducing a method called coarse-grain molecular dynamics (CGMD) in order to avoid difficulties in smooth transitions from atomistic to continuum in the presence of temperature effects. The bridging scale method to concurrently couple MD and FEM was proposed by Wagner and Liu [189]. The advantage of this method is that the FEM does not need to be graded down in spacing, which allows the usage of different time steps in the FEM domain than in the MD domain. The two-dimensional version was extended to three dimensions by Park et al. [146]. A recent summary on the method can be found in Reference [121].

Methods developed for MD-FEM coupling are only applicable to solids, where the positions of atoms relative to each other do not change significantly. In the case of fluids, the situation is more complicated because of the random atomic motion of the fluid particles, which causes continuous mixing through molecular diffusion. This seems one reason why there is only a relatively small number of studies dealing with the development of hybrid molecular-continuum methods for fluid flow.

DSMC-CFD

Initially, the development of hybrid methods for fluid flow has been focused on dilute gases, because these are easier to handle than dense fluids, where the boundary imposition is more complicated. The fastest molecular method for the simulation of dilute gas flows is DSMC,

developed by Bird [20]. As a statistical molecular method, it is of several orders of magnitude more efficient than MD, however remains less efficient than CFD methods. For DSMC-CFD hybrid methods the coupling is established through overlapping buffer cells. This is very similar to the function of MD-CFD hybrid methods, which is explained detail in the later chapters of this work. Wadworth and Erwin first demonstrated the function of DSMC-CFD hybrid schemes for a one-dimensional shock wave [186] and for a two-dimensional slit flow [187]. The schemes were developed further by Eggers and Beylich [44] and others. In a pioneering work, Garcia et al. [59] suggested the AMAR (Adaptive Mesh and Algorithm Refinement) method that uses a Navier-Stokes solver in conjunction with mesh refinement as a natural framework for the introduction of the DSMC simulation at a certain refinement level. A review on DSMC-CFD methods can be found in Reference [195].

MD-CFD

MD-CFD hybrid methods for dense fluids are probably the most challenging methods because of the difficulties involved in applying the boundary conditions onto the MD domain. At the same time, it is the scheme with potentially the broadest range of applications. While, DSMC-CFD methods can only be applied for the simulation of dilute gas flows and MD-FEM methods are restricted to solid state, a MD-CFD method may be used for liquids, solids (providing the respective equations of state for the solid material are available) and for dense gases.[2]

Similar to the other two hybrid methods, the coupling between the MD and CFD domain along the interface is established through overlapping cells which are used by a coupling scheme to join the two differently modelled domains to one single physical space. This part is the heart of a MD-CFD method and referred to as *hybrid solution interface* (HSI). The HSI prescribes the geometrical arrangement on the interface (for example, how far the domains overlap), and the protocol (or algorithm) of the coupling scheme. Essentially, all coupling schemes measure some quantities on the molecular and on the continuum side of the interface. Following a transformation into molecular or continuum formulation, these quantities are then applied as boundary conditions on the respective sites of the molecular and continuum domain. In a fully coupled scheme, the process is necessarily working in both directions, i.e. from continuum to molecular and vice versa.

On which type of quantity the coupling should be based on has been a subject of some debate. Some researchers advocate *coupling by state* [111, 195], where the exchanged quantities are state variables, such as pressure, velocities or temperatures. The alternative approach *coupling by fluxes*, preferred by others [39, 55], is based on the exchange of fluxes, i.e. mass, momentum and energy fluxes. While the state coupling approach is restricted to incompressible steady state problems, the flux coupling approach can be used for compressible unsteady flows as well. However, it is true that flux coupling methods are limited in the computable time span by the fact that they are only spatial multiscale methods for which the computable time span is still dictated by the molecular simulation

[2] MD simulations are most efficient for dense systems. The lower the density the less efficient the MD method becomes, compared to the number of simulated atoms.

method, i.e. MD [195].

The first state coupling approaches were only applicable to scenarios without mass or energy transfer across the interface. O'Connel and Thompson [142] simulated a one-dimensional incompressible Couette flow using a hybrid MD-CFD scheme for liquids based on an overlapping region between the CFD and MD domains for coupling of velocities using constrained Lagrangian dynamics.

Hadjiconstantinou and Patera [69] proposed two-dimensional state coupling scheme. To enable mass flow across the MD-CFD interface, the rectangular MD region is embedded into a bigger simulation box with periodic boundary conditions that serves as particle reservoir. A 'Maxwell Demon' is used to enforce the continuum velocities on the atoms at the MD-CFD interface. The Schwarz alternating scheme for the incompressible formulation was used as a time marching scheme to converge the solutions of the MD and CFD domains to a steady state. This hybrid method was first applied to Poiseuille and Couette flows of supercritical Argon [69]. Later, the method was extended to two species and applied to the moving contact line problem [71].

The state coupling hybrid MD-CFD method was further developed by Werder et al. [193]. They disposed the reservoir region and used specular walls and consistent body forces to confine the atoms within the molecular region. Particle insertion and removal algorithms was used to ensure constant density in the boundary region and to realise in and out flow of mass. The scheme was demonstrated on the flow of a Lennard-Jones fluid past a carbon nanotube.

Hybrid methods based on flux-coupling are the latest type of hybrid MD-CFD methods to be developed. The pioneering work was done by Flekkoy et al. [54] and Wagner et al. [188]. They developed a flux coupling method for incompressible flows. The scheme couples the mass and momentum fluxes on the interface between the CFD and MD domain. A reservoir region on the boundary of the MD region is used to implement the fluxes onto the molecular domain. Mass fluxes are implemented by the insertion or removal of atoms into or away from the reservoir region. The momentum fluxes are transferred through an external force that is acting on the atoms in the reservoir region. The method was tested for incompressible Couette and Poiseuille flows [53, 188].

Delgado-Buscalioni and Coveney [40] improved the flux coupling approach and extended it to compressible flows through incorporating the coupling of energy fluxes. This required the development of a search algorithm, referred to as USHER [39], that seeks insertion sites for atoms and molecules at a desired energy level. The MD flux boundary conditions were verified for longitudinal and transversal waves in an one-dimensional system [40]. The fully coupled scheme was then tested on oscillatory wall flows [38], though only for an incompressible isothermal fluid. The method was also applied to the simulation of a single tethered polymer in a solvent, subjected to oscillatory flow [17].

Later, the USHER algorithm was extended to incorporate the insertion of molecules such as water [37]. Improved flux boundary conditions for the MD domain were achieved by ensuring that the flux imposition fulfills the second law of thermodynamics and thus minimises the entropy production. Accounting for fluctuations of the state variables on the continuum side at nanoscale by introducing fluctuation terms into the continuum solver,

Fabritiis et al. [47] presented the respective MD-CFD flux coupling method which was tested on sound waves travelling in bulk water reflected by a lipid monolayer.

Reviews and discussions on the hybrid MD-CFD methods have been published by Wijesinghe and Hadjiconstantinou [195] and Koumoutsakos [111]. Additional information on the discussed methods can be found in [70, 194].

1.3 Alternative methods

Beside hybrid molecular-continuum methods, there are also other methods that may be used for the simulation of certain types of meso- and nanoscale flows. Most of them are based on the Boltzmann transport equation and can broadly be classified into pseudo particle methods and higher order moment methods. Generally, the validity of the Boltzmann transport equation for a specific material, a gas or liquid, depends on the employed collision integral. Thus any model derived from this equation can strictly only be applied within the validity envelope of the collision integral. In case of the classical Boltzmann transport equation, the collision integral is based on the kinetic gas theory and thus assumes binary collisions, molecular chaos, and point mass molecules. Hence, it is strictly only valid for gases and not for liquids.

Algorithms to solve the Boltzmann transport equation itself are very complicated to implement for general applications and are thus restricted to geometrically simple cases, such as channels [58]. Higher order moment methods can be derived from the Boltzmann equation. Unlike their lower order counterparts, the Euler equation (zeroth order) and Navier-Stokes equation (first order), these equations contain higher order terms to account for departures from equilibrium. They are also known as extended hydrodynamic equations (EHE) [1]. The most often used EHE is the Burnett equation, which was originally employed for rarified gas flows in the upper atmosphere, but has also been applied for microscale gas flows [122]. However, solving EHE is challenging and involves significant numerical and computational difficulties that restrict the application of EHE models. Furthermore, the EHE fail to capture some important microscopic phenomena, such as the Kundsen layer [123].

Pseudo particle methods include the dissipative particle dynamics (DPD) method [66], the direct simulation Monte Carlo (DSMC) method [20] and the lattice Boltzmann method (LBM) [29]. DPD is basically a coarse grained MD simulation. The fundamental particles are beads that comprise a number of atoms. All forces between particles smaller than the bead are assumed have averaged out and only the resulting forces between the beads are modelled. Three forces types are modelled: (i) a harmonic conservative force, (ii) a dissipative force representing the viscous drag (energy drain) and (iii) a random force (energy input). Originally, DSMC has been developed for high-speed rarified gas flows. It is not as effective for the simulation gas flows at micro- and nanoscale, because it suffers from slow convergence and large statistical noise under these conditions [58]. Most importantly, the

DSMC is limited to the simulation of rarified gas dynamics and can not be used for dense gases and liquids. The LBM has a relatively simple algorithm that moves pseudo particles along a lattice. Two particles experience a collision when moving on the same lattice point. The method has be applied for rarified and continuum gas flows. It can also be used in the incompressible limit and for liquid flow. However, it is a relatively new method and all implementations known to the author employ a collision model that is not valid for liquids; thus the validity of the LBM for liquid flow is still questionable.

The methods described above are based on the simulation of average values that comprise the individual properties of a more or less large number of atoms. That makes these methods unsuitable for all situations in which the behaviour can only be modelled appropriately by taking into account the interactions between individual atoms, such as it occurs on interfaces, where several species are present or in the examples for continuum breakdown given in the previous section. Therefore, these methods do not substitute the need for hybrid molecular-continuum methods.

1.4 Scope and objectives

The most complicated part of any hybrid MD-CFD method is the realisation of the MD side of the hybrid solution interface, in particular the imposition of fluxes or states as MD boundary conditions [40]. This is due to the fundamental difference in degrees of freedom between the continuum and molecular formulation. The missing microscopic information, i.e. atomic positions, velocities and/or forces, need to be reconstructed or reinvented based on few macroscopic variables. Therefore, the design and implementation of the MD boundary conditions is the most crucial step in the development of any hybrid MD-CFD method. Moreover, a functioning scheme to impose MD flux boundary conditions is not only the most important part of the HSI for MD-CFD coupling, but can also be used for the realisation of boundary conditions in pure MD simulations. For those reasons, the main focus of this PhD is directed towards the analysis, design and development of MD flux boundary conditions for hybrid MD-CFD methods.

Despite the great potential of hybrid MD-CFD methods, this field of research is still in its infancy. Only few methods, by a limited number of research groups, have been published and many open questions remain. So far, fully coupled MD-CFD simulations have only been performed for incompressible flows, even if MD flux boundary conditions that include the energy flux imposition, which is required for compressible flows, have been proposed [40]. The knowledge about the most crucial part, the MD boundary conditions for MD-CFD coupling, is still very limited. Little is known about different numerical designs and their stability, accuracy, efficiency and physical correctness. To shed some light on this issue, one purpose of this work is to give a structured overview over different flux imposition schemes for the individual flux (fluxes by convection, by stress) and the combinations of those.

To implement the momentum fluxes (these arise for example from static pressure) in

MD boundaries, previous flux-coupling schemes use an external force to transfer the momentum to the boundary atoms. The external force creates a relaxation zone at MD boundary of the MD-CFD interface in which the density falls from its bulk value to zero. However, this approach has a number of undesired side effects, which, for example, cause stability problems. To overcome these issues, a detailed analysis on the implementation of the momentum flux by an external force is carried out in this work; alternative schemes are suggested.

Based on the above, the aims and objectives of this PhD can be given on a more detailed level:

- To give an overview of, and analyse, different design possibilities of the HSI for hybrid MD-CFD methods as spatial multiscale simulations.
- To carry out a detailed investigation on MD boundary conditions and the required flux coupling schemes, including:
    - Analysis of existing schemes for flux imposition on the MD boundary;
    - Development of alternative schemes to overcome the deficiencies of the existing schemes;
    - Analysis and comparison of the existing and the alternative flux imposition schemes by performing MD simulations to subject these to standard flow situation in order to:
        - verify them for standard flow situations and explore their stability, accuracy, physical correctness and efficiency;
        - find their application range in terms of velocity and material phase, i.e. gas, liquid or solid;
        - explore the resulting characteristics of the molecular system on the boundary, for example profiles of state variables across the boundary;
        - identify the most suitable flux imposition schemes for usage within the HSI.

Hybrid MD-CFD coupling schemes pose a number of special requirements on the employed MD code.[3] Traditionally, MD codes simulate a constant number of atoms. For the purpose of an MD-CFD coupling scheme, where an inflow or outflow of mass may occur, the number of atoms is typically not constant. This excludes most available open source MD codes for coupling with a CFD solver. Thus, one important object was to find a suitable MD code that can accommodate MD flux boundary conditions and to implement those into the MD code.

1.5 Publications

Whist writing the thesis, a book chapter and several journal papers have been written and submitted:

[3] By code, the source code of a software is understood. In the case of an MD code the software can be used to perform MD simulations.

- D. Drikakis and M. Kalweit, *Computational Modelling of Flow and Mass Transport Processes in Nanotechnology*, Handbook of Theoretical and Computational Nanotechnology (eds. M. Rieth, W. Schommers), American Scientific Publishers, 2006.
- M. Kalweit and D. Drikakis, *Molecular Dynamics of Colliding Nanoclusters*, J. Comput. Theo. Nano Sci., pp. 367 – 377, 2004.
- M. Kalweit and Dimitris Drikakis, *Collision dynamics of nanoscale Lennard-Jones clusters*, Phys. Rev. B 74, 235415, 2006.
- M. Kalweit, D. Drikakis, *Molecular-continuum computational models for meso and nanoscale flows and materials*, Proceedings of the Institution of Mechanical Engineers, Part C, Journal of Mechanical Engineering Science, 2007 (under review).
- M. Kalweit, D. Drikakis, *Multiscale Methods for Micro/Nano Flows and Materials*, J. Comput. Theo. Nano Sci., 2007 (accepted).

Additionally, a conference paper and a poster have been written and presented:

- M. Kalweit and D. Drikakis, *Computational Nanoclusters*, in Proceedings of the 4th European Congress on Computational Methods in Applied Sciences and Engineering, ECCOMAS, Jyvaeskyla, 2004.
- M. Kalweit and D. Drikakis. *Computational Nanotechnology*, poster at Symposia at Shrivenham 21 – 22 February 2005, Shrivenham, 2005.

1.6 Outline of this work

To complete the introduction, the reader is provided with an overview of the thesis including a brief description of the content of the individual parts and chapters. The thesis is organised in eight chapters: (1) Introduction; (2) Models and computational methods; (3) Calculation of macroscopic quantities; (4) The molecular dynamics code and its validation; (5) Hybrid MD-CFD methods; (6) Test cases; (7) MD flux boundary conditions; (8) Conclusions. Beside introduction and conclusion, the thesis can be split into two parts. The first comprises of chapters 2 – 4 and introduces the relevant computational simulations techniques, underlying models, description of the employed codes and their validation. The second part, constituting of chapters 5 – 7, describes the investigation of molecular-continuum multiscale simulation methods for geometrical decomposition. The focus is on the construction of the molecular dynamics boundary conditions that are the key part of any MD-CFD coupling schemes. The following paragraphs give a brief description of the content of the individual chapters.

The second chapter, 'Models and computational methods', describes the models and computational methods at nanoscale that are being used within this work. The chapter is divided into two parts; the first is concerned with molecular models and the molecular dynamics simulation method, the second with the continuum.

The purpose of the third chapter is to introduce the formulas and schemes used for the calculation of macroscopic quantities from the molecular description. It also describes

relevant statistical considerations, including estimates of the expected fluctuations of the calculated quantities.

Chapter 4 introduces LAMMPS [148], the MD code which has been used in this work. The description covers the modules of the code that have been implemented during the work. The remaining sections provide a description of the performed validation cases, which have been carried out to ensure the correct functioning of LAMMPS.

The fifth chapter introduces multiscale MD-CFD methods based on geometrical decomposition. Several different approaches and designs for constructing the interface between the molecular and continuum domain (HSI) are discussed.

In chapter 6, the test cases used for the investigation of the molecular dynamics flux boundary conditions are described. Finally, chapter 7 contains a detailed description and analysis of the molecular dynamics flux boundary conditions using a number of different schemes that have been investigated based on the test cases described in chapter 6.

CHAPTER 2

Models and Computational Methods

This chapter introduces models and computational methods for numerical simulations at nanoscale. The first part focuses on the classical molecular model. The underlying assumptions, equations of motion, interaction models and force fields are described. This is followed by an explanation of the molecular dynamics method, including different time integration schemes, force and energy calculation algorithms, initialisation procedures and important techniques which have been employed in this work.

The second part of the chapter is concerned with a description of the model of flow and mass transport processes in the context of continuum mechanics. The origins and approximations of the continuum equations are presented. For more detailed information on this topic, the reader is recommended to consult one of the standard books on CFD, such as References [10, 43, 184].

2.1 Classical molecular model

The classical molecular model has its origin in the traditional visual representation of molecular structures. Chemists and physicists used balls connected by sticks to model molecules in order to visualize the molecular three-dimensional structure. The model considers atoms to be the smallest elements, which can form bigger structures, such as molecules or crystals. Starting from this general description, the classical molecular model relies on the following approximations:

Atoms as point masses: Atoms are modelled as point masses at the position of the nucleus, where 99.9 percent of the atom's mass is concentrated. In quantum mechanics, this is known as the 'Born Oppenheimer approximation' [16] which separates the movement of the nucleus and the electrons. The classical model considers the rapid movement of the electron cloud to be averaged out.

Interaction via potential: The interaction of atoms is modelled by a potential $\mathcal{V}$. All forces, whether intra- or inter-molecular, classical or non-classical have to be mod-

elled by potential functions[1] which depend on the atomic positions only. Having their origin in the interaction of sub-atomic particles, the exact potentials are only approximated by analytic functions in most of the cases. The shape of these functions imitates the characteristics of the corresponding real-world forces. The accuracy of the simulation results is mainly determined by the accuracy of these approximations.

Modelling of bonds: According to the picture of the classical bond-stick model, a molecule is considered to be a collection of atoms kept together by rigid or semi-rigid bonds. The reality is, however, different as a molecular bond is the result of the electron density distribution between the atoms and the forces caused by the interaction of the electrons and the protons (in the nucleus) which keep molecules together.

Bond forming intra-molecular forces are, at least, an order of magnitude greater than inter-molecular forces. Compared to the molecular motion, vibrations of molecular bonds are extremely rapid and their amplitude is very small. This justifies the use of rigid or semi-rigid bond models in many cases within the regimes of normal temperatures [153].

Bonds can be subject to different types of deformations or movements corresponding to the internal degrees of freedom:

- Stretching and compression (*bond length*);
- bending of two bonds relative to each other (*bond angle*);
- torsion of a bond by rotating the neighbouring two bonds towards each other (*torsion or dihedral angle*).

Molecular bonds can be modelled at different degrees of approximations. In principle one can:

- Freeze the movement, which means a certain property is kept fixed, for example, the bond length;
- define a potential function that determines the bond forces;
- impose no restriction at all. This is not possible for the bond length, because it would mean that there is actually no bond;

It is reasonable to model very small compact molecules as rigid structures with complete unmovable bonds. However, for large molecules, especially chain molecules such as proteins where the shape of the molecule is changing in time, fixed bonds would be highly unrealistic. In this case it is essential to use parameterised potential functions.

[1] An exception that consists of a further simplification is the case of fixed bonds. Fixed bonds are modelled by constraints that are inserted into the molecular equations of motion.

Classical movement: The foundation of quantum mechanics, the uncertainty principle, was formulated by Heisenberg [89]. It relates the uncertainty of a particle's momentum δp_x to the uncertainty of it's position δx:

$$\delta x \cdot \delta p_x \geq \frac{\hbar}{2} , \tag{2.1.1}$$

where $\hbar = h/2\pi$ with h as Plank's constant. The direct consequence of Equation (2.1.1) is that the momentum and position of a particle can never be exactly known.

The classical molecular model disregards the uncertainty principle. The canonically conjugate variables position and momentum of a particle are precisely known at any point in time. With this approximation, the motion of molecules and atoms can be described by the classical equations of motion.

The aforementioned constraints may limit the application of the molecular model, but they provide the advantageous possibility of describing the complete system through the equations of classical mechanics. These equations can be solved numerically with much less effort than using the quantum mechanical equations of the system. By integrating the equations in time, one can calculate, for example, the detailed trajectory of a particle, the conformation of a chain molecule or macroscopic properties, such as temperature. In fact, molecular models comply with the view of the physical world in the late nineteenth-century. However, even though the model is not accurate in quantum mechanical terms, within its validated limits it is an outstanding useful model and an effective description of the nanoscale world.

2.1.1 Simulation types of molecules

Taking into account the considerations described above one can divide molecular models into different groups:

Single atoms: They are considered structureless and spherical symmetrical. A typical example is Argon, which may be the most often simulated element in the early years of MD [2]. Because the movement of monatomic molecules is purely translational, Newton's second law and a simple potential function are sufficient to compute their motion.

Rigid molecules: For small compact molecules like water, it is often reasonable to consider them as rigid structures that do not change their shape. That means that all bonds are frozen and the consistent atoms do not change their position relative to each other. The motion can be translational as well as rotational and any applied force will act on the molecules as a whole. The Newton-Euler equations are the governing equations of motion which are not discussed here. The greatest advantage of rigid molecules is that the time step does not need to be adapted to the rapid bond vibrations, thus allowing the use of a relatively large time step [153].

Flexible molecules: In contrast to rigid molecules, the atoms of flexible molecules are restricted as little as possible with respect to the movement towards each other. The only necessary restriction is to keep the distance between bonded atoms constant. This can be realized either by a soft-potential or by imposing constraints that keep bonded atoms at a specified distance. For the former one needs only Newton's second law, but for the latter the Lagrangian equations with constraints are necessary to describe the molecules' motion. Flexible molecules are mainly used to simulate polymers especially one of chain-like shapes [153]. Sometimes, the aligning hydrogen atoms on a carbon atom are thought to be absorbed and are treated as part of the carbon atom.[2]

Molecules with potential based bonds: The most realistic models involve the use of molecules where all bond movements are governed by potential functions representing the underlying quantum-mechanical effects that create the binding and restricting intra-molecular forces. Such models are often used for the simulation of large molecules, like proteins, to investigate their conformation where the bond-movement are extremely important. There are cases in which it is sufficient to consider some of the bond parameters, for example length, as constant where beside Newton's second law the constrained Lagrangian equation of motion should be applied.

2.1.2 Equations of motion for atomic systems

One of the primary application of the classical molecular model is to simulate how the system progresses in time. The aim is to generate atomic trajectories which can be analyzed and compared with experimental data. In order to advance the system in time, one needs to integrate the system's equations of motion. The following section gives an overview of the basic equations for different types of classical molecular systems. In the simplest case, for a system containing only single atoms, Newton's second law is required. The simulation of rigid molecules requires the use of the Euler equations which may be best expressed by Hamilton's quaternion. For flexible molecules which are subject to internal constraints, the Lagrange description is employed [153]. Introduction into the mathematical background can be found in classical mechanics textbooks (for example, Goldstein et al. [62] provides a comprehensive discussion of the subject).

From the classical point of view the state of a conservative system[3] of particles (atoms) interacting via a potential $\mathcal{V}$, can be described by a set of N independent generalized coordinates[4], q_i, and velocities, $\dot{q}_i$, [62]. The governing equation of motion may be expressed in various ways. A fundamental form is the Lagrangian equation of motion

$$\frac{\mathrm{d}}{\mathrm{d}t}\left(\frac{\partial \mathcal{L}}{\partial \dot{q}_i}\right) - \frac{\partial \mathcal{L}}{\partial q_i} = 0 \ , \quad i = 1, ..., N \ . \tag{2.1.2}$$

[2]This avoids the use of a mixed, rigid and flexible structure.

[3]A conservative system is a system in which all forces (except forces caused by constraints) are derived from a potential energy function $\mathcal{V}$ and the potential energy function is an explicit function of positions only [62].

[4]Generalized coordinates can have different physical meanings, such as length or angle.

The Lagrangian $\mathcal{L}$ is a function of q_i and $\dot{q}_i$, i.e. $\mathcal{L} = \mathcal{L}(\boldsymbol{q}, \dot{\boldsymbol{q}})$, defined by the kinetic ($\mathcal{K}$) and potential ($\mathcal{V}$) energies as

$$\mathcal{L} = \mathcal{K} - \mathcal{V} \, . \tag{2.1.3}$$

Hamilton's principle is a fundamental principle for the motion of monogenic mechanical systems[5] and holds independent of the specific choice of coordinates. It states that the actual path of the system in time is distinguished from all the other possible paths by the fact that the line integral (action integral)

$$S = \int \mathcal{L} \, \mathrm{d}t \tag{2.1.4}$$

is an extremum. It is possible to prove that Lagrange's equation (2.1.2) follows from Hamilton's principle [62].

Newton's second law which governs the motion of a monatomic system can be derived from the Lagrangian equation of motion. Considering $\boldsymbol{r}_i$ to be the Cartesian coordinate of an atom and using the kinetic energy

$$\mathcal{K}(\dot{\boldsymbol{r}}_i) = \sum_i \frac{1}{2} m_i \dot{\boldsymbol{r}}_i^2 \, , \tag{2.1.5}$$

the Lagrangian $\mathcal{L}$ is written as

$$\mathcal{L} = \sum_i \frac{1}{2} m_i \dot{\boldsymbol{r}}_i^2 - \sum_i \mathcal{V}(\boldsymbol{r}_i) \, . \tag{2.1.6}$$

For a single atom i Equation (2.1.2) reduces to Newton's equation of motion

$$m_i \ddot{\boldsymbol{r}}_i = \boldsymbol{f}_i \, , \tag{2.1.7}$$

where m_i is the mass and

$$\boldsymbol{f}_i = \nabla_{\boldsymbol{r}_i} \mathcal{L} = -\nabla_{\boldsymbol{r}_i} \mathcal{V} \tag{2.1.8}$$

the force on atom i. For a system of N atoms one obtains N Equations (2.1.7), which have to be solved numerically.

The fundamental equations of motion can also be written in the form of Hamilton's equations. The generalized velocities $\dot{q}_i$ are replaced by generalized momentums[6], p_i, defined by

$$p_i = \frac{\partial \mathcal{L}}{\partial \dot{q}_i} \, , \quad i = 1, .., N \, . \tag{2.1.9}$$

The variables $\dot{q}_i$ and p_i are called *canonical variables*. Eliminating $\dot{q}_i$ with the aid of p_i, the Hamiltonian $\mathcal{H}$ is defined by

$$\mathcal{H}(q_i, p_i, t) = \sum_{i=1}^{N} \dot{q}_i p_i - \mathcal{L}(q_i, \dot{q}_i, t) \, , \quad i = 1, .., N \, . \tag{2.1.10}$$

[5] A monogenic system is a system in which all forces (except forces caused by constraints) are derived from a generalized scalar potential. The potential may be a function of time, position or velocity.

[6] The terms *conjugate momentum* or *canonical momentum* are also often used for p_i.

The Hamilton $\mathcal{H}$ depends on the generalized co-ordinates, on the conjugate momentums, p_i, and on the time, t. If $\mathcal{H}$ is time-independent ($\dot{\mathcal{H}} = 0$) then it represents the total energy (sum of kinetic and potential energy) of the system [62].

The Hamilton equations of motion are the time derivatives of generalized co-ordinates and momentums, that is,

$$\dot{q}_i = \frac{\partial \mathcal{H}}{\partial p_i} \quad \text{and} \quad \dot{p}_i = -\frac{\partial \mathcal{H}}{\partial q_i} \; . \tag{2.1.11}$$

For Cartesian coordinates and potential functions independent of velocities and time Equation (2.1.11) yields

$$\dot{\boldsymbol{r}}_i = \frac{\boldsymbol{p}_i}{m_i} \; , \tag{2.1.12}$$

$$\dot{\boldsymbol{p}}_i = -\nabla_{\boldsymbol{r}_i} \mathcal{V} = \boldsymbol{f}_i \; . \tag{2.1.13}$$

Computing the trajectories of a N-atomic system requires to solve either a system of $3N$ second-order differential Equations (2.1.7) or a set of $6N$ first-order differential Equations (2.1.12) and (2.1.13).

2.1.3 Equation of motion with constraints

There are cases in which it is desirable to keep certain bond properties constant in order to reduce the computational effort. Such restrictions are called constraints and appear as extra forces which do not derive from the potential, $\mathcal{V}$, in the equations of motion. The most basic type is a fixed bond length, which is often applied because the vibrational changes in the bond length are much faster than the changes of the bond or dihedral angle. A constraint fixing the distance between two atoms can be expressed by the condition [62]:

$$|\boldsymbol{r}_i - \boldsymbol{r}_j|^2 = {d_{ij}}^2 \; , \tag{2.1.14}$$

where i and j are the two atoms whose positions against each other are fixed. Using combinations of distance constraints one can fix the bond angle or dihedral angle where other restrictions like planarity of several atoms are handled in a different way. Given a total of n_c distance constraints applied on a molecule, these can be summarized as a set of equations [153]:

$$\sigma_k = \boldsymbol{r}^2_{i(k)\,j(k)} - {d_k}^2 = 0 \; , \; k = 1, ..., n_c \; , \tag{2.1.15}$$

where the kth constraint enforces atoms $i(k)$ and $j(k)$ at a distance d_k. Using this constraint, the Lagrangian equation of motion for atom i is given by

$$m_i \ddot{\boldsymbol{r}}_i = \boldsymbol{f}_i + \boldsymbol{G}_i \; . \tag{2.1.16}$$

This equation is identical to Newton's second law except $\boldsymbol{G}_i$ which is the extra force term caused by the constraints to keep the desired bond lengths constant. It is defined by

$$\boldsymbol{G}_i = -\sum_{k \epsilon C(i)} \lambda_k \nabla_i \sigma_k \; , \tag{2.1.17}$$

and stands for the set of constraints involving atom i. The parameters λ_k are the undetermined Lagrangian multipliers.

2.1.4 Potential energy surface (PES)

The term potential energy surface (PES) is derived from the visualization of a system's potential energy function. This function describes the potential energy $\mathcal{V}$ of a classical molecular system in terms of the atomic positions, a highly simplified assumption. In Figure 2.1, the potential energy function is displayed for a two-dimensional system as height over each point in the systems geometrical configuration, $\boldsymbol{q}$. The resulting surface is similar to a landscape in the real world where the potential energy corresponds to the the height over a datum plane.

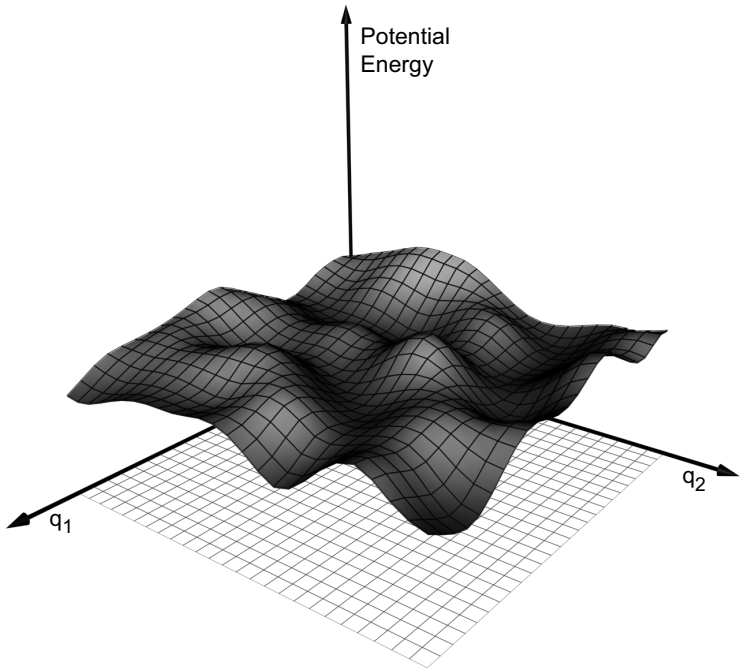


Figure 2.1: Potential energy surface for a two-dimensional system.

Because within the classical approximation the molecular motion is determined by forces, the system can be imagined as sliding over the frictionless PES subject to a force that is equal to the negative gradient of the potential energy function:

$$\mathbf{F} = -\nabla\mathcal{V}(\boldsymbol{q}) \; . \tag{2.1.18}$$

Significant features of the landscape are local potential minima (potential wells) and passes between them denoted as *cols* [42]. For example, a stiff and stable molecular structure resides in a well with steep walls. There may be accessible low *cols* to other wells by which the system can transit to another stable state provided that enough energy is available to

allow the system to reach the col.

To compute the potential energy, one needs a so called 'force field' which is built up from two components [182]:

- A set of equations which forms the potential function $\mathcal{V}$ and can be used to compute the potential energies and forces.
- The parameters used in the equation for every combinations of atomic types in the force field.

It is mainly the accuracy of the force field that determines the quality of the simulation results. This depends on how the form of the potential function coincides with the realistic behaviour, as well as on the quality of the parameters used in the equations. The parameters can be obtained either by experimental data or by ab initio calculations. The terms of the potential energy function can be classified into three groups:

- **Non-bonded interactions:** These are inter-molecular interactions potentials caused by the van der Waals forces and electro-magnetic Coulomb forces.
- **Bonded interactions:** These are intra-molecular interaction potentials of bond-stretching, angle-bending, improper dihedral and proper dihedral.
- **Special contributions:** This group stands for all contributions to the total potential that do not arise from physical interaction of atoms or pairs of atoms. They may be used to enforce boundary conditions or to incorporate forces that model other effects of the simulation.

The potential function $\mathcal{V}$ can be written as

$$\mathcal{V} = \mathcal{V}^{Non-bonded} + \mathcal{V}^{Bonded} + \mathcal{V}^{Special} \,, \tag{2.1.19}$$

where $\mathcal{V}^{Non-bonded}$ and $\mathcal{V}^{Bonded}$ are the potential functions corresponding to the non-bonded and bonded interactions, respectively. $\mathcal{V}^{Special}$ comprises all special contribution to the potential.

2.1.5 Non-bonded interactions

The non-bonded (intermolecular) interactions originally known as van der Waals forces encompass:

- Coulomb interaction between charged particles like ions,
- dipole-dipole interactions by permanent dipoles[7] and
- London dispersion forces[8] caused by temporarily fluctuating dipoles.

[7] Permanent dipoles are caused through different strong electronegativity of atoms inside a molecule.

[8] Today, the term 'van der Waals forces' is often used only for London dispersion forces. It was originally an expression for all inter-molecular forces.

Dipole-dipole interactions can be modelled by artificial charges inside molecules and are treated like Coulomb interaction potentials. Thus, the non-bonded potential is the sum of

$$\mathcal{V}^{Non-Bonded} = \sum_i \sum_{j>i} \mathcal{V}_{ij}^{Col} + \sum_i \sum_{j>i} \mathcal{V}_{ij}^{vdW} + \sum_i \sum_{j>i} \sum_{k>j} \mathcal{V}_{ijk}^{vdW} \ , \tag{2.1.20}$$

where the indicies i, j and k are running over all interactions sites, i.e., atoms and artificial charged points [182]. $\mathcal{V}_{ij}^{Col}$ is the Coulomb potential and $\mathcal{V}_{ij}^{vdW}$ the dispersion pair potential. This last term $\mathcal{V}_{ijk}^{vdW}$ represents the three-body interactions which despite significant contributions[9] are rarely included. The reason is the extreme high computational costs that involved to calculate them. However, the effect of the three-body interactions can be incorporated into the pair potential by defining a so called *effective* pair potential [2].

Coulomb interaction:

Caused by the attraction and repulsion forces between charged particles i and j the Coulomb interaction is displayed in Figure 2.2 and given by [112]

$$\mathcal{V}^{Col}(r_{ij}) = \frac{q_i q_j}{4\pi\epsilon_0 r_{ij}} \ , \tag{2.1.21}$$

where ϵ_p is the permittivity of free space[10] and r_{ij} is the distance between particles i and j. The force derived from this potential is [182]:

$$\boldsymbol{f}_i(\boldsymbol{r}_{ij}) = \frac{q_i q_j}{4\pi\epsilon_0 r_{ij}^2} \frac{\boldsymbol{r}_{ij}}{r_{ij}} \ . \tag{2.1.22}$$

Dispersion interaction:

Owing to the sensitivity of this interaction to different surrounding conditions various models have been developed. The models itself differ in terms of accuracy, application areas and computational cost.

Lennard-Jones potential:

The probably best known is the simple Lennard-Jones 12-6 Potential[11] [126, 130] originally proposed for liquid Argon. The potential energy of a pair of atoms i and j is

$$\mathcal{V}^{LJ}(r_{ij}) = 4\epsilon \left[\left(\frac{\sigma}{r_{ij}} \right)^{12} - \left(\frac{\sigma}{r_{ij}} \right)^{6} \right] \ , \tag{2.1.23}$$

[9] For Argon, the contribution of three-body interactions has been found to be about 10 percent of the lattice energy. For more polarisable species the contribution is expected to be even higher.

[10] Note, that for dense systems ϵ_p has to be adapted.

[11] The expression '12-6' specifies the exponents of the repulsive and the attractive terms in the potential equation. Beside the common 12-6 version, other combination are possible.

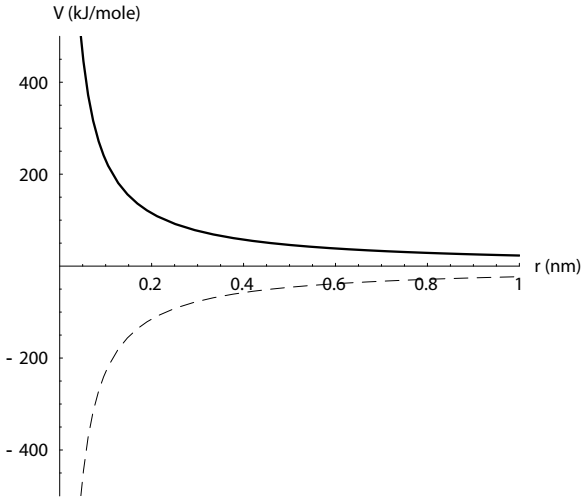


Figure 2.2: Coulomb potential: The solid and dashed lines show potential values for particles with equal and opposite signed charges, respectively.

where $r_{ij} = |\boldsymbol{r}_i - \boldsymbol{r}_j|$ is the distance between the two atoms. The parameter ϵ defines the strength of the interaction and σ is the length scale. As can be seen in Figure 2.3, the potential is characterized by a strong repulsion arising from the overlapping of the electron clouds and the attractive tail representing the dispersion interactions.

By choosing the appropriate parameters, the LJ-potential can be used to model a variety of elements, including Argon, Helium, Carbon, Nitrogen, Oxygen with varying accuracy. For systems with a single species modelled by the LJ-potential, early MD simulations non-dimensionalised Equation (2.1.23) to reduce the computational effort through eliminating the parameters σ and ϵ. In that case, all values in the simulation are in reduced LJ-units. The respective relations are shown later in Section 6.5 (q.v. Table 6.1). The argument of reduced computational effort is almost obsolete for nowadays computers and complex simulations codes. However, results are still given in LJ-units so that they can easily be converted to any material for which LJ-parameters exist.

Buckingham potential:

The Buckingham potential's repulsive term is more realistic than the Lennard-Jones potential [3]. This is, however, associated with more expensive computing costs. The interaction is calculated by:

$$\mathcal{V}^{Bh}(r_{ij}) = A_{ij} \exp\left(-B_{ij} r_{ij}\right) - \frac{C_{ij}}{r_{ij}^6} \,, \tag{2.1.24}$$

where A_{ij}, B_{ij} and C_{ij} are parameters similar to the Lennard-Jones potential which have to be carefully specified for each combination of atoms i and j.

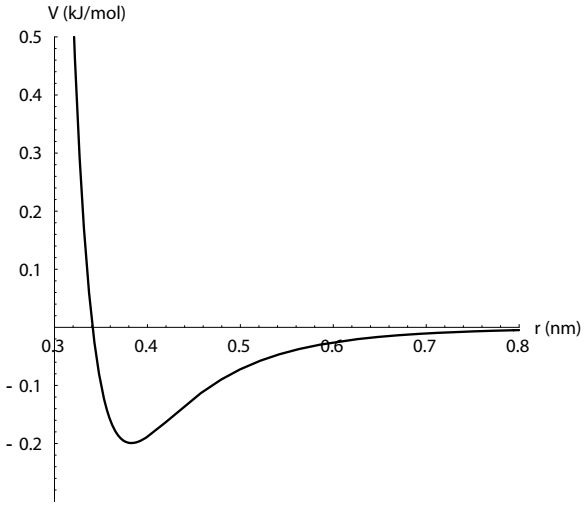


Figure 2.3: Lennard-Jones potential.

Other dispersion potentials:

For some purposes it may be sufficient to use especially simple and highly idealized potential forms like the soft-sphere potential [2]

$$\mathcal{V}^{SS}(r_{ij}) = \epsilon(\sigma/r_{ij})^{\nu} = a r_{ij}^{-\nu} \ , \tag{2.1.25}$$

where ν determines the hardness of the spheres and ϵ (as well as a) is the strength of the interaction.

2.1.6 Special potentials

Harmonic potential

Any boundary condition which is non-periodic imposes some restriction to keep the atoms within the simulation domain. In cases where the boundary consists of solid material, the system can be confined by preventing the boundary atoms, i e. atoms near the boundary, to move away from their initial position. The easiest way to achieve this is freezing them, which will prevent them from moving at all. A better solution is to assign a spring force to each boundary atom that pulls it to its original position. The advantage is that these atoms still vibrate and therefore the wall can be used as a heat sink or source. The spring potential for an atom i is defined as

$$\mathcal{V}^{Spring}(\boldsymbol{r}_i) = \frac{1}{2} k(\boldsymbol{r}_i - \boldsymbol{r}_{0i})^2 \ , \tag{2.1.26}$$

where $\boldsymbol{r}_i$ is the current position of the atom i; its initial position is $\boldsymbol{r}_{0i}$. The spring constant k defines the magnitude of the force acting on the atoms. Based on Equation (2.1.26), the force is given as

$$\boldsymbol{f}_i = k(\boldsymbol{r}_i - \boldsymbol{r}_{0i}) \ . \tag{2.1.27}$$

2.1.7 Bonded interactions

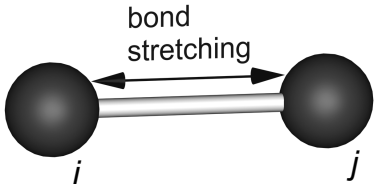
Bonded interactions represent the forces created by chemical bonds between atoms keeping the molecule together and are the primary cause of the molecular shape. The bonded (intramolecular) potential is the sum of many different interaction types, where each interaction can involve a different number of atoms such as pair potentials, three- or four-body potentials. The most important bonded interactions are *bond stretching* (two-body; $\mathcal{V}_{ij}^{stretch}$), *bond angle* (three-body; $\mathcal{V}_{ijk}^{bend}$), *dihedral torsion* (four-body; $\mathcal{V}_{ijkl}^{tors}$) and *improper dihedral* (four-body; $\mathcal{V}_{ijkl}^{impdi}$). The set of interactions can be written as:

$$\mathcal{V}^{Bonded} = \sum_{bonds} \mathcal{V}_{ij}^{stretch} + \sum_{angles} \mathcal{V}_{ijk}^{bend} + \sum_{\substack{proper \\ dihedrals}} \mathcal{V}_{ijkl}^{tors} + \sum_{\substack{improper \\ dihedrals}} \mathcal{V}_{ijkl}^{impdi} \,. \qquad (2.1.28)$$

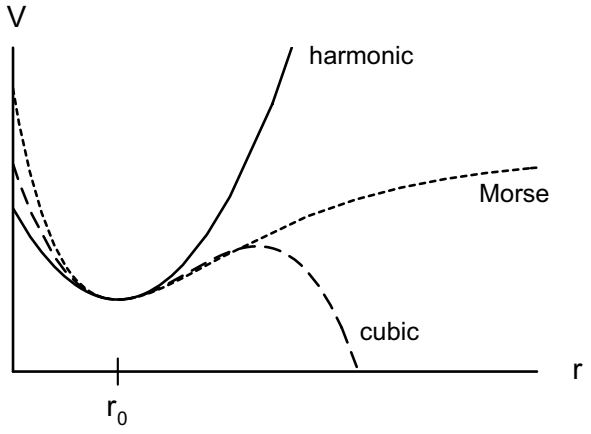
Note that there are also more interactions types available which can be found in the references cited in Section 2.1.7.

Bond stretching:

Covalent bonds resist compression and stretching by 'trying' to maintain an equilibrium length. The principle is shown Figure 2.4a and can be modelled in various ways by two-body potentials with different degrees of realism [27].



(a) Principle of bond stretching.



(b) Harmonic, cubic and Morse potential for bond stretching, where r_0 is the equilibrium bond length.

Figure 2.4: Bond stretching.

Harmonic bond stretching potential:

The harmonic potential (Figure 2.4b) is a simple and idealistic function form based on the behaviour of a spring

$$\mathcal{V}^{stretch}(r_{ij}) = \frac{1}{2} k_{ij}^{stretch} (r_{ij} - r_{ij0})^2 \,, \qquad (2.1.29)$$

where r_{ij} is the distance between the bonded atoms i and j, r_{ij0} the equilibrium distance and $k_{ij}^{stretch}$ the force constant [182]. The potential approximation is only good close to the equilibrium length, whereas for simulations outside this regime, more complicated function forms are needed.

Cubic bond stretching potential:

A more realistic potential can be obtained by adding an non-harmonic cubic term to the harmonic potential [42, 182]:

$$\mathcal{V}^{stretch}(r_{ij}) = \frac{1}{2}k_{ij}^{stretch}(r_{ij} - r_{ij0})^2 - \frac{1}{2}k_{ij}^{stretch}k_{ij}^{cubic}(r_{ij} - r_{ij0})^3 \ , \tag{2.1.30}$$

where k_{ij}^{cubic} is the cubic correction value that may be invariant in some force fields. As can be seen in Figure 2.4b, the model breaks down if r moves too far from r_0, i.e., the bond overstretches, and the potential becomes infinitely low.

Morse bond stretching potential:

For bonds under large tensile loads where the length does not remain near the equilibrium, the harmonic and cubic forms are clearly inadequate [42]. In this case, it is common to use the Morse potential [135]:

$$\mathcal{V}^{Morse}(r_{ij}) = D_{ij}\left[1 - e^{(-\beta_{ij}(r_{ij}-r_{ij0}))}\right]^2 \ , \tag{2.1.31}$$

where D_{ij} is the depth of potential well, β_{ij} is the steepness of the well and r_{ij} is the equilibrium distance. The value of D_{ij} can also be considered as the amount of energy required to break the bond at zero K, but it cannot be measured directly [42]. The parameter β_{ij} can be written as

$$\beta_{ij} = \sqrt{\frac{k_{ij}}{2D_{ij}}} \ , \tag{2.1.32}$$

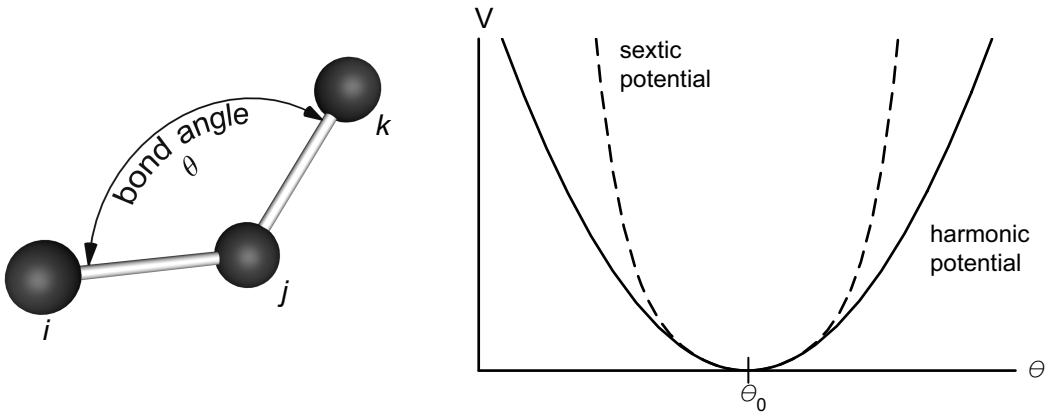
with $k_i j$ being the adjustable force constant. For small deviations the exponential term in Equation (2.1.31) can be approximated by a Taylor expansion. Omitting the higher order terms, the harmonic and the cubic functional forms can be derived [182].

Bond angle bending:

A bond angle θ_{ijk} is defined between the bonds ij and jk which share one atom j (Figure 2.5a). If more bonds share one atom then additional bond angles arise from the constellation:

- two bonds define one angle;
- three bonds define three angles;
- four bonds define six angles.

For each angle θ_{ijk} exists a preferred value θ_{ijk0} where the potential energy is lowest and all displacements are encountered by restoring forces [42]. There are several possibilities for the approximation of the potential energy:



(a) Principle of bond bending between the bonds *ij* and *jk*.

(b) Harmonic and sextic potential for bond bending.

Figure 2.5: Bond bending.

Harmonic bond bending potential:

The bond angle movement between the atoms *i, j* and *k* can be represented by the harmonic potential of the angle θ_{ijk}:

$$\mathcal{V}^{bend}(\theta_{ijk}) = \frac{1}{2} k_{ijk}^{bend} (\theta_{ijk} - \theta_{ijk0})^2 \ . \tag{2.1.33}$$

The form of the potential, as seen in Figure 2.5b, is similar to the harmonic stretch potential.

Sextic bond bending potential:

A more complicated form that contains an additional sixth-order term weighted by the parameter k_{ijk}^{sextic} is given by [3]:

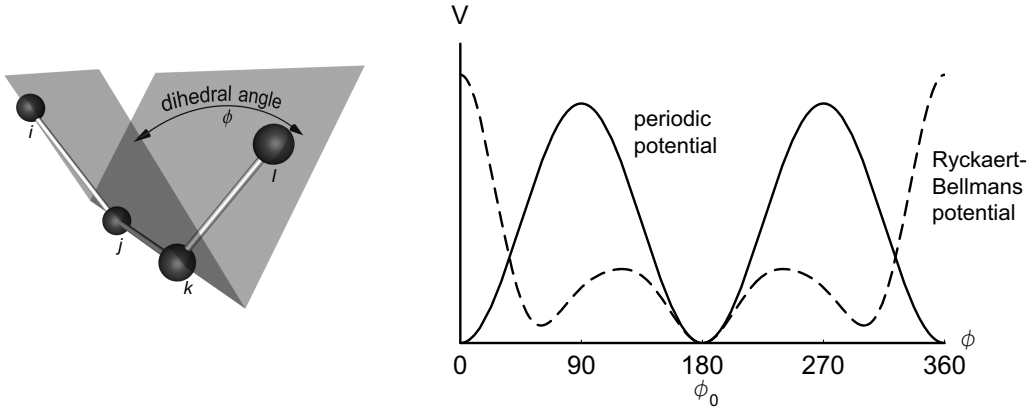
$$\mathcal{V}^{bend}(\theta_{ijk}) = \frac{1}{2} k_{ijk}^{bend} (\theta_{ijk} - \theta_{ijk0})^2 \left[1 + k_{ijk}^{sextic} (\theta_{ijk} - \theta_{ijk0})^4\right] \ . \tag{2.1.34}$$

Proper dihedral potential:

A rotation about a bond is not free of restricting forces. Each rotation angle ϕ is associated with a value for potential energy. The resulting periodic potential energy function may have several local minima and energy barriers between them.

According to the definition of the IUPAC[12] a dihedral torsion angle ϕ_{ijkl} is defined between the planes *ijk* and *jkl*, where $\phi_{ijkl} = 0$ corresponds to the constellation where the atoms *i* and *l* are on the same side [145, 158] (Figure 2.6a).

[12] International Union of Pure and Applied Chemistry.



(a) Principle of bond torsion: the angle ϕ is defined between the plane ijk and jkl.

(b) Periodic and Ryckaert-Bellmans potentials for bond torsion (proper dihedral).

Figure 2.6: Bond torsion.

Periodic torsion potential:

There are several approaches to model the proper dihedral potential energy. One way is to use a Fourier series periodic function type (Figure 2.6b) which is, for example, applied by the OPLS[13]-force field [101] (first four Fourier terms). The general form is given by

$$\mathcal{V}^{tors}(\phi_{ijkl}) = \sum_n \frac{1}{2} V^{tors}_{ijkln} \left[1 + \cos(n\phi_{ijkl} - \sigma_n)\right], \tag{2.1.35}$$

where V^{tors}_{ijkln} is the nth order energy constant for the $ijkl$ dihedral combination, ϕ_{ijkl} is the dihedral angle and σ_n is the phase angle. The sum may run up to $n = 12$ whereby σ_n is defined in a way that yields a minima at 180^o for terms with positive V^{tors}_{ijkln}:

$$\sigma_n = \begin{cases} 0^o & \text{for odd } n \\ 180^o & \text{for even } n \end{cases} . \tag{2.1.36}$$

Figure 2.6b shows an example for $n = 1$. For this simple type of function some force fields include an extra Lennard-Jones 1-4 potential [182].

Ryckaert-Bellemans potential:

The Ryckaert-Bellemans functional form, often used for alkanes, is based on:

$$\mathcal{V}^{tors}(\phi_{ijkl}) = \sum_n C_n \cos^n(\psi) , \tag{2.1.37}$$

where $\psi = \phi - 180^o$. The potential function for $n = 5$ is shown in Figure 2.6b, where the two-fold energy barrier is outstanding [14]. The Ryckaert-Bellemans functional form may be transformed into the periodic type by multiplying each coefficient C_n with $(-1)^n$.

[13] Optimized Potentials for Liquid Simulations.

[14] The value of ψ is defined according the *polymer convention*, where $\psi_{trans} = 0$ [182].

Improper dihedral potential:

A special form of dihedral interaction the (*improper dihedral*) is used at two occasions, to prevent molecules from flipping over to their mirror image (opposite chirality) and to keep planar groups (for example aromatic rings) [182]. It can be modelled either by a periodic torsion potential function with only the $n = 2$ term [30] or by using a simple harmonic potential

$$\mathcal{V}^{impdi}(\xi_{ijkl}) = k^{xi}(\xi_{ijkl} - \xi_0)^2 \ . \tag{2.1.38}$$

The functional form is similar to the bond stretch potential of Figure 2.4b. Because it does not provide periodicity ξ_0, it should be defined as far away from 180^o as possible [182].

Current force fields

Force fields have been developed by following one of the two main approaches [30]. The first is to develop force fields that accurately predict molecular structures and properties. The best known example for this approach may be the 'Molecular Mechanics' force field family developed by Allinger and co-workers. The force fields MM1 [5] and MM2 [3] have been replaced by the more sophisticated versions MM3 [6, 7, 117, 118] and MM4 [4, 8, 9, 138–140]. These force fields were created by using experimental data such as vibrational frequencies and formation heats.

In addition, the five interactions that are common to most force fields MM3 has additional interactions for stretch-bend, stretch-torsion and bend-bend that are also called *crossed terms* [6]. The MM4 force field includes additional crossed terms for stretch-stretch, torsion-bend, bend-torsion-bend, torsion-torsion, torsion-improper and torsion-torsion-improper interactions [139].

The second approach is directed towards the modelling and simulation of large macromolecules like proteins. Here, the conformation of the long polymer chain consisting of several hundreds or thousand atoms is priority and this necessitates simpler forms of the interaction potential functions. For example Van der Waals forces are handled by the simple Lennard-Jones or bond-stretch potentials by harmonic approximations. Even though they are simpler, these force fields give good results for condensed phase properties [30]. Typical representatives are the AMBER force fields [32, 147, 190–192] which comprise a history of several versions; the CHARMM (Chemistry at HARvard Molecular Mechanics) force field [26, 125, 134]; and the OPLS (Optimized Potentials for Liquid Simulations) force fields [36, 101, 102, 106].

There have also been attempts to close the gap between the two main directions. The MMFF (Merck Molecular Force Field) [73–77, 79] is a good example because the complexity of the functions and the number of different interactions lies between AMBER and MM3/MM4. The parameters are completely determined by ab initio calculations rather than experimental data. It produces good results for small organic molecules [78] but not for simulations in a condensed phase. Comparisons and evaluations of different force fields can be found in [46, 67, 68, 80, 91, 107].

2.2 Molecular dynamics

This section provides a description of the most important aspects of the molecular dynamics method.[15] As a deterministic simulation method, molecular dynamics calculates the evolution of a molecular system in time by computing the trajectory of the systems constituents based on the classical molecular model. The model determines the forces acting on the constituents through the potential $\mathcal{V}$ (Section 2.1.4), which is determined by a set of approximate equations that mimic the forces of quantum mechanical origin.[16]

The motion of the constituents, i.e. atoms and molecules, are described by the Hamiltonian equations of motion (Equation (2.1.2)) on the basis of different types of classical molecules:

- mono-atomic molecules: Newton's equation of motion (Equations (2.1.7) and (2.1.8));
- multi-atomic molecules without constraints: Bonds are modelled by potentials and Newton's equation (Equations (2.1.7) and (2.1.8));
- multi-atomic molecules partly constraint: Lagrangian equations of motion (Equations (2.1.16) and (2.1.17)).

In any case, the equations of motion lead to a system of differential equations that cannot be solved analytically; a numerical integration method must be used to advance the system from time t to time $t + \delta t$.

2.2.1 Integration methods

The time integration method (or short integrator) is the engine of MD that moves the particles and creates their trajectories. Numerical methods for time integration are usually based on finite difference approaches. Knowing the molecular position, velocities and other dynamic information at time t, the algorithm calculates position, velocities etc. for a time $t + \delta t$ where the step size δt is mostly kept constant. There are many integration methods, but not all are suitable for MD. Which algorithm fits best may be decided by basic criteria often used in numerical simulations [2, 157]: efficiency, accuracy and stability.

Efficiency: (of the simulation) is measured by the required resources, i.e. computing time and memory, to perform the simulation. While memory may only be a limiting factor for very large MD studies, computing time is of paramount importance.

Compared to the time-consuming force calculation (or calculation of the potential) the speed of the algorithm itself becomes insignificant [2]. Hence, the key for an efficient algorithm lies in a low ratio of force calculations per simulation time. The only way to achieve this is to minimize the number of force calculations per time step and enable at the same time the application of a large time step δt. This excludes algorithms like Runge-Kutta variants.

Often, algorithms are assessed in terms of efficiency by the size of the applicable time step [157]. It should be noted that size of the time step δt also depends on the

[15] A more comprehensive overview has been published by the author in Reference [43].

[16] Note that the potential can also be calculated by quantum mechanical means.

steepness of the potential gradient. As steep gradients result in fast changes of the acting forces they limit the size of the applicable time step.

Stability: The algorithm should be stable in a sense that it conserves energy. Small deviations or perturbations should not result in instabilities [174]. For each method stability is a function of several parameters such as step size and steepness of the potential gradients.

Accuracy: The algorithm should create trajectories that resemble the physical correct ones as closely as possible. Together with stability, accuracy can be assessed on simple models where an analytical solution is available. A good example is the harmonic oscillator [60, 61, 93].

Truncation and round-off errors: Other important numerical issues concern truncation and round-off errors.

Truncation errors are deviations from the exact solution caused by the approximation of the numerical method. Finite difference methods are mostly based on a Taylor series expansion that is cut off after some terms resulting in truncation errors. These errors are intrinsic to the method and independent of the implementation.

Round-off-errors on the other hand are mostly caused through the finite number of digit figures handled by computers. These errors are associated with the particular implementation of the numerical method and the specific machine which is being used for the simulation. Truncation errors are dominant for large δt, but decrease quickly as δt is reduced. For small δt round-off-errors are more significant, especially when the terms taking place for arithmetic operations are of different orders of magnitude.

2.2.2 The Verlet algorithm

A widely used time integration method is the *Verlet* algorithm [183] which is a direct solution of the second-order Equation (2.1.7). The method can be derived from two third-order Taylor expansions about $\boldsymbol{r}(t)$, one forward and one backward in time:

$$\boldsymbol{r}(t+\delta t) = \boldsymbol{r}(t) + \delta t\boldsymbol{v}(t) + \frac{1}{2}\delta t^2\boldsymbol{a}(t) + \frac{1}{6}\delta t^3\boldsymbol{b}(t) + O(\delta t^4) \ , \tag{2.2.1}$$

$$\boldsymbol{r}(t-\delta t) = \boldsymbol{r}(t) - \delta t\boldsymbol{v}(t) + \frac{1}{2}\delta t^2\boldsymbol{a}(t) - \frac{1}{6}\delta t^3\boldsymbol{b}(t) + O(\delta t^4) \ , \tag{2.2.2}$$

where $\boldsymbol{v} = \mathrm{d}\boldsymbol{r}/\mathrm{d}t$, $\boldsymbol{a} = \mathrm{d}^2\boldsymbol{r}/\mathrm{d}t^2$ and $\boldsymbol{b} = \mathrm{d}^3\boldsymbol{r}/\mathrm{d}t^3$. Adding Equation (2.2.1) to Equation (2.2.2) gives

$$\boldsymbol{r}(t+\delta t) = 2\boldsymbol{r}(t) - \boldsymbol{r}(t-\delta t) + \delta t^2\boldsymbol{a}(t) + O(\delta t^4) \ , \tag{2.2.3}$$

which is the basic version of the Verlet method. It is based on the knowledge of $\{\boldsymbol{r}_i(t),\ \boldsymbol{a}_i(t),\ \boldsymbol{r}_i(t-\delta t)\}$ and does not use velocities to compute the trajectory. As for the calculation of the kinetic energy, temperature, total energy etc., velocities are necessary and may be obtained by using a *central difference method*:

$$\boldsymbol{v}(t) = \frac{\boldsymbol{r}(t+\delta t) - \boldsymbol{r}(t-\delta t)}{2\delta t} \ . \tag{2.2.4}$$

As can be seen the inherent truncation error of Equation (2.2.3) is of the order δt^4, whereas the velocities in Equation (2.2.4) are of the order δt^2. The calculation sequence for each step is then:

1. Calculate the accelerations $\boldsymbol{a}(t)$ with Equation (2.1.11).
2. Calculate the positions $\boldsymbol{r}(t + \delta t)$.
3. Optionally: Calculate the velocities $\boldsymbol{v}(t + \delta t)$ to obtain the kinetic energy,

One disadvantage of the Verlet algorithm is the circuitous handling of the velocities. Another is that in Equation (2.2.3) a small term $O(\delta t^2)$ is added to a difference of two large terms $O(\delta t^0)$ entailing an unnecessary round-off-error [35, 183]. To eliminate these problems several modifications have been developed. Two of them, the so-called 'Leap-Frog' scheme and the velocity form of the Verlet algorithm, will be discussed below.

2.2.3 The Leap-Frog scheme

The *Leap-Frog* scheme [50, 92] is a half-step scheme and is based on the triplet $\{\boldsymbol{r}_i(t),\ \boldsymbol{v}_i(t-\frac{1}{2}),\ \boldsymbol{a}_i(t)\}$. It is named after the calculation sequence where position and velocity 'leap over' each other alternatively:

$$\boldsymbol{v}(t + \frac{1}{2}\delta t) = \boldsymbol{v}(t - \frac{1}{2}\delta t) + \delta t \boldsymbol{a}(t) , \tag{2.2.5}$$

$$\boldsymbol{r}(t + \delta t) = \boldsymbol{r}(t) + \delta t \boldsymbol{v}(t + \frac{1}{2}\delta t) . \tag{2.2.6}$$

The current velocities can be calculated from the midstep values:

$$\boldsymbol{v}(t) = \frac{\boldsymbol{v}(t - \frac{1}{2}\delta t) + \boldsymbol{v}(t + \frac{1}{2}\delta t)}{2} . \tag{2.2.7}$$

By eliminating the velocities the basic Verlet algorithm can be obtained. The calculation sequence is the following:

1. Calculate the acceleration $\boldsymbol{a}(t)$ by use of Equation (2.1.11).
2. Calculate the velocities $\boldsymbol{v}(t + \frac{1}{2}\delta t)$.
3. Calculate the positions $\boldsymbol{r}(t + \delta t)$.
4. Optionally: calculate the velocities $\boldsymbol{v}(t)$ to obtain the kinetic energy.

2.2.4 The velocity Verlet scheme

Both, the Verlet and the Leap-Frog scheme have the disadvantage that the positions and velocities are not available at the same time. In contrast, this variant of the Verlet method uses positions, velocities and accelerations defined at the same time, $\{\boldsymbol{r}_i(t),\ \boldsymbol{v}_i(t),\ \boldsymbol{a}_i(t)\}$ and minimises the round-off-error [176]. It has the form

$$\boldsymbol{r}(t + \delta t) = \boldsymbol{r}(t) + \delta t \boldsymbol{v}(t) + \frac{1}{2}\delta t^2 \boldsymbol{a}(t) , \tag{2.2.8}$$

$$\boldsymbol{v}(t + \delta t) = \boldsymbol{v}(t) + \frac{1}{2}\delta t \Big[\boldsymbol{a}(t) + \boldsymbol{a}(t + \delta t)\Big] . \tag{2.2.9}$$

An efficient implementation of the scheme can be obtained through a two-stage algorithm:

1. Advance the velocity by a half step:

$$\boldsymbol{v}(t + \frac{1}{2}\delta t) = \boldsymbol{v}(t) + \frac{1}{2}\delta t \boldsymbol{a}(t) \; . \tag{2.2.10}$$

2. Advance the position to $\boldsymbol{r}(t + \delta t)$ using Equation (2.2.8).
3. Calculate the accelerations $\boldsymbol{a}(t + \delta t)$.
4. Complete the velocity calculation by another half step:

$$\boldsymbol{v}(t + \delta t) = \boldsymbol{v}(t + \frac{1}{2}\delta t) + \frac{1}{2}\delta t \boldsymbol{a}(t + \delta t) \; , \tag{2.2.11}$$

   and calculate the kinetic energy based on $\boldsymbol{v}(t + \delta t)$.

As for the Leap-Frog scheme, the basic Verlet algorithm may be recovered by eliminating the velocities [2]. There is, however, an alternative way of deriving it [174].

Another closely related scheme is the so-called called *position Verlet scheme* which is not described here. Both the velocity- and position-based Verlet schemes have the property to propagate velocities and positions at half time steps. Because the schemes split the calculation process into an applied force (acceleration) term and a free-flight (no acceleration) term, the velocity- and position-based Verlet schemes are also called *half-kick*/*drift*/*half-kick* and *half-drift*/*kick*/*half-drift*, respectively [174].

2.2.5 Integration methods for constraint molecules

For all cases where the computational effort demands to fix some degrees of freedom, the Lagrangian equation of motion with constraints described in Section 2.1.3 need to be solved. Most often the bond length will be fixed since their vibrational motion is the fastest. In principle, Equation 2.1.16 may be solved exactly, but small errors, which are inherent to numerical integrators, would accumulate to significantly large errors in the constraints. For that reason, a number of algorithms that compute approximate solutions, such as the SHAKE and RATTLE, have been proposed [162]. Here, only the SHAKE algorithm, which is incorporated into the code used, is described.

Developed by Ryckaert et al. [161], SHAKE may be the most widely applied algorithm and must be used in connection with Verlet integrators. In the first step, the unconstrained atomic positions are calculated by

$$\boldsymbol{r}_i^{uc}(t + \delta t) = 2\boldsymbol{r}_i(t) - \boldsymbol{r}_i(t - \delta t) + \frac{\delta t^2}{m_i}\boldsymbol{f}_i \; , \tag{2.2.12}$$

where m_i is the mass of the atom i and $\boldsymbol{f}_i$ is the force applied to it. The true positions, however, are

$$\boldsymbol{r}_i(t + \delta t) = 2\boldsymbol{r}_i(t) - \boldsymbol{r}_i(t - \delta t) + \frac{\delta t^2}{m_i}(\boldsymbol{f}_i - \boldsymbol{g}_i) \tag{2.2.13}$$

and can be written using Equation (2.1.17) as

$$\boldsymbol{r}_i(t+\delta t) = \boldsymbol{r}_i^{uc}(t+\delta t) - \frac{\delta t^2}{m_i}\sum_{\alpha=1}^{M}\lambda_\alpha\left(\frac{\partial\sigma_\alpha(\boldsymbol{r}(t))}{\partial\boldsymbol{r}_i}\right), \tag{2.2.14}$$

where the sum runs over all M constraints with λ_α being the undetermined multipliers. The atomic positions at time $t+\delta t$ must satisfy the set of M constraint equations:

$$\left.\begin{array}{c}\sigma_1\left(\boldsymbol{r}(t+\delta t)\right)=0\\ \vdots\\ \sigma_M\left(\boldsymbol{r}(t+\delta t)\right)=0\end{array}\right\} \tag{2.2.15}$$

which are solved by the SHAKE algorithm. After obtaining the unconstrained positions $\boldsymbol{r}_i(t+\delta t)$ the algorithm adjusts them iteratively until the constraints are satisfied within a given tolerance [31, 161].

2.2.6 Potential energy calculation and optimisation method

The calculation of the atomic and molecular interactions is often denoted as the heart of molecular dynamics simulations. MD simulations require the computation of the configurational energy and their derivative (Equations (2.1.19) and (2.1.18)). Because it is the most time consuming part of the simulation, optimization is imperative. As specified in Section 2.1.4, Equation (2.1.19) contains intra- and inter-molecular (bonded and non-bonded) terms. Computationally, the most time consuming part is the calculation of the inter-molecular terms since the number of interactions to be evaluated at every time step scales with $n_f = N^m$ for a system of N atoms, where m is the number of interactions per atom. This is a tremendous number if one takes into account that nowadays simulations contain several millon atoms, thus involving huge computational costs. For this reason, most simulations neglect three atomic interaction and use effective pair-potentials, which limits the number of interaction to be evaluated to $n_f = N^2$, and considering only distinguishable interactions to $n_f = N(N-1)/2$.

In this section, the calculation of the inter-molecular interactions will be described, including the applied optimisation techniques. The description is limited to short-range forces, since only these have been used in this work. There are many other methods for the computation of long range forces, which can be found in the review [43].

The calculation of intra-molecular forces does not pose a methodological challenge in terms of exploding computational costs for larger systems. Instead, it only requires an appropriate data structure to store the information about the molecular bonds in a way that enable a convenient evaluation of the arising energies and forces.

Basic calculation

The basic potential and force calculation is explained for the case of the simple Lennard-Jones pair potential (Equation 2.1.23) which is used for most simulations in this work:

$$\mathcal{V}(r_{ij}) = 4\epsilon\left[\left(\frac{\sigma}{r_{ij}}\right)^{12} - \left(\frac{\sigma}{r_{ij}}\right)^{6}\right] .$$

Since the model defines the potential between all pairs of atoms, the total potential energy has to be calculated through summation over all individual pairs

$$\mathcal{V} = \sum_{i=1}^{N-1}\sum_{j>i}^{N}\mathcal{V}(r_{ij}) , \tag{2.2.16}$$

where r_{ij} is the distance between atoms i and j. Equation (2.2.16) may easily be computed in a double loop while each individual pair is only evaluated once. The potential of a single atom i is the sum of all potential interactions in which atom i is involved, i.e.,

$$\mathcal{V}_i = \sum_{j\neq i}^{N}\mathcal{V}(r_{ij}) . \tag{2.2.17}$$

Substituting this into (2.1.8) one obtains the force acting on atom i

$$\boldsymbol{f}_i = -\nabla_{\boldsymbol{r}_i}\mathcal{V}_i = -\sum_{j\neq i}^{N}\nabla_{\boldsymbol{r}_{ij}}\mathcal{V}(r_{ij}) = -\sum_{j\neq i}^{N}\frac{1}{r_{ij}}\left(\frac{\mathrm{d}(r_{ij})}{\mathrm{d}r_{ij}}\right)\boldsymbol{r}_{ij} . \tag{2.2.18}$$

For the Lennard-Jones potential (Equation 2.1.23) this transforms into

$$\boldsymbol{f}_i = \sum_{j\neq i}^{N}\frac{24\epsilon}{{r_{ij}}^2}\left[2\left(\frac{\sigma}{r_{ij}}\right)^{12} - \left(\frac{\sigma}{r_{ij}}\right)^{6}\right]\boldsymbol{r}_{ij} . \tag{2.2.19}$$

The force $\boldsymbol{f}_i$ can be calculated easily for all molecules within a double loop in parallel with the potential energy $\mathcal{V}_i$. Every atomic pair (i, j) needs to be evaluated only once, since $\boldsymbol{r}_{ij} = -\boldsymbol{r}_{ji}$.

Because such a simple 'brute force' approach is highly inefficient and computational expensive, it is limited to systems of small size and short simulation times. Several optimisation techniques have been proposed to reduce the computational effort. The ones employed in this work are presented below. More details as well as other approaches can be found in [2, 153, 162].

Cut-off distance and shifted potential

To reduce the number of interactions to be evaluated it is useful to distinguish between short- and long-range forces. While long-range forces are more difficult to handle, short-range forces offer effective possibilities for optimisation. Short-range forces like the Vander-Waals force decrease rapidly with increasing distance and are only significant in the

atoms nearest vicinity. Thus, contributions that arise from atoms further away than a certain cut-off distance, r_c, may be omitted. This reduces the number of potentials and forces to be calculated tremendously, but it also introduces a discontinuity at cut-off distance r_c, which affects the energy conservation as well the atomic trajectories. The discontinuity may be smeared out by changing the potential function, which should, however, be done carefully because it is the potential that determines the physical model. Thus, altering the potential function will affect the physical properties of the model as well. Another way that is commonly used (also in this work) to remove the discontinuity is to use the so called *shifted* potential. Here, the potential function is shifted so that its value at the cut-off distance r_c becomes exactly zero [153]. The shifted potential is then given by

$$\mathcal{V}(r) = \mathcal{V}(r) - \mathcal{V}(r_c) - \left.\frac{\mathrm{d}(r)}{\mathrm{d}r}\right|_{r=r_c} (r - r_c) \,, \tag{2.2.20}$$

where the last term on the right side removes the kink at r_c that emerges if the gradient of $\mathcal{V}$ at r_c is non-zero. In order to compensate the effects of neglecting the atoms further away, one could also introduce additional approximate *long range corrections* into the potential function [162, 174]. When applying periodic boundary conditions (Section 2.2.7), the use of a cut-off distance r_c smaller than half of the smallest size of the simulation box is necessary in order to meet the minimum image convention [2].

Beside computational constraints, there is also a physical justification for adopting a cut-off radius. Quantum mechanics dictates that energy can not be transferred continuously but only in quantised amounts. Thus, the potential energy function of an interatomic interaction should not approach zero asymptotically (compare Equation 2.1.23), but rather jump discontinuously to zero at the smallest transferrable amount of energy.

Neighbour list method

Though the use of a cut-off distance reduces the number of potentials to be computed, the radii of all pairs still have to be calculated at each time step. Verlet [183] suggested to reduce the computational effort further by using a list of neighbours. The list is constructed by calculating the distance of each individual atomic or molecular pair. Starting with an empty list, all pairs whose distance is less than a predefined sphere of radius r_l are added to the list. As shown in Figure 2.7, r_l must be bigger than r_c because the space in between serves as a buffer for the molecules that are just outside the cut-off sphere (or circle) and may come closer than r_c within the next few steps. Therefore, only the distances of the pairs in the neighbour list must be calculated to find out whether they are closer than r_c. This reduces the computational effort substantially [2]. Usually, the list is updated every 10 – 20 steps since the atoms from outside the buffer zone may have moved inside the cut-off sphere. The optimal number of time steps depends on the ratio r_c/r_l.

Fincham and Ralston [51] proposed to update the list automatically. This is accomplished by accumulating the sum of the magnitudes of the two largest movements of every time step. If this value exceeds the radial thickness of the buffer zone($r_l - r_c$) an atom from outside the buffer zone may have entered the cut-off sphere and the neighbour-list must be

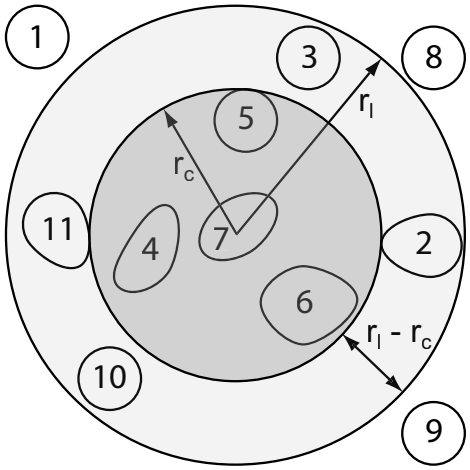


Figure 2.7: The cut-off sphere (r_c) and its buffer zone ($r_l - r_c$) around molecule 7; except 1, 8 and 9 all molecules are on the neighbour list of molecule 7, where at this point in time only the interactions with the molecules 4, 5 and 6 are taken into account.

recalculated. The method reduces the number of force calculations for N atoms to

$$n_f = \frac{N^2}{2\delta_s} + N_l N \ , \tag{2.2.21}$$

where N_l is the average number of atoms within the neighbour list and δ_s is the number of calculation steps between the list updates.

Cell subdivision (cell-method)

With increasing number of molecules ($N >= 1000$) the creation of the neighbour-list (scaling with N^2) becomes computationally expensive because the distance of all possible interaction-pairs must be calculated. To make the distance evaluation process more efficient, a *linked cells list* method was proposed by Quentrec and Brot [152].

In this method the computational box of size $\boldsymbol{L} = (L_x, L_y, L_z)$ is divided into cells as shown in Figure 2.8. The number of cells in each dimension $\boldsymbol{M} = (M_x, M_y, M_z)$ determines the cell-size in the corresponding dimension as $\boldsymbol{L}_C = (L_x/M_x, L_y/M_y, L_z/M_z)$. If $\boldsymbol{M}$ is chosen in a way that the cell-size $\boldsymbol{L}_C$ is in all dimensions slightly bigger than the cut-off distance, then interactions within cut-off distance can only be found between atoms that are inside the same cell or inside neighbouring cells. This means that for the calculation of the total force acting on an atom, only the distances to all other atoms of the same cell and immediate neighbour cells must be calculated and compared with the cut-off distance r_c. Furthermore, only half of the neighbour cells need to be evaluated since the interactions to the other half is handled automatically by these cells. This is achieved by using a neighbour-cell link list[17] that is created before the simulations starts for each cell. A two-dimensional example can be seen in Figure 2.8 where the four dark shaded cells are linked by the cell

[17] A list is a basic programming element that is described in almost every programming textbook.

13. The four light shaded cells, applying the same scheme as cell 13, have a link to cell 13 in their own list. The handling of the boundary cells depends on the applied boundary condition. The cells from the opposite boundary are the neighbouring cells, if periodic boundary conditions are employed.

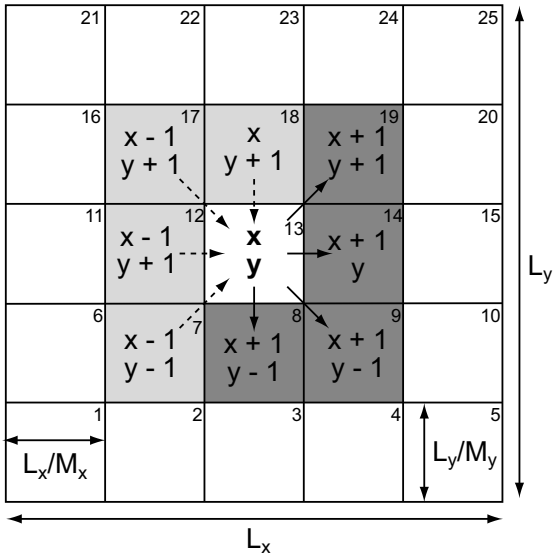


Figure 2.8: Illustration of a two-dimensional cell subdivision; the cell number is printed in the upper right corner of each cell. Cell number 13 (x, y) is highlighted. It retains links (solid arrows) to the dark shaded neighbouring cells while each of the light shaded neighbouring cells store a link (dashed arrows) to cell (x, y).

Another list that each cell stores contains the indices of all atoms[18] that are currently inside the cell. This list can easily be assembled every step by going through all atoms and calculate the list index by

$$i = M_x M_y \frac{r_z}{LC_z} + M_x \frac{r_y}{LC_y} + \frac{r_x}{LC_x} \ . \tag{2.2.22}$$

Once the atoms have been sorted into the linked cells, the complete force/potential computation starts with a loop over all cells. For each cell, the algorithm successively processes all atoms in the atomic list. To calculate the total force and/or potential of an atom, the distance between this atom and all following atoms of the list is calculated, and if closer than r_c the force/potential is computed. Next, the same is done for all atoms in the atomic list of the linked neighbour cells.

Using the cell-method (and periodic boundary conditions) the number of potential/force calculations scales with

$$n_f \approx (n_n + 1) N_c N \ , \tag{2.2.23}$$

[18] A reference list stores the index of (or pointer to) the place (memory, array) where the actual information (in this case data about atoms or molecules) are found. Like the neighbour list method the algorithm can also be based on molecules, but the atomic version is more accurate.

where n_n denotes the number of neighbouring cells that is $n_f = 8$ for two dimensions and $n_f = 26$ for three dimensions. It has been shown that in some circumstances the use of neighbour-list may have some advantages over the cell method, though it is not suitable for vector processors [154]. The cell method is, however, vastly superior for large number of atoms and/or low densities [57]. The best approach, however, is to use the link-cell method for creating an atomic neighbour list (Section 2.2.6) which can then be used for few time steps.

2.2.7 Boundary conditions

Boundary conditions are an important and necessary part of all numerical simulations. Inherently, a simulation can only be performed in a limited computational domain requiring conditions to be defined at the boundaries. The simplest boundary condition for a MD simulation is an infinite vacuum in all directions. This is sufficient for simulations where the material is kept together by inter-molecular forces, such as in the case of nanoclusters. The situation changes when the material within the simulation domain is surrounded by more material outside, whose effects have to be modelled through the boundary conditions, as is normally the case. A brief description of the most commonly used boundary conditions is given below.

Periodic boundary conditions

Traditionally, the most often applied boundary conditions in molecular dynamics simulations are periodic boundary condition because they allow the simulation of a homogenous infinite system with a limited number of atoms, which were the only systems that could be simulated with early computers. Periodic boundary conditions copy identical images of the computational box aligned to a space-filling array throughout the space, as sketched in Figure 2.9. The approach mimics an infinite computational domain with a finite number of molecules. To implement this condition the following has to be applied:

1. An atom leaving the box in one direction enters it at the same time from the opposite side. Therefore, one needs to check regularly whether an atom has crossed the boundary and to correct its position in that case.
2. An atom whose position is within interaction distance of the boundary interacts with the images of the atoms from the opposite side of the simulation box [153]. This wraparound effect must be taken into account when calculating the interatomic potentials or forces.

While the periodic boundary condition circumvent elegantly the problem of surface atoms, they create another. For calculating the potentials or forces, it would strictly be required to evaluate the interaction between all pairs of the computational box and their images, which is an infinite number of interactions. For short-range forces this problem is overcome through the *minimum image convention* which is based on the usage of a cut-off distance. It dictates that only the closest periodic image of an interaction between two atoms must be regarded. Figure 2.10 shows an example in two dimensions.

Periodic boundary conditions may also be applied for other geometrical shapes of the simulation domain. While the box is extensively used because of its simplicity all space-filling bodies can be used in conjunction with periodic boundary conditions [2]. Two-dimensional systems may also be mapped on the surface of a sphere [83] and three-dimensional systems on a hypersphere [113]. Complications caused by the non-Euclidean character of the geometry, become less important with bigger systems as the curvature of the geometry decreases.

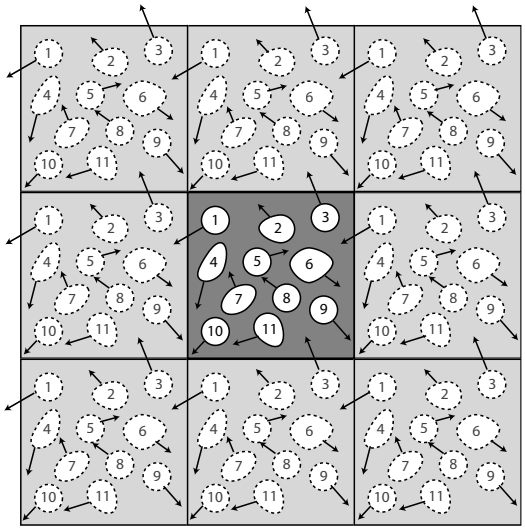


Figure 2.9: Periodic boundary condition in two dimensions; the simulation box in the middle is dark and the images are light grey, molecules that leave that box enter it at the same time from the opposite side.

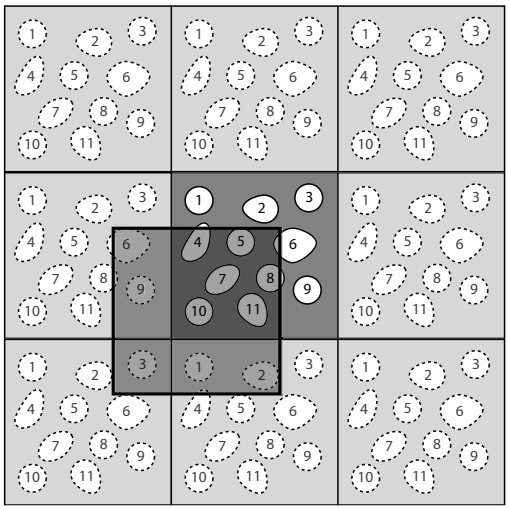


Figure 2.10: Minimum image convention for molecule 10 that only interacts with the molecules and images that lie inside the shaded box

Wall boundary conditions

Many interesting scientific questions are related to surface interactions. This may be a liquid flowing through a channel or a cluster impacting on a surface. In such cases boundary conditions that mimic or simulate the properties of the wall must be applied. Instead of reproducing real walls, various approximations can be used:

- *Stochastic walls* [180] reflect all molecules that try to cross the wall back to the interior. The resulting velocity vector $\boldsymbol{v}$ of the reflected particle is determined by the wall properties. Its direction may be perfectly reflected, randomly generated, adapted to a specific direction (moving wall) or be a blend of several weighted parameters. The change in absolute amount of the velocity $|\boldsymbol{v}|$ is governed by the temperature of the wall. It may be set to a certain value, heated or cooled.
- *Fixed wall* atoms that are arranged in layers may also be used to represent a wall. Through this arrangement a desired roughness can be achieved [15].

- Since atoms are in reality not immovable, *tethered atoms* [15] can be used to mimic the thermal vibrations of the wall atoms. By accelerating or decelerating their movements, one can transfer heat to or from the interior.

Flux boundary conditions

The prescription of mass, momentum and energy fluxes on the boundaries of MD simulation are potentially very useful and can form the basis of boundary conditions as they are usually applied in continuum simulations, such as Neumann or Dirichlet boundary conditions [11]. Flux boundary conditions have been used for the application of heat fluxes in simulations of heat transfer. Other fluxes, in particular mass fluxes have only been used in few cases in MD simulations, due to the difficulties involved in their application. The type of the hybrid MD-CFD method that is investigated in this thesis requires as core element MD flux boundary conditions for mass, momentum and energy. The algorithms and numerical techniques to impose them are discussed in detail in chapter 7.

2.2.8 Initialisation

Before any MD simulation can be started, the content of the simulation, i.e. the atoms or molecules have to be generated. Generating means that the necessary structures are allocated in the memory and each atom is assigned with a position and velocity. In case of molecules, the bonds between the atoms that form the individual molecules have to be designated. The initialisation may require certain additional steps regarding the integration algorithm, specific potentials, optimisation procedures or the initialisation of the hybrid solution interface. After all necessary initialisation have been done, the actual simulation, i.e. the time integration of the atomic trajectories, can commence.

Even if all parts of the initialisation procedure are important for a functioning simulation, the atoms are the essence of a MD simulation and the discussion here will concentrate only on the initialisation of positions and velocities. The first of the following section is focused on the generation of the atomic positions and the second one addresses the handling of the velocities.

Initialise atomic positions

The initial positions have to be generated as closely as possible to the natural structure of the respective elements. Molecules and mono-atomic substances will either occur in a well ordered solid state or a less ordered liquid or gaseous state. Whereas for the initialisation of solid state materials the *lattice filling scheme* is the first choice, liquid or gaseous states can be treated more easily by the *random filling scheme*. Both schemes are discussed in this section for mono-atomic cases only; they can however be extended to the initialisation of molecules.

Lattice filling scheme Before an explanation of the lattice filling method, a few words about solid state must be said to provide the fundamental definitions which are used in the

algorithm. In solid state, most materials form a crystal structure, which is characterised by the regular repetition of identical structural units, such as atoms, molecules or groups of these. This enables the description of the solid structure through a spacial *lattice* (a mathematical grid of point coordinates) and the structural units which are placed at every point of the lattice. The structural unit of a crystal is called *base* and has the same orientation at each point of the lattice.

The standard way of describing the lattice is the definition of a *unit cell*, which is a parellelepiped that contains a small part of the lattice points. By pure translational displacement the unit cell can be used to reconstruct the complete lattice. Thus, the lattice is formed as a three-dimensional array of identical unit cells. In Figure 2.11a such an array is sketched with one unit cell highlighted in blue. The red marked unit cell vectors $\boldsymbol{a}_x$, $\boldsymbol{a}_y$ and $\boldsymbol{a}_z$ define the shape of the unit cell. In general, the angle between the unit cell vectors is arbitrary, and the shape of the unit cell ranges from triclinic to cubic. It is advantageous to choose the unit cell based on symmetry groups. In total there are fourteen so called *Bravais* lattices with the respective unit cells that cover all possible symmetries [96].

In this work, only materials that form a face centred cubic (FCC) lattice with a simple atomic base are used. Figure 2.11b shows the respective unit cell of a FCC lattice. Atoms are placed at each corner and in the middle of the each of the faces. Thus, each corner atom belongs to four and each atom on a face to two adjacent cells. For generating a crystal structure, one needs to assign each atom to one cell. Here, the non-transparent atoms are atoms of the displayed cell and the transparent atoms belong to the neighbour cells.

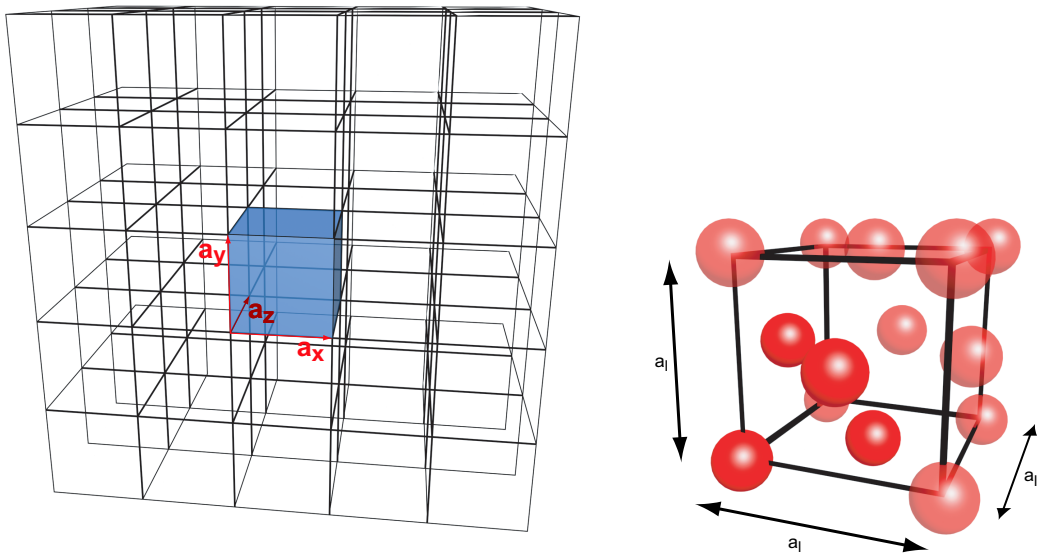
The length of the unit cell vectors are called lattice parameters. They depend on the density of the material and the type of the unit cell. In a FCC unit cells all vectors have the same length and one lattice parameter a_l is sufficient to define the size of the unit cell: $|\boldsymbol{a}_x| = |\boldsymbol{a}_y| = |\boldsymbol{a}_z| = a_l$. The lattice parameter itself is determined through the number density of the system and is calculated by

$$a_l = \sqrt[3]{\frac{4}{n}} \tag{2.2.24}$$

for a FCC lattice.

For pure bulk materials, the size of the molecular domain $\boldsymbol{L}$ can be chosen to comprise exactly the crystal lattice: $\boldsymbol{L} = a_l \boldsymbol{L}_l$, where $\boldsymbol{L}_l$ is the size of the system in unit cells. This enables also to create relatively small periodic systems with the exact density and without any discontinuities.

To fill an arbitrary region R with a crystal structure one has to select the origin $\boldsymbol{r}_o$ and to rotate the unit cell vectors ($\boldsymbol{a}_x$, $\boldsymbol{a}_y$, $\boldsymbol{a}_z$) and the base atom coordinates to fit them to the desired orientation. The atoms can now easily be generated as copies of the base atoms at the lattice points which are obtained through addition of whole-numbered multiples of the unit cell vectors. Algorithm 1 describes the procedure for a three-dimensional lattice from $\boldsymbol{0}$ to $\boldsymbol{L}_l$ unit cells. The position of the unit cell is called $\boldsymbol{r}_u$. The number of lattice points per unit cell is N_p and the relative position of a lattice point to the unit cell origin $\boldsymbol{r}_u$ is designated with $\boldsymbol{r}_p$. For the FCC lattice, $N_p = 4$ with $\boldsymbol{r}_1 = (0, 0, 0)$, $\boldsymbol{r}_2 = (1/2, 1/2, 0)$, $\boldsymbol{r}_3 = (1/2, 0, 1/2)$ and $\boldsymbol{r}_4 = (0, 1/2, 1/2)$. The number of atoms in the base is N_b with the



(a) Grid of unit cells.

(b) Unit cell of a cubic FCC lattice.

Figure 2.11: Unit cell and grid of a FCC crystal lattice.

relative coordinates of the base atoms being $\boldsymbol{r}_b$. For the mono atomic case $N_b = 1$ and $\boldsymbol{r}_b = (0, 0, 0)$.

Problems arise when using the lattice filling method on the boundaries of non-trivial geometries or when two or more different lattice constants are used in the same molecular domain as it may occur for simulations of more than one material. Especially in conjunction with periodic boundary conditions, the domain size can only be adapted to one lattice parameter, so that one material fits exactly into the domain. If the other material needs to be fitted into the periodic domain then its lattice parameter needs to be slightly increased or decreased, in order to avoid overlapping atoms or defects in the material. This problem is especially prominent for regions of small size in any direction.

For liquids or gases, one is often interested in filling a region of the molecular domain with a certain density. The lattice parameter which corresponds to that density rarely fit to the size of the region. As a result, the density in the region deviates from the desired density. It is, therefore, better to use the random filling scheme for liquids and gases.

Random filling scheme The *random filling scheme* is particularly useful for filling an arbitrarily complex region with atoms in liquid or gaseous state, because it delivers exactly the required density. As the name suggest, the scheme generates the coordinates of the atoms randomly. Each randomly generated coordinate must be tested as to whether it is inside the region R and whether the closest distance to any other atom r_{close} is bigger than a minimum distance r_{min} to avoid overlapping of atoms. The random number generator must be based on a uniform distribution to achieve an evenly filled region.

Algorithm 1: Lattice filling algorithm [153].

```
i ← 0;
for l ← 0 to L_l do
    r_u ← l_x a_x + l_y a_y + l_z a_z (Calculate the unit cells position)
    for p ← 0 to N_p do
        r_l ← r_u + a_l r_p (Calculate the lattice point position)
        for b ← 0 to N_b do
            r_i ← r_o + r_l + r_b (Calculate the atoms position)
            if r_i is within the region R then
                generate atom i at r_i
                i ← i + 1
```

Especially for large systems, the computation of the minimum distance r_{close} is computationally expensive. For this reason a random filling scheme that uses a density grid superimposed on the region R has been devised during this work. This cartesian grid divides the region into evenly spaced cells. For each cell C, the number of atoms inside is stored together with a linked list to each of the respective atoms. To insert a new atom, the density grid is used to find the cell with the lowest density and a new coordinate is generated inside this cell through a uniform random generator. Using the linked atom list of the respective cell and its neighbour cells, only possible candidates for the closest atom need be accessed for the calculation of r_{close}. This enables a rapid calculation of r_{close} because the number of atoms that need to be checked is dramatically reduced.[19] The principle operation mode of the random filling scheme is given in Algorithm 2. The number of atoms N that need to be generated is calculated from the number density $n = \rho/m$ (m is the molecular mass) and the volume V_R of the region R.

Algorithm 2: Random filling algorithm.

```
Generate the initial density grid
N ← nV_R
i ← 0;
while i < N do
    Find grid cell C with lowest density
    repeat
        Generate new random position r_i within the cell C
        Calculate closest distance r_close to other atoms in cell C neighbor cells
    until r_close > r_min ;
    Generate atom i at r_i
    i ← i + 1
    Add contribution of atom i to density of cell C
```

[19] The algorithm is similar to the linked cell list algorithm used for the force calculation (q.v. Section 2.2.6).

When using the random filling scheme for higher densities such as liquids, it has turned out to be computationally more efficient to use a relatively small minimum distance r_{min} and to resolve the issue of overlapping atoms through an energy minimisation stage before proceeding with the equilibration.

Initialise atomic velocities

There are two common ways of generating the initial velocities. The simplest is that of a fixed magnitude and the other the sampling of a velocity distribution. Whatever algorithm is chosen, the generated velocities have to meet the conservation law of momentum [20] that states that the overall momentum is zero:

$$\boldsymbol{v}_{sys} = \sum_{i=1}^{N} m_i \boldsymbol{v}_i = \boldsymbol{0} \ , \tag{2.2.25}$$

$$\boldsymbol{\omega}_{sys} = \sum_{i=1}^{N} m_i \boldsymbol{v}_i \times \left(\boldsymbol{r}_i - \boldsymbol{r}_{sys} \right) = \boldsymbol{0} \ , \tag{2.2.26}$$

where the reference point is the system's centre-of-mass $\boldsymbol{r}_{sys}$ which is calculated by

$$\boldsymbol{r}_{sys} = \frac{1}{M_{sys}} \sum_{i=1}^{N} m_i \boldsymbol{r}_i \ , \tag{2.2.27}$$

with M_{sys} being the total mass of the system. For statistical reasons, the translational momentum $\boldsymbol{v}_{sys}$ and the angular momentum $\boldsymbol{\omega}_{sys}$ will never be exactly zero. Hence, $-\boldsymbol{v}_{sys}$ and $-\boldsymbol{\omega}_{sys}$ need to be added to all of the system's atoms after their velocity vectors have been generated.

Distribution sampling: The natural velocity distribution of an atomic or molecular system in equilibrium state is the *Maxwell-Boltzmann* velocity distribution [19], which gives the probability for the one-dimensional velocity components of an atom depending on the temperature T and the atom mass m_i. For each dimension $\alpha = x, y, z$, the probability density $f(v_{i\alpha})$ of the velocity $v_{i\alpha}$ is given by

$$f(v_{i\alpha}) = \left(\frac{m_i}{2\pi k_B T} \right)^{\frac{1}{2}} \exp\left(-\frac{m_i v_{i\alpha}^2}{2 k_B T} \right) . \tag{2.2.28}$$

The distribution is basically a parameterised *Gaussian* distribution. Hence, it is possible to generate Maxwell-Boltzmann distributed velocities by creating random numbers $R_{i\alpha}$ according to a Gaussian distribution with unit variance ($R_{i\alpha} \sim N(0,1)$. These random numbers can be transformed to obtain the velocities

$$v_{i\alpha} = \sqrt{\frac{k_B T}{m_i}} R_{i\alpha} \ . \tag{2.2.29}$$

[20] Here it is assumed that the overall velocity of the system is zero.

Fixed value: As a simple alternative, one can assign the same magnitude of the velocity to all atoms and run the simulation until an equilibrium has been reached with the velocity distribution corresponding to the natural Maxwell-Boltzmann distribution. In that case, the magnitude of all translational vectors is set to

$$|\boldsymbol{v}_i| = v_i = \sqrt{\frac{3k_BT}{m_i}} \ . \tag{2.2.30}$$

The direction of the velocity vector $\boldsymbol{v}_i$ is *uniformly* distributed. A simple method to generate isotropic distributed vectors is the acceptance-rejection technique [185]. The technique creates a unit vector $\boldsymbol{e}_u$ by

$$\boldsymbol{e}_u = \frac{1}{n}\begin{pmatrix} 1-2n_1 \\ 1-2n_2 \\ 1-2n_3 \end{pmatrix},\ n = \sqrt{n_1^2+n_2^2+n_3^2},\ n < 1 \ , \tag{2.2.31}$$

where the numbers n_1, n_2 and n_3 are sampled from a uniform distribution in the range $[0, 1]$. The generated vector $\boldsymbol{e}_u$ is accepted if n is smaller than one [157]. An improved version has been suggested by Allen and Tildesley [2]:

$$\boldsymbol{e}_u = \begin{pmatrix} 2(1-2n_1)\sqrt{1-n^2} \\ 2(1-2n_2)\sqrt{1-n^2} \\ 1-2n^2 \end{pmatrix},\ n = \sqrt{n_1^2+n_2^2},\ n < 1 \ . \tag{2.2.32}$$

Simulations of a fluid at different densities have been performed in order to demonstrate that the fixed initialization approach leads to the Maxwell-Boltzmann velocity distribution. The mono-atomic fluid was modelled by the Lennard-Jones potential. Units have been made dimensionless by basing them on the choice of the atomic mass m and the Lennard-Jones parameter σ and ϵ. The resulting replacements are then $r \to r\sigma$ for length, $e \to e\epsilon$ for energy, and $t \to t\sqrt{m\sigma^2/\epsilon}$.

The liquid was simulated by adapting an example from [153] in a two-dimensional 150×150 cell (22500 atoms). A Leap-Frog integrator with a time step of 0.001 was used. At the start of the simulation every atom had the same velocity $|\boldsymbol{v}|$. Figure 2.12 shows the plot of the velocity distribution at different time points. It can clearly be seen that it approaches the theoretical Maxwell-Boltzmann distribution which is displayed in the background of the figure.

Another possibility for surveying the velocity distribution is to use the Boltzmann H-function [130] which is defined as

$$H(t) = \int f(\boldsymbol{v}, t) \log f(\boldsymbol{v}, t) \mathrm{d}\boldsymbol{v} \ . \tag{2.2.33}$$

It can be shown that $\langle \mathrm{d}H/\mathrm{d}t \rangle \leq 0$ is only valid if $f(\boldsymbol{v}, t)$ is the Maxwell distribution. Figure 2.13 shows the time development of the analysis for the simulated system at different densities. The convergence is faster for higher densities. For low density the convergence does not change before the atoms come in interaction range.

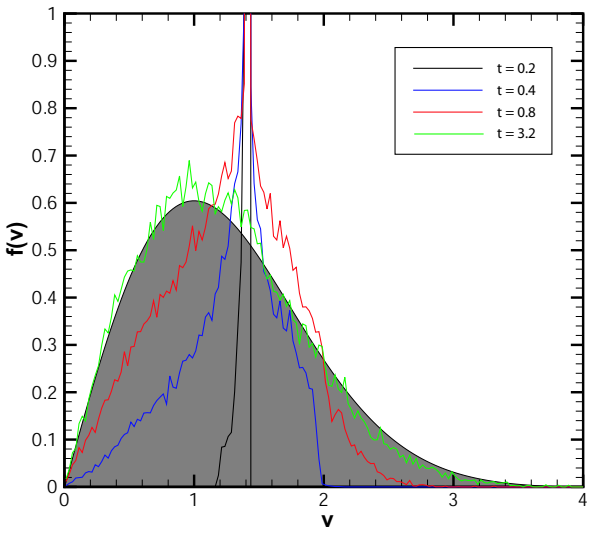


Figure 2.12: Velocity distribution for different times 0.2, 0.4, 0.8 and 3.2. The shaded curve is the theoretical Maxwell-Boltzmann distribution. With the initial value being a spike at $v = \sqrt{2}$ the graphs approach the theoretical distribution as the simulation evolves. The density of this run was 0.4.

2.2.9 Equilibrium and equilibration

The most important state of a molecular system in MD simulations is the state of thermodynamic equilibrium. This is obviously true for equilibrium MD (EMD) simulations, but also for non-equilibrium MD (NEMD) simulations. NEMD simulation, like most of the simulations performed for this work, are started from a well defined state, which is in most case an equilibrium state.

A molecular system is considered to be in thermal equilibrium if the macroscopic state does not tend to change in time [19]. In an isolated system, i.e. with constant number of atoms, volume and energy (NVE), the equilibrium condition is that the entropy must be constant, i.e. $\delta S = 0$. The equivalent condition for systems with constant number of atoms, pressure and temperature (NPT) is that Gibbs free energy is constant, i.e. $\delta G = 0$. Even if the system is in equilibrium, there will be a constant exchange between kinetic and potential energy which causes these energies to fluctuate around an average value. The effect is more noticeable for small systems; for large systems the magnitude of the fluctuations relative to the total energy approaches zero.

It is practically impossible to initialise even a simple molecular simulation in a state of equilibrium. Rather, the system will be more or less far from equilibrium after each atom has been assigned with a position and velocity vector. Therefore, the initialisation is normally followed by an equilibration phase. The equilibration phase is a normal MD run during which the temperature, i.e. the kinetic energy of the system, is controlled. In the

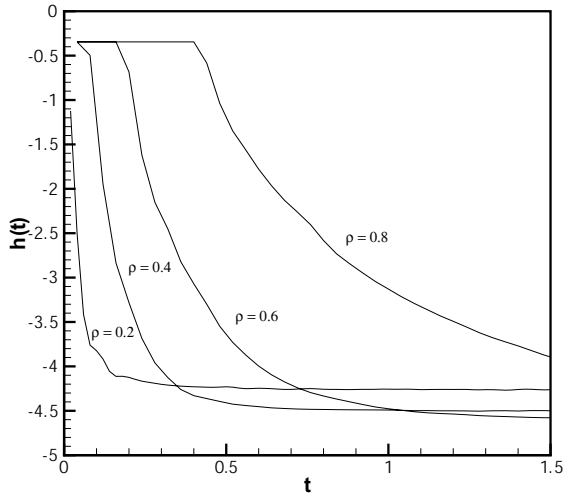


Figure 2.13: Boltzmann H-function in time for different densities 0.2, 0.4, 0.6 and 0.8; the denser the faster the convergence.

course of the equilibration, the system will gradually approach an equilibrium state. The temperature control scheme is explained in the following section.

There are several indications as to whether a system has reached equilibrium. A simple way is to monitor the instantaneous values of potential, kinetic and total energy [2, chap. 2]. Here, a successful equilibration is indicated by the asymptotic course of energies to steady mean values around which they oscillate in the equilibrium state. When using any kind of flexible boundary conditions, like in some of the simulations performed for this work, the systems shows large oscillations that arise from oscillations in volume. In such cases, it is necessary to record the energies and the volume of the system in order to check for successful equilibration according to the condition $\delta G = 0$. Another way of checking for equilibration is to verify whether the velocity distribution function has converged to its predicted form, the Maxwell distribution [153, chap. 2], or to monitor structural order parameters [2, chap. 5].

For large systems it may be necessary to control the temperature of smaller parts of the system individually.

Temperature control

A range of schemes have been developed to accomplish this task. These include simple velocity scaling, heat-bath coupling, Anderson thermostat and the Nosé-Hoover thermostat [162]. In this work, only simple velocity scaling is used. This scheme scales the velocities of the atoms by a factor to adjust the current temperature

$$T = \frac{1}{3Nk_B} \sum_i m_i \boldsymbol{v}_i^2 \tag{2.2.34}$$

to the desired temperatures T_D. The temperature of the system can be brought to T_D by multiplying the velocity vectors $\{\boldsymbol{v}\}$ of the atoms with $\sqrt{T_D/T}$. Because of the relatively small size and the inherent fluctuations, velocity scaling should not be performed if the temperature is within a small window of the desired temperature, i.e. $|T - T_D| < \epsilon$. Also, velocity scaling influences the trajectories significantly, which is why an additional factor f_s is normally used to moderate the strength of the scaling. The scaled velocities are given by

$$\boldsymbol{v}_i' = \boldsymbol{v}_i \sqrt{f_s \frac{T_D}{T}} \; . \tag{2.2.35}$$

2.2.10 Energy transfer scheme

Several schemes that are discussed in this work require the transfer of energy to or from a set of atoms. Since the energy of each atom has a potential and kinetic part, the energy can, at least in principle, be transferred to or from either of them. However, changes in the potential part would require the repositioning of atoms, which is difficult and entails a number of side effects. On the other hand, manipulating the kinetic part is relatively uncomplicated; kinetic energy can be inserted or removed by scaling the atomic velocity vectors. This is common practise in MD, but has the disadvantage of an involved momentum transfer due to the change of the velocity vectors. For several schemes used in this work it is crucial to eliminate any transfer of momentum when transferring energy. For this reason, an adapted energy transfer scheme is suggested that allows the transfer of kinetic energy only, without any transfer of momentum.

Let $\mathcal{S}$ be the set of atoms to which the energy E_{trans} is supposed to be transferred to. The mass $M_\mathcal{S}$ and the velocity $\boldsymbol{v}_\mathcal{S}$ of the set $\mathcal{S}$ are given by

$$M_\mathcal{S} = \sum_{i \in \mathcal{S}} m_i \quad \text{and} \quad \boldsymbol{v}_\mathcal{S} = \frac{1}{M_\mathcal{S}} \sum_{i \in \mathcal{S}} m_i \boldsymbol{v}_i \; . \tag{2.2.36}$$

These can be used to determine the total, internal and external kinetic energies, $E_{k\mathcal{S}}$, $E_{i,k\mathcal{S}}$ and $E_{e,k\mathcal{S}}$, by

$$E_{k\mathcal{S}} = \sum_{i \in \mathcal{S}} \frac{1}{2} m_i \boldsymbol{v}_i^2 \; , \quad E_{e,k\mathcal{S}} = E_{k\mathcal{S}} - \frac{1}{2} M_\mathcal{S} (\boldsymbol{v}_\mathcal{S})^2 \quad \text{and} \quad E_{i,k\mathcal{S}} = E_{k\mathcal{S}} - E_{e,k\mathcal{S}} \; . \tag{2.2.37}$$

In order for the total momentum of the set $\mathcal{S}$ to stay constant, the energy can only be transferred into or from internal kinetic energy. This is achieved by scaling the velocity vectors of the atoms in $\mathcal{S}$:

$$\boldsymbol{v}_i^{new} = \boldsymbol{v}_i^{old} f + \boldsymbol{c} \; , \tag{2.2.38}$$

with the scaling factor f. The additional correction vector $\boldsymbol{c}$ ensures the conservation of overall momentum. It can be shown that the scaling factor and the correction vector need to be

$$f = \sqrt{1 - \frac{E_{trans}}{E_{i,k\mathcal{S}}}} \; , \quad \text{and} \quad \boldsymbol{c} = \boldsymbol{v}_\mathcal{S}(f - 1) \tag{2.2.39}$$

for transferring the energy E_{trans}. It should be noted that Equation 2.2.39 is undefined if $E_{trans} < E_{i,k}$. This corresponds to the case where more energy shall be removed than is actually available as internal kinetic energy in the system. The energy transfer is impossible in that case.

In some of the investigated schemes, like the one described in Section 7.9, it is necessary to transfer the energy to a specific dimension. In that case, the energy to be transferred is a vector itself, $\boldsymbol{E}_{trans}$, describing which amount of energy is to be transferred into which dimension. Here, the scaled atomic velocities are

$$v_{i_\alpha}^{new} = v_{i_\alpha}^{old} + c_\alpha \ , \tag{2.2.40}$$

where the respective dimension is $\alpha = \{x, y, z\}$. The scaling vector $\boldsymbol{f}$ and the correction vector $\boldsymbol{c}$ are defined respectively as

$$f_\alpha = \sqrt{1 - \frac{E_{trans_\alpha}}{E_{i,k_{\mathcal{S}\alpha}}}} \quad \text{and} \quad c_\alpha = v_{\mathcal{S}\alpha}(f_\alpha - 1) \tag{2.2.41}$$

with $E_{i,k_{\mathcal{S}\alpha}}$ being the internal kinetic energy of $\mathcal{S}$ in dimension α that is calculated by

$$E_{i,k_{\mathcal{S}\alpha}} = \frac{1}{2} \sum_{i \in \mathcal{S}} m_i {v_{i_\alpha}}^2 \ . \tag{2.2.42}$$

The implemented scheme for the version with separated dimensions is given in Algorithm 3. Both, the general and the dimensional energy transfer can easily be proven analytically. The implemented algorithms have been tested on sample systems and proved their functionality on various test cases.

2.3 Computational fluid dynamics: The continuum model

The continuum model is not concerned with the motion of individual atoms. Instead, the properties of a large number of atoms are summarised to macroscopic quantities, where the most important are density ρ, flow velocity $\mathbf{u}$, energy e, temperature T and pressure p. Each quantity A is a function of position $\boldsymbol{x}$ in space and time t. In Cartesian form this is $A(\boldsymbol{x}, t) = A(x, y, z, t)$ for scalar quantities, i.e. $\rho = \rho(\boldsymbol{x}, t)$, $e = e(\boldsymbol{x}, t)$ and $T = T(\boldsymbol{x}, t)$. The equivalent notation for vector quantities, such as velocity is $\boldsymbol{u} = \boldsymbol{u}(\boldsymbol{x}, t) = \boldsymbol{i}u(x, y, z, t) + \boldsymbol{j}v(x, y, z, t) + \boldsymbol{k}w(x, y, z, t)$, where the velocity components u,v and w are combined to the vector $\boldsymbol{u} = (u, v, w)$. The Cartesian base vectors are $\boldsymbol{i}$, $\boldsymbol{j}$ and $\boldsymbol{k}$. All extensive quantities are given per unit volume, i.e. mass as density, momentum as $\rho\boldsymbol{u}$ and energy e is energy per unit volume. e is the total energy density which is the sum of internal and external energy density ($e = e_i + e_e$), where the external energy density e_e accounts for the motion of the fluid: $e_e = \frac{1}{2}\rho(u^2 + v^2 + w^2)$. The internal energy density e_i consists of a potential and kinetic part, $e_i = e_p + e_{i,k}$, and the total kinetic energy is $e_k = e_e + e_{i,k}$. The physics of the continuum model is based on the conservation of mass, momentum and energy, which is formulated in a set of equations known as the Navier-Stokes equations.

Algorithm 3: Energy transfer algorithm that takes into account the individual dimensions.

$M_{\mathcal{S}} \leftarrow 0$;
$\boldsymbol{v}_{\mathcal{S}} \leftarrow \mathbf{0}$;
$\boldsymbol{E}_{k\mathcal{S}} \leftarrow \mathbf{0}$;
foreach *atom i in* $\mathcal{S}$ **do**
  $M_{\mathcal{S}} \leftarrow M_{\mathcal{S}} + m_i$;
  $\boldsymbol{v}_{\mathcal{S}} \leftarrow \boldsymbol{v}_{\mathcal{S}} + m_i \boldsymbol{v}_i$;
  $\boldsymbol{E}_{k\mathcal{S}} \leftarrow \boldsymbol{E}_{k\mathcal{S}} + (1/2) m_i \boldsymbol{v}_i^2$;
$\boldsymbol{v}_{\mathcal{S}} \leftarrow \boldsymbol{v}_{\mathcal{S}} / M_{\mathcal{S}}$;
$\boldsymbol{E}_{i,k\mathcal{S}} \leftarrow \boldsymbol{E}_{k\mathcal{S}} - (1/2) M_{\mathcal{S}} \boldsymbol{v}_{\mathcal{S}}{}^2$;
if $-\boldsymbol{E}_{i,k\mathcal{S}} < \boldsymbol{E}_{trans}$ **then**
  Output error message that not enough internal energy is available;
  return;
for $\alpha = x, y, z$ **do**
  $f_\alpha = \sqrt{1 - E_{trans\alpha} / E_{i,k\mathcal{S}_\alpha}}$;
  $c_\alpha = v_{\mathcal{S}\alpha}(f_\alpha - 1)$;
foreach *atom i in* $\mathcal{S}$ **do**
  $v_{ix} \leftarrow v_{ix} f_x + c_x$;
  $v_{iy} \leftarrow v_{iy} f_y + c_y$;
  $v_{iz} \leftarrow v_{iz} f_z + c_z$;

These will be presented below. A more extensive presentation can be found in Anderson [11], Drikakis and Rider [43] or Versteeg and Malalasekera [184].

The equation that describes the conservation of mass is the *continuity equation*, which is given in its differential form as:

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \boldsymbol{u}) = 0 \; . \tag{2.3.1}$$

For a small fluid element, the first term on the left is the rate of change in time of density. The second term is the *convective term*, which describes the net out flow of mass across the boundaries of the fluid element.

Starting point for the formulation of the momentum equation is Newton's second law. The time rate of the change of the momentum $\rho \boldsymbol{u}$ of a fluid element with the volume V enclosed by surface S is equal to the total force acting on the volume V plus the net momentum across the surface S. Thus, the conservation law for the momentum can be written as

$$\begin{aligned} \frac{\partial}{\partial t} \int\int\int_V \rho \boldsymbol{u} \, \mathrm{d}V \;\; &= \;\; - \int\int_S \boldsymbol{u} (\rho \boldsymbol{u} \cdot \boldsymbol{n}) \, \mathrm{d}S \\ &\quad + \int\int\int_V \boldsymbol{F}_V \, \mathrm{d}V + \int\int_S \boldsymbol{F}_S \, \mathrm{d}S \; . \end{aligned} \tag{2.3.2}$$

where the first term on the right hand side is the convection of momentum across the surface S. The normal vector $\boldsymbol{n}$ is the outward pointing unit vector at each point of S. The volume forces $\boldsymbol{F}_V$ originate from inertial, gravitational or electromagnetic forces. The surface forces account for the fluid stresses at each point of the surface S. They are given as $\boldsymbol{F}_S = \boldsymbol{n} \cdot \boldsymbol{\Pi}$, where $\boldsymbol{\Pi} \equiv \sigma_{ij}$ $(i, j = x, y, z)$ is the stress tensor. Fluid stresses are caused by the static pressure p and the viscous stresses. Thus, the pressure tensor $\boldsymbol{\Pi}$ can be written as

$$\boldsymbol{\Pi} = -p\mathbf{I} + \mathbf{T} \,, \tag{2.3.3}$$

where $\mathbf{I}$ is the identity matrix. The viscous stress tensor $\boldsymbol{T}$ is defined as

$$\mathbf{T} = \begin{pmatrix} \tau_{xx} & \tau_{xy} & \tau_{xz} \\ \tau_{yx} & \tau_{yy} & \tau_{yz} \\ \tau_{zx} & \tau_{zy} & \tau_{zz} \end{pmatrix} , \tag{2.3.4}$$

It should be noted that $\sigma_{ij} = \tau_{ij}$ for $i \neq j$. Normally, the stress tensor, $\mathbf{T}$, is symmetrical, but in cases like in the presence of electrostatic fields, the symmetry may be broken [43].

Applying Gauss's theorem to transform a surface integral into a volume integral, Equation 2.3.2 can be recast into the differential form

$$\frac{\partial(\rho \boldsymbol{u})}{\partial t} + \nabla \cdot (\rho \boldsymbol{u} \otimes \boldsymbol{u}) = -\nabla p + \nabla \cdot \boldsymbol{\Pi} + \mathbf{F}_V \,. \tag{2.3.5}$$

The notation of the operators is explained in the appendix A. The tensor product $\boldsymbol{u} \otimes \boldsymbol{u}$ is given by

$$\boldsymbol{u} \otimes \boldsymbol{u} = \begin{pmatrix} u^2 & uv & uw \\ vu & v^2 & vw \\ wu & wv & w^2 \end{pmatrix} . \tag{2.3.6}$$

The third equation ensures the conservation of energy and describes how energy is transported. It is, therefore, called *energy equation* and given as

$$\frac{\partial \rho e}{\partial t} + \nabla \cdot (\boldsymbol{u} \rho e) = -\nabla \cdot p\boldsymbol{u} + \boldsymbol{u} \cdot \nabla \cdot \mathbf{T} + \nabla \cdot \boldsymbol{q} \,, \tag{2.3.7}$$

where the flux of heat is $\boldsymbol{q} = \lambda \nabla T$ with λ being the thermal conductivity.

The continuity, momentum and energy equations (Equations (2.3.1), (2.3.5) and (2.3.7)) that describe the motion of a fluid in three dimensions contain the variables pressure p and temperature T. These have not yet been linked to the other unknowns. Assuming that the fluid is locally in a thermodynamic equilibrium [184], the equilibrium *equations of state* from thermodynamics can be used to establish the missing links. Using ρ and e as state variables, the pressure p and temperature T are given by

$$p = p(\rho, e_i) \quad \text{and} \quad T = T(\rho, e_i) \,. \tag{2.3.8}$$

For a perfect gas, in which the potential energy density is zero ($e_p = 0$), these equations are

$$p = \rho RT \quad \text{and} \quad T = \frac{e_i}{c_v} , \tag{2.3.9}$$

where $e_i = e - \frac{1}{2}\rho(u^2 + v^2 + w^2)$. For a single atomic material, such as the LJ-fluid, the temperature is simply

$$T = \frac{2e_i m}{k_B \rho} . \tag{2.3.10}$$

Beside $p = p(\rho, e_i)$ and $T = (\rho, e_i)$, hybrid MD-CFD methods also require the knowledge of the relations between the potential, kinetic and internal energy densities:

$$e_p = e_p(\rho, e_i) \quad \text{and} \quad e_{i,k} = e_{i,k}(\rho, e_i) . \tag{2.3.11}$$

More complex fluids, such as multi-component fluids and/or fluids consisting of large molecules, require even more equations of state to describe the state of the fluid macroscopically.

The viscous stress tensor, $\mathbf{T}$, contains two different type of viscous stresses τ_{ij}, the shear stresses for $i \neq j$ and the normal stresses for $i = j$. In Newtonian fluids, the viscous stresses are related linearly to the strain rate, i.e. velocity gradients. For such fluids, Stokes obtained for the normal and shear stresses

$$\tau_{xx} = 2\mu\frac{\partial u}{\partial x} + \lambda_v(\nabla \cdot \boldsymbol{u}) \tag{2.3.12a}$$

$$\tau_{yy} = 2\mu\frac{\partial v}{\partial y} + \lambda_v(\nabla \cdot \boldsymbol{u}) \tag{2.3.12b}$$

$$\tau_{zz} = 2\mu\frac{\partial w}{\partial z} + \lambda_v(\nabla \cdot \boldsymbol{u}) \tag{2.3.12c}$$

$$\tau_{xy} = \tau_{yx} = \mu\left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x}\right) \tag{2.3.12d}$$

$$\tau_{xz} = \tau_{zx} = \mu\left(\frac{\partial u}{\partial z} + \frac{\partial w}{\partial x}\right) \tag{2.3.12e}$$

$$\tau_{yz} = \tau_{zy} = \mu\left(\frac{\partial v}{\partial z} + \frac{\partial w}{\partial y}\right) \tag{2.3.12f}$$

where μ is the dynamic (first) viscosity and λ_v is the bulk (second) viscosity. The bulk viscosity is difficult to obtain, but for gases a good working approximation is $\lambda_v = -\frac{2}{3}\mu$. For incompressible cases, e.g. liquids, $\nabla \cdot \boldsymbol{u} = 0$, so that the normal viscous stresses are zero.

Finally, the equation system that governs the fluid flow is given the conservative form in Table 2.1. In each of the Equations (2.3.13a) – (2.3.13c), the term on the left side is the change of the respective quantity in time. On the right hand side, the first terms are the convective terms, and the second terms are the diffusion terms.

$$\frac{\partial \rho}{\partial t} = -\nabla \cdot (\rho \mathbf{u}) \tag{2.3.13a}$$

$$\frac{\partial \rho \mathbf{u}}{\partial t} = -\nabla \cdot (\rho \mathbf{u} \otimes \mathbf{u}) - \nabla \cdot \mathbf{\Pi} \tag{2.3.13b}$$

$$\frac{\partial e}{\partial t} = -\nabla \cdot (e\mathbf{u}) - \nabla \cdot (\mathbf{\Pi} \cdot \mathbf{u}) - \nabla \cdot \mathbf{q} \tag{2.3.13c}$$

The stress tensor $\mathbf{\Pi}$ for a Newtonian fluid, the heat flux vector $\boldsymbol{q}$ and the equations of state for p and T given by

$$\mathbf{\Pi} = p\mathbf{I} - \lambda_v(\nabla \cdot \mathbf{u})\mathbf{I} - \mu[(\nabla \mathbf{u}) + (\nabla \mathbf{u})^T] , \tag{2.3.13d}$$

$$\boldsymbol{q} = \lambda \nabla T , \tag{2.3.13e}$$

$$p = p(\rho, e_i) \quad \text{and} \quad T = T(\rho, e_i) . \tag{2.3.13f}$$

Table 2.1: The complete Navier-Stokes equations in differential form.

CHAPTER 3

Calculation of Macroscopic Quantities

This chapter is concerned with the calculation of macroscopic variables. Different schemes for the assignment of atoms to grid points are explained. The necessary relations to calculate the macroscopic variables from detailed atomic information is given. Furthermore, fluctuations that arise from these calculations are discussed and the relevant estimations for the fluctuations strengths are discussed.

3.1 Particle to Mesh

The connection from the molecular model to the continuum model is accomplished by transferring the molecular variables into the macroscopic hydrodynamic variables. This is known as statistical physics. Statistical physics relates the properties of many individual atoms, i.e. their mass, position, velocities, forces et cetera to the macroscopic variables such as density, pressure or velocity. The quantities can be calculated globally for the entire simulation or locally for a small volume around a point. Using local calculations, a grid of points can be laid into the simulation domain and the calculations be performed locally for each grid point. A two-dimensional example is illustrated in Figure 3.1. By this, one may generate profiles, cross-sections, flow fields and similar graphs, which can be related to the results of continuum simulations.

Assigning the atomic properties to the grid nodes allows to translate the microscopic state into a grid based macroscopic presentation that is used by the CFD solver. This is a key element for the hybrid MD-CFD scheme and therefore some attention should be given to this. Schemes for assigning particles to grid points have been investigated in connection with particle-mesh techniques [94]. Originally, the particle-mesh technique was invented by Harlow [84] for the simulation of compressible flows. It is widely used in particle type simulation with long range interactions, such as electrostatic or gravitational forces. The main applications are simulations of plasmas, gravitational N-body systems and incompressible fluid flows (vortex method [115]). These methods use the field, such as the electrostatic or gravitational field, which is presented by the values at the grid nodes, to compute efficiently far field interactions that otherwise had to be obtained through an

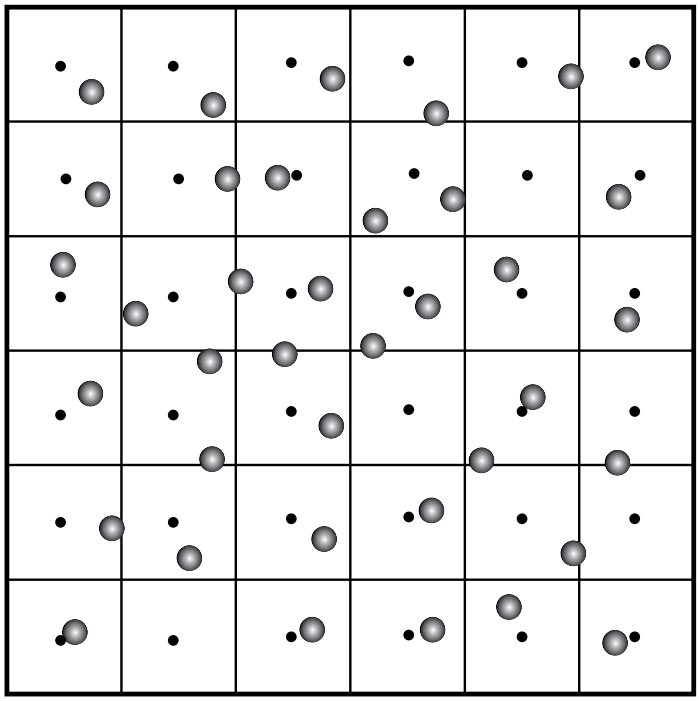


Figure 3.1: Illustration of a mesh laid over a molecular simulation domain. The grid nodes are displayed as small filled circles at the centre of each cell of the grid. The bigger shaded circles represent the atoms.

expensive particle to particle evaluation.

Using a three-dimensional cartesian grid, a node is identified by the indexes i, j, k; its coordinate is $\boldsymbol{x}_{(i,j,k)}$. Generally, a quantity $\mathcal{A}_{(i,j,k)}$ of a grid node i,j,k is calculated as an *instantaneous average* over a finite number of atoms N by

$$\mathcal{A}_{(i,j,k)} = \langle \mathcal{A}_{(i,j,k)} \rangle_w = \sum_{s=1}^{N} \mathcal{A}_s W(\boldsymbol{d}, \boldsymbol{l}) \tag{3.1.1}$$

where $\mathcal{A}_s$ is the property $\mathcal{A}$ of atom s, $\boldsymbol{d}$ the distance of atom s to node i,j,k ($\boldsymbol{d} = \boldsymbol{r}_s - \boldsymbol{x}_{(i,j,k)}$) and $\boldsymbol{l}$ the grid cell width. The weighting function $W(\boldsymbol{d}, \boldsymbol{l})$ is evaluated for each dimension separately:

$$W(\boldsymbol{d}, \boldsymbol{l}) = W(d_x, l_x) W(d_y, l_y) W(d_z, l_z) \; . \tag{3.1.2}$$

It depends on the distance d_α of the particle to the node and on the cell size l_α in the respective dimension α. Three types of weighting functions are commonly used: *nearest-grid-point* (NGP), *cloud-in-cell* (CIC) and *triangular-shaped-cloud* (TSC).

For NGP, which is the simplest scheme, the properties of an atom are accounted com-

pletely to the nearest node. The NGP weighting function is defined as

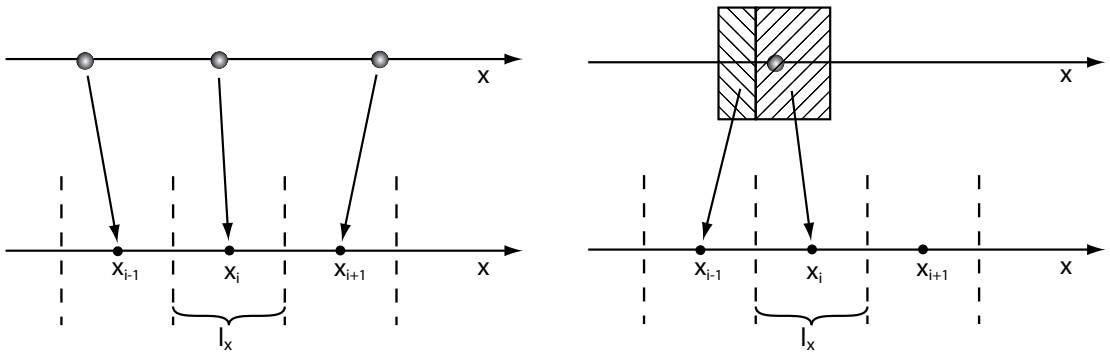
$$W_{NGP}(d_\alpha, l_\alpha) = \begin{cases} 1 & d_\alpha \leq \dfrac{l_\alpha}{2} \\ \\ 0 & \text{otherwise} \end{cases} . \tag{3.1.3}$$

As an example, the assignment of three atoms is illustrated on the left hand side of Figure 3.2. The scheme requires minimum computing time, but because it is of zeroth order, it shows a discontinues behaviour when atoms move from one cell to another cell. This introduces noise into the calculations.

In the CIC scheme, atomic properties are considered as a uniform distributed cloud centered about the atomic position. This is illustrated for one atom on the right hand side of Figure 3.2. The scheme is linear and the weighting function is given as

$$W_{CIC}(d_\alpha, l_\alpha) = \begin{cases} 1 - \dfrac{d_\alpha}{l_\alpha} & d_\alpha \leq l_\alpha \\ \\ 0 & \text{otherwise} \end{cases} . \tag{3.1.4}$$

Though the computational costs are slightly higher than for the NGP scheme, the numerical properties are better because it reduces spacial and temporal fluctuations [94]. Particle to mesh algorithms are also used for the calculation of long range forces from the first derivative of the potential field. When using the CIC scheme the first derivative is discontinues and introduces jumps in the forces. For this reason, the TSC scheme was derived. By assuming a triangular shape distribution of the atomic properties, it is continuous also for the first derivative. However, more computational effort is required and because no forces need to be calculated from the potential field, this scheme is not considered here.



(a) NGP **(b)** CIC

Figure 3.2: Weighting function for assigning atoms to a grid node. The left hand side represents the NGP scheme and the right hand side the CIC scheme.

3.2 Calculation of specific quantities

Basic quantities

In the previous section, the assignment of atoms to grid nodes was discussed. This part deals with the concrete formulas for the calculation of the macroscopic quantities from the properties of the individual atoms. Table 3.1 contains the calculation formulas for the basic quantities for a cell node i, j, k from the collectivity of the all atoms, which are indicated by the subscript s. The weighting function $W(\boldsymbol{d}, \boldsymbol{l})$ has been replaced by W_s for reasons of brevity.

Quantity	Symbol		Formula	Equation
mass	$M_{(i,j,k)}$	=	$\sum_s W_s m_s$	(3.2.1)
density	$\rho_{(i,j,k)}$	=	$\frac{M_{(i,j,k)}}{V_{(i,j,k)}}$	(3.2.2)
velocity	$\boldsymbol{u}_{(i,j,k)}$	=	$\frac{1}{M_{(i,j,k)} V_{(i,j,k)}} \sum_s W_s m_s \boldsymbol{v}_s$	(3.2.3)
potential energy density	$e_{p_{(i,j,k)}}$	=	$\frac{1}{V_{(i,j,k)}} \sum_s W_s E_{p_s}$	(3.2.4)
kinetic energy density	$e_{k(i,j,k)}$	=	$\frac{1}{V_{(i,j,k)}} \sum_s W_s m_s \frac{1}{2} \boldsymbol{v}_s^{\,2}$	(3.2.5)
external kinetic energy density	$e_{e,k_{(i,j,k)}}$	=	$\frac{1}{2} \rho_{(i,j,k)} \, \boldsymbol{v}_{(i,j,k)}^{\,2}$	(3.2.6)
internal kinetic energy density	$e_{i,k_{(i,j,k)}}$	=	$e_{k(i,j,k)} - e_{e,k_{(i,j,k)}}$	(3.2.7)
total energy density	$e_{(i,j,k)}$	=	$e_{p_{(i,j,k)}} + e_{k(i,j,k)}$	(3.2.8)
degrees of freedom	$f_{(i,j,k)}$	=	$\left(\sum_s 3 W_s \right) - 3$	(3.2.9)
temperature	$T_{(i,j,k)}$	=	$\frac{2 e_{i,k_{(i,j,k)}} \, V_{(i,j,k)}}{f_{(i,j,k)} k_B}$	(3.2.10)

Table 3.1: Calculation of the molecular quantities.

3.2.1 Pressure and stress tensor

For hybrid MD-CFD methods the calculation of the scalar pressure and the stress tensor of molecular system are fundamental and critical. One reason is that the pressure obtained from the equation of state (Equation 2.3.13f, $P = P(\rho, e_i)$) is used in the continuum formulation and must fit the pressure actually exerted by the molecular system for the same state, which is defined by density ρ and internal energy e_i. Generally, $P(\rho, e_i)$ is calculated

in advance for the respective molecular system and it is, therefore, important that the pressures calculated in that stage are correct. Depending on the coupling scheme, it may also be necessary to calculate the local pressure or stress tensor during the simulation. Unlike in the macroscopic limit, where the stress tensor is well defined, its definition from the molecular point of view is less clear. Here, the stress tensor arises from atomic velocities and interatomic forces, which result from n-body interactions (pair interactions, bond bending, bond torsion, ...). Schemes for the calculation of the local and average stress tensor have been a subject of research since the early MD simulations [99] and still are [88, 203]. The different schemes that have been suggested include the method of Irving and Kirkwood [99], the method of planes [178], and the virial theorem [127].

The most often applied method is the one based on the virial theorem, from which the instantaneous stress tensor is given by [88]

$$\mathbf{\Pi} = \frac{1}{V} \sum_i m_i \boldsymbol{v}_i \otimes \boldsymbol{v}_i + \sum_i \boldsymbol{f}_i \cdot \boldsymbol{r}_i \ , \tag{3.2.11}$$

with the first sum being the kinetic pressure and the second sum the pressure due to forces. The advantage of Equation (3.2.11) is that it uses the total force on each atom and is therefore easy to implement and computationally efficient. Unfortunately, it is only appropriate for the calculation of the average pressure of a simulation box with periodic boundary conditions and not for local stress evaluations [203]. Schemes suitable for the calculation of the local stress tensor are based on the evaluation of the individual forces between atoms. For example, Irving and Kirkwood [99] evaluate the forces across an imaginary plane to obtain the pressure due to forces. Each element $\Pi_{\alpha\beta}$ of the pressure tensor is given for a small volume $\delta V = (\delta\alpha)^3$ with the side length $\delta\alpha$ by

$$\Pi_{\alpha\beta} = \frac{1}{2A_\alpha \delta\alpha} \sum_i m_i v_{i\beta} v_{i\alpha} + \frac{1}{A_\alpha} \sum_{\boldsymbol{r}_{ij} \cap A_\alpha} f_{ij\beta} \ . \tag{3.2.12}$$

where the first sum is the kinetic pressure. The second sum, the pressure due to forces, runs over all pair interactions $\boldsymbol{r}_{ij}$ that act across an imaginary plane A_α ($\boldsymbol{r}_{ij} \cap A_\alpha$). The plane A_α is located at the centre of δV, its surface normal vector is parallel to α and its surface area is A_α. Equation (3.2.12) is only valid for pair interactions. Heinz et al. [88] extended the method to n-body interactions. The disadvantage of these methods is the computationally expensive force evaluation, which is why they are not considered in this work.

Independent on the specific calculation of the pressure tensor, the scalar pressure is obtained from the stress tensor by averaging the main diagonal:

$$P = \frac{\Pi_{xx} + \Pi_{yy} + \Pi_{zz}}{3} \ . \tag{3.2.13}$$

Another method, employed here, for calculating the scalar pressure makes use of the setup which has two reflecting walls that enclose the molecular region in one dimension (the other two dimensions are periodical). The pressure on the walls can be calculated by

integrating the momentum change of the particles that are reflected. The scalar pressure is then given by

$$P = \frac{1}{A\delta t_m} \sum_i 2m_i \boldsymbol{v}_i \ , \tag{3.2.14}$$

where the sum is over the reflected atoms i. The velocity $\boldsymbol{v}_i$ is the one before the reflection which reverses $\boldsymbol{v}_i$ to $-\boldsymbol{v}_i$. It should be noted that this approach is not applicable for negative pressures.

Because the virial based method is only suitable for periodic boundary conditions and a system using a HSI is essentially non-periodic, the virial method can not be applied in that case. To confirm this, pressure calculations of a system with 5600 LJ-atoms in gaseous, liquid and solid state have been performed. First, periodic boundary conditions were applied in all three dimension. Afterwards, periodic boundary conditions were only applied in y- and z-dimension, whereas the x-dimension was constrained by two reflective walls. The scalar pressures calculated by Equation (3.2.11) for both types of boundary conditions differed by approximately 9%. On the other hand, the scalar pressure calculated by Equation (3.2.14) did correspond to the pressure obtained by Equation (3.2.11) for the fully-periodic system. These values also agreed with the equation of state published by Meier [131]. Therefore, Equation (3.2.14) was used for the pressure calculation in this work, especially to obtain the pressure values for the equation of state, $P(\rho, e_i)$.

3.3 Statistical considerations

Because of the small scales considered here, calculations of any hydrodynamic quantity $\mathcal{A}$, such as density, velocity or pressure, within one grid cell is obtained from a relatively small number of atoms. Therefore, these calculated quantities will be subject to fluctuations, both in space and time. These originate from the thermal motion of the atoms and may also be called thermal fluctuations or noise.

It is important to be able to quantify these fluctuations for several reasons. When analysing results of any simulation, one needs to give the statistical error as a grade of the precision, just like in any experiment. In fact, molecular simulations are computational experiments. In some cases, for example when measuring the flow velocity for small Mach-numbers, the fluctuations may exceed the actual quantity by several orders of magnitude. For MD-CFD coupling, the fluctuations will be transferred from the MD-domain to the continuum-domain where it may cause stability problems for the CFD-solver. In both cases, the method of calculation has to be adapted in order to keep the fluctuation below a desired level.

For reducing the strength of the fluctuations, one can increase the number of atoms from which the respective quantity is calculated. This is achieved by increasing the volume of the cell for which the calculation is performed. However, this approach is limited due a possible required minimum grid resolution, or in relation to the MD-CFD hybrid simulation, through the thickness of the HSI-layer or the requirements of the coupling scheme. Often, simulations are run in three dimensions even if the problem is effectively of one or

two-dimensional nature. In such cases, the cells can be stretched in the third dimension to increase the volume. The only limitations are the computational resources available. Using a particle to mesh scheme like CIC, as described in the previous section, also helps to reduce the fluctuations.

The next three sections focus on the reduction and estimation of fluctuations. The discussion is started with time averaging and the involved time correlation issues. Subsequently, estimates for the fluctuation strength are given.

3.3.1 Time averaging

Another way of reducing statistical fluctuations is to apply additional averaging for which two types are available. Firstly, one can average over a number of molecular systems. This is called *ensemble average* and is obtained when a simulation or an experiment is carried out repeatedly with slightly different initial conditions. The ensemble average is more useful for probabilistic simulations, like DSMC, but for MD it would necessitate running the same simulation with alternated initial positions and/or velocities. This is not practical and therefore the second type of averaging, which is *time averaging*, is usually employed. Here, the instantaneous values of $\mathcal{A}$ are averaged over a period of time δt_{av}. For steady state problems, the only limit of the length of the period, are computational resources. For unsteady problems, the characteristic time scales of the changes in $\mathcal{A}$ pose an upper limit to δt_{av}. When ensemble and time averages are identical, then the molecular motion is called *ergodic*. In this work, only time averaging is used.

Averaging over a period of time δt_{av}, a macroscopic quantity $\mathcal{A}$ at time t_m is given by

$$\langle \mathcal{A} \rangle_t = \frac{1}{\delta t_{av}} \int_{t_m}^{t_m+\delta t_{av}} \mathcal{A}(t)\mathrm{d}t = \frac{1}{N_m} \sum_{\tau_m=1}^{N_m} \mathcal{A}(\tau_m) \,, \tag{3.3.1}$$

where the second equation is the discretised form, for which $\mathcal{A}$ is calculated N_m times during δt_{av}. The averaging time is related to the number of calculations by $\delta t_{av} = N_m \delta t_m$. The measuring time step δt_m is a multiple of the simulation time step: $\delta t_m = \tau_{mes} \delta t$. The relation factor τ_{mes} is any positive integer. In practise, its minimum value is dictated by the correlation time.

3.3.2 Correlation time

For time averaging, one also has to take into account that subsequent time steps in molecular simulations are correlated. Especially in MD simulations at low densities, the changes from one time step to the next are small and therefore macroscopic variables are correlated over many subsequent time steps. Hence, it is inefficient to perform the calculation at every simulation time step ($\delta t_m = \delta t$). Additionally, there is a danger of underestimating the variance of the fluctuations [2]. For these reasons, τ_{mes} should be chosen large enough so

that subsequent calculations are uncorrelated.

A simply way of obtaining the correlation time is to read it off the autocorrelation function over time lags. The autocorrelation function of a variable $\mathcal{A}$ for a series of time lags is calculated by

$$\phi(\mathcal{A}, \tau_c) = \frac{1}{N_m - \tau_c} \sum_{\tau_m=1}^{N_m-\tau_c} \mathcal{A}(\tau_m)\mathcal{A}(\tau_m + \tau_c) \; , \tag{3.3.2}$$

where τ_c is the time lag and the sum runs over the products of the variable of interest at the times τ_m and $\tau_m + \tau_c$. N_m is the number of total calculations available. The most useful form is probably the correlation coefficient depending on the time lag. It is defined as the covariance $cov(\mathcal{A}, \tau_c)$ over the variance $\sigma^2(\mathcal{A})$:

$$\rho(\mathcal{A}, \tau_c) = \frac{cov(\mathcal{A}, \tau_c)}{\sigma^2(\mathcal{A})} \; , \tag{3.3.3}$$

$$cov(\mathcal{A}, \tau_c) = \frac{1}{N_m - \tau_c} \sum_{\tau_m=1}^{N_m-\tau_c} (\mathcal{A}(\tau_m) - \langle\mathcal{A}\rangle_t)(\mathcal{A}(\tau_m + \tau_c) - \langle\mathcal{A}\rangle_t) \; . \tag{3.3.4}$$

3.3.3 Fluctuation strength and confidence interval

A useful measure of the strength of fluctuations in time is the standard deviation of $\langle\mathcal{A}\rangle_\tau$. It describes the average deviation of the calculated and time averaged value $\langle\mathcal{A}\rangle_\tau$ from the true value of $\mathcal{A}$. Only the necessary relations to calculate the fluctuation strength are included in this chapter. For a detailed description please refer to Allen and Tildesley [2], Chap. 6, Rapaport [153], chap. 4.2 and Hadjiconstantinou et al. [72].

The underlying assumption is that fluctuations in time of $\mathcal{A}(t)$ are Gaussian distributed. If the individual calculated quantities $\mathcal{A}(\tau)$ are independent from each other, i.e. they are not correlated and have been obtained at time steps which are separated more than the correlation time ($\tau_m > \tau_c$), the standard deviation of $\langle\mathcal{A}\rangle_\tau$ is defined by

$$\sigma(\langle\mathcal{A}\rangle_t) = \sqrt{\sigma^2(\langle\mathcal{A}\rangle_t)} \; , \tag{3.3.5}$$

with $\sigma^2(\langle\mathcal{A}\rangle_t)$ being the variance of $\langle A\rangle_t$ that is simple given by

$$\sigma^2(\langle\mathcal{A}\rangle_t) = \frac{\sigma^2(\mathcal{A})}{N_m} \; , \tag{3.3.6}$$

where $\sigma^2(\mathcal{A})$ is the variance of the individual values $\mathcal{A}$. When obtained from a sufficient large number of time steps, instead of the strictly unknown variance $\sigma^2(\mathcal{A})$ the sample variance $s^2_{N-1}(\mathcal{A})$ can be used [25, chap. 16.3]. The variance and the unbiased squared sample variance are then given by

$$\sigma^2(\mathcal{A}) = s^2_{N_m-1}(\mathcal{A}) = \frac{1}{N_m - 1} \sum_{\tau_m=1}^{N_m} (\mathcal{A}(\tau_m) - \langle\mathcal{A}\rangle_t)^2 \; . \tag{3.3.7}$$

The average deviation from the mean, the standard deviation, is $\sigma(\mathcal{A}) = \sqrt{\sigma^2(\mathcal{A})}$ for the instantaneous values and $\sigma(\langle\mathcal{A}\rangle_t) = \sqrt{\sigma^2(\langle\mathcal{A}\rangle_t)} = \sigma(\langle\mathcal{A}\rangle_t)/\sqrt{N_m}$ for the time average respectively.

Confidence interval

To obtain a statistically significant average of $\mathcal{A}$, the minimum number of values $N_{m,min}$ from which the average needs to be taken, can be calculated for a chosen confidence level P_{cl} and a relative confidence interval δl_{ci}. For the confidence interval

$$\mathrm{CONF}_{P_{cl}}\left\{\langle\mathcal{A}\rangle_t(1-\delta l_{ci}) \leq \mathcal{A} \leq \langle\mathcal{A}\rangle_t(1+\delta l_{ci})\right\} , \tag{3.3.8}$$

the minimum sample size is obtained from the relation

$$N_{m,min} = 2\,\sigma^2(\mathcal{A})\left(\frac{\mathrm{erf}^{-1}(P_{cl})}{\delta l_{ci}\langle\mathcal{A}\rangle_t}\right)^2 , \tag{3.3.9}$$

where $\mathrm{erf}^{-1}(P_{cl})$ is the inverse error function [114]. The corresponding minimum number of MD time steps is then $N_{\tau,min} = N_{m,min}\tau_c$, where τ_c is the correlation time.

Fluctuations

In the MD-CFD hybrid schemes, the state variables density ρ, velocity $\boldsymbol{u}$ and energy density e are calculated in the molecular dynamics domain and imposed as boundary conditions on the continuum domain. The knowledge about fluctuations in ρ, $\boldsymbol{u}$ and e are therefore necessary for the construction of the HSI. For a time averaged quantity $\langle\mathcal{A}\rangle_t$, the average fluctuation strength can be defined as the fractional error, which is the standard deviation of $\langle\mathcal{A}\rangle_t$ over its true value:

$$F_{\langle\mathcal{A}\rangle_t} = \frac{\sigma(\langle\mathcal{A}\rangle_t)}{|\mathcal{A}|} = \frac{\sqrt{\sigma^2(\langle\mathcal{A}\rangle_t)}}{|\mathcal{A}|} . \tag{3.3.10}$$

Now, the most interesting question is: over how many values M must one average to reduce the fractional fluctuation below an acceptable level, for instance below 5 % ($F_{\langle\mathcal{A}\rangle_t} \leq 0.05$). To this end, Equation (3.3.10) can be rearranged to give the minimum value for $N_{m,min}$:

$$N_{m,min} = \frac{\sigma^2(\mathcal{A})}{F^2{\mathcal{A}_0}^2} , \tag{3.3.11}$$

where $\mathcal{A}_0$ is the true value or the limit of the average: $\mathcal{A}_0 = \lim_{t\to\infty}\langle\mathcal{A}\rangle_t$.

Of course, the fluctuations can be computed in preliminary test runs before the actual simulations, but this effort may be saved by using the theoretical derived expressions of

Hadjiconstantinou et al. [72] for fractional errors in gases and dense fluids. The predictions are given for: density, ρ; velocity in one dimension, u_α; internal energy density e; temperature, T; and pressure, P. Since the fluctuation strength is based on the standard deviations $\sigma(A)$ of the variables (q.v. Equation (3.3.10)) one has to start with those definitions:

$$\sigma(\rho) = \frac{m\sqrt{N}}{VAc} \tag{3.3.12}$$

$$\sigma(u_\alpha) = \sqrt{\frac{k_B T}{mN}} \tag{3.3.13}$$

$$\sigma(e_{int}) = \sqrt{\frac{k_B T^2 c_{V(P)} N}{V^2}} \tag{3.3.14}$$

$$\sigma(T) = \sqrt{\frac{k_B T^2}{c_{V(P)} N}} \tag{3.3.15}$$

$$\sigma(P) = \frac{k_B T Ac\sqrt{\gamma N}}{V} \,. \tag{3.3.16}$$

It is assumed that the variables of interest are calculated from N atoms. The acoustic number Ac is the ratio of the speed of sound of the fluid, c, to the speed of sound of an ideal gas, a^i: $Ac = a/a^i$. k_B is Boltzmann's constant, m the molecular mass, $c_{V(P)}$ the isochoric heat capacity per particle and γ the ratio of the heat capacities: $\gamma = c_{P(P)}/c_{V(P)}$.

From the instantaneous standard deviations of Equations (3.3.12) – (3.3.16) one easily obtains the time averaged standard deviation (q.v. Equation (3.3.7)). These can be related to the absolute variable values: ρ_0 for the density; $u_{0\alpha}$ for velocity in dimension α; e_{int0} for the internal energy density; T_0 for temperature; and P_0 for pressure, to give a prediction for the fluctuation strength (q.v. Equation (3.3.10)):

$$F_{\langle\rho\rangle_t} = \frac{\sigma(\langle\rho\rangle_t)}{\rho_0} = \frac{1}{\sqrt{N_m N}} \frac{1}{Ac} \tag{3.3.17}$$

$$F_{\langle u\rangle_t} = \frac{\sigma(\langle u_\alpha\rangle_t)}{|u_{0\alpha}|} = \frac{1}{\sqrt{N_m N}} \frac{1}{AcMa\sqrt{\gamma}} \tag{3.3.18}$$

$$F_{\langle e_{int}\rangle_t} = \frac{\sigma(\langle e_{int}\rangle_t)}{e_{i0}} = \frac{1}{\sqrt{N_m N}} \frac{TN\sqrt{k_B c_{V(P)}}}{e_{int0} V} \tag{3.3.19}$$

$$F_{\langle T\rangle_t} = \frac{\sigma(\langle T\rangle_t)}{T_0} = \frac{1}{\sqrt{N_m N}} \sqrt{\frac{k_B}{c_{V(P)}}} \tag{3.3.20}$$

$$F_{\langle P\rangle_t} = \frac{\sigma(\langle P\rangle_t)}{P_0} = \frac{1}{\sqrt{N_m N}} \frac{k_B T AcN\sqrt{\gamma}}{P_0 V} \,, \tag{3.3.21}$$

where Ma is the local mach-number: $Ma = u_{0\alpha}/c$. For the cases considered here, i.e. mono atomic substances, $c_{V(P)} = (3/2)k_B$.

Rearranging Equations (3.3.17) – (3.3.21), the minimum number of calculation, $N_{m,min}$, over which one needs to average to reduce the fractional fluctuations F below a desired level can be determined:

$$N^{\rho}_{m,min} = \frac{1}{F^2}\frac{1}{NAc^2} \tag{3.3.22}$$

$$N^{u}_{m,min} = \frac{1}{F^2}\frac{1}{NAc^2Ma^2\gamma} \tag{3.3.23}$$

$$N^{e_i}_{m,min} = \frac{1}{F^2}\frac{k_B T^2 c_{V(P)} N}{(e_{i0})^2 V^2} \tag{3.3.24}$$

$$N^{T}_{m,min} = \frac{1}{F^2}\frac{k_B}{Nc_{V(P)}} \tag{3.3.25}$$

$$N^{P}_{m,min} = \frac{1}{F^2}\frac{{k_B}^2 T^2 Ac^2 N^2 \gamma}{P_0 V} \; . \tag{3.3.26}$$

3.4 Automatic calculation simulation mode (ACSM)

For calculating a quantity, it is necessary to know about the accuracy of the calculation. This is true for experiments as well as for simulations. For equilibrium MD simulations, the accuracy can be increased arbitrarily, within the computational resources, to a desired level by running the simulation longer and averaging over more calculated values. A practical problem one faces is that the variance $\sigma(\mathcal{A})$ and correlation time τ_c are usually not known in advance. Hence, the required minimum number of values $N_{m,min}$ for achieving the required accuracy is unknown and thus the minimum number of time steps $N_{\tau,min}$ which need to be performed as well. Thus, one needs to run the simulation for a period of time $N_{\tau,mp}$ which seems appropriate and to calculate the mean, correlation time, variance and confidence level from the values obtained during this run. If the confidence level is too low, the simulation has to be extended to the estimated minimum number of integration time steps (Equation 3.3.9) with a new analysis being carried out. This procedure is repeated until the confidence level has reached the required level. Several iterations may be necessary, since the estimates variance, correlation time and confidence level change as the number of calculated values increases.

Executing the described procedure manually, by using statistical software tools[1] for the analysis is elaborate and time consuming. Therefore, an *automatic calculation simulation mode* (ACSM) has been developed by the author and implemented into the MD-code. It provides a fast and convenient way to calculate variables in equilibrium MD simulations without the requirement of human interaction. The basic principle is to extended the MD simulation automatically until the mean value of the calculated variable satisfies the chosen statistical confidence criteria. As input parameters, the procedure requires

1. the calculation frequency τ_{mes};

[1] Such as the mathematical software package MATLAB [119]

2. the calculation period $N_{\tau,mp}$;
3. the threshold of the time correlation coefficient ρ_{max};
4. the confidence interval δl_{ci};
5. the confidence level P_{cl}.

The flow chart of the ACSM operation mode is plotted in Figure 3.3. During the first part, the molecular system is initialised and run until a thermal equilibrium has been reached. Subsequently, the simulation is run in equilibrium state for $N_{\tau,mp}$ time steps. Thereby, the variable of interest is calculated every τ_{mes} time step and appended to an array that contains all calculated values. The simulation run is followed by the statistical analysis of the recorded array of values to calculate the mean $\langle \mathcal{A} \rangle_t$, variance $\sigma^2(\mathcal{A})$, correlation time τ_c, minimum number of values $N_{m,min}$ and eventually minimum number of time steps to run, $N_{\tau,min}$. The results of the statistical analysis are printed into a file to inform the user about the progress of the calculation and to provide an estimate of how many time steps are still to be performed. The ACSM is finished if the total number of integrated time steps is bigger than the calculated minimum ($N_\tau > N_{\tau,min}$). Otherwise, the simulation is run for another $N_{\tau,mp}$ time steps.

The correlation time is determined by calculating the correlation coefficient $\rho(\mathcal{A}, \tau_c)$ using Equation 3.3.3 for τ_c increasing stepwise from $\tau_c = 1$ by 1 until $\rho(\mathcal{A}, \tau_c) < \sigma_{max}$. For $\tau_c = 0$, the correlation coefficient is unity ($\rho(\mathcal{A}, 0) = 1$) and will approach zero for increasing τ_c.

The ACSM can be used to conveniently calculate any quantity from a MD simulation in thermal or dynamic equilibrium to the desired confidence level. It should be noted that the procedure conditions the calculated variable to be, in principle, constant in time, apart from the fluctuations. Situations in which this condition is not satisfied will lead to meaningless results.

3.4.1 State variables for LJ-material

In later chapters, the state variables: potential energy density e_p, kinetic energy density e_k and static pressure P_s are required for different combinations of density and temperature of the LJ potential modelled material. These variables have been calculated, by using the ACSM to obtain the desired accuracy. All simulation parameters were chosen to match the ones that were used for the following simulations in order to be able to use the calculated quantities there. The simulations were run in reduced LJ-units (q.v. Section 2.1.5). The LJ potential was shifted to give zero at the cut-off distance ($r_c = 2.5$, $\mathcal{V}^{LJ}(r_c) = 0$). The employed integration time step was $\delta t = 5 \times 10^{-3}$. It should be noted that for ensuring a perfect total energy conservation, the time step would have to be chosen one order of magnitude smaller, i.e. $\delta t = 5 \times 10^{-4}$. However, for all subsequent simulations $\delta t = 5 \times 10^{-3}$ was employed to limit the computational effort. Therefore, this time step was used here as well. To compensate the minor total energy drifts, a weak temperature scaling was employed every 1,000 time steps with a scaling factor $f_s = 0.1$. A comparison run with $\delta t = 5 \times 10^{-4}$

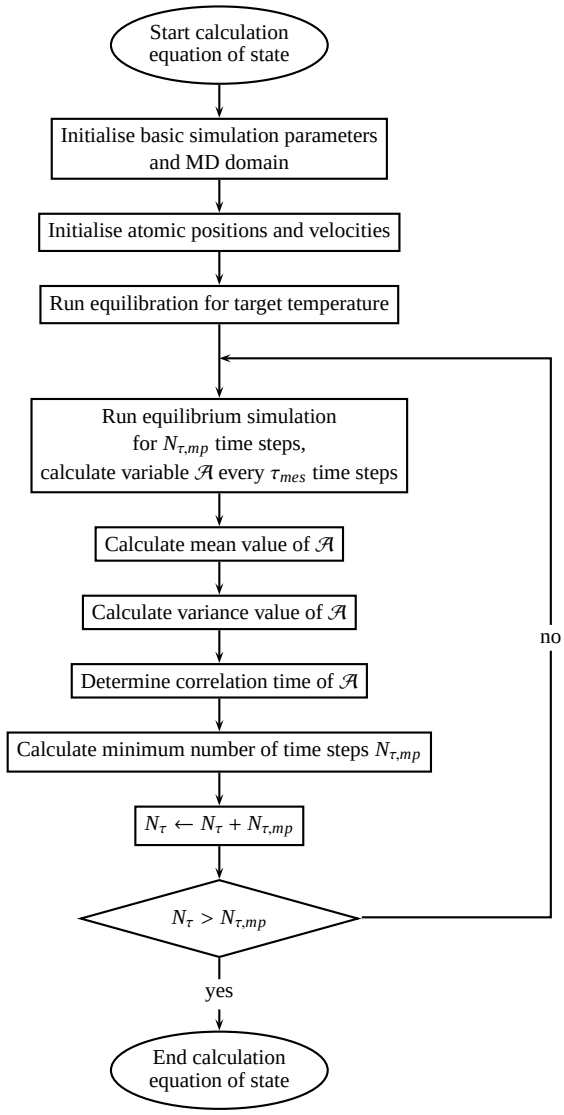


Figure 3.3: Flow chart of the automatic calcuation procedure that runs the simulation until the calculated value satisfies the confidence criteria.

showed basically no differences in the results.

For each state the system was initialised with a fcc lattice with $\boldsymbol{L}_l = (20, 5, 5)$ unit cells, resulting in 2,000 atoms. The lattice filling scheme that was used for this task is explained in Section 2.2.8. To calculate the bulk energies, periodic boundary conditions have been applied in all dimensions. For the calculation of the pressure through the wall method (q.v. Section 3.2.1), the system was confined by two reflective planes in the x-dimension. The setup is identical to the one used for the static test case in Section 6.1 which is illustrated in Figure 6.1. For each state, the system was equilibrated to the respective temperature before the ACSM procedure was started. The calculation frequency

was set to $\tau_{mes} = 10$ time steps and the simulation intervals were $N_{\tau,mp} = 10,000$. The potential energies were computed by using Equation (3.2.4) and the static pressure from the reflective wall through Equation (3.2.14).

In Table 3.2 the calculated values for the variables $e_p(\rho, T)$, $e_k(\rho, T)$ and $p(\rho, T)$ are given. The values for the kinetic energy density are exact through the prescribed temperature, $e_k = \frac{3}{2} n k_b T$. The correlation time τ_{corr} is the time step τ_c, where the threshold of the time correlation coefficient has been $\rho_{max}(\mathcal{A}, \tau_c) = 0.1$. The chosen confidence level for all cases was 99 %. The left and right sided confidence interval is given relative to the mean value and ranges from $\delta l_{ci} = 1\,\%$ to $\delta l_{ci} = 0.01\,\%$. The concrete size of the confidence interval was chosen based on the computational costs to achieve the required accuracy and the absolute value of the respective variable. Considering a concrete value from table 3.2, for example, $p(\rho = 0.1, T = 2) = 0.19$, it implies that the mean value of a new series of calculations would be within the interval $[0.188, 0.191]$ with 99 % certainty.

State		Variable	ACSM statistical analysis					
ρ	T		Mean	Std. dev	τ_{corr}	δl_{ci}	$N_{m,min}$	$N_{\tau,min}$
0.1	2	e_p	-0.0555	1.83×10^{-6}	24	0.1 %	4,001	960,240
		e_k	0.3	— exact value —				
		P_s	0.19	0.037	1	1 %	67,694	676,940
0.8	1	e_p	-3.7512	0.0139	2	0.01 %	9,086	181,720
		e_k	1.2	— exact value —				
		P_s	1.63	0.865	1	1 %	21,562	215,620
1.2	1	e_p	-6.659	0.0239	1	0.01 %	8,525	85,250
		e_k	1.8	— exact value —				
		P_s	17.25	51.4	1	0.1 %	188	18,800

Table 3.2: Calculated state variables for the used LJ-material including the statistical accuracy and minimum time steps. All quantities are given in LJ-units (q.v. Table 6.1).

The last two columns contain the minimum number of calculations $N_{m,min}$ and integration time steps $N_{\tau,min}$. For each considered state, the simulation has been run for the respective maximum $N_{\tau,min}$. From the data it is obvious that for all states energies can be calculated with much higher precision than the pressure with comparable computational effort. When comparing the number of required integration time steps, one needs to bear in mind that the computational effort per time step is less for lower densities than for higher ones, because of the lower number of interactions that need to be computed.

The ACSM has shown to be an excellent tool for calculating any equilibrium variable with a desired precision. After defining the desired correlation time and confidence criteria the procedure runs independently and the investigator can devote their time to other tasks.

CHAPTER 4

The Molecular Dynamics Code and Its Verification

This chapter describes LAMMPS, the molecular dynamics code, that has been used for this work. The most important fundamental properties of LAMMPS are presented including the modules that have been added within this work. LAMMPS has been verified, based on momentum and energy conservation, the radial distribution function of Argon and the collision dynamics of binary cluster collisions.

The implementation of hybrid MD-CFD coupling methods with geometrical coupling poses a number of special requirements on the MD code. This is due to the fact that the boundary of the MD domain at the interface to the CFD domain allows in- and outflow of material. For this reason, the number of atoms within the MD domain is not constant throughout the simulation.

Traditionally, MD codes have been developed to simulate a fixed number of atoms. The internal data and source code structure of these is based on a constant number of atoms. For this reason, it would require an inappropriately high programming effort to change such MD codes in a way that they could handle a varying number of atoms. Consequently, such MD codes are not suitable for hybrid MD-CFD simulations.

It was necessary to find a MD code that allowed the insertion and removal of atoms during the actual simulation. At the beginning of the PhD, a MD code with the desired properties was not available. Thus, the author started the development of a MD code that would be suitable for geometrical coupling of MD and CFD. However, the development and especially the verification of a MD code that is flexible, parallelised and can accommodate potentials for a broad range of materials is an extensive task, in particularly within the time frame of a PhD. Therefore, after the publication of LAMMPS [148], it was decided to use this MD code. The reason why LAMMPS is suitable for hybrid MD-CFD coupling originates in the way in which the parallelisation is achieved. In LAMMPS, the MD domain is basically subdivided into a cartesian grid of individual sub domains, where each sub domain is simulated by one processor as a self-contained MD simulations. The connection between neighbouring sub domains is established by including the atoms interaction distance as 'ghost' atoms. When crossing the boundaries between the sub domains, atoms are passed from one processor to another one. Thus, the code structure of LAMMPS

enables the insertion and removal of atoms innately. It has been adapted for the purposes of this work, extended by several modules and used for all performed simulations.

4.1 LAMMPS

LAMMPS was developed at and is distributed by Sandia National Laboraties. The source code, examples and documentation can be found at [148]. For integrating the atomic motion in time, a velocity Verlet algorithm is employed. LAMMPS uses the previously explained optimisation techniques for the computation of the interatomic interaction: cut-off distance, shifted potential, neighbour lists and linked cell list algorithm for generating the neighbour lists. Simulations are controlled through a script file, that contains all necessary commands for setting up and running the simulation. The code itself is written in C++, using an object-orientated structure. This allows a relatively uncomplicated extension of additional modules into the existing code.

Implementation of additional parts

Several additional parts have been implemented into LAMMPS to be able to perform the simulations necessary for this work:

ACSM: The automatic calculation simulation mode (ACSM) runs an MD simulation in static or dynamic equilibrium until a desired precision for a calculated quantity has been achieved. It is described in Section 3.4.

Grid analysis: This is an analysis module that can be used to relate the molecular states to the continuum description. For using it, one defines a virtual grid of cells throughout the molecular domain. For each grid cell, or the grid node in the centre of the cell, the chosen macroscopic variables are calculated and averaged in time. These may comprise mass, density, number density, momentum, velocity, temperature, potential energy, internal kinetic energy, external kinetic energy, total energy and temperature. The results are printed into a file for further postprocessing. The details of the calculations are described in Section 3.1. To assign atoms to grid nodes, the NGP or CIC scheme can be chosen.

Fragment identification: The purpose of this module is to identify fragments of connected atoms. It has been used in the study of colliding cluster, that is being described in Section 4.3. The algorithm is described in [104].

Reflective wall: This module is based on the algorithm described in Section 7.7. It poses a boundary for atoms, which are reflected perfectly on it.

Flux boundary conditions: This module is the implementation of the MD flux boundary conditions that are part of the HSI between MD domain and CFD domain. It is the central part of this work and was implemented with flexibility as a primary design target to allow the investigation of the different flux imposition methods. These can

be switched on or off through a number of parameters. The investigated parameters are described in chapter 7.

4.2 Basic verfication

Even though LAMMPS is being used by a relatively large community of researchers, the version that was used in this work was verified with several test cases. For basic verification a three-dimensional atomic system modelled with the LJ-potential and periodic boundary conditions was used. The used time step was $\delta t = 0.005$ in LJ-units (Compare table 6.1) and the cut-off distance $r_c = 5.0$.

Conservation of total momentum and energy

The most basic verification case is to check the conservation of total momentum and total energy in an equilibrium simulation. For equilibrium simulations with constant number of atoms, volume and energy, a so called NVE ensemble, the total momentum and energy must be conserved in time. The tests were performed for gaseous, liquid and solid (fcc lattice structure) states. In all performed cases, the total momentum and total energy of the system stayed constant in time.

Radial distribution function

The second basic verification case is concerned with the radial distribution function (RDF), a structural correlation function that is characteristic of the internal atomic organisation of a material at a specific state, i.e. at a certain density and temperature. Because the RDF can be measured experimentally, for example by neutron scattering experiments [201], and calculated from MD results it is a suitable for verification and validation of a MD code.

Imagine a particle at the position $\boldsymbol{r}_1$ in a volume element $\mathrm{d}\boldsymbol{r}_1$ as part of a N particle system in a statistical ensemble. The probability that, at the same time, another particle is at position $\boldsymbol{r}_2$ in a volume element $\mathrm{d}\boldsymbol{r}_2$ (regardless where the remaining $N-2$ particles are located) is proportional to the *pair distribution* function $g(\boldsymbol{r}_1, \boldsymbol{r}_2)\ \mathrm{d}\boldsymbol{r}_1\mathrm{d}\boldsymbol{r}_2$. If the system is in a homogeneous equilibrium then $g(\boldsymbol{r}_1, \boldsymbol{r}_2)\mathrm{d}\boldsymbol{r}_1\mathrm{d}\boldsymbol{r}_2$ depends only on the distance $\boldsymbol{r} = \boldsymbol{r}_1 - \boldsymbol{r}_2$ between both particles [157]. Furthermore, if isotropy is assumed, g is only a function of the magnitude of the distance $r = |\boldsymbol{r}|$. This function is known as radial distribution function $g(r)$ and is a measurement for the probability of finding a particle at the distance r from another particle.

For a simple mono-atomic system, in which only pair-interactions are modelled by a potential $\mathcal{V}(r)$, the radial distribution function is given by [167]

$$g(r) = V^2 \frac{\int \mathrm{d}\boldsymbol{r}_3 \cdots \mathrm{d}\boldsymbol{r}_N \exp\left[-\frac{1}{k_B T}\frac{1}{2}\sum_i^N \sum_{j\neq i}^N \mathcal{V}\left(|\,\boldsymbol{r}_i - \boldsymbol{r}_j|\right)\right]}{\int \mathrm{d}\boldsymbol{r}_1 \cdots \mathrm{d}\boldsymbol{r}_N \exp\left[-\frac{1}{k_B T}\frac{1}{2}\sum_i^N \sum_{j\neq i}^N \mathcal{V}\left(|\,\boldsymbol{r}_i - \boldsymbol{r}_j|\right)\right]}\ . \tag{4.2.1}$$

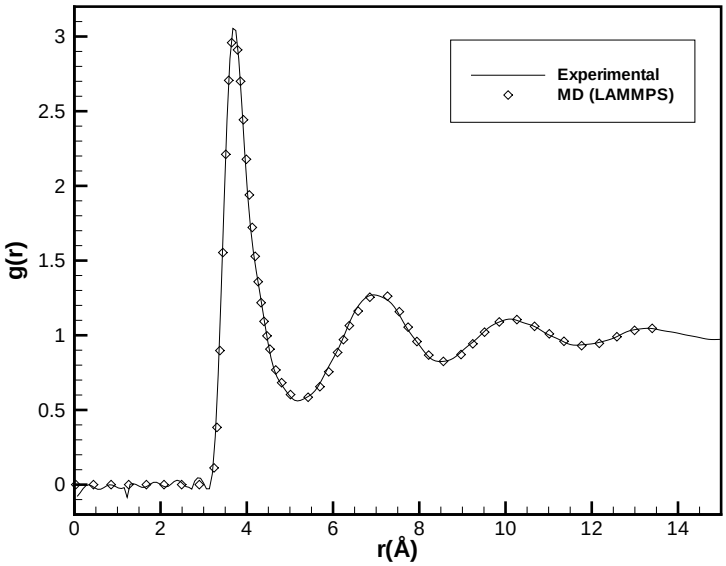


Figure 4.1: Comparison of the radial distribution function of Argon ($T = 85\,\mathrm{K}$, $n = 0.02125\,1/\text{Å}^3$) between experimental measurements by Yarnell et al. [201] and MD simulation results using LAMMPS.

The computation of (4.2.1) poses difficulties. However, using particle positions that have been computed by a molecular simulation, one can determine $g(r)$ directly by an algorithm which is formally written as [157]:

$$g(r) = \frac{V}{N}\frac{n(r,\delta r)}{4\pi r^2 \delta r}\,, \tag{4.2.2}$$

where $n(r,\delta r)$ is the number of all particles within a spherical shell that ranges from the radius r to $r+\delta r$ around a randomly selected particle.

The verification was done for Argon at a temperature of 85 K and number density of $0.0213\,1/\text{Å}^3$. To model Argon, the Lennard-Jones potential (Equation 2.1.23) with the parameters $\epsilon = 1.654 \times 10^{-21}\,\mathrm{J}$ and $\sigma = 0.341\,\mathrm{nm}$ was used. The simulated system was a three-dimensional box with periodic boundary conditions containing 27,000 atoms. The RDF was calculated from all atoms using Equation 4.2.2 up to a distance of $r = 13.6\,\text{Å}$, every 500 time steps and averaged over 20 independent calculations. In Figure 4.1, the calculated RDF function is plotted together with the RDF function obtained experimentally by Yarnell et al. [201] in neutron scattering measurements. The plots show excellent agreement and it can assumed that LAMMPS is verified for simulations using the Lennard-Jones potential.

4.3 Study on binary cluster collisions

As a further verification study, an investigation on the binary collision of spherical clusters was performed. The collision results have been compared with models that have been developed based on the results of binary droplet collisions experiments. Beside the purpose of validating the MD code, the investigation also has a number scientific contributions which are discussed in the following sections. The detailed results of the study have been published in [105].

The interaction of particles, i.e. drops, droplets or clusters, play an important role in various natural phenomena and technical applications. Understanding of collision dynamics of binary droplets under different conditions is pertinent to a broad range of processes including nucleation and growth of aerosols in the atmosphere [14, 151]; technical sprays used in material manufacturing processes, internal combustion engines, surface treatment and coating [108, 149, 151]; sintering during which bulk material is generated through inter-particle collisions, where the properties of the material depend strongly on the shape of the coalescence or agglomerated particles [52, 202]; growth, handling and utilisation of atomic and molecular clusters [100], particularly as applied in cluster beams, where the growth of clusters is driven by monomer addition and cluster coalescence [169]. Furthermore, research studies have been conducted to understand the fundamental mechanisms associated with coalescence of liquid drops [45] as well as drop deformation and breakup in viscous flows [172].

Binary collisions of particles can occur at different scales ranging from macroscopic particles, such as raindrops, down to nano-sized particles, i.e. clusters and nanodroplets. In the context of macroscopic scales, binary collision of droplets have been the subject of extensive investigations over the last few decades. Experimental studies of Ashgriz and POO [14], Brazier-Smith et al. [23], Brenn et al. [24], Qian and Law [151], Willis and Orme [196, 197], have identified four main collision modes: coalescence, reflexive separation, stretching separation and shattering. Models predicting the outcome of a collision for a given set of initial conditions have been developed and enhanced using experimental results, e.g. Arkhipov et al. [13], Ashgriz and POO [14], Brazier-Smith et al. [23], Gopinath and Koch [63], Ko and Ryou [108], Post and Abraham [149]. However, the details of the collision processes are difficult to obtain through experiments, especially at small length and time scales.

As an alternative, numerical simulations can be employed to shed light into the details of collisions. Previous studies on three-dimensional simulations of binary droplet collisions have been published by Rieber and Frohn [156], based on the incompressible Navier-Stokes equations in conjunction with the volume-of-fluid method, and Schelke and Frohn [164] based on the Lattice Boltzmann equation method. Mashayek et al. [128] used the Galerkin finite element method to simulate the coalescence of liquid droplets in two dimensions. The method is limited to coalescence and cannot handle separation. Meleán and Sigalotti [132] employed smoothed particle hydrodynamics (SPH) to simulate the binary collisions of equal-sized drops in two dimensions, while simulations based on the incompressible Navier-Stokes equations covering main modes of binary droplet collisions have also been recently published by Pan and Suga [144].

At nanoscale, the atomic structure of the particles becomes important since they consist of a few hundred or thousand of atoms. At present, these scales are not directly accessible by experiments, hence, numerical simulations become even more important. Quantum mechanical methods such as the density functional theory (DFT) would give the most accurate results, but they are computationally very expensive and can only be applied to small clusters. MD methods based on phenomenological potential functions to model interatomic interactions, have turned out to be an excellent tool for conducting numerical experiments at small scales. Continuum methods would reduce the computational effort even further, but they are not valid for the simulations of binary collisions in the nanoscale range.

Greenspan [64] and Greenspan and Heath [65] presented two- and three-dimensional molecular dynamics simulations of droplets consisting of 1,128 molecules interacting via a Lennard-Jones potential. The simulations showed that coalescence, stretching separation and shattering modes occur for droplet collision at nanoscale. Wyatt [199] presented results from MD simulations of binary collision of $(H_2O)_{400}$ droplets using the simple point charge model (SPC) and a new derived hybrid model (HYB) for the water molecules. The first systematic investigations of binary collisions were performed by Ming et al. [133] for Ar_{1000} clusters, for different collision velocities and angles, confirming the modes found in previous studies, i.e. coalescence, stretching separation and shattering. Svanberg et al. [175] employed the same setup for the simulation of $(H_2O)_{125}$ and $(H_2O)_{1000}$ and performed simulations for two different initial temperatures of 160 K and 300 K in conjunction with the SPC water model. They found that the boundary between coalescence and stretching separation is in good agreement with macroscopic models. However, reflexive separation which is observed at macroscale was not found in any of the above investigations. In a previous study, the authors have investigated the collision dynamics of the head-on impact of a small cluster (Ar_{309}) onto a large cluster (Ar_{10973}) for a broad range of impact speeds [104]. Both clusters had an initial temperature of 5 K. In addition to coalescence, the collisions resulted in partial scattering, eversion and total disintegration (shattering).

For sintering, collisions between clusters at thermal energies, i.e. at very low speeds, which are likely to lead to the coalescence of the involved particles, play a key role at nanoscale level. MD simulations of these processes are performed by starting with two equilibrated clusters, initially placed in single point contact with each other. As the simulation advances, the attracting interatomic forces cause the coalescence of the two clusters, which takes place through two distinguishable stages: Initially a neck, following a power-law growth, forms between the clusters. The neck disappears later and one particle is formed. Several authors have compared MD results with phenomenological models for coalescence time, neck growth, surface area reduction, temperature change and change of gyration radii. Simulation for coalescence of Si_{15} clusters were carried out by Blaisten-Barojas and Zachariah [21]. A combination of MD and DFT was used by Schmidt et al. [166] to study the fusion and reaction cross sections of small metal-metal cluster collisions. Zachariah and Carrier [202] performed MD simulations to investigate the morphology of particles in binary collision of spherical silicon particles, created by coalescence of zero impact factor (see § 4.3.2 for definition), and validate phenomenological models of particle growth. They found that the shape of the generated particles (from 30 up to 480 atoms) depends on the size and temperature of the original particles. Similar simulations were

performed by Hawa and Zachariah [85], for hydrogen-neutralised silicon particles, in order to investigate possibilities of controlling the size of silicon particles generated through vapour phase. Hawa and Zachariah also developed a mathematical model for the coalescence of coated particles [86] and extended the investigations to coalescence of unequal sized particles [87]. Rogan et al. [159] used the embedded atom method (EAM) potential to model binary collisions between small gold clusters and observed fusion, fragmentation and scattering collision modes. Arcidiacono et al. [12] has performed MD simulations of coalescence of solid gold nanoparticles for sintering processes. The clusters were modelled by using the glue potential. They reported that coalescence of gold clusters with radii greater than 20Å agreed well with macroscopic phenomenological models, whereas for smaller particles the models do not hold.

One aim of the present study is to investigate a variety of phenomena occurring during the binary collision between spherical particles at nanoscale (henceforth called clusters) for a broad range of collision angles and impact speeds. Each cluster consists of approximately 10,000 atoms, which is about one magnitude bigger than the largest cluster that has been used in previous binary collision studies. This is an attempt to shed light on the collision dynamics at nanoscales and examine similarities between nanoscales and macroscopic scales. Qualitative comparisons with previous MD studies and analytical prediction models derived for macroscopic droplet collisions, are also presented. In addition to the collision identification approaches that have been used in previous MD studies, an alternative identification scheme is proposed, which enables classification of the collision phenomena into main modes and sub-modes with respect to different collision angles and impact speeds. Finally, it is investigated whether the collision mode referred to as reflexive separation, which has been observed in macroscopic droplet collisions, occurs at nanoscale.

4.3.1 Molecular model and definitions

MD has been employed in several previous investigations of cluster-cluster [65, 104, 133, 159, 175, 199] and cluster-surface collisions [22, 136, 150, 179, 200].

The clusters of the present study consist of atoms that are modelled by the widely used 12-6 Lennard-Jones (LJ) potential. A distance of $5 \times \sigma$ has been used as cut-off distance in the present simulations. All units are given in LJ-units (see Table 6.1). Similar to previous investigations [104], a dimensionless time step of 0.0023 has been used. Different criteria have been used for analysing the computed trajectories. Collision fragments that are formed during the collision process are classified according to their size. The absolute size, S^{α}, of a fragment α is defined as the number of atoms in the fragment. The relative size, s^{α}, is the number of atoms in fragment α relative to the initial number of atoms in one cluster N ($s^{\alpha} = S^{\alpha}/N$). Using the relative size, fragments have been classified into four categories, very large, large, medium and small (Table 4.1). The largest fragment is labelled as F^1 with a relative size s^1; the second and third, ... and N_fth largest fragments are labelled as F^2, F^3, ... F^{N_f} with relative size s^2, s^3, ... s^{N_f}.

Type	Abbrev.	s	S
very large	F_{vl}	$10\% \leq s$	$1000 \leq S$
large	F_l	$1\% \leq s < 10\%$	$100 \leq S < 1000$
medium	F_m	$0.1\% \leq s < 1\%$	$10 \leq S < 100$
small	F_s	$s < 0.1\%$	$S < 10$

Table 4.1: Classification of fragments according to their size; s and S are the relative and absolute size of the fragments, respectively.

Depending on the collision dynamics, there may be one or two main fragments. Per definition, scattered atoms are all atoms that do not belong to the main fragments:

$$N_{sca} = \begin{cases} 2N - S^1 & : \text{ one main fragment} \\ 2N - (S^1 + S^2) & : \text{ two main fragments} \end{cases} . \tag{4.3.1}$$

For the identification of fragments the nearest neighbour distance criterion of Stoddard [171] has been applied with a critical atom separation of $r_{cl} = 1.76$. A description of the fragments detection algorithm can be found in [104].

The analysis of collision dynamics involves the calculation of several fragment and cluster properties; henceforth, the word fragment will apply to clusters as well. For a fragment F^α the centre $\boldsymbol{r}^\alpha$ is considered as the centre of mass. The translational and angular velocities are $\boldsymbol{v}^\alpha$ and $\boldsymbol{\omega}^\alpha$ respectively. $\boldsymbol{\omega}^\alpha$ is calculated from angular momentum $\boldsymbol{l}^\alpha$ and the inertia tensor $\mathbf{I}^\alpha$. In the analysis, both internal and external energies have been considered. The internal potential energy $E^\alpha_{p,int}$ of a fragment F^α is calculated by the sum of the interactions between the atoms within the cluster. Furthermore, the internal, $E^\alpha_{k,int}$, and external, $E^\alpha_{k,ext}$, kinetic energies are used in the analysis of the results. $E^\alpha_{k,int}$ comprises of the linear, $E^\alpha_{kl,ext}$, and the angular, $E^\alpha_{ka,ext}$, parts. The calculation of these variables is standard mechanics and can be found in textbooks, e.g. Goldstein et al. [62].

The fragments temperatures are obtained from the internal kinetic energy by

$$T^\alpha = \frac{E^\alpha_{k,int}}{2k_B f^\alpha} , \tag{4.3.2}$$

where k_B is Boltzmann constant and f^α are the degrees of freedom of the fragment; $f^\alpha = 3S^\alpha - 6$ for an atomic fragment.

4.3.2 Setup of the nanoclusters

The initial setup for the two binary clusters, C^A and C^B, is shown in Figure 4.2. Each cluster consists of $N = 10,973$ atoms, has a diameter of $d \approx 27$, a temperature of 0.33 and an internal potential energy of $E_{p,int} \approx 8 \times 10^4$. They have been spherically cut from a face-centred cubical crystal and equilibrated over 100,000 time steps.

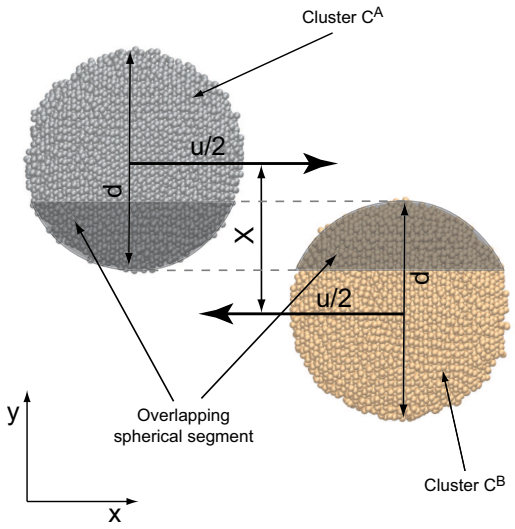


Figure 4.2: Initial setup of the binary nanoclusters. Both clusters are assigned a speed of $u/2$ in opposite direction. The right cluster is displaced by X in the negative y direction.

The collision dynamics is investigated with respect to relative speed u and impact parameter X. The absolute velocities are given by

$$\boldsymbol{v}^{\mathrm{A}} = \frac{1}{2}\begin{pmatrix} u \\ 0 \\ 0 \end{pmatrix}, \quad \boldsymbol{v}^{\mathrm{B}} = -\frac{1}{2}\begin{pmatrix} u \\ 0 \\ 0 \end{pmatrix}, \tag{4.3.3}$$

for the clusters C^{A} and C^{B}, respectively. The linear momentums are $\boldsymbol{p}^{\mathrm{A}} = Nm_a\boldsymbol{v}^{\mathrm{A}}$ and $\boldsymbol{p}^{\mathrm{B}} = Nm_a\boldsymbol{v}^{\mathrm{B}}$; the initial relative momentum is $\boldsymbol{p} = 2Nm_a\boldsymbol{u}$; and the total linear momentum is $\boldsymbol{p}_t = \boldsymbol{p}^{\mathrm{A}} + \boldsymbol{p}^{\mathrm{B}} = 0$.

The impact parameter, X, determines how far off-centre the collision occurs and is defined as the distance from the centre of one cluster to the relative velocity vector $\boldsymbol{u}$ originating from the centre of the other cluster [14]. Non-dimensionalising X yields x that can be easily related to the collision angle α (Figure 4.3),

$$x = \frac{X}{d}, \quad x = \sin(\alpha)\,. \tag{4.3.4}$$

For $x = 0$ ($\alpha = 0^o$) both clusters are subject to head-on collision, whereas for $x = 1$ ($\alpha = 90^o$) the collision is marginally avoided.

The displacement X must be perpendicular to the relative velocity vector $\boldsymbol{u}$ that is aligned to the horizontal direction. Therefore, the right cluster is moved by X in the negative y direction and the two initial positions of C^{A} and C^{B} are set as

$$\boldsymbol{r}^{\mathrm{A}} = \frac{1}{2}\begin{pmatrix} -17 \\ 0 \\ 0 \end{pmatrix}, \quad \boldsymbol{r}^{\mathrm{B}} = \frac{1}{2}\begin{pmatrix} 17 \\ -X \\ 0 \end{pmatrix}. \tag{4.3.5}$$

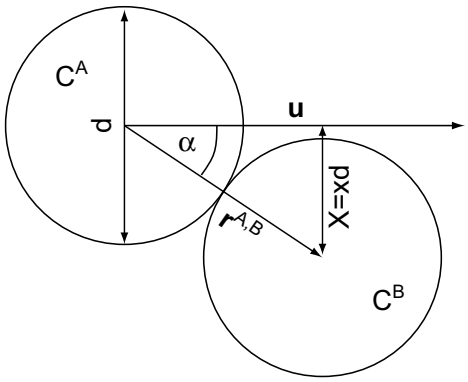


Figure 4.3: Relation of impact factor x and collision angle α.

For $x \neq 0$ an angular momentum $\boldsymbol{l}$ in the z direction is introduced, which is conserved throughout the simulation:

$$\boldsymbol{l}_z = \frac{1}{2} N m_a u x \; . \tag{4.3.6}$$

4.3.3 Investigation of the collision dynamics

The collision dynamics of two LJ_{10973} clusters was investigated for different values of the impact parameter x in the range of 0 to 1 and at different speeds, u, in the range of 1.58 to 6.33 (non-dimensional). Initially, simulations were carried out for all combination of $x = \{0, 0.13, 0.24, 0.36, 0.42, 0.5, 0.6, 0.71, 0.87, 0.97, 0.99\}$ and $u = \{1.58, 3.17, 4.75, 6.33\}$, in order to identify the main collision modes. Furthermore, simulations were performed for selected x and u, to obtain a clearer understanding of the transition between the collision modes and to render more precisely the location of the transition lines. Simulations were conducted for up to 100,000 time steps (a non-dimensional time interval of 232). For $x = 0.13$ and $u = 5.33$ computations were performed for 250,000 time steps (a non-dimensional time of 580) in order to investigate the evaporation processes that follow a collision with relatively high energy. The trajectories have been visualised by using the VMD software [95].

During the simulation the state of fragmentation was monitored, i.e. the number of very large, N_f^{vl}, large, N_f^l, medium, N_f^m, and small fragments, N_f^s; the number of atoms in each class N_a^l, N_a^b, N_a^m, N_a^s; and the number of scattered atoms, N_{sca}. Other parameters were recorded in relation to the two largest fragments F^1 and F^2 including the total sizes, S^1 and S^2; the relative sizes, s^1 and s^2; the fragments positions, $\boldsymbol{r}^1$ and $\boldsymbol{r}^2$; the velocities, $\boldsymbol{v}^1$ and $\boldsymbol{v}^2$; their angular velocities, $\boldsymbol{l}^1$ and $\boldsymbol{l}^2$; the linear external kinetic energies, $E^1_{kl,ext}$ and $E^2_{kl,ext}$; the angular external kinetic energies, $E^1_{ka,ext}$ and $E^2_{ka,ext}$; the internal kinetic energies, $E^1_{k,int}$ and $E^2_{k,int}$; the internal potential energies, $E^1_{p,int}$ and $E^2_{p,int}$; the external potential energies, $E^1_{p,ext}$ and $E^2_{p,ext}$; and the temperatures, T^1 and T^2.

An overview of the collision dynamics for different impact factors, velocities, and times is shown in Figure 4.4. The scale varies between the images in order to enable visualisation of the spatial distribution of the resulting fragments. For low x and/or low u the collision

x\u	1.58	1.9	2.53	3.17	3.93	4.75	5.38	6.33
0.00	116			116		116	174	174
0.13	116			116	116	174	174	174
0.24	116			116	116	174	290	174
0.36	116	116	116	116	290	174	232	174
0.42	116	116	116	116	174	116	232	116
0.50	116	116	116	116		116		116
0.60	116	116	116	116		116		116
0.71	116	116	116	116		116		116
0.87	116			116		116		116
0.97	116			35		35		116
1.00	116			35		35		35

Table 4.2: Non-dimensional times at which the fragment parameters were calculated (see Tables 4.3 – 4.5).

results in one main fragment whose shape depends on the x and u values. This collision mode is referred to as *coalescence* (henceforth labelled as (I)). For higher x and/or u values the collision yields two main fragments and in some cases it may also include satellite droplets. As described later in more detail, the temporarily formed cluster separates after being stretched, thus the mode is referred to as *stretching separation* (labelled as (II)). A third collision mode, *shattering* (labelled as (III)), which corresponds to the destruction of both clusters, occurs for high velocities, u, in connection with low impact factors, x. In the following section, classification of the collision modes is presented in detail.

Modes Classification

Different techniques and criteria can be used for identifying the collision modes. One possibility is to classify the dynamics visually as done in experiments [14]. The same procedure can be used in the context of MD simulations and, additionally, one can obtain a more thorough analysis of the dynamics using the computed trajectories. To identify the collision modes the following parameters were employed: the relative size of the largest, s^1, and second largest fragment, s^2; and the difference of the relative sizes of the two largest fragments $s^{|1-2|} = |s^1 - s^2|$. These parameters are calculated at different times (Table 4.2) depending on the duration of the collision. The values of these parameters for different impact factors and velocities are given in Tables 4.3 – 4.5.

Identification scheme A: One possibility is to adapt the scheme used by Ming et al. [133], which compares s^1 and s^2 with threshold values $c_{A1} = 70\%$ and $c_{A2} = 20\%$, respectively. These values provide the best agreement with the visualisations. The collision modes can then be classified as:

(I) Coalescence for $s^1 > c_{A1} = 70\%$.
(II) Stretching separation for $s^2 > c_{A2} = 20\%$).
(III) Shattering for $s^1 < c_{A1} = 70\%$ and $s^2 > c_{A2} = 20\%$.

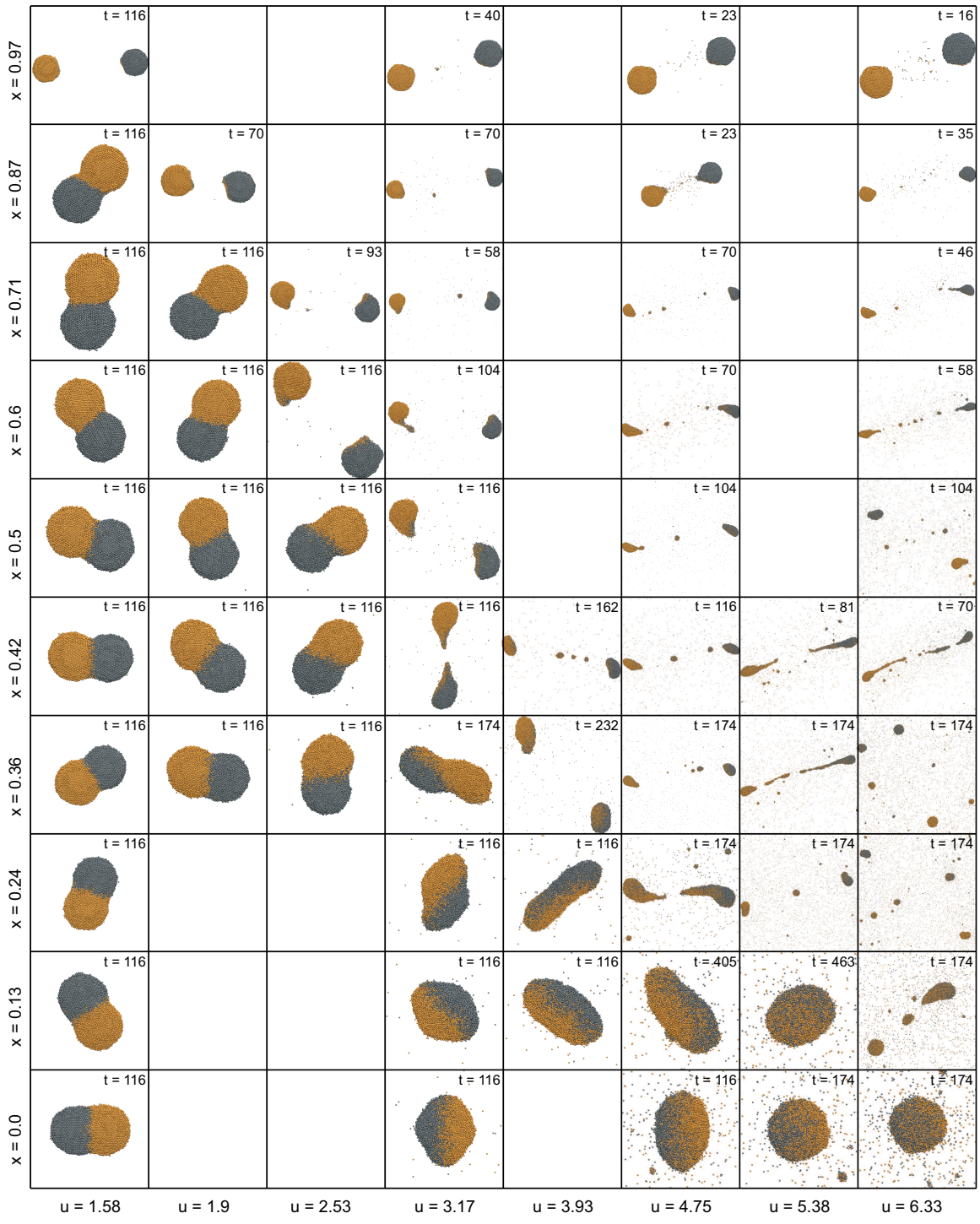


Figure 4.4: $x - u$ matrix showing an overview of collision outcomes for different impact factors and velocities. The scale varies between the individual images because the distribution of material in space is different in each case (see Figures 4.5a, 4.5b and 4.6 for classification of modes and sub-modes). All numbers are non-dimensional.

x\u	1.58	1.9	2.53	3.17	3.93	4.75	5.38	6.33
0	100.0			99.5		90.7	78.2	58.7
0.13	100.0			99.5	98.4	89.5	77.0	39.6
0.24	100.0			99.6	98.8	48.8	38.5	20.8
0.36	100.0	100.0	99.9	99.7	48.9	43.1	35.8	26.1
0.42	100.0	100.0	99.9	51.3	47.3	45.0	44.6	32.7
0.5	100.0	100.0	99.9	50.4		47.2		40.2
0.6	100.0	100.0	50.1	50.5		47.8		44.2
0.71	100.0	100.0	50.7	49.7		47.8		47.4
0.87	100.0	50.1		49.7		49.8		49.3
0.97	50.1			50.0		49.9		49.9
1	50.0			50.0		50.0		50.0

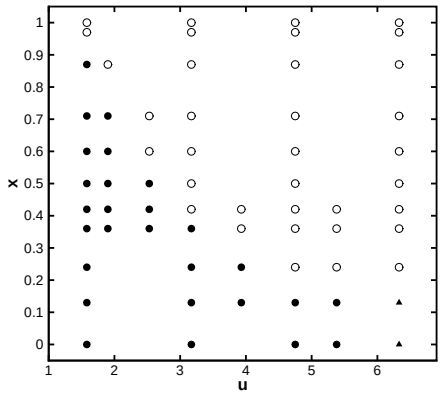
Table 4.3: Relative size of the largest fragment, s^1, for different collision speeds, u and impact factors, x.

x\u	1.58	1.9	2.53	3.17	3.93	4.75	5.38	6.33
0.00	.0			.0		1.1	1.5	3.8
0.13	.0			.0	.0	1.3	4.8	17.6
0.24	.0			.0	.0	41.4	29.1	20.4
0.36	.0	.0	.0	.0	47.4	43.1	33.1	22.6
0.42	.0	.0	.0	48.1	45.8	43.5	42.0	32.7
0.50	.0	.0	.0	48.4		37.0		38.8
0.60	.0	.0	49.8	48.9		46.9		42.5
0.71	.0	.0	49.0	48.8		47.8		47.0
0.87	.0	49.9		49.5		49.8		49.2
0.97	49.9			49.9		49.9		49.8
1.00	50.0			50.0		50.0		49.9

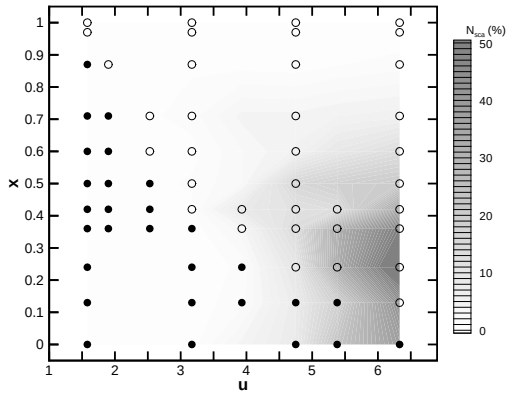
Table 4.4: Relative size of the second largest fragment, s^2, for different collision speeds, u and impact factors, x.

x\u	1.58	1.9	2.53	3.17	3.93	4.75	5.38	6.33
0	100.0			99.5		89.6	76.7	54.9
0.13	100.0			99.5	98.4	88.3	72.2	22.1
0.24	100.0			99.6	98.8	7.5	9.4	.4
0.36	100.0	100.0	99.9	99.7	1.5	.0	2.7	3.7
0.42	100.0	100.0	99.9	3.1	1.4	1.5	2.6	.1
0.5	100.0	100.0	99.9	2.0		10.3		1.4
0.6	100.0	100.0	.4	1.6		.9		1.6
0.71	100.0	100.0	1.7	.9		.0		.5
0.87	100.0	.2		.2		.0		.1
0.97	.2			.1		.0		.1
1	.0			.0		.0		.0

Table 4.5: Values of $s^{|1-2|}$, for different collision speeds, u and impact factors, x.



(a) Collision modes depending on the impact factor, x, and velocity, u based on the identification schemes. Coalescence is denoted by (•), stretching separation by (○) and shattering by (▲).



(b) Collision modes depending on the impact factor, x, and velocity, u based on the identification schemes B. Coalescence is denoted by (•), stretching separation by (○). The shaded area displays the strength of scattering.

Figure 4.5: Collision modes for two different identification schemes.

Even though these criteria allow, in theory, a collision to belong simultaneously in coalescence and stretching separation modes, none of the simulated cases resulted in this scenario. Applying the above criteria in conjunction with the computed s^1 (Table 4.3) and s^2 (Table 4.4), yields the collision modes of Figure 4.5a. The transition between coalescence and stretching separation is clearly identified. The shattering mode is somewhat arbitrarily defined due to the diffusive nature of the transition towards shattering compared to the other two collision modes.

Identification scheme B: This scheme uses $s^{|1-2|}$, i.e. the size difference of the largest and second largest fragment, to distinguish between coalescence and stretching separation modes based on a threshold value, $c_2 = 15\%$. It also examines the percentage of scattered atoms N_{sca}/N instead of defining the shattering mode. The criteria for scheme B are listed as follows:

(I) Coalescence for $s^{|1-2|} > c_B = 15\%$).
(II) Stretching for $s^{|1-2|} < c_B = 15\%$.

Figure 4.5b demonstrates application of the above criteria on the present MD results. The shading denotes the intensity of scattering (N_{sca}/N), where the dark areas correspond to strong scattering. The collision dynamics discussed in the next sections is based on the identification scheme B. The boundaries between the collision modes are shown in Figure 4.6 including various submodes, which are explained in detail in the following sections.

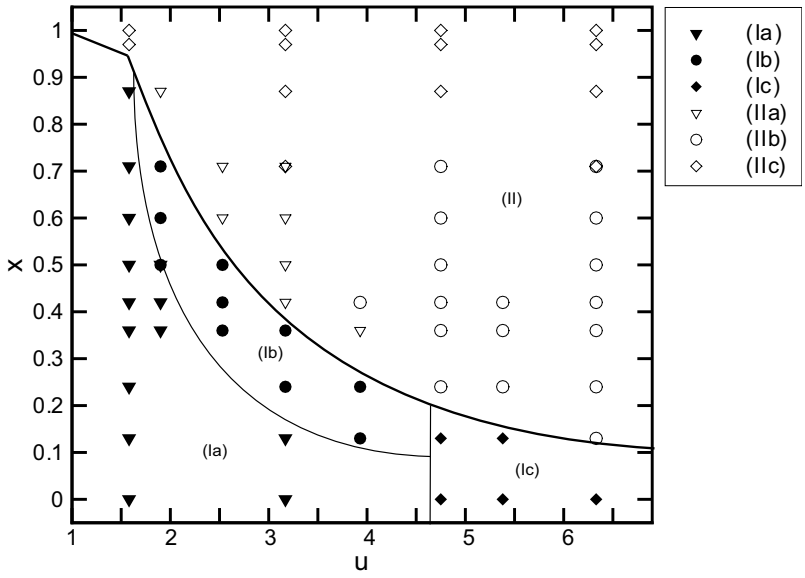


Figure 4.6: Overview of the collision modes including submodes in *x*-*u*-space: (Ia) sticking; (Ib) sliding and locking; (Ic) droplet; (IIa) normal stretching separation; (IIb) stretching separation with satellite droplets; (IIc) shearing-off.

4.3.4 Coalescence (I)

Generally, the coalescence mode prevails at low velocities, u, and/or small impact parameters, x. Common to all collisions belonging to this mode is the formation of one main cluster or a droplet that is created by the two original clusters. Scattering of atoms occurs only at high velocities. The structure, shape, temperature and motion of the final aggregate, as well as the dynamics of the collision process, differ within the coalescence mode. The present simulations revealed three different submodes: (a) *sticking mode*, (b) *slide and locking mode* and (c) *droplet mode*. The locations of these submodes are illustrated in the *x*-*u* diagram of Figure 4.6. Transition between submodes occurs gradually, thus collisions occurring in the vicinity of borderlines encompass features of two different submodes.

Sticking mode (Ia)

Figure 4.6 shows that for $u < 1.58$ the sticking submode covers the entire range of impact factor x, except for $x > 0.87$, where the collision results in both clusters sticking together, while their internal structures remaining intact. The shape of the resulting cluster resembles that of a dumb-bell (Figure 4.4) thus it is also referred to as dumb-bell mode [65].

The collision dynamics of a typical (Ia) case, e.g. $x = 0.36$ and $u = 1.58$, is shown in Figure 4.7. During impact, only the material near the contact interface is deformed. Once the dumb-bell cluster is created, the two clusters are 'locked' to each other. The translational motion is transformed into rotation. The trajectories of the two original clusters, C^A and C^B, are plotted in Figure 4.8; starting on the left side for C^A and on the right side for C^B, respectively. For the sticking mode case ($x = 0.36, u = 1.58$), the initially straight path

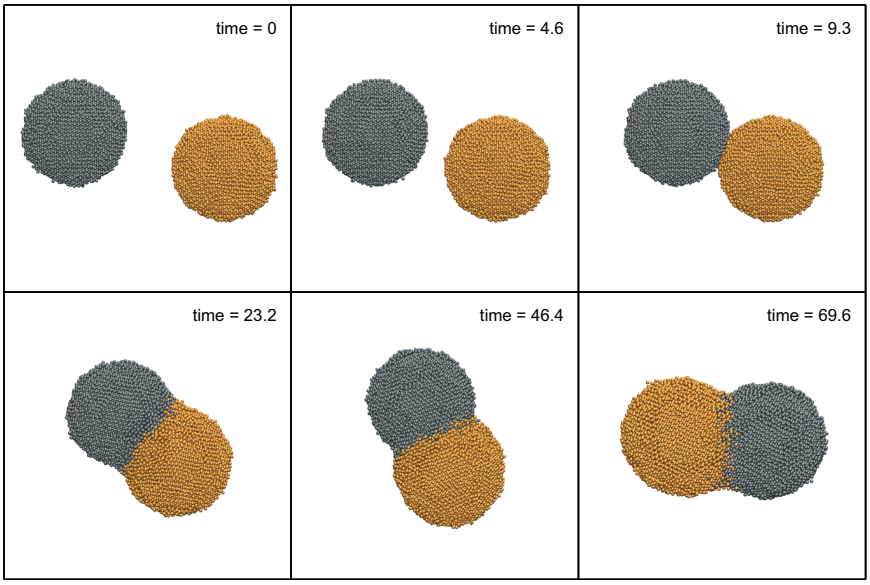


Figure 4.7: Collision typical to coalescence mode ($x = 0.36$, $u = 1.58$) with the resulting cluster taking a dumb-bell shape.

is followed by a circular one (dashed line in Figure 4.8). The point at which the motion changes is marked as contact point and this is exactly the position where the two clusters come into contact. The transition from linear to circular motion occurs in a very short period of time. The distance between the mass centres of the two initial clusters, $r^{A,B}$, remains constant after the clusters have come into contact with each other (Figure 4.9).

Figures 4.10 and 4.11 show the variation of the angular velocity ω_z^1 and angular momentum l_z^1 in time of the dumb-bell cluster F^1. Before the initial clusters come in contact there is no rotation. Following impact, the angular momentum, w_z^1, steeply rises and reaches a plateau value, while, as expected, the angular momentum, l_z, of the resulting cluster is conserved (Figure 4.11) .

For impact factors $x < 0.2$, the collision can still be characterised as 'sticking mode', even for velocities $u > 2$, with the original clusters being flattened at impact. The dumpbell shape appearing at lower u is replaced by an elliptical or a spherical shape. A typical example is the collision for $x = 0$ and $u = 3.17$ (Figure 4.4). The change of $r^{A,B}$is shown in Figure 4.9. The distance between the clusters' centres is smaller compared to the case of $(x = 0.13, u = 1.58)$. By further increasing the impact velocity ($u > 4$), the collision mode will gradually take on the characteristics defining the droplet mode. The collision in sticking mode is governed by transformation of the initial external linear kinetic energy $E_{kl,ext}$ into: internal kinetic energy $E_{k,int}$; internal potential energy $E_{p,int}$ corresponding to the structural changes of the deformation and angular kinetic energy $E_{ka,ext}$.

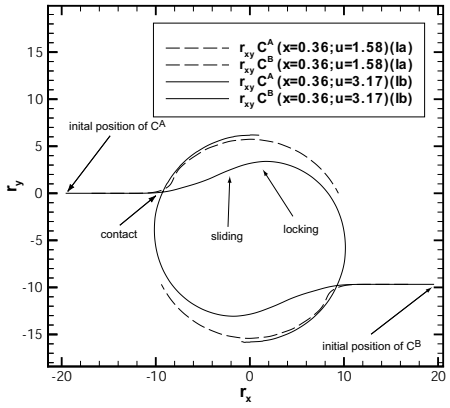


Figure 4.8: Trajectories of the mass centres r^{A} and r^{B} for coalescence mode, sticking (Ia) and slide-and-locking (Ib) submodes; C^{A} and C^{B} trajectories start on the left and right sides, respectively, and are traced for $t = 0 - 116$ (non-dimensional)

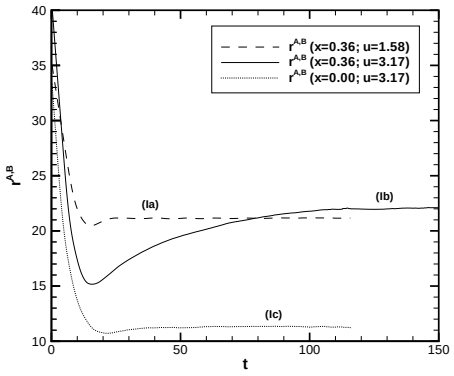


Figure 4.9: Distance $r^{\mathrm{A,B}}$ between the centre of masses r^{A} and r^{B} of the original clusters C^{A} and C^{B} for sticking (Ia); slide-and-locking (Ib); and droplet (Ic) submodes.

Slide-and-locking mode (Ib)

The slide-and-locking mode forms a narrow (transition) region in $x - u$ space between coalescence and stretching separation (Figure 4.6). The collision dynamics for a typical case ($x = 0.36, u = 3.17$) is shown in Figure 4.12. The clusters are flattened at impact ($t = 4.6 - 11.6$) and the initial contact point evolves to a flat, circular interface. The heat generated at the contact interface liquifies the material in this region and a (liquid) layer with reduced friction is formed. This layer acts as a cushion between the clusters facilitating their sliding along each other. The direction of their motion is slightly deflected outwards and aligns parallel to the contact interface. During sliding the heat is dissipated away from the interface to the rest of the cluster thus cooling down the interface and increasing friction. The linear motion is transformed into rotation of the entire complex. Eventually, the interface 'locks' and the sliding motion completely stops, leaving behind a rotating cluster. The shape of the resulting cluster depends mainly on the extent of flattening and sliding of the original cluster.

The motion of the clusters can be traced by plotting the position of the mass centres projected onto a two-dimensional plane (solid line, Ib case in Figure 4.8), where the contact and locking points as well as sliding area are shown. After the initial contact the clusters' trajectories remain parallel to each other, while symmetry is preserved throughout. As the friction increases, the trajectories enter a circular motion and the contact interface is locked thus not permitting further sliding of the clusters. The distance of the mass centres, $r^{\mathrm{A,B}}$, (Figure 4.9) shows that the closest approach occurs immediately after the clusters have come in contact. The distance $r^{\mathrm{A,B}}$increases during the sliding phase and approaches asymptotically a plateau value as sliding continues. The angular momentum, l_z^1, of the formed cluster is conserved (Figure 4.11) and the angular velocity, ω_z^1, decreases (solid

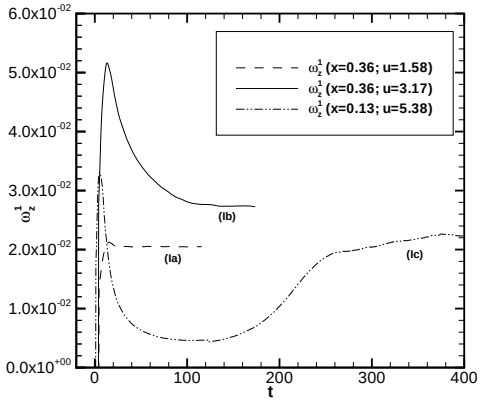


Figure 4.10: Angular velocity ω_z^1 of the resulting cluster at late time for sticking (Ia); slide-and-locking (Ib); and droplet (Ic) submodes.

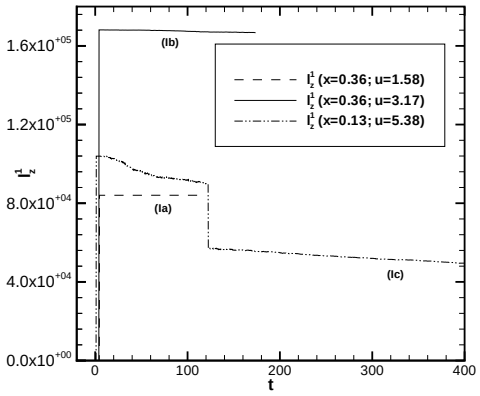


Figure 4.11: Angular momentum l_z^1 of the cluster at late time for sticking (Ia); slide-and-locking (Ib); and droplet (Ic) submodes.

lines in figures 4.9 and 4.10). The initial, linear kinetic energy, $E_{kl,ext}$, is transformed into: (i) internal potential energy, $E_{p,int}$, causing deformation and increasing pressure; and (ii) internal kinetic energy, $E_{k,int}$, in the vicinity of the contact interface. During the sliding-and-locking phase, the remaining part of $E_{kl,ext}$ is transformed into $E_{k,int}$ and angular kinetic energy, $E_{ka,ext}$, of the entire cluster.

Increasing the velocity u results in faster and longer sliding motion, which eventually leads to the separation of both clusters; this is the transition to the stretching separation mode. Higher values of x entail a smaller contact interface. Consequently, the transformation of impact energy, $E_{kl,ext}$, into heat, $E_{k,int}$, and deformation, $E_{p,int}$, takes place more slowly and therefore leads to stretching separation.

Further comments are given on how the classification of sticking mode (Ia) and slide-and-locking modes (Ib) has been obtained. The trajectories of the two clusters are projected onto a two-dimensional $(x-y)$ plane and, if the path follows a straight line after the clusters come in contact, the collision mode is considered to be (Ib), otherwise it is classified as (Ia). An example is given in Figure 4.13 for constant $u = 3.17$ and increasing x. Because the trajectories of C^A and C^B are symmetrical, only those of C^A are displayed. The path for $x = 0.0$ is a straight line (central collision), but because there is no sliding, the collision is characterised as (Ia). The path for $x = 0.13$ shows no distinct linear section and is therefore considered to belong to mode (Ia). Note that the short straight line is due to compression and flattening of the cluster, thus it does not account for sliding. In increasing the values of x to 0.24 and 0.36, a distinct linear segment can be found in the trajectories following the contact point. Hence, both cases are regarded as mode (Ib). The classification of the collision modes was also checked by examining simulation movies created using the computed trajectories.

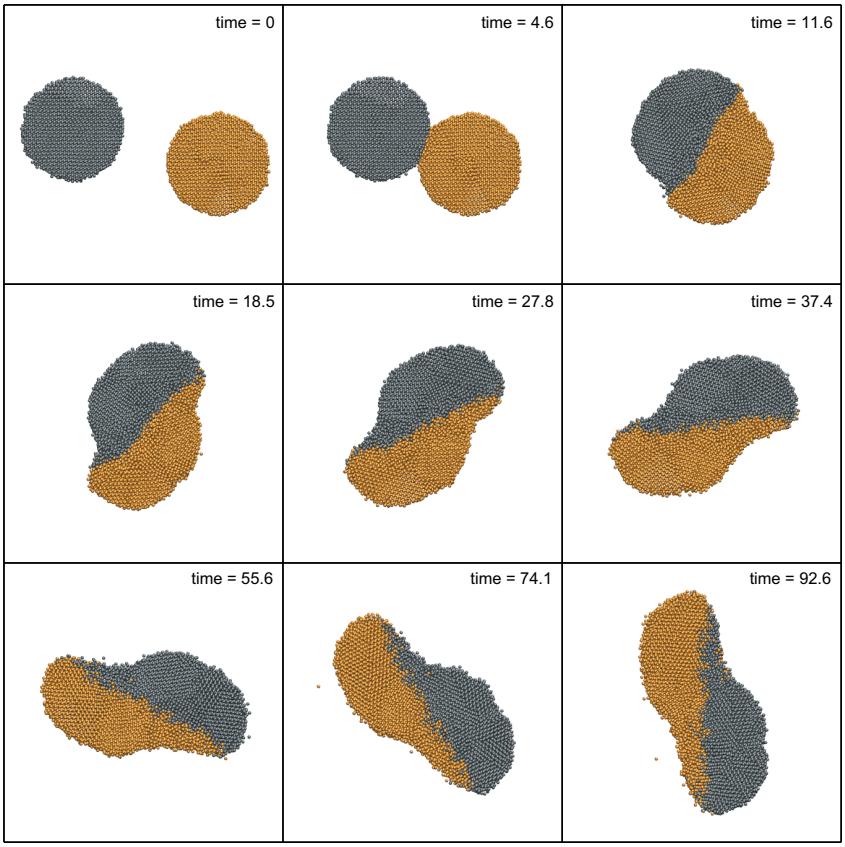


Figure 4.12: Development of coalescence featuring slide-and-locking mode ($x = 0.36$, $u = 3.17$).

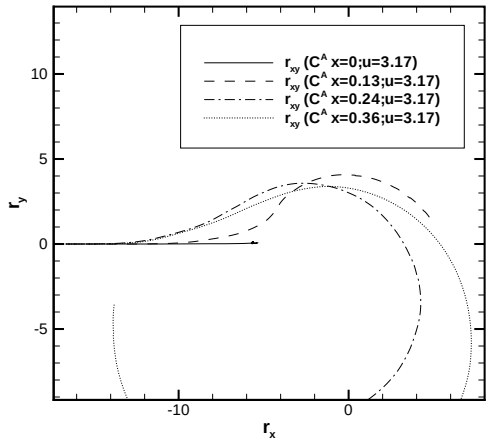


Figure 4.13: Trajectories of the clusters C^A for $u = 3.17$ and different impact factors x. For $x = \{0, 0.13\}$ the collision belongs to sticking mode (Ia), while for $x = \{0.24, 0.36\}$ belongs to slide-and-locking mode (Ib).

Droplet mode (Ic)

For impact factors $x < 0.2$ and velocities u greater than approximately 4.5, the collision dynamics is classified as droplet mode (Figure 4.6). The high impact energy results in one main fragment in liquid phase. In contrast to the previous cases, where the original clusters can still be identified within the newly formed aggregate, in the droplet mode the material of the two initial clusters is mixed. The droplet mode also involves significant scattering and evaporation as can be seen in Tables 4.3 – 4.5.

The collision dynamics for a typical example ($x = 0.13$ and $u = 5.33$) is shown in figures 4.14a and 4.14b, where each column shows a different view: from the side (along the z-axis), along the velocity vector $\boldsymbol{u}$ and from a perspective view. The clusters are flattened along a planar contact interface ($t = 4.6$). Radial, disc-like scattering originating from the contact interface is observed at $t = 4.6$ and $t = 9.3$. The scattering is due to the transformation of the largest part of impact energy into heat at the contact interface. The internal pressure force acts outwards and is opposed by an inward pressure force due to the interatomic and deceleration forces. Because the pressure acting perpendicular to $\boldsymbol{u}$ is much smaller than the one acting in the direction of $\boldsymbol{u}$, the contact interface expands radially. The expansion overcomes the binding forces and results in asymmetric scattering, which reaches a peak value at $t \approx 14$. While the contact interface widens out into a circular plane, asymmetric momentum distribution forces the planar interface to bend and take an S-like shape (see, for example, side view at $t = 23.2$ in Figure 4.14a). The onset of this behaviour is already visible in the slide-and-locking case (Figure 4.12). At later times an elongated droplet is formed ($t = 138.9$ in Figure 4.14b), which finally approaches a spherical shape (last row of Figure 4.14b).

During the collision process, evaporation and condensation of the liquid and gas atoms, respectively, occur. The process gradually leads to reduced size and temperature of the formed droplet, while the evaporation rate is also gradually reduced. The effect of evaporation is encountered by condensation of the vapour phase surrounding the droplet. When the rate of evaporation equals the rate of condensation an equilibrium is achieved. To examine evaporation and confirm the spherical shape of the final fragment, the simulations have been extended for $x = 0.13, u = 5.38$ until $t = 925$. The change of absolute size S^1 and temperature T^1 of the largest fragment, F^1, are shown in Figure 4.15. The initial stages are dominated by scattering and from $t = 34$ onwards the decay in S^1 is caused by evaporation. The sudden change at $t = 122$ is due to the separation of a satellite droplet; cf. $t = 69.5$ and $t = 138.9$ in Figure 4.14b. The reduction of the absolute size S^1 shows an asymptotic behaviour towards the end of the simulation.

Effects of the impact factor

Further investigations have been dedicated to the dependence of angular momentum, angular velocity and distance of the clusters' centres on the impact factor x. According to (4.3.6) the angular momentum of the final fragment is determined by the initial setup. The l_z^1 values obtained from the simulations are shown in Figure 4.17 and agree exactly with the values calculated by (4.3.6), i.e., follow a linear variation with slope $\frac{1}{2}Nm_a u$. Because the

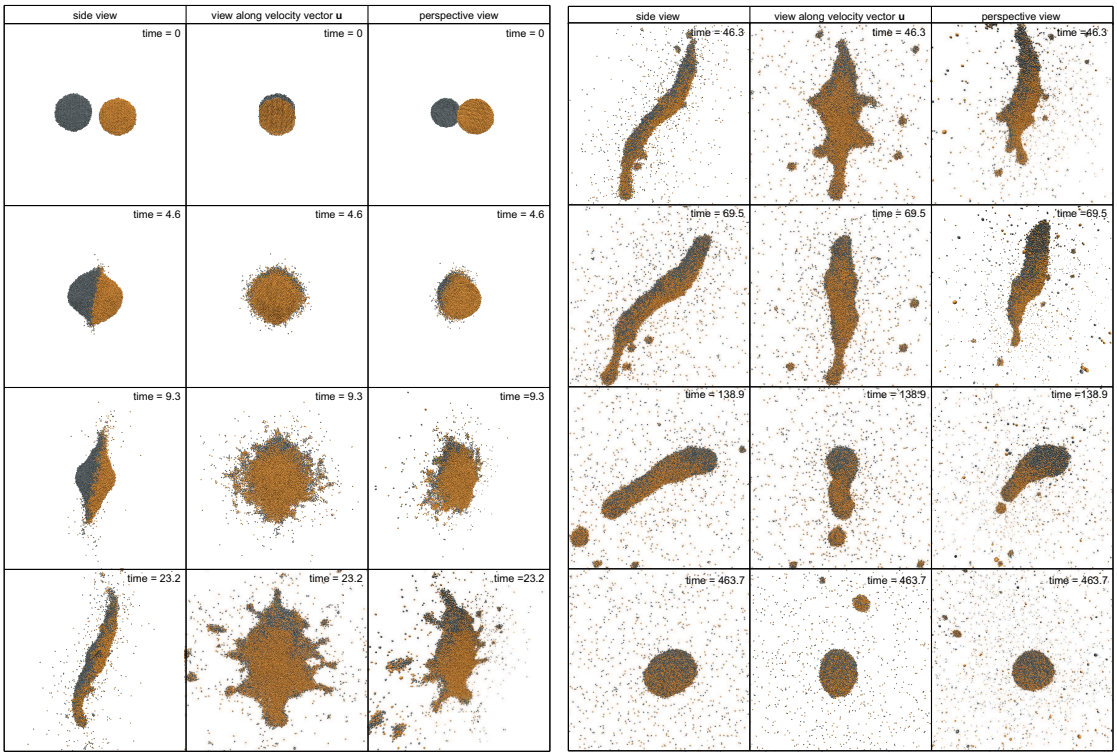


Figure 4.14: Collision dynamics in coalescence-droplet mode with strong scattering and evaporation, for $x = 0.13$ and $u = 5.33$.

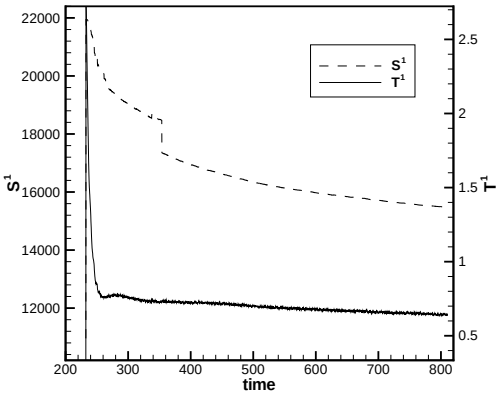


Figure 4.15: Size and temperature of the largest fragment for the case of $x = 0.13$ and $u = 5.38$ (droplet mode (Ic)).

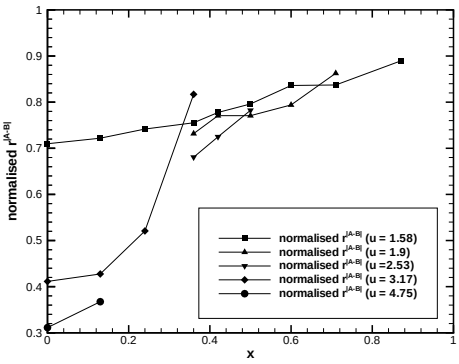


Figure 4.16: Normalised distance of the cluster centres $r^{|A\text{-}B|}$, i.e. of the two original clusters C^A and C^B inside the resulting cluster F^1 against the impact factor x for collisions in coalescence mode.

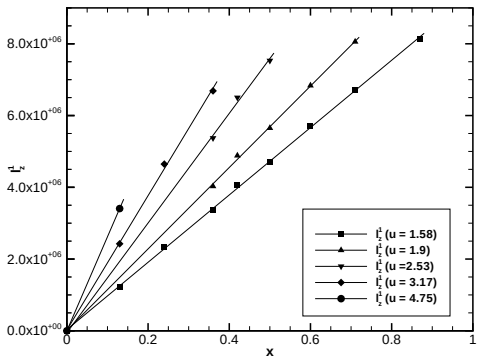


Figure 4.17: Angular momentum l_z^1 of the largest fragment F^1 against the impact factor x for collisions in coalescence mode.

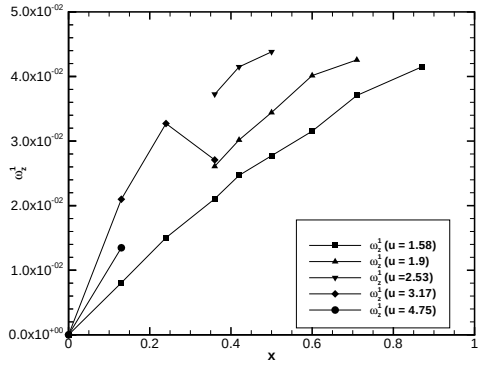


Figure 4.18: Angular velocity ω_z^1 of the largest fragment F^1 against the impact factor x for collisions in coalescence mode.

angular momentum is conserved the angular velocity ω_z^1 depends only on the shape of the final fragment. Figure 4.18 shows that larger impact factors result in larger ω_z^1. Deviation from this behaviour is observed for $u = 3.17$ and $x = 0.36$ because of the slide-and-locking effect (q.v. Figure 4.12), which results in high values of $\mathbf{I}_{zz}$, due to the elongated shape of the formed cluster. The angular velocity, ω_z^1, depends on the inertia of the main fragment, F^1, thus depending on $r^{A,B}$. The normalised distance $r^{|A,B|}$(normalisation factor is $1/d$) between the clusters' centres is plotted in Figure 4.16. For low velocities, $u = 1.58, 1.9$, $r^{|A,B|}$follows a linear dependency on x. At higher values (e.g., for $u = 3.17$) the effect of sliding reflects on the steep increase of $r^{|A,B|}$, especially for higher x.

4.3.5 Stretching separation (II)

In stretching separation, the common fragment, i.e. the cluster or droplet that is temporarily formed by the two original clusters, separates again into two main fragments and, possibly, into additional satellite fragments. Stretching separation occurs for high impact factors and/or at high velocities (see Figure 4.5b), and can be further classified into three submodes: (a) *normal stretching separation*, (b) *stretching separation with satellite droplets* and (c) *shearing-off*. A collision is considered to belong to submode (b) if the there is at least one large fragment beside the two main fragments, while the submodes (a) and (c) can be distinguished only by observation.

Normal stretching separation (IIa)

Collision dynamics in normal stretching separation share similarities with the slide-and-locking coalescence mode (Figure 4.6). The major difference is that for the normal stretching separation the sliding motion is so distinct that stretching of the temporarily formed cluster causes separation again. Below, the collision dynamics for a typical case corresponding to $x = 0.42$ and $u = 3.17$ (Figure 4.19) is explained. The first part of the collision is identical to that in slide-and-locking mode. Following the impact, both clusters are flat-

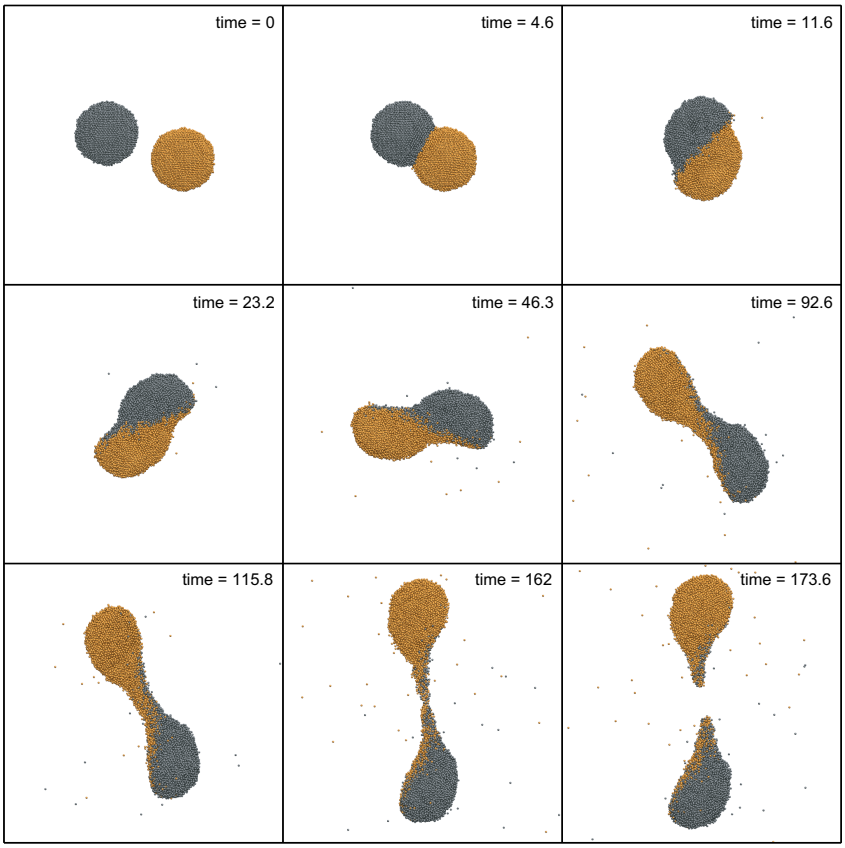


Figure 4.19: Collision dynamics in normal stretching separation mode (IIa) for $x = 0.42$ and $u = 3.17$.

tened along the interface plane and start to slide along each other. In contrast to mode (Ib) the sliding motion lasts longer because of higher velocity u and/or smaller contact interface due to higher x. As the sliding continues the distance between the original clusters increases. This leads to a narrower contact interface and reduced binding forces. Eventually, the contact interface shrinks until it breaks. At this point the temporary common fragment has turned around 135 degrees before splitting into two clusters again. The centrifugal forces prevail over the binding forces, whereas in mode (Ib) the situation is reversed. Figure 4.19 shows that from $t = 23.2$ onwards the interface starts to contract until it becomes very thin at $t = 115.2$ and eventually breaks apart.

The trajectories of the original clusters C^A and C^B, as well as the distance $r^{A,B}$are plotted in figures 4.20 and 4.21, respectively. The trajectories indicate sliding after the first contact of the clusters. The clusters enter a circular path and the distance $r^{A,B}$increases linearly (or almost linearly) for all cases in collision mode (II). Note that there is no discontinuity at the separation point.

The conversion of the initial translational energy $E_{kl,ext}$ is identical to mode (Ib). The only difference is that during stretching and separation the angular energy of the temporary

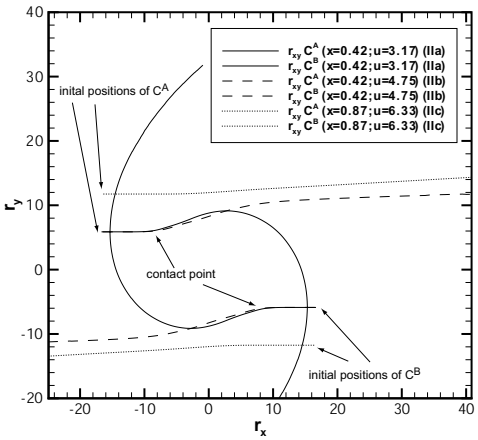


Figure 4.20: Trajectories of the clusters C^A and C^B for the (IIa), (IIb) and (IIc) submodes.

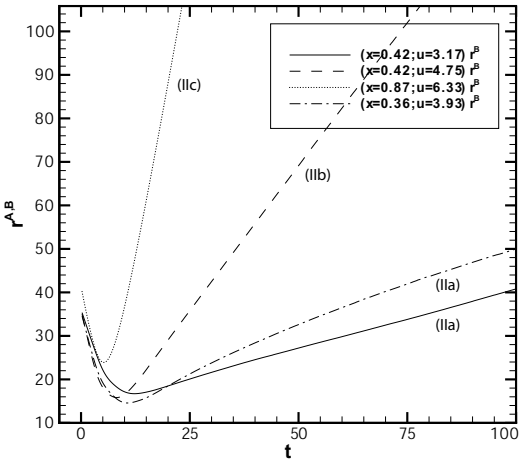


Figure 4.21: Distance $r^{A,B}$ between the mass centres r^A and r^B of the original clusters C^A and C^B as function of time for collision dynamics in stretching separation modes (IIa), (IIb and (IIc); see text for more details.

aggregate is transformed back into translational energy, $E_{kl,ext}$, of the two final main fragments.

Stretching separation with satellite droplets (IIb)

For high velocities ($u > 3.93$) and for a broad range of moderate impact factors ($0.13 \leq x \leq 0.7$), the collisions belong to stretching separation mode including production of satellite droplets (Figure 4.6). A satellite droplet is a large fragment which is not, however, one of the main fragments. Figure 4.22 shows the development of the collision for $x = 0.42$ and $u = 4.75$. Note that the impact parameter is the same as for mode (IIa), but at higher impact velocity. Similar to mode (IIa) the collision begins with flattening of the colliding clusters along a contact interface. Because of the higher initial impact energy in the present case, the process features scattering and faster stretching motion. Therefore, less rotation of the temporary common fragment occurs for (IIb) compared to (IIa).

The material at the contact interface is stretched into a long tubular mass. Around $t = 69$, one can observe the formation of three necks along the stretched connection. These are the points in which the connection breaks into two satellite droplets, while the two main fragments still continue to move straight in opposite directions (see Figure 4.22 for $t = 92.6$ and $t = 115.6$).

The trajectories of the main fragments are shown in Figure 4.20 (dashed lines). After a short sliding the main fragments continue to travel along a straight line and, therefore, the distance $r^{A,B}$increases linearly in time (Figure 4.21).

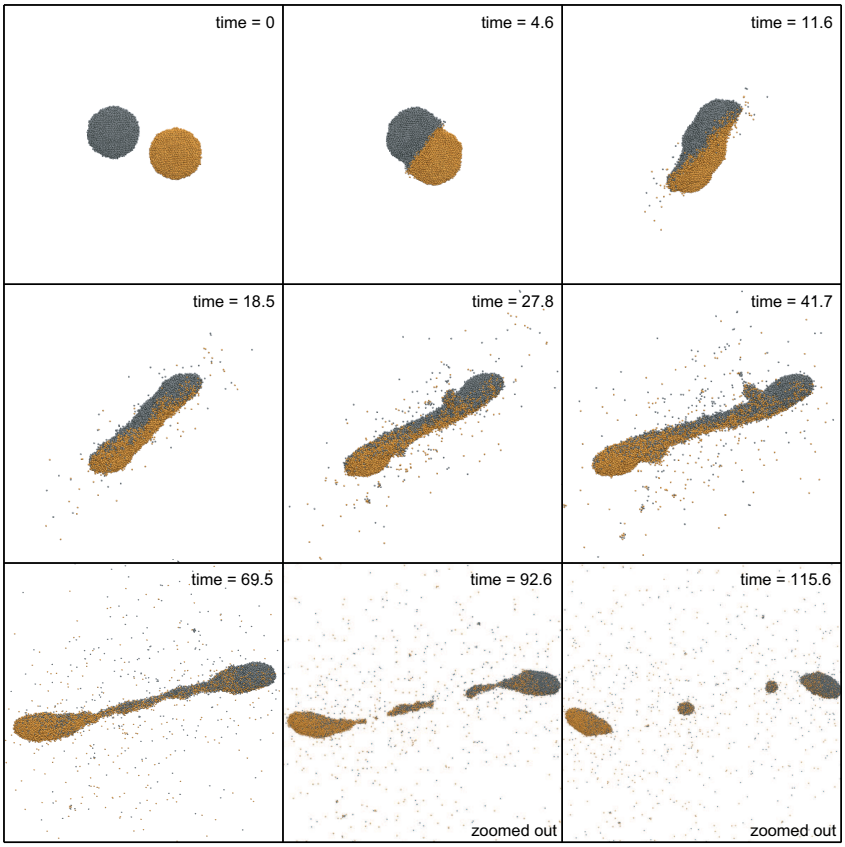


Figure 4.22: Collision dynamics for $x = 0.42$ and $u = 4.75$; the case belongs to stretching separation mode with satellite droplets (IIb).

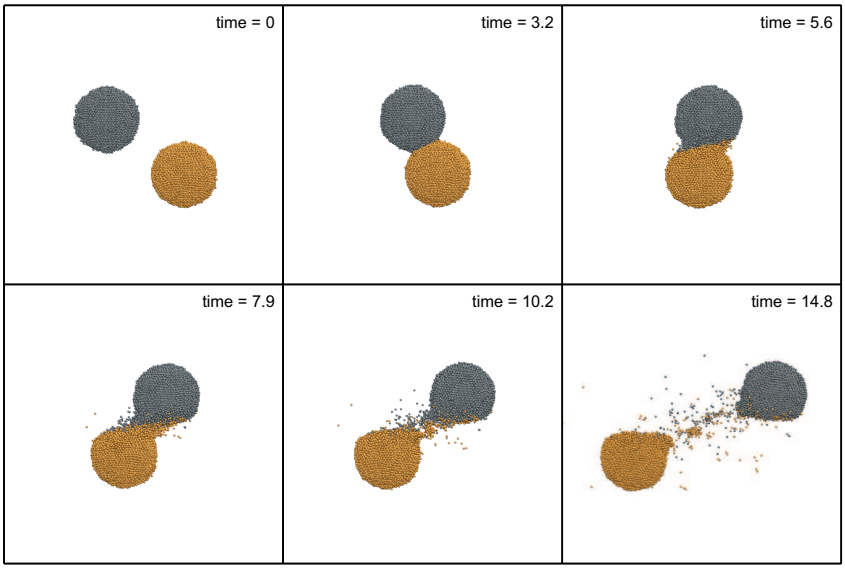


Figure 4.23: Collision dynamics for the shearing-off case ($x = 0.87, u = 6.33$).

Shearing-off (IIc)

For $x > 0.8$, the overlapping cross section of the original clusters (see Section 4.3.2) becomes very small thus the collision is minimised and the internal structure of the original clusters is not significantly altered. This case is referred to as shearing-off mode and a typical scenario is demonstrated in Figure 4.23 for $x = 0.87$ and $u = 6.33$. The trajectories are shown in Figure 4.20 and the distance of the mass centres in Figure 4.21. The binding forces of the material in the collision region are too small to withstand the inertia of the entire cluster, thus, fragments of the material are sheared-off. The collision also results in slight rotation of both clusters. In the shearing-off mode, a small percentage of $E_{kl,ext}$ is converted into potential energy, $E_{p,int}$, and kinetic energy, E_k, of the scattered atoms.

Effects of the impact factor on stretching separation mode

To assess the effects of the impact factor, x, on the clusters deflection and molecular mixing, two parameters have been employed: (i) the deflection, β, of a cluster, which is defined by the angle between the initial velocity vector, $\boldsymbol{u}$, and the velocity vector of the cluster after the collision; and (ii) the mixing parameter, mf, representing the mass fraction of a main fragment that consists of atoms from the other cluster.

The difference of β^{A} and β^{B} was insignificant for all cases: less than 5% for high u in connection with low x, otherwise less than 1%. Figure 4.24 shows the average of β^{A} and β^{B} against different impact factor values. As expected, β tends to be zero when $x \to 1$. For $u = 3.17$, β increases exponentially when decreasing x; lower x entails higher overlapping regions leading to stronger interaction with the highest value measured in the borderline to coalescence (for $x = 0.5$), where the common cluster rotates 135 degrees before it separates.

The mf values for the two main fragments differ less than 6% for low x combined with

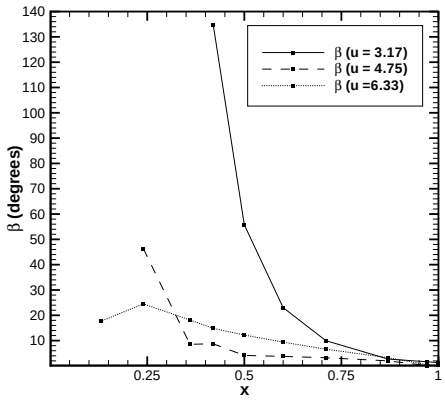


Figure 4.24: Average deflection β of clusters C^A and C^B against the impact factor, x, for collision dynamics in mode (II).

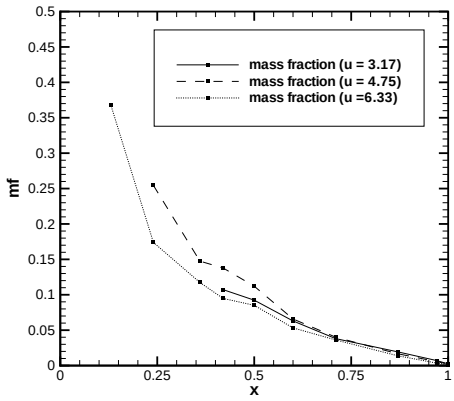


Figure 4.25: Average mass fraction mf against the impact factor x for collision dynamics in mode (II).

high u and less than 1% in all other cases. Figure 4.25 shows the averaged mass fraction mf against x. The three curves for the velocities almost coincide, suggesting that the impact velocity has no influence. Instead, the mixing depends primarily on the impact factor and increases faster for smaller values of x, reaching a peak value of $mf = 0.36$ for $x = 0.13$.

4.3.6 Discussion and relation to macroscopic droplet collisions

In the past, collision of droplets at macroscopic scales have been the subject of many experimental [13, 14, 23, 24, 151] and numerical [156, 164] investigations. The setup of these experiments is very similar to the present simulations. There are, however, two noticeable differences. Firstly, the length-scale of the present simulations is smaller by five orders of magnitude than the macroscopic droplet experiments. Secondly, in the macroscopic case the droplets are in liquid phase and retain the same temperature throughout the simulation. At nanoscale the temperature of the clusters varies significantly during the development of the collision: the clusters are initially in a solid phase and depending on the collision scenario they are transformed to liquid droplets. Like in previous studies [65, 133, 175, 199], similarities in the collision dynamics of droplets at macro and nanoscale are examined below, despite the mentioned differences.

Binary droplet collisions at macroscales are characterised by the impact factor, x, and the Weber number, $We = (\rho d/\gamma)\,u^2$, where ρ is the density and γ is the surface tension. Four modes have been observed: coalescence, stretching separation, reflexive separation and shattering. These modes correspond to the ones described in the previous section for collisions at nanoscale. The reflexive separation mode is observed at macroscales for almost central collisions ($x < 0.25$) at high velocities ($We \geq 19$) [14]. Here, the two droplets merge to a temporary common droplet, which is elastically flattened into a disc-like or a torus-like shape during impact. The subsequent reflexive motion, caused by surface ten-

sion, re-shapes the formed droplet into an elongated cylinder that eventually breaks up into two droplets. Depending on the values of the Weber number and impact factor satellite droplets may be formed.

For the length scales less than 10^{-8}m considered here, reflexive separation was not observed. This agrees with previous studies for clusters smaller than the ones considered here, e.g. LJ_{1000} [133], $(H2O)_{729}$ [199] and $(H2O)_{1000}$ [175]. It should be noted that the clusters in the present study are initially in solid state and, therefore, one cannot exclude the possibility that reflexive separation may occur for liquids at nanoscale. Because the clusters are immediately liquified one could argue that the initially solid state is not the reason for the absence of reflexive separation. Nevertheless, this is a subject of a separate investigation.

Because all macroscopic investigations have been carried out using liquid droplets, submodes associated with the coalescence mode have not been previously reported [14]. Comparison with the results of Brenn et al. [24] shows that the formation of satellite droplets in the stretching separation mode can be much more pronounced at macro than at nanoscales.

Previous investigations of macroscopic binary droplet collisions have pursued to find analytical criteria that describe the boundaries between coalescence and stretching separation [13, 14, 23]. Ming et al. [133] found satisfactory accuracy between MD simulation results for binary collisions of Ar_{1000} clusters and the analytical model predictions of Brazier-Smith et al. [23] . Brazier-Smith's model is based on the balance of centrifugal forces and forces originating from the surface tension. For droplets of the same diameter, d, the transition criterion is given by

$$x_{crit} = \left(c \sqrt{\frac{\gamma}{\rho d}} \right) \frac{1}{u} , \tag{4.3.7}$$

where different values of the constant c have been proposed: $c = 2.50$ by Brazier-Smith et al. [23] and $c = 3.47$ by Arkhipov et al. [13]. In the presented work, the density ρ of the initial cluster is $1.042\, m_a/\sigma^3$. In general, it is difficult to estimate the surface tension γ of Lennard-Jones clusters [173] since this depends on the internal structure and temperature of the clusters [168], which, however, changes during the collision process.

When using the same surface tension value as Ming et al. [133], i.e. $\gamma = 1.7\,\epsilon/\sigma^2$, one obtains similar predictions from the analytical model and simulations. However, the best agreement between the Brazier-Smith's/Arkhipov's model and the present results is obtained for $c\sqrt{\rho d/\gamma} = 1.2$ (Figure 4.26). For the d and ρ considered here, the value of $c\sqrt{\rho d/\gamma} = 1.2$ implies a surface tension γ in the range of $3.4 - 6.5\,\epsilon/\sigma^2$. These values are still within the range of the surface tension for Lennard-Jones clusters [97].

The discrepancies between the analytical models and simulations may be due to the following reasons:

For macroscopic particles, the temperature varies insignificantly compared to the nanoparticle case: the smaller the particle size the higher the temperature rises during the collision. Let us consider two pairs of particles a and b with diameters d_a and d_b, respectively, and a diameter ratio f: $d_b = d_a f$. For identical Weber numbers $u_b^2 = u_a^2/f$ thereby for smaller particles the impact velocity is higher. Using the definition of the Weber number, one

obtains

$$u_b^2 = \frac{We\,\gamma}{\rho d_a f}\ . \tag{4.3.8}$$

For central impact ($X = 0$), the external kinetic energy, $E_{kext} = \frac{1}{2}mu^2$, is converted entirely into internal energy. The temperature then changes as

$$\triangle T_b = \frac{E_{kextb}}{2cm} = \frac{mu_b^2}{4cm} = \frac{u_b^2}{4c} = \frac{We\,\gamma}{4c\rho d_a f}\ , \tag{4.3.9}$$

with c being the specific heat capacity, where u_b^2 has been replaced by 4.3.8. According to 4.3.9, $\triangle T_b \sim 1/f$, i.e. the smaller the particles, the higher the temperature rise (for binary collisions of comparable outcome).

As an example, compare the temperature rise for water droplets of diameter 500 μm, such as in Ashgriz's [14] experiments, with that of water particles of diameter 10 nm as used in the presented study. For a typical Weber number $We = 35$; density of water $\rho = 1003\,\mathrm{kg/m^3}$; surface tension $\gamma = 73{\times}10^{-3}\mathrm{N/m}$ and specific heat capacity $c_p = 4.187\,\mathrm{J/gK}$), the droplets of 500 μm colliding with a velocity of 2.26 m/s result in a temperature rise of 3×10^{-4}K. This temperature rise should not influence material properties such as surface tension and viscosity. However, for the nanoscale particles with $d = 10$ nm a collision with the same characteristics at 505 m/s will result in a temperature rise of 15 K. Monoatomic substances like the ones used in the present study have lower heat capacity and, therefore, the temperature rise will be even higher. For the cases considered in the present study, $We = 35$ and $X = 0$, the temperature rise was about 30 K (considering that the LJ-potential represents Argon cluster).

For the nanoparticles considered here, phase transformation occurs, with the temperature and phase of the material influencing the strength of the binding forces and, hence, the details of the collision dynamics including the transition regime.

Finally, scattering of atoms, especially at high impact velocities, reduces the size of the clusters significantly in some cases. This effect is not observed in macroscopic droplets and it is not, therefore, accounted for by (4.3.7). A better fit of the simulation data can be obtained by using $x_{crit} = au^b$ instead of (4.3.7), with $a = 2$ and $b = -1.5$ for best fitting. This result is plotted as a solid line in Figure 4.26.

4.3.7 Summary

MD offers the possibility of investigating the details of nanoparticle (molecular clusters or droplets) collisions, which are more difficult to study by experiments due to the small length and time scales involved. The collision dynamics of nanoclusters has been investigated for a broad range of impact factors and speeds. Similar to macroscopic droplets collision, coalescence and stretching were also found in the present case. The collision results obtained by MD simulations showed good agreement with collision models derived from results of experimental investigation of binary droplet collision. Reflexive separation, another mode observed in macroscopic droplet collisions, was not found for the collision of the initially solid nanoclusters.

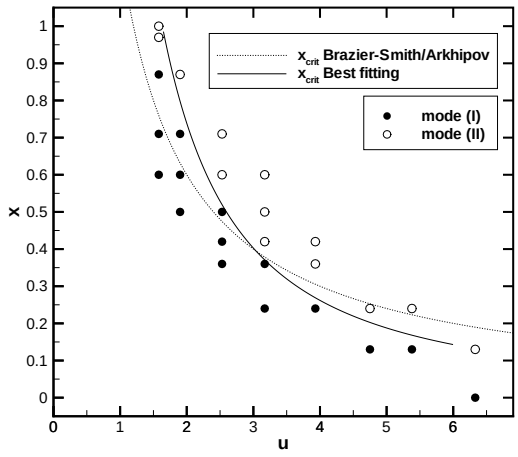


Figure 4.26: Comparison of the macroscopic analytical model (dashed line) of Brazier-Smith et al. [23] (Eq. 4.3.7) with the present MD simulation results as well as an alternative model $x_{crit} = au^b$, $a = 2$ and $b = -1.5$, (solid line) proposed here.

Detailed analysis of the collision processes has revealed different submodes that lead to coalescence, which are classified according to their dynamics into: sticking (Ia), slide-and-locking (Ib) and droplet (Ic) modes. Submodes also exist within stretching separation: normal stretching separation (IIa), stretching separation with satellite droplets (IIb) and shearing-off (IIc). Macroscopic analytical models that describe the borderline between coalescence and stretching separation were found to describe qualitatively the transition between the two modes at nanoscale. The development of a new analytical model that describes collision mode boundaries at nanoscale requires further understanding of the transition between modes (Ib) and (II). This requires simulations for larger clusters and droplets to be conducted in order to find the lower length-scale limit for the occurrence of reflexive separation. Future studies should also include a vapour atmosphere surrounding the cluster in order to capture the equilibrium between evaporation and condensation.

CHAPTER 5

Hybrid MD-CFD methods

This chapter discusses methods for multiscale modelling using CFD and MD methods in conjunction with geometrical decomposition. The principle function of geometrically coupled MD-CFD methods are explained, in particular the design of the hybrid solution interface (HSI). Different basic designs of the HSI, i.e. coupling by state or by flux, are described, including their advantages and deficiencies. The preferred design is selected and reasons for the choice are given.

In CFD the physical behaviour of the simulated material is modelled by the Navier-Stokes equations, which are based on the continuity assumption. For this reason, CFD can only be used for problems where the material within the entire simulation domain forms a continuum and for which an adequate macroscopic model can be defined on all boundaries. For the majority of macroscale problems, well-defined boundary conditions are available. However, when advancing to nanoscales, one inevitably reaches a point, where the continuity assumption breaks down because the atomic or molecular properties of the materials start to play a significant role. In such situations molecular simulation methods, such as MD, need to be used to obtain physically correct results. Theoretically, any problem, including macroscale problems, could be simulated using molecular methods, providing the computational effort is not prohibitive. Current computational resources only allow the simulation of systems smaller than a micrometer, for very short time intervals of few nanoseconds. Many interesting problems have larger characteristic length and time scales and are beyond the practical limits of MD methods. Still, in most situations, the continuum model is valid for the vast majority of the simulation domain and only small parts really require molecular modelling. In particular this is true for cases, where no macroscopic model for a boundary can be formulated. There is no convincing reason to waste computational resources by performing a complete MD simulation of a system that can be treated using CFD techniques for 99 % of the simulation domain. The obvious idea is to decompose the simulation domain and to use the fast CFD continuum methods for the large parts in which the continuum model is valid while the computationally expensive MD methods are only used within small parts for which the continuum model does not hold. An example of a conceivable setup is sketched in Figure 5.1, where the MD domain is a small part of the lower boundary. The red lines mark the interfaces between the molecular and continuum domains.

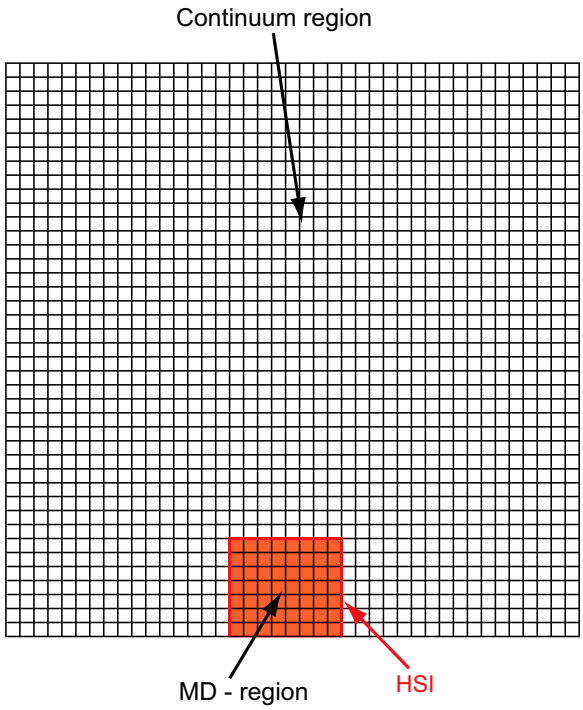


Figure 5.1: Schematics of a conceivable application of domain decomposition into continuum and molecular domain.

Within the individual domains the solution is computed independently through the according method. At the interfaces between the domains, a link needs to be established between the solutions of the two quite different simulation methods. It is the purpose of the *hybrid solution interface* (HSI) to create and maintain this link throughout the simulation and to join the domains into one single physical space. From the viewpoint of the individual simulation method, i.e. CFD or MD, it essentially acts as boundary condition. This perspective also points to another application of the MD-CFD coupling, which is to provide 'infinite' boundary conditions to a MD simulation. To this end, the MD simulation can be embedded completely or partly into the continuum domain to simulate the far field for the MD domain, which would be computationally far to expensive to be treated by MD. Such a setup would enable substantially more realistic MD boundary conditions than those currently available.

On question that arises is how to decide a priory, which parts of the simulation domain should be simulated by MD and which by CFD. Generally, there are two types of situations: (i) where it is unknown in advance which parts of the domain must be modelled molecularly or (ii) where geometrical features enable to determine the locations of MD regions. However, even if one knows where to place the MD regions, there is still the question of how large these regions need to be. Therefore, a sensible possible approach in situation (ii) is to perform a hybrid MD-CFD simulation with an initial guess and to correct the size of the MD domain if necessary. Alternatively and in case of situation (i), one can run the

entire domain as a CFD simulation and decide based on the simulation result, like the location of strong gradients, mixing regions or material interfaces, which regions need to be changed to MD. For DSMC-CFD hybrid methods, Garcia et al. [59] developed an adaptive mesh and algorithm refinement method that refines the mesh of the CFD solver depending on the flow situation and initiates a DSMC simulation in cells that are smaller than a certain critical size. For the MD-CFD hybrid methods a similar approach is conceivable, but the development of the necessary coupling methods has not progressed sufficiently far yet.

5.1 Principle HSI construction

The first step in constructing the HSI is to identify its required properties. Independent of the specific design, a number of requirements can be given:

- The conservation of mass, momentum and energy fluxes across the interface as the most basic physical principle must be satisfied. Any mass, momentum or energy flowing out at one side of the interface must enter on the opposite side of it.
- The gradients of the primitive variables must be continuous across the interface in space and time.
- The profiles of the primitive variables, such as density, velocity or temperature, across the interface must be physically correct. The position of the interface should not be recognisable from the solution of the simulation. This means that close to the interface, the hybrid MD-CFD solution should be identical to a solution obtained by using one simulation method, CFD or MD, exclusively.
- The design of the HSI should be as simple as possible.
- In terms of computational resources, the HSI must be inexpensive. Otherwise, the advantage in calculation speed over a pure MD simulation would be diminished.

In Figure 5.2 the principle construction of the HSI is sketched. At the interface the molecular and the continuum domain overlap; how far depends on the scheme used and will be explained in the subsequent sections of this chapter. The figure also indicates the two different ways of modelling the material in both domains, the continuum domain on the left and the molecular domain on the right. To solve the continuum domain, a finite volume method would be used, for which the domain is subdivided into cells. The state of the material, i.e. the primitive variables of density, momentum and energy density, is stored for each cell. The simulation is advanced by calculating the fluxes across the surfaces between the cells. The obtained fluxes are then integrated over the faces of each cell and added to the primitive variables that define the cell state. The cells adjacent to the interface, which are coloured blue in Figure 5.2, are called boundary cells. In the displayed case, their right side coincides with the interface to the molecular domain. The cells on the right of the boundary cell are so called ghost cells. These are auxiliary cells with a single purpose of imposing the boundary conditions onto the boundary cells.

In the molecular domain, the properties of each individual atom, i.e. its mass, position and velocity, are stored explicitly. These atoms move freely according to governing equations of MD and there is no explicit boundary confining them within the molecular domain.

Instead, the HSI needs to ensure that the atoms are kept within the molecular domain and do not move uncontrolled into the continuum domain. Because, the molecular dynamics method is grid less, *virtual cells* at the interface must be defined that can be mapped onto the respective continuum grid cells. The name 'virtual' symbolises that there are no real grid cells in the molecular domain; rather, the virtual cell specifies a volume and therefore the atoms inside the respective volume. For obvious reasons the virtual cells of the molecular side must be defined in such a way that they coincide with the ghost cells of the continuum domain. The coupling will be established between the respective continuum and virtual molecular cells whose normal projection on the interface coincide. The number of overlapping cell rows depends on the coupling scheme, with a minimum of two cells.

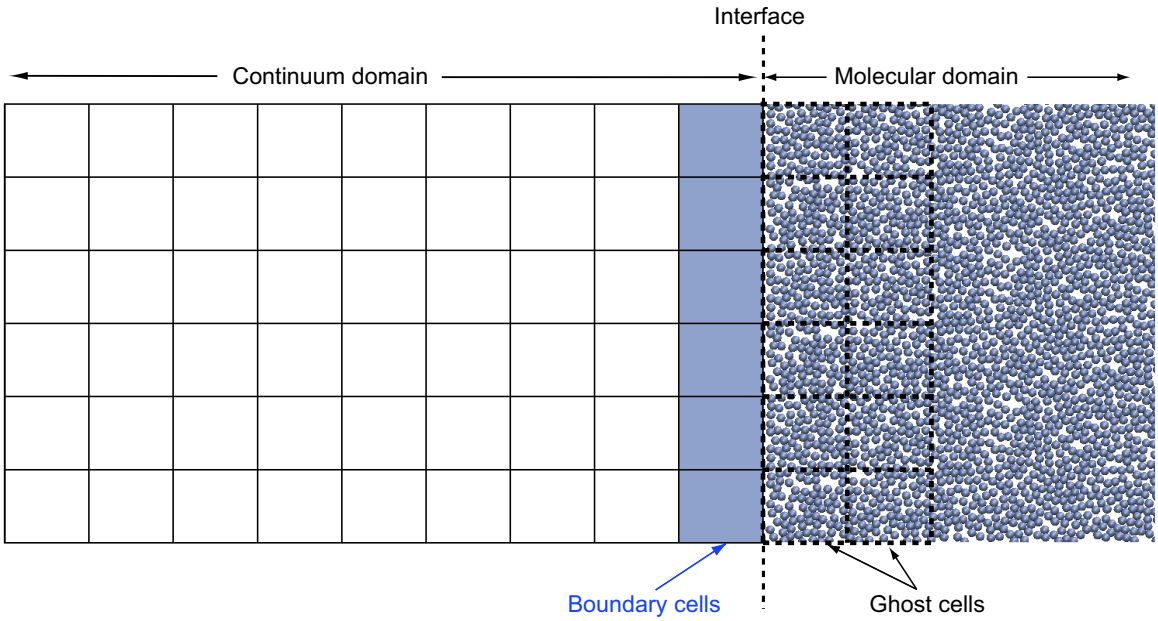


Figure 5.2: Principle overview of the HSI between CFD grid on the left side and atom based MD domain on the right.

In any conceivable MD-CFD coupling scheme the coupling must be established in two directions: from molecular to continuum (M → C) and from continuum to molecular (C → M). These two parts are largely independent of each other, which is easily understood when considering the case where one of the domains is kept constant, so that only the part of the coupling in direction of the non-constant domain has an effect. Of course, in a fully coupled situation both directions will influence each other, though indirectly, through changes in the states of the materials close to the interface.

For both directions, the coupling is performed in three steps: 1. Calculation of the necessary quantities on the source side; 2. Translation of these quantities into the formulation of the destination side; 3. Imposing the translated quantities onto the destination side. On which quantity the coupling should be based on, has been the subject of previous investigations on the coupling of continuum and molecular simulation methods. It has been found

that it can be based on cell states or on fluxes [38]. These two different approaches are discussed within the next two sections.

5.2 Coupling by state

Coupling by states means that the coupling is realised trough calculating, transferring and imposing cell states, i.e. density, momentum and energy [70]. Figure 5.3 is an illustration of the one-dimensional setup for coupling of states with the continuum domain on the left and the molecular domain on the right respectively. Physically, both domains form a single one-dimensional space. The exploded drawing fashion is used to give a better view on the functionality of this coupling approach. As can be seen in Figure 5.3, both domains are overlapping on the interface by one cell on each side. Generally, this could be extended to a higher number of cell pairs, for example when employing higher order CFD schemes. More than one cell one the molecular side would enable to enforce gradients and instead of pure states. However, for the purposes of explaining the principle, the minimum cell number is sufficient. The red arrows between the two domains symbolise the direction of the coupling. For the M → C coupling, the cell state of the inner virtual molecular cell is calculated, translated into continuum formulation and imposed onto the ghost cell of the continuum domain. Inversely, the C → M coupling is established by translating the cell state of the boundary cell of the continuum domain into molecular formulation and imposing it onto the outer virtual cell of the molecular domain.

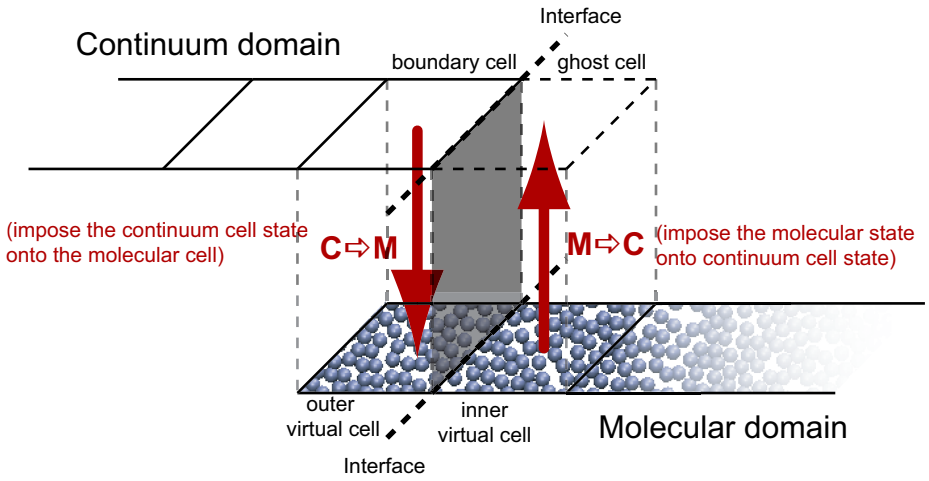


Figure 5.3: A one-dimensional illustration of coupling by state in both directions. The continuum and molecular domains are exploded to aid visualisation. The arrows symbolise the transfer of cell states.

To translate cell states between the continuum and molecular formulation, it is necessary to relate continuum and molecular cell quantities to each other. Table 5.1 compares the primitive variables[1] and the corresponding molecular formulas for the cell states. The sums in the molecular formulas are taken over all the atoms i within the cell C, which has

[1] In continuum formulation, variables that describe the cell state are denoted as primitive variables.

the volume V_C. The assignment of the atoms to the cell is done either by the NCP or CIC scheme. These are described in Section 3.1.

It is the fundamental assumption of the continuum formulation that each quantity represents the local average of a large number of atoms over a sufficient period of time. That means that the represented material is in a local equilibrium, i.e. microscopic fluctuations in space and time are negligible. On the molecular side, in most cases the number of atoms over which the sum is taken will be far below the required number of atoms for achieving practically zero fluctuations. Hence, the molecular side will be subject to fluctuations which have no equivalent expression on the continuum side. To compensate for this discrepancy one can introduce fluctuation terms into the continuum equations [48]. Without extra fluctuation terms, the standard way of reducing the fluctuations to an acceptable level is to average the molecular expressions in time, which is expressed by the angular brackets ($\langle \ldots \rangle_t$) in the equations 5.2.1 – 5.2.3. For a detailed discussion on time averaging, including choosing the appropriate parameters, such as sampling rate and minimum sample number, the reader should refer to the previous Section 3.3.

For reasons of lucidity the macroscopic model is based on a single species; multi species cases would require additional primitive variables. Such cases are not addressed in this work.

Both models, the continuum and molecular, describe the same cell state. Obviously, the degrees of freedom in the microscopic formulation of the molecular version are much greater than that of the macroscopic continuum one. For a mono atomic system, the molecular side has $Dof_M = 6N$ degrees of freedom, compared to $Dof_C = 5$ of the continuum side. In thermodynamic language this is the relation of the number of microscopic states that correspond to the macroscopic state, known as entropy. If the material is locally in equilibrium, then the entropy must be maximal.

The M $\rightarrow$ C coupling by state is relatively uncomplicated, as ρ, $\rho\boldsymbol{u}$ and e can be obtained directly from the molecular formulation. These values are simply assigned directly to the corresponding ghost cell. In terms of information transfer, the detailed microscopic description is condensed into five macroscopic primitive variables.

The C $\rightarrow$ M coupling is more difficult [193]. It requires the reconstruction of the microscopic state with $Dof_M = 6N$ from the macroscopic state with $Dof_C = 5$. The difference in degrees of freedom means that the microscopic state can not be generated from the macroscopic information without additional assumptions or stochastically generated values. While the number of atoms in the cell can easily be obtained by $N_C = \rho V_C$, the positions and velocities must be chosen in a way that equations 5.2.2 and 5.2.3 are satisfied. Still, a tremendously large number of possible combinations fulfill these requirements. In order to satisfy the second law of thermodynamics, only microscopic states with maximum entropy are allowed. Therefore, positions and velocities of the atoms in the outer virtual cell must be generated randomly in a way that satisfies equations 5.2.1 – 5.2.3 and the second law of thermodynamics. This is an extremely difficult task, algorithmically and in

Quantity	Continuum		Molecular	Equation
Density	ρ	$=$	$\frac{1}{V_C}\left\langle \sum_{i \in C} m_i \right\rangle_t$	(5.2.1)
Momentum	$\rho \boldsymbol{u}$	$=$	$\frac{1}{V_C}\left\langle \sum_{i \in C} m_i \boldsymbol{v}_i \right\rangle_t$	(5.2.2)
Energy	e	$=$	$\frac{1}{V_C}\left\langle \sum_{i \in C} E_{p_i} + E_{ki} \right\rangle_t$	
		$=$	$\frac{1}{V_C}\left\langle \sum_{i \in C} E_{p_i} + \frac{1}{2} m_i \boldsymbol{v}_i^2 \right\rangle_t$	(5.2.3)

Table 5.1: Primitive variables in continuum and molecular formulation.

terms of computational effort.

5.3 Coupling by fluxes

Contrary to coupling by state, coupling by fluxes does not exchange the state of material itself but fluxes of the properties that define the state, i.e. the conservative variables. Thus, coupling by fluxes is close to the functionality of a finite volume CFD-sover [54], where the algorithm calculates the fluxes between adjacent cells and changes the conservative variables by the sum of the these fluxes (compare Section 2.3). The flux of the conservative variables mass, momentum and energy is distinguished by the way of transport into flux by convection, stress and conduction [38]. A convective flux is the transport of a quantity bound to a fluid element which moves from one place to another through the flow field. Flux by stress and flux by conduction are the transport of a conservative quantity through interatomic forces, predominantly in solids, and/or diffusion, predominantly in gases. For a single species compressible material, there are six different fluxes in total:

- convective mass flux F_{mc},
- convective momentum flux $\boldsymbol{F}_{pc}$,
- convective energy flux F_{ec},
- momentum flux due to stress F_{ps},
- energy flux due to stress (dissipation) F_{es} and
- conductive energy flux F_{eq}.

For incompressible isothermal cases, the fluxes are reduced to mass flux and momentum fluxes by convection and by stress, which simplifies the coupling by fluxes significantly. For this reasons, most previous coupling investigations have been limited to isothermal incompressible flows [17, 47, 193].

Figure 5.4 gives an illustration of coupling by fluxes. Similar to the previous figure for coupling by states, the one-dimensional physical space is divided into a continuum and molecular domain, which overlap at the interface by one cell on each side.

For the C → M direction, the fluxes are calculated by the CFD-solver on the western face of the boundary cell. Then, these fluxes are translated into molecular formulation and imposed onto the outer virtual cell of the molecular domain. Reversely, the M → C coupling is established by calculating the fluxes across the eastern face of the inner virtual cell of the molecular domain and imposing them onto the ghost cell in the continuum domain. It should be noted that the described combination of cells is not the only possible one. Moreover, it is imaginable to establish flux coupling in other combinations as well. However, this work concentrates on the basic principles; and these are the same for every combination.

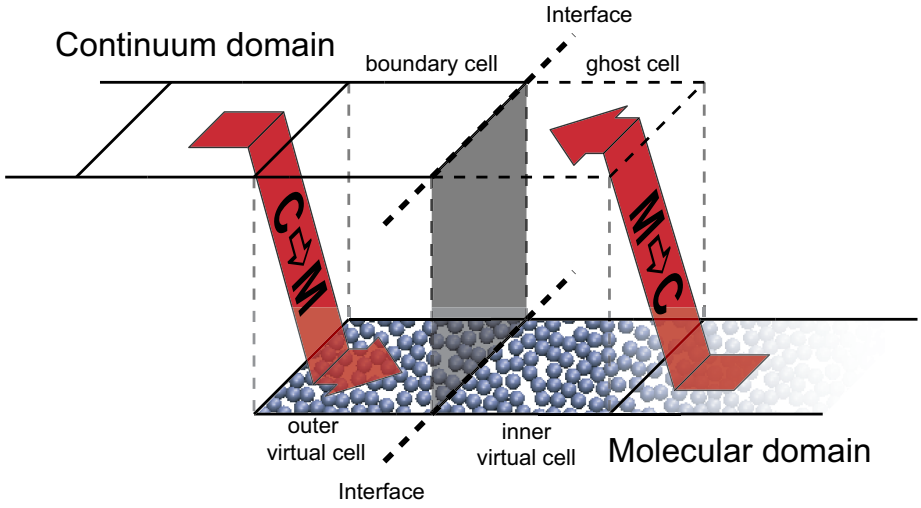
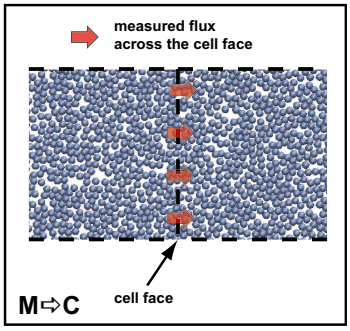


Figure 5.4: A one-dimensional illustration of coupling by fluxes in both directions. The continuum and molecular domains are exploded to aid visualisation. The arrows symbolise imposing the fluxes that have been calculated across the face between two cells of one domain onto the respective cell in the other domain.

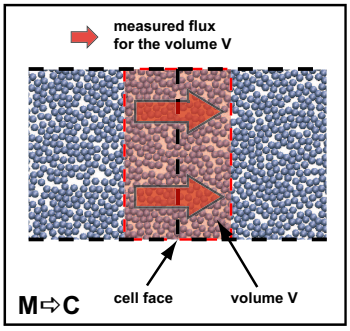
Before beginning to discuss the fluxes coupling, it is necessary to establish a number of fundamental conventions. The flux of any quantity, $\mathcal{A}$, can be given either for a volume, V, or across a surface S. For a volume, the flux is a vector quantity that states how much of $\mathcal{A}$ per volume and time flows in which direction:

$$\boldsymbol{F}_{\mathcal{A}}^{V} = \mathcal{A}\boldsymbol{n}\,, \tag{5.3.1}$$

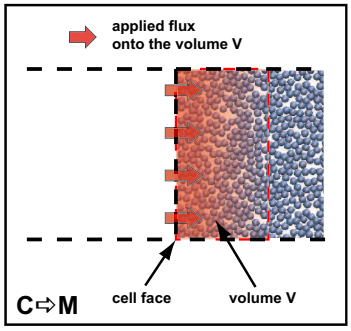
where the unity vector $\boldsymbol{n}$ points in the direction of the flux. $\mathcal{A}$ is the quantity per volume and time. The flux of $\mathcal{A}$ across the surface, $F_{\mathcal{A}}^{S}$, is a scalar that states how much of $\mathcal{A}$ flows



(a) Calculating molecular flux across the face



(b) Calculating molecular flux from a volume



(c) Imposing flux across a surface onto a molecular volume

Figure 5.5: Flux coupling schemes on the molecular side.

across S per volume and time. The volume and surface fluxes are connected through

$$F_{\mathcal{A}}^{S} = \boldsymbol{F}_{\mathcal{A}}^{V} \cdot \boldsymbol{n}_{S} \ , \tag{5.3.2}$$

where $\boldsymbol{n}_S$ is the unity normal vector of the surface S. Any flux in this document that is given without specific specification refers to a surface flux, i.e. $F_{\mathcal{A}} = F_{\mathcal{A}}^{S}$.

The fluxes of a quantity $\mathcal{A}$ across the surfaces that confine a cell are related to the change of $\mathcal{A}$ inside this cell. To this end, the directions of the cell faces must be defined and for this work, the surface vectors of the cell faces point inwards for each cell. For a three-dimensional cartesian grid, each cell is enclosed by six faces with the normal vectors $n_w = (1,0,0)^{\mathrm{T}}$ on the western face, $n_e = (-1,0,0)^{\mathrm{T}}$ on the eastern face, $n_s = (0,1,0)^{\mathrm{T}}$ on the southern face, $n_n = (0,-1,0)^{\mathrm{T}}$ on the northern face, $n_b = (0,0,1)^{\mathrm{T}}$ on the bottom face and $n_t = (0,0,-1)^{\mathrm{T}}$ on the top face. Thus, the rate of change of $\mathcal{A}$ inside a cell is the sum of all fluxes at each cell face $\boldsymbol{F}_{\mathcal{A}}^{V}$ times the respective cell surface vector n_S times the surface area A_S:

$$\frac{\partial \mathcal{A}}{\partial t} = \sum_{S} \boldsymbol{F}_{\mathcal{A}}^{V} \cdot (A_S \boldsymbol{n}_S) = \sum_{S} \boldsymbol{F}_{\mathcal{A}}^{S} A_S \ . \tag{5.3.3}$$

The sum is over the western, eastern, southern, northern, bottom and top cell faces, i.e. $S = \{w, e, s, n, b, t\}$.

5.3.1 M → C flux coupling

For the M → C direction of the flux coupling, there are two conceivable ways of calculating the fluxes on the molecular side. One approach, illustrated in Figure 5.5a, calculates the fluxes across the surface directly. The alternative, which is indicated in Figure 5.5b, is to obtain the fluxes from the atoms within a finite volume that encloses the surface of interest. The translation of the fluxes from the molecular to the continuum formulation differs in the two schemes.

Calculating molecular fluxes across the surface S: The molecular-continuum relations for the molecular mass, momentum and energy flux across the surface S are given by the

equations 5.3.4 – 5.3.9 in Table 5.2, where $\boldsymbol{n}_S$ is the normal vector of the surface across which the flux is calculated. For the reasons explained in the previous paragraphs on coupling by states, all molecular expressions are time averaged using Equation 3.3.1. It is important to note that the equated molecular and continuum expressions are no actual fluxes themselves, but rather the resulting increase or decrease of the flowing quantity in time, i.e. $\partial\mathcal{A}/\partial t$. This is due to the way in which the convective fluxes are calculated on the molecular side.

The convective fluxes in the equations 5.3.4 – 5.3.9 are obtained by summing over all atoms i that have crossed the surface S during the period of calculation. Their contribution is either positive or negative, depending on the direction in which they have moved across the surface S ($\text{sign}(\boldsymbol{v}_i \cdot \boldsymbol{n})$). Because the calculation is based on atoms that have crossed S, the convective fluxes are directly bound to the actual surface S.

The sum $\sum_{i,j}$ in the stress and dissipative flux terms stands for the double sum $\sum_i \sum_j$, where the former sum is over all atoms i on the side of the surface at which the surface vector $\boldsymbol{n}_S$ is pointing. The latter sum is over all atoms j on the other side of S. Here, only the pairwise interactions are included. For interactions between more atoms, the sum will be over the resulting forces on each atom i across the surface S.

Flux	Continuum	Molecular	Eq.
Mass			
convective	$\rho\boldsymbol{u}\cdot\boldsymbol{n}_S$	$= \frac{1}{A_S}\left\langle \sum_{\substack{i \text{ that} \\ \text{crossed } S}} m_i \,\text{sign}(\boldsymbol{v}_i\cdot\boldsymbol{n}_S)\right\rangle$	(5.3.4)
Momentum			
convective	$(\rho\boldsymbol{u}\otimes\boldsymbol{u})\cdot\boldsymbol{n}_S$	$= \frac{1}{A_S}\left\langle \sum_{\substack{i \text{ that} \\ \text{crossed } S}} m_i\boldsymbol{v}_i \,\text{sign}(\boldsymbol{v}_i\cdot\boldsymbol{n}_S)\right\rangle$	(5.3.5)
stress	$\boldsymbol{\Pi}\cdot\boldsymbol{n}_S$	$= \frac{1}{A_S}\left\langle \sum_{i,j} \boldsymbol{f}_{ij}\right\rangle$	(5.3.6)
Energy			
convective	$e\boldsymbol{u}\cdot\boldsymbol{n}_S$	$= \frac{1}{A_S}\left\langle \sum_{\substack{i \text{ that} \\ \text{crossed } S}} E_i \,\text{sign}(\boldsymbol{v}_i\cdot\boldsymbol{n}_S)\right\rangle$	(5.3.7)
dissipative	$(\boldsymbol{\Pi}\cdot\boldsymbol{u})\cdot\boldsymbol{n}_S$	$= \frac{1}{A_S}\left\langle \sum_{i,j} \boldsymbol{f}_{ij}\cdot\sum_i \boldsymbol{v}_i\right\rangle$	(5.3.8)
conductive	$\boldsymbol{q}\cdot\boldsymbol{n}_S$	$= \frac{1}{A_S}\left\langle \sum_{i,j} \boldsymbol{f}_{ij}\cdot\boldsymbol{v}_i - \sum_{i,j}\boldsymbol{f}_{ij}\cdot\sum_i \boldsymbol{v}_i\right\rangle$	(5.3.9)

Table 5.2: Molecular fluxes calculated across a surface S translated to continuum fluxes.

Calculating molecular fluxes from a volume V_S: Calculating the fluxes from a volume V_S that encloses the surface S, first requires the definition of the volume V_S [188]. In Figure 5.5b this volume is the red shaded area. The size of V_S must be chosen as a compromise

between minimising the fluctuations and accuracy in terms of position. In Table 5.3 the relations to translate the fluxes from molecular to continuum formulation are given. Unlike the flux calculation across the surface, these expressions are actual fluxes and stand for the flow velocity of the respective quantity. As for the previously given relations, the molecular fluxes are time averaged to cancel the intrinsic fluctuations. The double sums $\sum_{i,j\in V_S}$ in the formulations for stress and dissipative fluxes are over all pairs of atoms inside the volume V_S. The term $\boldsymbol{r}_{ij} \otimes \boldsymbol{f}_{ij}$ is effectively a viral [127] and may be substituted by other equivalent formulations for the local stresses in molecular system, such as the one from Zimmerman et al. [203]. The dissipative and conductive energy fluxes have been merged into Equation 5.3.14, because it was not possible to find separate expressions for each of the fluxes in molecular formulation for a finite volume. The reason for this is that both the energy transfer by dissipation and by heat conduction have their molecular origin in diffusion processes and energy exchange through interatomic forces. These can not be separated in a situation that involves both energy transfer mechanisms.

Flux	Continuum	Molecular	Eq.
Mass			
convective	$\rho\boldsymbol{u}$	$= \frac{1}{V_S}\left\langle \sum_{i \in V_S} m_i \boldsymbol{v}_i \right\rangle$	(5.3.10)
Momentum			
convective	$\rho\boldsymbol{u}\otimes\boldsymbol{u}$	$= \frac{1}{V_S}\left\langle \sum_{i \in V_S} m_i \boldsymbol{v}_i \otimes \boldsymbol{v}_i \right\rangle$	(5.3.11)
stress	$\boldsymbol{\Pi}$	$= \frac{1}{V_S}\left\langle \sum_{i,j \in V_S} \boldsymbol{r}_{ij} \otimes \boldsymbol{f}_{ij} \right\rangle$	(5.3.12)
Energy			
convective	$e\boldsymbol{u}$	$= \frac{1}{V_S}\left\langle \sum_{i \in V_S} E_i \boldsymbol{v}_i \right\rangle$	(5.3.13)
dissipative + conductive	$\boldsymbol{\Pi}\cdot\boldsymbol{u} + \boldsymbol{q}$	$= \frac{1}{V_S}\left\langle \sum_{i,j \in V_S} (\boldsymbol{r}_{ij} \otimes \boldsymbol{f}_{ij}) \cdot \boldsymbol{v}_i \right\rangle$	(5.3.14)

Table 5.3: Molecular fluxes calculated from a volume V_S translated to continuum fluxes.

5.3.2 C → M flux coupling

For flux coupling in C → M direction, the fluxes are calculated by the CFD-solver, translated to molecular formulation and imposed onto a virtual cell in the molecular domain. In Figure 5.5c, the C → M flux transfer is illustrated for the one-dimensional case of Figure 5.4. The cell face corresponds to the left cell face of the outer virtual molecular cell. It also forms the boundary of the molecular domain. Generally, no atoms are on the left side of the cell face, though this depends on the scheme which is used to impose the flux onto the molecular region. The flux imposing scheme must ensure that the exact values of

the mass, momentum and energy fluxes that have been calculated by the CFD-solver are introduced into the cell.

Table 5.4 is a summary of the translation from continuum into molecular formulation for the C → M coupling. Positive convective mass fluxes, $\rho \boldsymbol{u} \cdot \boldsymbol{n}$, correspond to an influx of atoms across the cell face into the outer virtual cell. Inversely, a negative convective mass flux means that atoms flow out of the cell across the cell face. Because the cell face also forms the outer boundary of the molecular domain, inflowing atoms must be generated and inserted. Respectively, outflowing atoms must be deleted. How this can be accomplished is the subject of the following chapter.

The atomic in- and outflows in Equation 5.3.15 are accounted by the sum over the generated or removed atoms. The term s_i, which is defined in Equation 5.3.21, stands for the increasing or decreasing contribution of the atom i, depending on whether the atom is inserted or removed. s_i also occurs in the convective momentum and energy flux equations 5.3.16 and 5.3.18.

Momentum transfer due to stresses are related in Equation 5.3.17. On a molecular level, the momentum flux by stress, $(\boldsymbol{\Pi} \cdot \boldsymbol{u}) \cdot \boldsymbol{n_S}$, corresponds to an external force, $\boldsymbol{f}^{ext}$, across the cell face on the atoms close to the surface. The distribution of the total force, $\boldsymbol{F}^{ext}$, into the individual forces, $\boldsymbol{f}_i^{ext}$, onto each atom i depends on the employed scheme. Applying the external force, f^{ext}, onto the atoms on the boundary an atom entails an energy transfer of $\sum_i \boldsymbol{f}_i \cdot \boldsymbol{v}_i$, which is required to match the dissipative energy flux, $(\boldsymbol{\Pi} \cdot \boldsymbol{u}) \cdot \boldsymbol{n}$ (Equation 5.3.19). As will be demonstrated in the following sections, matching these fluxes is one of the major difficulties in designing a HSI.

Finally, the conductive heat flux, $\boldsymbol{q} \cdot \boldsymbol{n}_S$, is translated in a molecular energy flux $\langle Q \rangle$. The molecular energy flux may be realised easily through velocity scaling or by employing a thermostat.

5.4 Which type of coupling?

The two previous sections gave a description of the different principle construction possibilities for the HSI between CFD and MD in both directions. The logical question, which one of them is best to be used, can currently not be answered with complete satisfaction, because the research field of hybrid MD-CFD methods is still in its infancy. This work focuses on a combination of coupling by state for the M → C direction and coupling by fluxes for the C → M direction. An illustration of the setup is given in Figure 5.6. The reason for this choice becomes clear when considering the advantages and the involved difficulties of different coupling types.

M → C direction: For the M → C direction, coupling by state is the most easy and uncomplicated version. Using equations 5.2.1 – 5.2.3, the states, i.e. the primitive variables, are easily calculated for the virtual molecular cells and can simply be assigned to the corresponding continuum cells. The procedure requires minimum computational effort, since

Flux	Continuum		Molecular	Equation
Mass				
convective	$\rho \boldsymbol{u} \cdot \boldsymbol{n}_S$	$=$	$\frac{1}{A_S}\left\langle \sum_i m_i s_i \right\rangle$	(5.3.15)
Momentum				
convective	$(\rho \boldsymbol{u} \otimes \boldsymbol{u}) \cdot \boldsymbol{n}_S$	$=$	$\frac{1}{A_S}\left\langle \sum_i m_i \boldsymbol{v}_i s_i \right\rangle$	(5.3.16)
stress	$\boldsymbol{\Pi} \cdot \boldsymbol{n}_S$	$=$	$\frac{1}{A_S}\left\langle \sum_i \boldsymbol{f}_i^{ext} \right\rangle$	(5.3.17)
Energy				
convective	$e\boldsymbol{u} \cdot \boldsymbol{n}_S$	$=$	$\frac{1}{A_S}\left\langle \sum_i E_i s_i \right\rangle$	(5.3.18)
dissipative	$(\boldsymbol{\Pi} \cdot \boldsymbol{u}) \cdot \boldsymbol{n}_S$	$=$	$\frac{1}{A_S}\left\langle \sum_i \boldsymbol{f}_i^{ext} \cdot \boldsymbol{v}_i \right\rangle$	(5.3.19)
conductive	$\boldsymbol{q} \cdot \boldsymbol{n}_S$	$=$	$\frac{1}{A_S}\langle Q \rangle$	(5.3.20)
Auxiliary terms				
convective	$s_i = \begin{cases} 1 & \text{for an inflown atom,} \\ -1 & \text{for an outflown atom} \end{cases}$			(5.3.21)

Table 5.4: Continuum fluxes across a surface translated to molecular fluxes and applied to a molecular volume.

it only involves the summation of basic atomic properties that are directly stored by the MD-solver, i.e. $\{m_i\}$, $\{\boldsymbol{r}_i\}$ and $\{\boldsymbol{v}_i\}$ are calculated each time step, i.e. $\{E_i\}$. The alternatives, which are to calculate the fluxes over a cell face or for a volume enclosing the cell face, pose some difficulties.

For calculating fluxes across the surface, problems arise in conjunction with the numerical integrator. When an atom crosses the surface that is interacting with other atoms ambiguities arise over the contribution of this atom to the stress and energy terms in equations 5.3.6 and 5.3.8. Another problem is that most time integration methods, like the velocity Verlet scheme, do not evaluate the positions, velocities and forces for the exactly same points in time, which also increases the uncertainty of the calculation. Unlike for the M $\rightarrow$ C coupling by state, the computational effort is significantly higher due to the requirement to evaluate the forces across the surface. Even if these forces are computed by the standard routines of the MD-solver, it is, at least, necessary to alter this highly optimised part of the MD-code.

A very high computational effort is also involved in calculating the fluxes from a volume by using equations 5.3.10 – 5.3.14. Especially, the calculation of the stress tensor is a computationally intensive task, because it requires either averaging over a large number

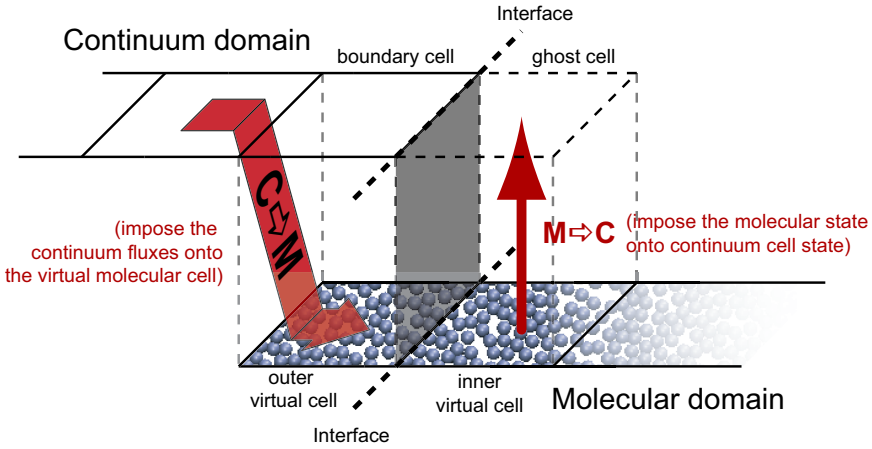


Figure 5.6: The investigated coupling scheme of the HSI between CFD grid on the left side and atom based MD domain on the right. Coupling by state is used for C → M direction and flux based coupling for the M → C direction [188].

of samples, the use of large cells sizes or the application of smoothing schemes like the thermodynamic field estimator [193]. Another important disadvantage is that not the exact flux, but only an average over the volume and the period of calculation, is obtained by using this method. Also, no separation between dissipative fluxes and conductive fluxes is possible.

C → M direction: The situation is different for coupling in the C → M direction. Generally, this direction is the most difficult part of any coupling scheme, due to the mentioned difference in degrees of freedom. If one decided to used C → M state coupling, it would require the reconstruction of density, velocity and energy density in each virtual molecular cell according to equations 5.2.1 – 5.2.3. For this sake, the microscopic state with its $6N$ degrees of freedom must be invented on the basis of five primitive variables. This is a very formidable task and can not be achieved easily, even for simple mono-atomic systems. Also, the computational effort is expected to be high.

On the other hand, the practicability of C → M flux coupling is less problematic. The main reason is that the motion of the atoms in the outer virtual molecular cells is still computed normally through the MD solver while the fluxes are imposed additionally. Calculating of fluxes across the cell faces is the core of a finite volume CFD solver, so that no additional effort for their calculation is required. Imposing these fluxes onto the virtual molecular cells poses difficulties as well, but contrary to the state version, the MD solver essentially calculates the motion of atoms within the virtual molecular cells in normal fashion. A scheme is required to impose the fluxes across the outer boundary onto the atoms inside the virtual molecular cells, and to add and delete atoms.

5.5 Development stages and development range

At this point, the practical and theoretical foundations for developing of a HSI between CFD and MD have been discussed. The different aspects can be separated into a number subtasks:

1. Development of a molecular dynamics code
2. Development or adaptation of a CFD code
3. Design and implementation of HSI for the CFD side
   (a) C → M direction
   (b) M → C direction (boundary conditions)
4. Design and implementation of the HSI for the MD side
   (a) M → C direction
   (b) C → M direction (boundary conditions)
5. Validation of the HSI for steady continuum boundary conditions
6. Validation of fully coupled MD-CFD simulation for steady and unsteady situations

The first stage has been covered in Section 2.2. The most important requirement of the HSI on the MD-solver is the capability to insert and remove atoms during the run of the simulation. LAMMPS fulfills this requirement and this is the major reason for choosing it as MD code.

The second stage is concerned with the development or adaption of a suitable CFD solver. The basic principles of continuum model and CFD solver have been discussed in Section 2.3. In principle, any CFD solver that is based on the finite volume method should be easily adaptable for being run in a coupled mode with a MD solver. The number of ghost cells beyond the boundary depends on the order of the numerical method and must be matched by the inner virtual cells in the MD region (see Figure 5.2). The implementation of the HSI design, as described in the previous section, requires the calculation of the fluxes across the boundary into the ghost cells and the transfer of this information to the MD solver (Point 3a). In the reverse direction, the primitive variables that are provided by the MD solver are assigned to the respective ghost cells (Point 3b). The required changes on the CFD solver are minimal.

In the fourth stage, the HSI is designed and implemented into the MD solver. The principles have been discussed in the previous sections of this chapter. For the chosen setup, i.e. , the direction M → C (Point 4a), the primitive variables have to be calculated for each virtual molecular cell. The details the of calculations are explained in chapter 3.

From all development stages, number 4b is the most critical. For the C → M flux coupling the task is to design and to implement a scheme for imposing mass, momentum and energy fluxes locally onto a MD simulation. More generally, one could regard this part of

the HSI scheme as MD flux boundary conditions. Because of its importance for the hybrid MD-CFD method and its challenges, the remainder of this work is dedicated to this Point 4b.

Stage 5 is essentially a test and validation programme of the MD flux boundary conditions for all standard flow situations. The test cases and their relevance are explained in Section 6. All test cases have been selected so that the continuum solution can be obtained analytically. Hence, the C $\rightarrow$ M fluxes can be calculated in advance and the MD flux boundary condition tested independently of the variations within the continuum domain. This is especially important in respect of finding a stable flux imposing design that does not rely on the CFD solver to compensate possible fluctuations or counterbalance instabilities of the flux imposition scheme.

The final stage 6 is beyond the scope of this work. The difficulties, which were involved in the pursuit of designing and validating MD flux boundary conditions have made it necessary to tackle this stage in the continued research programme, which is based on the work presented here. This includes all synchronisation issues that arise from the different time steps of the MD and CFD simulations. However, as stated before, the MD flux boundary conditions are the keystone of a successful hybrid MD-CFD simulation and it was therefore more prudent to achieve a stable and effective design before proceeding to fully coupled cases.

CHAPTER 6

Test Cases

The MD boundary conditions for the hybrid solution interface have been tested, analysed and validated for standard flow situations. A test case was devised for each flow situation. These are described in the coming chapter.

The flow on the boundary between continuum and molecular domain, i.e. at the HSI, can be subject to different flow situations. The multiplicity of conceivable situations can be reduced to a number of basic flow situations that may occur on the HSI:

- static case (fluid at rest),
- steady flows
  - flow parallel to the interface,
  - flow perpendicular to the interface,
- unsteady flows.

Any other more complex situation may be understood as a combination these and it is reasonable to assume that a HSI scheme that is validated for these basic situations will also work for other cases which are a mixture of the basic situations.

The static situation, i.e. where the fluid is at rest, is the most simple one. The only flux across the interface is the momentum flux due to the static pressure.

For steady flows, two fundamental situations can be distinguished: flows with the velocity vector parallel and flows with the velocity vector perpendicular to the interface. Flows parallel to the HSI do not involve mass flux across the interface. Thus, the only fluxes beside the momentum flux that is already present in the static situation are momentum and energy fluxes arising from viscous forces. The distinctive feature of flows perpendicular to the HSI is the transport of mass across the interface, which is represented by the convective mass flux that entails convective momentum and energy fluxes as well.

Unsteady flows comprise cases such as oscillatory flows, longitudinal waves, transversal waves, shock waves, turbulence and similar phenomena. No other fluxes than the ones that are already present for steady flows are involved. Higher complexity is caused by the time dependency of the fluxes. Unsteady flows are outside the scope of this work and must

be addressed in further investigations.

For steady flow situations, i.e. the static case, flow parallel and perpendicular to the boundary, suitable test cases have been devised to investigate the different schemes for the realisation of the HSI. Because the continuum side is constant in all employed test cases (steady), the fluxes across the HSI are constant too and can be precalculated based on the analytic solution, i.e. no CFD is used at this stage. From the continuum point of view, the basic flow situations are of one-dimensional nature. This is beneficial, because the investigation of the fundamental properties of coupling schemes should be done for situations with a minimum of complexity. Still, on the molecular side all simulations are three-dimensional. The connection to the one-dimensional continuum is enabled through the fact that only the boundaries in one dimension of the molecular domain are coupled to the continuum while the boundaries in the other two dimensions are realised through periodic boundary conditions.

For all basic flow situations, test simulations have been performed to validate the $\mathrm{C} \rightarrow \mathrm{M}$ coupling of the HSI for different coupling methods. In the subsequent sections, the test cases and respective validation criteria are described. The concrete geometrical dimensions of the test systems depend on the specific matter of state. Because each test case has been performed for different matter of states, no geometrical dimensions are given in the descriptions of this chapter. Instead, these are summarised in Table B.1 (Appendix B) for the performed simulations.

6.1 Static test case

The natural starting point for the investigation is the static situation. The setup for this test case is shown in Figure 6.1. The constant pressure boundary conditions are applied on the left side of the molecular domain (x dimension). The right side of the molecular domain is realised by a perfectly reflecting wall (see Section 7.7). Periodic boundary conditions have been applied in the y and z dimension.

The only flux across the HSI is the momentum flux that originates from the static pressure:

$$\boldsymbol{F}_{ps} = -p\mathbf{I} \cdot \boldsymbol{n} \tag{6.1.1}$$

where the pressure p can be calculated or measured in advance for the chosen density and temperature. For the concrete setup the momentum flux over the western face onto the molecular domain is

$$\boldsymbol{F}_{ps} = \begin{pmatrix} 0 \\ p \\ 0 \end{pmatrix} . \tag{6.1.2}$$

It can be expected that the HSI is working correctly if the ρ and T inside the molecular domain keep constant, the interface stays at its initial position and the profiles of ρ, $\boldsymbol{u}$ and T are constant across the molecular domain.

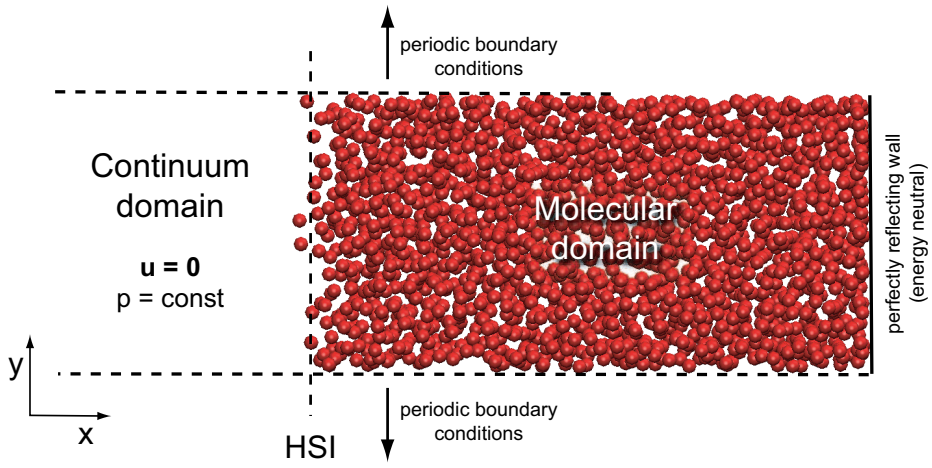


Figure 6.1: Schematic of the static test case.

6.2 Couette flow

A fundamental test case for the flow parallel to the interface is the Couette flow [10]. Originally, the Couette is the laminar flow of a viscous fluid between two parallel plates, where one plate is moving relative to the other. A distinguishing feature of the flow field between the plates or walls is that the only velocity component is parallel to the plates changing linearly from between the walls. All other velocity components are zero, which makes the Couette flow an ideal validation case of the HSI for the situation of flow parallel to the interface.

To ensure that the flow is parallel to the HSI, it needs to be aligned parallel to the plates. The basic setup is shown in Figure 6.2. The flow is from left to right, in x direction. Periodic boundary conditions are applied x and z dimension. The HSI forms the upper boundary in y dimension of the molecular domain; it is positioned at y_e. On the lower side, the liquid is confined by a wall of three atomic layers of Lennard-Jones atoms arranged in a fcc lattice. The interface between wall and liquid is at y_w, so that D is the distance between the lower wall and the HSI ($D = y_e - y_w$). In order to keep the wall in place, each atom is fixed by a spring force to its initial position. The energy and force model is described by the Equations (2.1.26) and (2.1.27) in Section 2.1.6. The wall is kept at a constant temperature, T_w, by velocity scaling of the wall atoms (see Section 2.2.9).

For the steady test case considered here, it is assumed that the flow speed in x direction, u_x, at y_e is u_e. With the velocities u_y and u_z being zero, the flow is parallel to the HSI and exclusively in x direction with the velocity u_x increasing linearly from $u_x = 0$ on lower side to maximum on the upper side. This is only exactly true for $Kn < 10^{-3}$. For $Kn > 10^{-3}$, the number of interatomic collisions is not high enough equilibrate the flow locally near the wall. In that case the velocity profile close to the wall is not linear and slip occurs. Due to viscous dissipation, the kinetic energy of the flow will partly transformed into heat through which the temperature rises within the flow field [10]. Because the lower wall is kept at constant temperature, the heat is conducted towards the wall which acts as

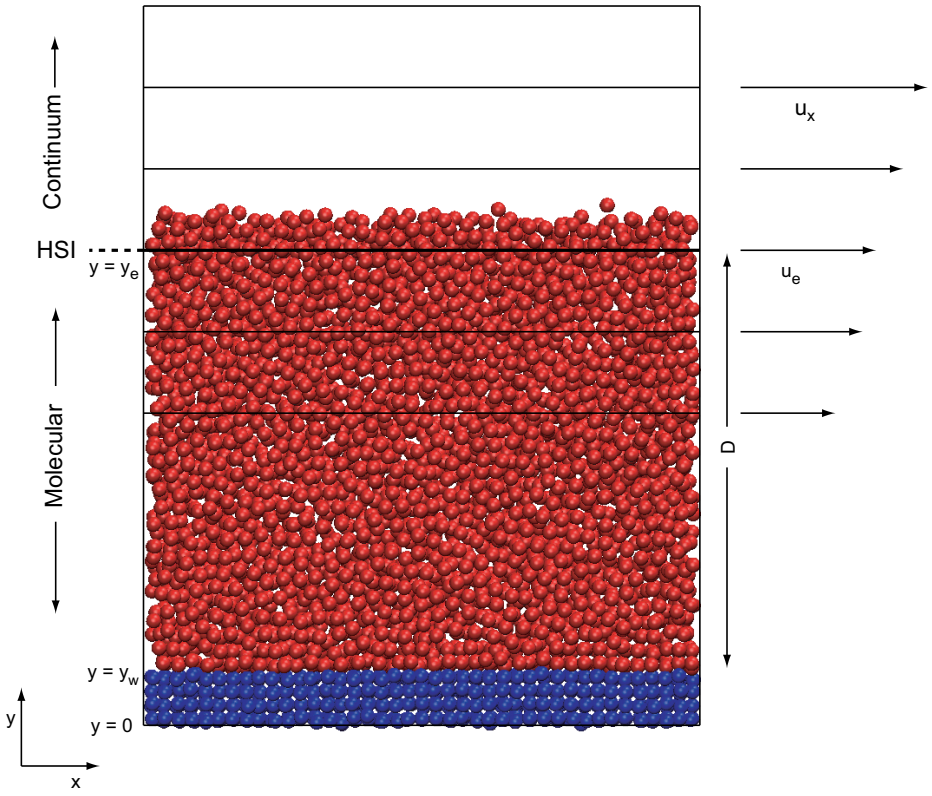


Figure 6.2: Schematic of the Couette test case for flow parallel to the HSI. The arrows on the right side symbolise the flow speed in x direction.

a heat sink. For moderate temperature gradients in connection with relative small D, the transport properties, i.e. viscosity μ and thermal conductivity λ, will be nearly identical throughout the flow field. In that case, an analytical solution can be given for the velocity and temperature profiles [10]:

$$u_x(y) = u_e \frac{y - y_w}{D} , \tag{6.2.1}$$

$$T(y) = T_w + \frac{\mu u_e^2}{2\lambda} \left[1 - \left(\frac{y_e - y}{D} \right)^2 \right] . \tag{6.2.2}$$

where the velocity profile of is obviously linear. The temperature profile is quadratic and determined by the ratio of viscosity to thermal conductivity, with the viscosity standing for how much heat is generated and the thermal conductivity how fast the heat is transported towards the wall. The temperature at the upper boundary, i.e. the HSI, is designated by T_e. To separate heat flux issues from the momentum flux validation, it is assumed that no heat is conducted across the HSI, i.e. that the interface is adiabatic. This is also a necessary condition for Equation (6.2.2). The temperature on the interface, T_e, is then obtained by

$$T_e = T_w + \frac{\mu u_e^2}{2\lambda} . \tag{6.2.3}$$

For flows parallel to the interface, the momentum flux across the HSI arises from two contributions. One part originates from the static pressure and is identical to the static situation in the previous test case. The other part represents the momentum flux due to the viscous forces which are caused by the velocity gradient in y direction. This momentum transfer also entails an energy transfer. The fluxes are given by

$$\boldsymbol{F}_{ps}^{n} = -\boldsymbol{\Pi} \cdot \boldsymbol{n}_n \tag{6.2.4}$$

$$\boldsymbol{F}_{es}^{n} = -\boldsymbol{\Pi} \cdot \boldsymbol{u} \cdot \boldsymbol{n}_n \tag{6.2.5}$$

where the stress tensor $\boldsymbol{\Pi}$ is

$$\boldsymbol{\Pi} = p\mathrm{I} - \lambda(\nabla \cdot \boldsymbol{u})\mathbf{I} - \mu\left[(\nabla \boldsymbol{u}) + (\nabla \boldsymbol{u})^T\right] . \tag{6.2.6}$$

For the described test case, one can calculate the concrete steady momentum and energy fluxes over the upper face into the molecular domain:

$$\boldsymbol{F}_{ps} = \begin{pmatrix} \mu\frac{u_e}{D} \\ -p \\ 0 \end{pmatrix} \quad \text{and} \quad \boldsymbol{F}_{es} = \begin{pmatrix} \mu\frac{u_e^2}{D} \\ 0 \\ 0 \end{pmatrix} . \tag{6.2.7}$$

Due to the periodic boundary conditions in y and z direction, it is not of importance whether the HSI is divided several cells or a single cell is used. It can be assumed that the momentum and energy transfer through the HSI is validated if

- the simulation converges to a steady solution,
- the density is equal along the y direction,
- the velocity u_x at y_e equals u_e,
- the velocity profile of u_x in y direction is linear and follows Equation (6.2.1),
- the temperature at y_e equals T_e and
- the temperature profile in y direction follows the theoretical prediction of Equation (6.2.2).

6.3 Steady free flow test cases

The most demanding situation for the HSI is flow across the interface. Depending on the direction of the flow, this is either flow into or out of the molecular domain. A steady free flow test case that contains both inflow and outflow is schematically shown in Figure 6.3. The simulation domain comprises two continuum domains which enclose the molecular domain between them. Both continuum domains are effectively one-dimensional and in a steady state, the fluxes at the MD boundary are constant and can be precalculated. The MD domain is however three-dimensional with periodic boundary conditions in the y and z dimension. The simulated material may be gaseous, liquid or solid with a uniform density ρ and temperature T. The flow is in positive x direction, i.e. from the left to the right,

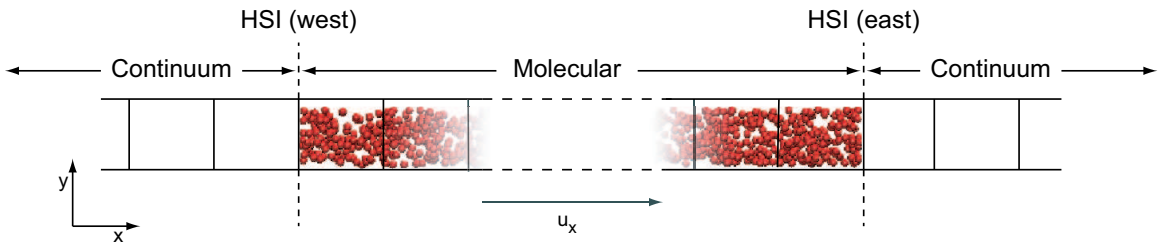


Figure 6.3: Schematic of the free flow test case, where a molecular domain is enclosed by two continuum domains.

with the speed u_x, so that the western HSI is subject to an inflow and the eastern HSI to an outflow situation.

The HSI is expected to be validated for in and out flow, if

- the solution is steady, i.e. what flows in on the western side flows out on the eastern side,
- density, velocity and temperature of the fluid across the molecular domain are constant and match the expected values.

On the first glance this may look trivial, but, as will be seen later, there are substantial issues that need to be resolved.

In a free flow, the fluxes are the same at every position of the flow field, hence also at the western and eastern interface. Like for the two previous test cases the static pressure means a momentum transfer. However, through the velocity, u_x, which is now perpendicular to the HSI, this momentum transfer entails an energy flux as well, even if the flow velocity is constant. The general formulation of $\boldsymbol{F}_{ps}$ and $\boldsymbol{F}_{es}$ has already been given in the Equations (6.2.4) and (6.2.5).

Because of the in- and outflow of mass this test case involves convective mass, momentum and energy fluxes:

$$F_{mc} = -\rho \boldsymbol{u} \cdot \boldsymbol{n} \tag{6.3.1}$$
$$\boldsymbol{F}_{pc} = -\rho (\boldsymbol{u} \otimes \boldsymbol{u}) \cdot \boldsymbol{n} \tag{6.3.2}$$
$$F_{ec} = -\rho e \boldsymbol{u} \cdot \boldsymbol{n} \tag{6.3.3}$$

For the steady case, the constant fluxes can be given:

$$F_{mc} = n_s \rho u_x \quad , \quad \boldsymbol{F}_{pc} = n_s \begin{pmatrix} \rho u_x^2 \\ 0 \\ 0 \end{pmatrix} \quad \text{and} \quad F_{ec} = n_s e u_x \, . \tag{6.3.4}$$

The factor n_s stands for the direction and originates from the normal surface vector of the face. For the western side $n_s = 1$, whereas for the eastern side $n_s = -1$. The energy density e is the total energy, i.e. potential, internal kinetic and external kinetic energy, per volume: $e = e_p + e_{i,k} + \frac{1}{2}\rho \boldsymbol{u}^2$.

6.4 Simulation procedure

The simulation procedures are very similar for all test cases and C → M flux coupling schemes. Thus, it is convenient to describe the general procedure for cases here. In some cases, deviations from the general procedure is necessary and additional steps are given as appropriate. On the top level, the procedure is separated into three stages: (1) Calculation of the state variables, (2) Preparation and (3) Investigation.

All simulations have been performed with an adapted and extended version of LAMMPS (see Section 4.1). Atomic visualisations have been produced by using the freely available software *VMD* [95]. For the analysis of results, the mathematical software *MATLAB* [119] and the visualization and postprocessing software *Tecplot* [98] have been used.

6.4.1 Stage 1: Calculation of the equation of state

The purpose of the first stage is to calculate the state variables of the LJ-fluid for the respective states which are used in stages (2) and (3). The required state variables are the potential and kinetic energy densities of the fluid at rest (e_p, e_k) and the static pressure, P_s, for the used density, ρ, and temperature, T, combinations. The LJ-potential has been used extensively in the past and a wealth of numerical simulation data is available on its phase diagram [28, 120]. However, the usage of different cut-off distance, time step or time integration algorithm may result in deviations of the state variables. To avoid possible implications, the required state variables have been calculated by using the ACSM procedure. The ACSM procedure is explained in Section 3.4, together with the relevant results for this part.

6.4.2 Stage 2: Preparation

The aim of the preparation stage is to establish the system to the point where the actual investigation can start. The procedure is similar for all test cases and the general flow chart is plotted in Figure 6.4. The individual points have been marked with numbers and are explained in the subsequent list:

1. Definition of the molecular domain including dimensions; simulation box: subregions, i.e. walls and fluid areas; interatomic potentials with parameters; force calculation routines; optimisations; the time integration scheme and all other necessary simulation parameters.
2. Definition of the boundary conditions as given in the descriptions of the individual test cases:
   (a) Assign periodic boundary conditions to the respective dimensions.
   (b) If the case involves atomic wall boundary conditions, the wall atoms are placed by using the appropriate lattice structure and density. In all cases, the walls are made of LJ-atoms whose solid crystal structure is fcc.
   (c) Place the HSI on the appropriate boundaries.

3. The initialisation of the atoms:

   (a) If the material is actually solid, the lattice filling scheme is applied. In all cases the FCC crystal structure is used in connection with the lattice parameter corresponding to the density.

   (b) For liquid or gaseous material state, the random filling scheme is used to generate the appropriate number of atoms in the molecular domain.

   (c) In case that some of the created atoms are overlapping, i.e. the generated configuration has an excessive high energy gradient, an energy minimisation run is performed to relax the systems configuration.

4. The velocities are sampled from a uniform distribution to match the selected temperature (q.v. Section 2.2.8) (The velocity distribution approaches the natural Maxwell-Boltzmann distribution normally rapidly within the first few thousand time steps).

5. Equilibration: a MD run with temperature scaling is performed to lead the system into a stable thermal equilibrium.

6. The total momentum is set to zero to enable an exact measurement of inserted and extracted momentum into and from the system through the MD flux boundary conditions and to prevent a constant drift of the entire system through any initial overall movement.

7. The prepared system is saved to a restart file which can be reloaded for the actual investigation.

As result, the preparation stage delivers a restart file of the molecular system in thermal equilibrium which can be reloaded and serves as starting point for the actual investigation stage.

6.4.3 Stage 3: Investigation

In the investigation stage, the prepared system is subjected to different conditions in order to test the MD flux boundary conditions. For each test case and each set of parameters the procedure, as shown in Figure 6.5, is conducted. A detailed explanation of these steps follows:

1. The restart file, which has been generated in the preparation stage, is loaded to restore the complete molecular system in thermal equilibrium state.

2. If the test case involves an overall initial velocity, this velocity is imposed by adding the velocity vector to each atom. For free flow test cases, the same velocity vector is added to each atom. In case of the Couette flow, the additional velocity per atom depends on the distance of the atom to the wall: $\delta u = u_e(r_{iy} - y_w)/D$, where $r_{iy} - y_w$ is the distance of atom i to the wall (For $Kn < 10^{-3}$, which is satisfied for all selected fluid states).

3. The simulation is run with the active MD flux boundary conditions and calculations of macroscopic quantities and/or analysed variables are carried out. By standard, the

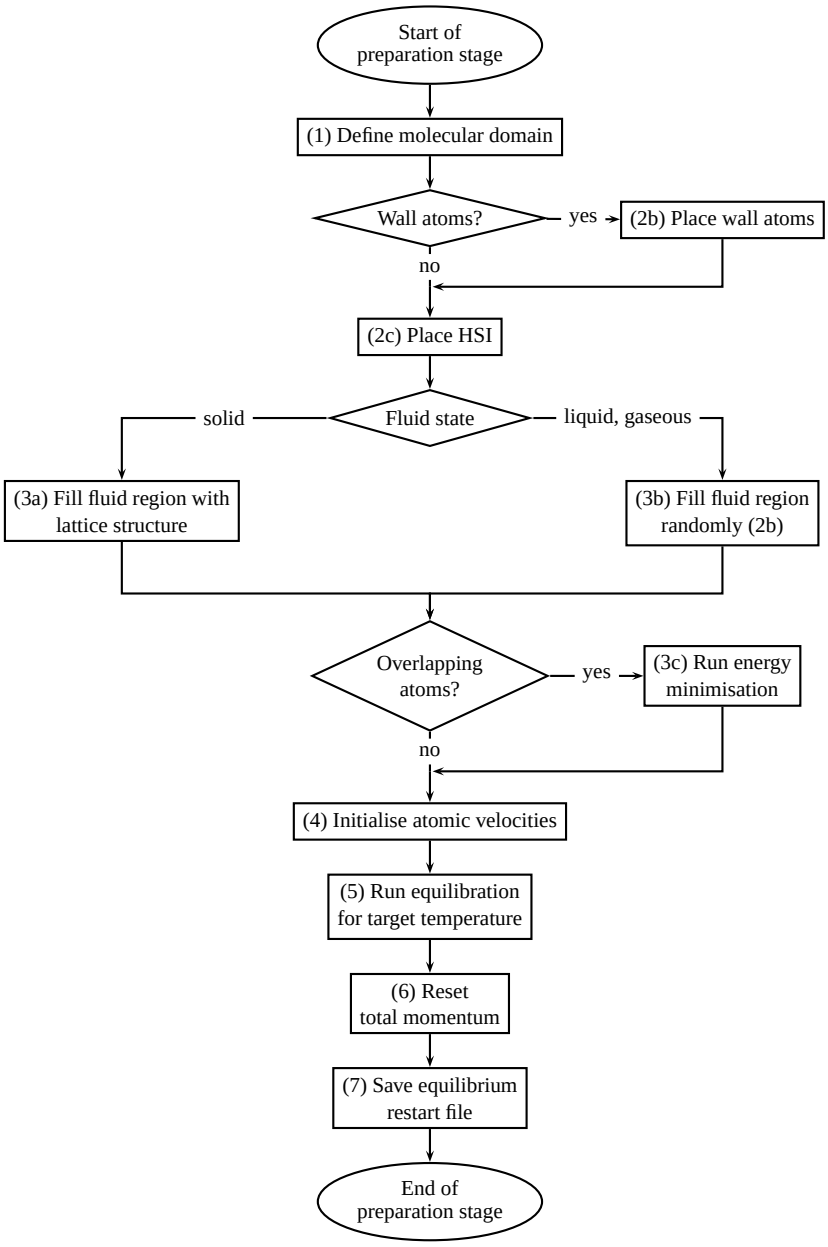


Figure 6.4: Flow chart of the preparation procedure to obtain the starting point for investigation of the MD flux boundary conditions schemes.

calculations include the overall density, momentum, velocity, temperature, potential, kinetic and total energies of the entire system. These quantities are also calculated for a grid superimposed on the molecular domain to obtain the local states and gradients across the domain.

4. The recorded measurements are analysed to determine whether the MD flux boundary conditions have worked as expected.

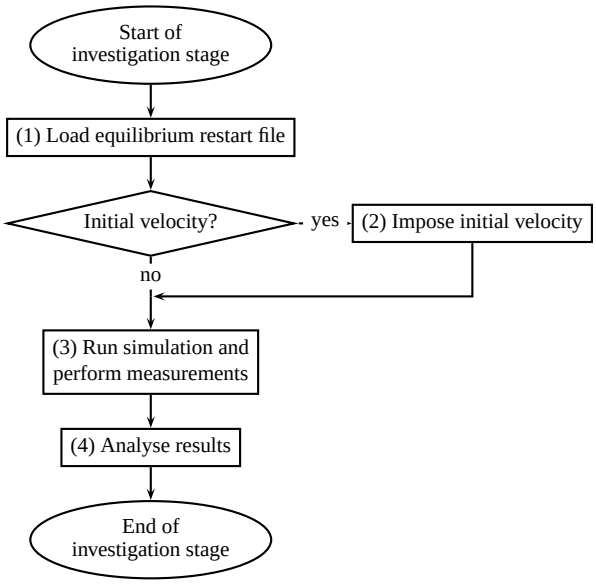


Figure 6.5: Flow chart of the investigation procedure test the MD flux boundary conditions schemes.

6.5 Fluid model, units and simulation parameters

A simple LJ-material has been used for all the fundamental test cases. Even if the LJ-model is only accurate for noble gases, its simplicity makes it an ideal model for basic test cases. Also, tabulated transport quantities, such as specific heats, speed of sound and viscosity, are available for the LJ-model for many areas of the phase space [131]. Thus, these quantities did not have to be calculated or measured for the usage in this work.

All simulations have been performed in reduced LJ-units. The non-dimensionalisation of relevant variables are listed in the second column of table 6.1. The third column contains the respective quantities for Argon whose LJ-parameters are $\sigma = 0.341\,\text{nm}$ and $\epsilon = 1.654 \times 10^{-21}\,\text{J}$.

Depending on density and temperature the LJ-material is gaseous, liquid, solid or supercrititical state.

6.5.1 Shifted potential

It has been found that it is important to use shifted interatomic potential functions (q.v. Section 2.2.6) to avoid an energy jump at the cutoff distance. If non-shifted potential were used, it was extremely difficult in any test case to check through the change in total energy of the system, in order to know whether the correct amount of energy was introduced or removed from the system and to trace energy ‘leaks’. There is, however, no convincing reason why the investigated methods and schemes should not work for unshifted potentials in the way

Physical quantity	Factor	Unit for Argon
mass	m_a	39.948 amu
length	σ	0.341 nm
energy	ϵ	1.654×10^{-21} J
time	$\sigma\sqrt{m_a/\epsilon}$	2.16 ps
velocity	$\sqrt{\epsilon/m_a}$	158 m/s
angular velocity	$\sqrt{\epsilon/(m_a\sigma^2)}$	0.463 THz
linear momentum	$\sqrt{m_a\epsilon}$	6.31×10^3 $\frac{\text{amu m}}{\text{s}}$
angular momentum	$\sigma\sqrt{m_a\epsilon}$	2.15×10^{-6} $\frac{\text{amu m}^2}{\text{s}}$
temperature	ϵ/k_B	119.8 K

Table 6.1: Non-dimensionalisation based on the atomic mass and the Lennard-Jones potential parameters. The symbol amu denotes the atomic mass unit.

they do for the shifted ones.

6.5.2 Transport properties

For the calculation of gradients and to estimate the fluctuation strength, the knowledge of a number of transport coefficients is necessary: specific heat capacity at constant temperature, c_v; specific heat capacity at constant pressure, c_p; speed of sound, a; internal energy density, e_{int}; and adiabatic coefficient, γ. Because these properties vary with density and temperature, Table 6.2 gives the state variables and transport coefficients for all used states of the LJ-fluid. Please note that the employed cut-off distance is $r_c = 2.5$. The values of specific heat capacity, speed of sound, and viscosity have been taken from Meier [131]. The relative error in these values was reported to be 5 % for the viscosity μ, less than 0.5 % for the isochoric heat capacity c_v and less than 0.7 % for the isothermal pressure coefficient γ_V. It should be noted that slight discrepancies arise from the fact that in the referenced work a cut-off distance (r_c) of twice the size was used. The smaller value of r_c, used in this work, gives satisfactory results and the minimal improvement in accuracy, which would be gained by using a larger r_c, does not justify the entailed computational effort.

Values for the thermal conductivity, λ, have been published by Nasrabad et al. [137]. However, values for temperature of $T = 2$ were only available for densities $\rho \geq 0.2$ so that the value for $\rho = 0.1$ had to be obtained through extrapolation via fitting an exponential function. The relative error of the values of λ that are given in Nasrabad et al. [137] are about 0.4 for gaseous states and 0.34 for liquid states. This is a surprisingly high values, but other measurements for the required states were not available.

For the adiabatic coefficient, γ, no tabulated values could be found. However, γ can be obtained from the relationship:

$$\frac{a^2}{c_v\gamma(\gamma-1)} = T \ . \tag{6.5.1}$$

where the values of $c_{P(P)}$ have been obtained from Heyes [90].

Property	Symbol	dense gas	liquid
Density	ρ	0.1	0.8
Temperature	T	2	1
Isochoric heat capacity	$c_{V(P)}$	1.6	2.38
Speed of sound	a	1.87	5.5
Ratio of specific heats	γ	1.71	1.8
Acoustic number	Ac	1	4.1
Internal energy density	e_{int}	0.245	-2.55
Viscosity	μ	0.24	2.0
Thermal conductivity	λ	≈ 1.1	6.41

Table 6.2: State variables and transport properties of a Lennard-Jones-fluid for the test cases.

CHAPTER 7

MD Flux Boundary Conditions

This chapter describes the investigation of the MD flux boundary conditions. In the first part, basic conventions of fluxes in molecular terms and techniques fundamental for the flux imposition are introduced. These include the methods for imposing convective fluxes, the required algorithms for the insertion and removal of atoms and the definition of the conditions of a breakdown of the flux boundary conditions. The second part of the chapter contains the discussion on different combinations of the investigated schemes for imposing the mass, momentum and energy fluxes onto the MD domain. Each combination is evaluated for its applicability and validated for the three basic flow situations at different states of matter.

On the molecular side, the C → M flux coupling is realised through imposing flux boundary conditions onto the molecular system. The fluxes are either provided by the continuum solver in the coupled cases or, alternatively, may be calculated based on analytical or empirical models. In both cases, flux boundary conditions imitate a much larger system beyond the boundary of the molecular system. In the next sections different designs of the flux boundary conditions are discussed and validated, based on the test cases which have been described in Section 6. All formulations of this section are related to a single virtual boundary cell of the molecular dynamics region and apply in the same manner to multiple cells, even if not stated explicitly in each case.

In Figure 7.1, a one-dimensional example of the situation on a MD flux boundary is illustrated. The normal vector of the boundary surface, $\boldsymbol{n}_S$, points into the MD domain. The blue shaded area is the outer virtual cell, which is called *reservoir*. The atoms inside the reservoir are outside of the actual MD domain, i.e. beyond the boundary. Henceforth, they are referred to as boundary atoms and it is these atoms, onto which the fluxes are imposed. The velocity of the boundary atoms is given by

$$\boldsymbol{v}_b = \frac{1}{m_b} \sum_{i=1}^{N_b} m_i \boldsymbol{v}_i \quad \text{with} \quad m_b = \sum_{i=1}^{N_b} m_i \,, \tag{7.0.1}$$

where m_b is the mass of the boundary atoms. Some of the flux imposition methods that are discussed in the following part require the access to the boundary atoms in order of their

distance to the boundary surface. To this end, each outer virtual cell has a *boundary atom list* (BAL) into which the boundary atoms are sorted in descending order of their distance to the boundary surface. For a general surface orientation, the distance d_{bi} of a boundary atom i is calculated by $d_{bi} = -\boldsymbol{n}_S \cdot (\boldsymbol{r}_i - \boldsymbol{r}_b)$, where $\boldsymbol{r}_b$ is an arbitrary point on the surface of the boundary. The boundary atom list is generated each time step before the fluxes are applied. The sorting is done while inserting the atoms into the list by comparison of the distances.

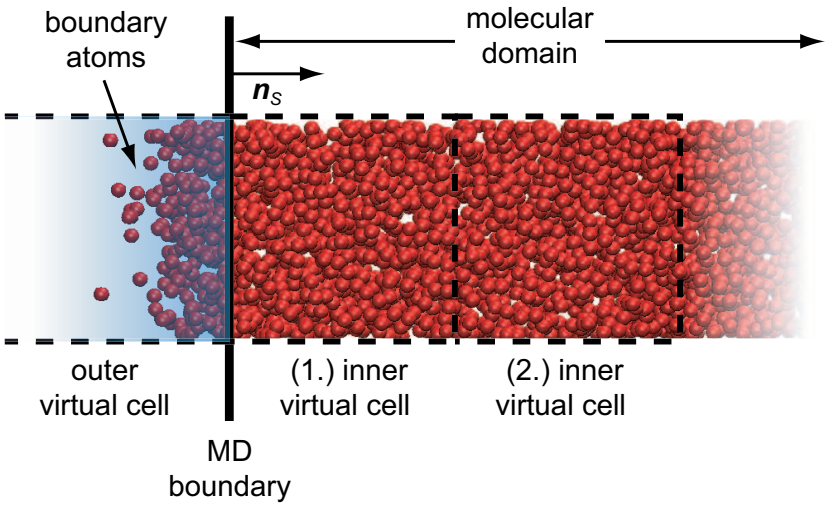


Figure 7.1: Illustration of the MD flux boundary conditions for an one-dimensional example.

7.1 Fluxes

As described in Section 5.3.2, the fluxes across the boundary comprise of mass, momentum and energy fluxes, where the transport may occur through convection, stress or conduction. The convective fluxes $\boldsymbol{F}_c$ are the convective mass flux F_{mc}, the convective momentum flux $\boldsymbol{F}_{pc}$ and the convective energy flux F_{ec}. From Table 5.4, the fluxes across the boundary are defined in molecular formulation

$$F_{mc} = \frac{1}{A}\left\langle \sum_i m_i s_i \right\rangle_t \tag{7.1.1}$$

$$\boldsymbol{F}_{pc} = \frac{1}{A}\left\langle \sum_i m_i \boldsymbol{v}_i s_i \right\rangle_t \tag{7.1.2}$$

$$F_{ec} = \frac{1}{A}\left\langle \sum_i E_i s_i \right\rangle_t \tag{7.1.3}$$

with

$$s_i = \begin{cases} 1 & \text{for an inflown atom,} \\ -1 & \text{for an outflown atom} \end{cases} \tag{7.1.4}$$

where each of the sums in each equation is over the atoms i, which have been inserted into, or removed from, the boundary cell. The convective energy flux F_{ec} consists of a potential part, F_{ecp}, and a kinetic part, F_{eck}. The ratio between these is determined by the relation of kinetic and potential energy at the respective state of matter at the boundary:

$$\frac{F_{ecp}}{F_{eck}} = \frac{e_p(\rho, T)}{e_{i,k}(\rho, T) + e_e} = \frac{e_p(\rho, T)}{e_{i,k}(\rho, T) + \frac{1}{2}\rho \boldsymbol{u}^2} \ . \tag{7.1.5}$$

The Fluxes caused by stress are the momentum flux, F_{ps}, and the energy flux, F_{es}, which are related to the N_b boundary atoms as

$$\boldsymbol{F}_{ps} = \langle \sum_{i=1}^{N_b} \boldsymbol{f}_i^{ext} \rangle_t \ , \tag{7.1.6}$$

$$F_{es} = \langle \sum_{i=1}^{N_b} \boldsymbol{f}_i^{ext} \cdot \boldsymbol{v}_i \rangle_t \ . \tag{7.1.7}$$

The energy flux, F_{es}, is of kinetic nature and corresponds to the energy gain or loss due to the acceleration of the boundary atoms due to $\boldsymbol{F}_{ps}$. It is bound to the direction of the momentum flux, $\boldsymbol{F}_{ps}$, and the velocity, $\boldsymbol{v}_b$, of the boundary atoms. Thus, F_{es} is the sum of three energy fluxes, one from each dimension:

$$F_{es} = |\boldsymbol{F}_{es}| \quad \text{where} \quad \boldsymbol{F}_{es} = \begin{pmatrix} F_{ps_x} v_{b_x} \\ F_{ps_y} v_{by} \\ F_{ps_z} v_{bz} \end{pmatrix} , \tag{7.1.8}$$

where the components of $F_{es\alpha}$ stand for the energy transfer in the respective dimension α. The distribution of the external force, $\boldsymbol{f}_i^{ext}$, is discussed below.

Finally, energy can also be transported by conduction, where the conductive energy flux is

$$F_{eq} = \langle Q \rangle_t \tag{7.1.9}$$

with $\langle Q \rangle$ being the heat transfer per unit time and area to or from the boundary atoms atoms.

In principle, fluxes can be imposed in two distinctive ways, either *continuously* or *integrated*. Continuous flux imposition is achieved through the application of external forces, which are then automatically incorporated into the numerical integration algorithm. Otherwise, fluxes can be integrated in time and imposed at discrete time steps for which the numerical integrator calculates the solution. The latter applies, in particularly, to convective fluxes, where the translation of continuous density values into discrete atoms requires the integration of the mass flux, since no 'half' atoms can be inserted or removed. How the integrated flux imposition is incorporated into the numerical time stepping algorithm depends on the specific scheme. In general, the integrated fluxes are imposed every τ_f time steps, where τ_f is a positive integer. For a period of time from t_s to $t_s + \delta t_f$, with $\delta t_f = \delta t \tau_f$, an integrated flux across a constant surface with the area A is

$$F_{\alpha\beta}^{int} = A \int_{t_s}^{t_s + \delta t_f} F_{\alpha\beta}(t) \, \mathrm{d}t = A \, \delta t_f \sum_{\tau=\tau_s}^{\tau_s + \tau_f} F_{\alpha\beta}(\tau) \ , \tag{7.1.10}$$

where δt is the time step of the numerical integrator and the subscripts $\alpha = \{m, p, e\}$ and $\beta = \{c, s\}$ denote the flux quantity and the way of transport respectively. The integrated fluxes from Equation (7.1.1) – (7.1.9) are then

$$F_{mc}^{int} = A\,\delta t_f \sum_{\tau=\tau_s}^{\tau_s+\tau_f} F_{mc}(\tau) = \sum_i m_i s_i a;, \tag{7.1.11}$$

$$\boldsymbol{F}_{pc}^{int} = A\,\delta t_f \sum_{\tau=\tau_s}^{\tau_s+\tau_f} \boldsymbol{F}_{pc}(\tau) = \sum_i m_i v_i s_i\,, \tag{7.1.12}$$

$$F_{ec}^{int} = A\,\delta t_f \sum_{\tau=\tau_s}^{\tau_s+\tau_f} F_{ec}(\tau) = \sum_i E_i s_i\,, \tag{7.1.13}$$

$$\boldsymbol{F}_{ps}^{int} = A\,\delta t_f \sum_{\tau=\tau_s}^{\tau_s+\tau_f} \boldsymbol{F}_{ps}(\tau) = A\,\delta t_f \sum_{\tau=\tau_s}^{\tau_s+\tau_f} \sum_{i=1}^{N_b} \boldsymbol{f}_i^{ext}\,, \tag{7.1.14}$$

$$\boldsymbol{F}_{es}^{int} = A\,\delta t_f \sum_{\tau=\tau_s}^{\tau_s+\tau_f} \boldsymbol{F}_{es}(\tau) = A\,\delta t_f \sum_{\tau=\tau_s}^{\tau_s+\tau_f} \sum_{i=1}^{N_b} \boldsymbol{f}_i^{ext} \boldsymbol{v}_i\,, \tag{7.1.15}$$

$$F_{eq}^{int} = A\,\delta t_f \sum_{\tau=\tau_s}^{\tau_s+\tau_f} F_{eq}(\tau) = \sum_{\tau=\tau_s}^{\tau_s+\tau_f} Q(\tau)\,. \tag{7.1.16}$$

The integrated fluxes constitute the prescribed mass, momentum and energy that need to be imposed onto the reservoir region. They do not, necessarily, match the actual implemented quantities, which will be referred to as $F_{\alpha\beta}^{imp}$, with α and β defined as before. The aim of the flux imposition is to match the integrated and actually implemented fluxes ($F_{\alpha\beta}^{int} = F_{\alpha\beta}^{imp}$).

7.2 Overview of flux imposition

There are a number of possible ways in which each of the individual fluxes, i.e. fluxes due to convection, fluxes by stress and fluxes by conduction, can be imposed onto the reservoir region. Some of these have been designed and used in previous works on coupling [40, 70, 111]. Different schemes for different fluxes can be combined in various ways, however, it is not necessary and would be very elaborate to investigate all possible combinations. Thus, only some schemes and combination have been selected and applied to the test cases of Section 6; these are listed in Table 7.1.

The two fundamentally different options for imposing the convective fluxes are the *direct* (DC) and the *indirect* (ICS,ICC) mass flux imposition. More variety exists for momentum fluxes caused by stress, which can be imposed through: a combination of a *reflective plane* (MTRP) and *force* (MTF) scheme; force or *direct velocity change* (MTDVC) scheme; or by the *reverse velocity* (MTRV) scheme. The imposition of energy by stress depends primarily on the chosen scheme for the momentum flux by stress. When imposing the momentum flux by a reflective plane, force or direct velocity change scheme for momentum fluxes, the energy flux is imposed *indirectly*. However, for some of the schemes

Combination	fluxes		
	convective	*momentum by stress*	*energy by stress*
DC-MTRP	direct	static part: reflective plane viscous part: force	indirect
DC-MTF		force or direct velocity change	
DC-MTRV		reverse velocity	direct
ICS-MTRV	indirect	reverse velocity (separated)	
ICC-MTRV		reverse velocity (coupled)	

Table 7.1: Overview of flux impositions schemes.

it may deviate from the calculated value and requires correction if an exact energy flux is desired. Since the reverse velocity scheme is energy neutral, the energy needs to be imposed *directly*. The conductive fluxes are not considered here, but they are easily imposed using the direct energy transfer algorithms that are employed to impose the energy by stress.

The discussion is continued with an explanation of how the flux imposition schemes are integrated into the time integration method of the MD-code. This is followed by a description of how atoms are inserted and removed. By then the fundamentals are introduced and the selected combinations of flux impositions schemes can be discussed, each in a separate section. The functionality of the individual schemes is explained in the section of their first usage.

7.3 Flux imposition within the time integration algorithm

To implement the MD flux boundary conditions within the MD code, the flux imposition schemes need to be integrated into the time integration method of the MD code. The time integration method may use several intermediate steps to advance positions and velocities in time. It is important that the fluxes are imposed only at steps which enable exact insertion and removal of the flux quantities (mass, momentum and energy). Which step of the algorithm is suitable, depends on the chosen flux imposition scheme and time integration algorithm itself. For most time integration methods, positions, velocities and forces are not valid for exactly the same points in time at each intermediate steps. For example, in the Leap-Frog time stepping method, velocities and positions are never evaluated for the same point in time, but always differ by half a time step. Initial tests have shown that only if the positions and velocities are available for the same point in time can an exact imposition of mass, momentum and energy be successful. Attempts to introduce any of these quantities at intermediate steps where positions and velocities were not advanced to exactly the same time resulted in significant differences between the quantities intended to insert and remove to the ones actually inserted and removed.

The formulated condition makes it clear that there would be no easy efficient way to

apply flux boundary condition in conjunction with the Leap-Frog method. The situation is better for velocity Verlet method, which is used in LAMMPS. In the velocity Verlet method, positions and velocities are available at the start and the end of each time step. Therefore, convective fluxes, fluxes due to stresses and conduction can be applied before the first step of the velocity Verlet algorithm. This includes the flux imposition schemes MTRP, MTDVC and MTRV for the imposition of momentum transfer due to stress.

If the momentum flux is imposed through external forces, it is best to add this external force on the boundary atoms after all interatomic forces have been calculated. The momentum change is then applied automatically within the time stepping method.

In Figure 7.2, the flow chart of a single time step of the adapted velocity Verlet algorithm is shown. The algorithm includes the different flux imposition schemes, which have been investigated. Depending on the case, the respective schemes which are being used are activated while others are bypassed. The individual steps are explained in the following subsections.

At the beginning of each time step, the fluxes are integrated in space and time (*Integrate fluxes*). To this end, each boundary cell stores a variable for each integrated flux. The integrated flux variables are named according to Equations (7.1.11) – (7.1.16) with $F^{int}_{\alpha\beta}$, where α is the flux quantity and β the way of transport. They basically contain the absolute amount of the according quantity that needs to be inserted ($F^{int}_{\alpha\beta} > 0$) into or removed ($F^{int}_{\alpha\beta} < 0$) from the respective reservoir region. One could consider the integrated flux variables as a buffer of flux quantities which needs be imposed. The buffer is filled or evacuated through integrating the fluxes and evacuated through adding or removing the flux quantities into or from the reservoir region. In Algorithm 4, the flux integration is provided. The surface A_S is the surface of the face boundary cell over which the fluxes $F_{\alpha\beta}(C,\tau)$ flow during the time step τ.

Algorithm 4: Flux integration.

foreach *boundary cell C* **do**
  foreach *flux* $F_{\alpha\beta}$ **do**
    $F^{int}_{\alpha\beta_C} \leftarrow F^{int}_{\alpha\beta_C} + A_S \delta t F_{\alpha\beta}(C,\tau)$

Some flux imposition schemes require the treatment of atoms which have crossed a limiting surface (*Handle crossed atoms*). The limiting surface may coincide with the outer surface of the boundary cell or may be another parallel surface in a certain distance. Different treatments of the atoms that have crossed the limiting surface are required for the different flux imposition schemes and have been investigated for their operability. In the case of the MTRP scheme, the crossed atoms are basically reflected. For the MTF or MTDVC schemes, several different treatments were tested. These include the removal and reinsertation of the respective atoms further inside the boundary cell or the restriction of the movement of the atoms across the limiting surface. The details are explained in the subsequent sections. In any case, these algorithms involve the manipulation of the velocity

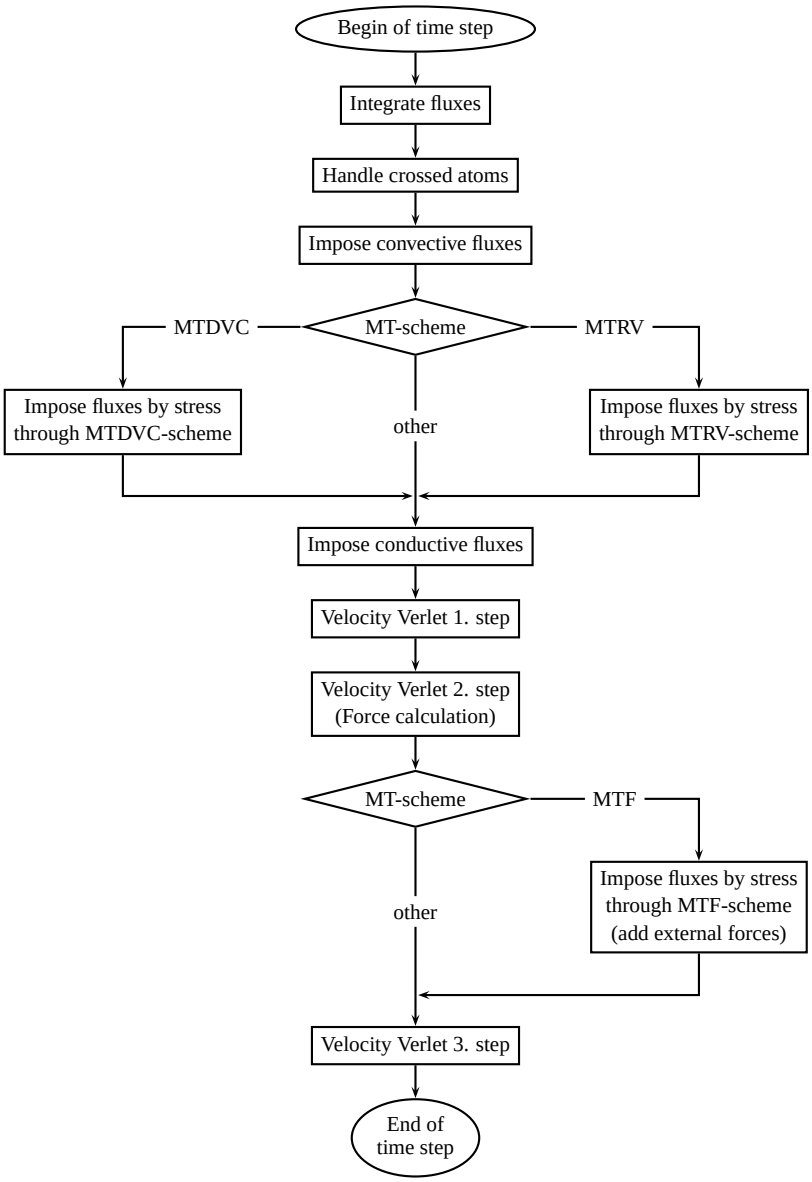


Figure 7.2: Flow chart of one time step of the Verlet algorithm with incorporated flux imposition scheme.

vectors or positions. For reasons given before, this needs to be done before the first step of the Verlet algorithm. The most suitable place is before any other other flux is imposed.

The step *impose convective fluxes* handles the insertion and removal of atoms and the entailed momentum and energy transfer. Followed by the convective fluxes imposition, the fluxes due to stresses are imposed depending on whether the MTDVC or the MTRV is used. Subsequently, the conductive fluxes are transferred. Then, the first and second step of the normal velocity Verlet algorithm are performed. The force calculation in the second step

uses a number of optimisation schemes to accelerate the computational costly pair evaluation. If a neighbour list method is used, it must be rebuild each time after atoms have been inserted or removed during the imposition of the convective fluxes. When using the MTF scheme, the external forces are added to the already calculated forces on the atoms in the reservoir region. The time step is completed by the third velocity Verlet step.

The implemented method of the described algorithm, includes the calculation, collection, preparation and output of a large number of statistical and debug information. This enabled the error free implementation of the flux imposition scheme and the analysis of their performance. Because it has no direct contribution to the methodology, the part has been omitted from description above.

7.4 Imposition of convective fluxes

The two methods of convective flux imposition considered here are the DC (direct convective flux imposition) and IC (indirect convective flux imposition).

Direct flux imposition

The idea of DC is to insert and remove the exact amount of mass, momentum and energy prescribed by the convective fluxes F_{mc}, $\boldsymbol{F}_{pc}$ and F_{ec}. Because no fractional, but only entire, atoms can be inserted or removed, the imposition of convective fluxes is based on the integrated fluxes F_{mc}^{int}, $\boldsymbol{F}_{pc}^{int}$ and F_{ec}^{int}. A new atom is inserted if $F_{mc}^{int} > m$, where m is the molecular mass of the inserted atom. Respectively, an atom is removed if $F_{mc}^{int} < m$. The remaining fractional fluxes, which could not be applied, are stored within the integrated fluxes and applied whenever $|F_{mc}^{int}|$ becomes greater than one. The principle operation for DC is given in Algorithm 5. The two subsequent sections contain detailed descriptions on finding a new position, $\boldsymbol{r}_{new}$, and generating a velocity vector, $\boldsymbol{v}_{new}$.

Algorithm 5: DC (Direct convective flux imposition)

if $F_{mc}^{int} > m$ **then**
- find new position $\boldsymbol{r}_{new}$;
- generate new velocity vector $\boldsymbol{v}_{new}$;
- insert atom i at position $\boldsymbol{r}_{new}$ with velocity $\boldsymbol{v}_{new}$;
- $F_{mc}^{int} \leftarrow F_{mc}^{int} - m_i$;
- $\boldsymbol{F}_{pc}^{int} \leftarrow \boldsymbol{F}_{pc}^{int} - m_i \boldsymbol{v}_i$;
- $F_{ec}^{int} \leftarrow F_{ec}^{int} - E_{p_i} - (1/2) m_i \boldsymbol{v}_i^2$;

if $F_{mc}^{int} < m$ **then**
- determine atom i to be removed;
- $F_{mc}^{int} \leftarrow F_{mc}^{int} + m_i$;
- $\boldsymbol{F}_{pc}^{int} \leftarrow \boldsymbol{F}_{pc}^{int} + m_i \boldsymbol{v}_i$;
- $F_{ec}^{int} \leftarrow F_{ec}^{int} + E_{p_i} + (1/2) m_i \boldsymbol{v}_i^2$;
- remove atom i;

Indirect flux imposition

The principle of indirect convective flux imposition is to impose the convective fluxes not directly, but to obtain them as a result of material moving in or out of the reservoir volume. This is achieved by keeping the number of atoms N_b in the reservoir within chosen limits ($N_{b,min} \leq N_b \leq N_{b,max}$) through removing or inserting atoms whenever the number of atoms in the reservoir is too high or too low. The properties of new atoms correspond to the convective fluxes and the properties of the removed atoms must correspond to that of an average atom in the reservoir region or virtual boundary cell.

IC is not a pure unidirectional flux coupling scheme, because the convective fluxes are not implemented strictly; rather they are a result of the simulation. The actual realised convective fluxes are determined by the properties of the atoms that have entered or left the boundary. Those concrete values are summed up automatically in the integrated convective fluxes, F_{mc}^{int}, F_{pc}^{int} and F_{ec}^{int}, which can be transferred back to the continuum solver during each synchronisation. There, they can easily be incorporated by the continuum solver when updating the conserved variables of the cells from the fluxes over the cell faces.

From a number of different IC schemes that have been investigated for their applicability, two gave stable results as MD-boundary conditions:

(1) Separated momentum fluxes (ICS): Here, the convective momentum flux and the momentum flux by stress are still imposed separately. This means that the implemented momentum flux depends on the number of atoms that enter the system.

(2) Combined momentum fluxes (ICC): In this IC scheme, the momentum fluxes by convection and stress are combined by applying the coupling of integrated fluxes by convection and by stress, which is explained in the following section. The advantage of the scheme is that the total prescribed momentum flux ($\boldsymbol{F}_{pc} + \boldsymbol{F}_{ps}$) is always imposed, independently on the number of in- or outflowing atoms.

In Algorithm 6 the indirect convective flux imposition algorithm is shown. The momentum flux coupling in the last part is discussed in the next section.

7.4.1 Coupling of integrated fluxes by convection and by stress (FCCS)

As already mentioned, some flux imposition designs require the coupling of the integrated convective fluxes with the integrated fluxes by stress, which will be referred to as FCCS. In this work, FCCS is the transfer of the integrated convective fluxes to the integrated fluxes by stress. The transfer can apply to momentum and/or energy fluxes. In terms of when the transfer takes place, three types are applicable: at every time step, after inserting a new atom or after removing an atom. Additionally, it is necessary to distinguish between in- and outflow conditions ($F_{mc} > 0$ or $F_{mc} < 0$). The different possibilities for flux type, transfer time and flow condition result in many conceivable combinations. Here, only the four most useful have been considered:

IMP: The integrated momentum fluxes are transferred at every time step for inflowing conditions ($F_{mc} > 0$). The transfer is accomplished by the following rule that is

Algorithm 6: IC (Indirect convective flux imposition)

while $N_b < N_{b,min}$ **do**
  find new position $\boldsymbol{r}_{new}$;
  generate new velocity vector $\boldsymbol{v}_{new}$;
  insert atom i at position $\boldsymbol{r}_{new}$ with velocity $\boldsymbol{v}_{new}$;
  $F_{mc}^{int} \leftarrow F_{mc}^{int} - m_i$;
  $\boldsymbol{F}_{pc}^{int} \leftarrow \boldsymbol{F}_{pc}^{int} - m_i \boldsymbol{v}_i$;
  $F_{ec}^{int} \leftarrow F_{ec}^{int} - E_{p_i} - (1/2) m_i \boldsymbol{v}_i^2$;
while $N_b > Nb, max$ **do**
  determine atom i to be removed;
  $F_{mc}^{int} \leftarrow F_{mc}^{int} + m_i$;
  $\boldsymbol{F}_{pc}^{int} \leftarrow \boldsymbol{F}_{pc}^{int} + m_i \boldsymbol{v}_i$;
  $F_{ec}^{int} \leftarrow F_{ec}^{int} + E_{p_i} + (1/2) m_i \boldsymbol{v}_i^2$;
  remove atom i;
if *Coupling of momentum fluxes for inflow(IMP)* **then**
  if $F_{mc} > 0$ **then**
    $\boldsymbol{F}_{ps}^{int} \leftarrow \boldsymbol{F}_{ps}^{int} + \boldsymbol{F}_{pc}^{int}$;
    $\boldsymbol{F}_{pc}^{int} \leftarrow \boldsymbol{0}$;
if *Coupling of momentum fluxes for outflow(OMP)* **then**
  if $F_{mc} < 0$ **then**
    $\boldsymbol{F}_{ps}^{int} \leftarrow \boldsymbol{F}_{ps}^{int} + \boldsymbol{F}_{pc}^{int}$;
    $\boldsymbol{F}_{pc}^{int} \leftarrow \boldsymbol{0}$;

executed every time step:

$$\boldsymbol{F}_{ps}^{int} \leftarrow \boldsymbol{F}_{ps}^{int} + \boldsymbol{F}_{pc}^{int} , \tag{7.4.1}$$

$$\boldsymbol{F}_{pc}^{int} \leftarrow \boldsymbol{0} \tag{7.4.2}$$

The IMP transfer is implemented as subsequent step after the imposition of the convective fluxes (q.v. Algorithm 6).

OMP: The transfer is identical to the preceding IMP transfer, with the only difference that it applies for outflows ($F_{mc} < 0$). The OMP transfer is also implemented directly after the IMP transfer (q.v. Algorithm 6).

OMAR: The integrated momentum fluxes are transferred every time after an atom has been removed. The purpose of the transfer is to circumvent the difficulties of matching the convective momentum fluxes on the outflow. This issue is explained in Section 7.4.3. The update rule is identical to the IMP and OMP rule:

$$\boldsymbol{F}_{ps}^{int} \leftarrow \boldsymbol{F}_{ps}^{int} + \boldsymbol{F}_{pc}^{int} , \tag{7.4.3}$$

$$\boldsymbol{F}_{pc}^{int} \leftarrow \boldsymbol{0} \tag{7.4.4}$$

The implementation of the rule follows immediately the removal of an atom.

OEAR: The integrated energy fluxes are transferred each time an atom is removed. The purpose is similar to the previous point and enables it to match the convective energy fluxes on the outflow. The update rule is

$$\boldsymbol{F}_{es}^{int} \leftarrow \boldsymbol{F}_{es}^{int} + \begin{pmatrix} F_{ec}^{int}/3 \\ F_{ec}^{int}/3 \\ F_{ec}^{int}/3 \end{pmatrix} , \tag{7.4.5}$$

$$\boldsymbol{F}_{ec}^{int} \leftarrow \boldsymbol{0} \tag{7.4.6}$$

Because the energy fluxes by stress are given for each dimension individually, the convective energy needs to be split into three parts.

It is stated at the appropriate sites in the text which of the FCCS schemes are employed for the individual test cases.

7.4.2 Insertion of atoms

For inflow conditions on the boundary, new atoms enter the molecular system which need to be assigned with a position and a velocity. The position of a new atom determines its potential energy, the velocity its momentum and kinetic energy. These values are prescribed by the convective fluxes.

Assuming that the flow across the boundary, with the surface vector $\boldsymbol{n}_S$, is in a local equilibrium the average properties of the inserted atoms can be calculated. The average velocity of an inserted atom i is then

$$\langle \boldsymbol{v} \rangle = \boldsymbol{v}_{iext} = \frac{\boldsymbol{F}_{pc}}{F_{mc}} = \frac{\rho \boldsymbol{u}(\boldsymbol{u} \cdot \boldsymbol{n}_S)}{\rho(\boldsymbol{u} \cdot \boldsymbol{n}_S)} = \boldsymbol{u} , \tag{7.4.7}$$

where $\boldsymbol{v}_{iext}$ is the external velocity of the atom i that equals the velocity $\boldsymbol{u}$ of the inflowing material. Knowing $\boldsymbol{u}$, the density can be recovered through

$$\rho = \frac{F_{mc}}{(\boldsymbol{u} \cdot \boldsymbol{n}_S)} = \frac{\rho(\boldsymbol{u} \cdot \boldsymbol{n}_S)}{(\boldsymbol{u} \cdot \boldsymbol{n}_S)} , \tag{7.4.8}$$

and the average total energy is calculated by

$$\langle E_{tot} \rangle = \frac{F_{ec}}{(\boldsymbol{u} \cdot \boldsymbol{n}_S)} \frac{m}{\rho} = \frac{e(\boldsymbol{u} \cdot \boldsymbol{n}_S)m}{(\boldsymbol{u} \cdot \boldsymbol{n}_S)} = e\frac{m}{\rho} , \tag{7.4.9}$$

with m being the mass of an inserted atom. The average total energy of an inserted atom comprises of the average external kinetic energy $\langle E_{k,ext} \rangle$ (this is bound to velocity $\boldsymbol{u}$ with which the material flows across the boundary) and the average internal energy $\langle E_{int} \rangle$ of an atom. It is given by

$$\langle E_{int} \rangle = \langle E_{tot} \rangle - \frac{1}{2} m \langle \boldsymbol{v} \rangle . \tag{7.4.10}$$

The internal energy is the sum of a potential and kinetic part: $\langle E_{int} \rangle = \langle E_p \rangle + \langle E_{k,int} \rangle$, where the average potential energy of an inserted atom, $\langle E_p \rangle$, is related to the potential energy

density, e_p, by $\langle E_p \rangle = (m/\rho)e_p$. The potential energy density, e_p, is a state function, which must be measured in advance for any occurring combination of density and internal energy: $e_p = e_p^{mes}(\rho, e_i)$. Alternatively, an analytical equation of state may be used instead: $e_p = e_p^{ana}(\rho, e_i)$. For all cases in this work, the respective values for e_p have been measured by using the AMSM scheme described in Section 3.4.1.

Knowing $\langle E_p \rangle$, the internal kinetic energy is obtained through

$$\langle E_{k,int} \rangle = \langle E_{int} \rangle - \langle E_p \rangle \,, \tag{7.4.11}$$

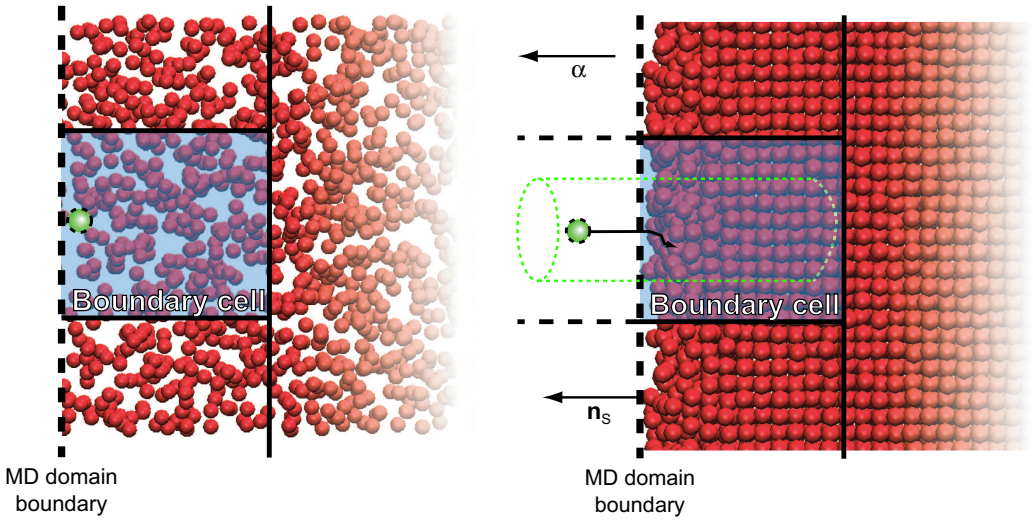
and must be used to generate the internal velocities of the new atoms.

At this point, all variables necessary to describe the average inserted atom that satisfies the convective flux conditions are available. It remains to describe the generation of concrete positions and velocities of the inserted atoms based on the average values. The explanation starts with the determination of the positions and is followed by the equivalent description for the velocities. Instead of the fluxes $F_{\alpha\beta}$, in some cases the corresponding integrated fluxes $F_{\alpha\beta}^{int}$ are used. This is noted where necessary.

Determination of the position

From the description above it is clear that each new atom has to be placed at a position in the molecular system at which its potential energy corresponds to $\langle E_p \rangle$. Also, since the flux imposition always considers a boundary cell, the atom has to placed somewhere inside the corresponding cell or wherever the insertion region of this cell is defined. A typical example is shown in Figure 7.3a. The schematic is based on a case with a reflective plane as flux imposition scheme. One boundary cell has been highlighted through blue shading. Depending on the size of the boundary cell, the region of insertion may comprise the entire volume of the boundary cell or be more restricted to its outer side.

Which scheme is employed for insertion depends, first of all, whether it is a single atom that needs to be inserted or a molecule. Here, only single atoms acting through a pair potential are considered, but a scheme for inserting polar molecular has been published by De Fabritiis et al. [37]. Depending on the density of the medium into which the atom is to be inserted, different schemes may be used. For low densities, such as in gaseous states, the potential energy of an atom is close to zero and the potential energy gradients are very flat. A typical example is the gaseous state of the LJ-fluid ($n = 0.1$, $T = 2$), which is used in many test cases throughout this work. Under these conditions it is efficient to use *random position sampling* with a minimum distance check in conjunction with a density grid. The minimum distance to any other atom ensures that the energy of the inserted atom is not too high. The scheme is the same as for the initialisation procedure and explained in Section 2.2.8. It has been used in all test cases of a gaseous state and convective fluxes across the boundary. For higher densities, such as in liquids or solids, the structure of the energy landscape becomes very complex, even for a simple pair potentials. For typical moderate temperatures, the atoms reside in deep potential energy wells surrounded by steep gradients of the energy surface. Therefore, the potential energy of an average



(a) Insert atoms via density grid scheme or USHER

(b) Insert atoms via modified USHER scheme

Figure 7.3: Schematic display of the insertion of atoms into the boundary cell. In each case, the new atom is coloured green. The respective boundary cell is shaded blue.

atom $\langle E_p \rangle$ is extremely low compared to the average energy level [39]. To insert atoms in a dense medium, one has to find these extremely low energy sites. Inserting atoms at other sites with high energies values and steep gradients would lead to an excess of introduced energy and cause the system to 'explode'. The problem one faces is how to find the energy holes. Simply using the density grid and random sampling is extremely inefficient and would lead to a situation where most of the simulation time is spent in sampling random positions. Hence, this scheme can not be used for inserting atoms into dense fluids. Instead, the USHER algorithm can be used to locate the low energy holes in the potential energy landscape. For dense fluids, this algorithm is much more efficient than the random sampling scheme. Also, the algorithm is easy to implement.

USHER scheme

Developed by Delgado-Buscalioni and Coveney [39], the USHER algorithm is essentially a steepest-descent scheme with the step width adapted to the local topology of the potential energy surface. The algorithm starts with generating an initial position $\boldsymbol{r}^{(0)}$ randomly within the inflow zone. It then performs a number of successive iterations. Each iteration moves the current position, $\boldsymbol{r}^{(n)}$, to a new position, $\boldsymbol{r}^{(n+1)}$, according to the update rule

$$\boldsymbol{r}^{(n+1)} = \boldsymbol{r}^{(n)} + \frac{\boldsymbol{f}^{(n)}}{f^{(n)}} \delta s^{(n)} \,, \tag{7.4.12}$$

where $\boldsymbol{f}^{(n)}$ is the force on the atom, which is equal to the negative potential energy gradient: $\boldsymbol{f}(\boldsymbol{r}) = -\nabla \mathcal{V}(\boldsymbol{r})$. The positions of the existing atoms are constant during all the time. The displacement, $\delta s^{(n)}$, depends on the potential energy and the force:

$$\delta s^{(n)} = \begin{cases} \delta s_{ovlp} & \text{if } \mathcal{V}^n > \mathcal{V}_{ovlp} \,, \\ \min\left(\Delta s, \frac{\mathcal{V}^n - \mathcal{V}_0}{f^n}\right) & \text{if } \mathcal{V}^n \leq \mathcal{V}_{ovlp} \,. \end{cases} \tag{7.4.13}$$

$\mathcal{V}_0$ is the target potential energy of the atoms ($\mathcal{V}_0 = \langle E_p \rangle$).

Depending on the randomly generated starting position, the iterations need to be downhill ($\mathcal{V}^{(0)} > \mathcal{V}_0$) or uphill ($\mathcal{V}^{(0)} < \mathcal{V}_0$). Both cases can be unified through mirroring the uphill case into a downhill one. This is easily achieved by scaling the potential energy as $\mathcal{V}(\boldsymbol{r}) \longrightarrow \mathrm{sgn}\mathcal{V}(\boldsymbol{r})$, where $\mathrm{sgn} \equiv (\mathcal{V}^{(0)} - \mathcal{V}_0)/|\mathcal{V}^{(0)} - \mathcal{V}_0)|$. This step redefines the forces as well, so that the uphill case ($\mathrm{sgn} = -1$) is converted into a downhill one. For the entire discussion on the USHER algorithm, the potential energy and forces are considered to be scaled by sgn even if 'sgn' is not explicitly included in the equations. Thus, only the downhill case is discussed below, which obviously covers the uphill case as well.

The following explanation on how the USHER algorithm works follows the original article of Delgado-Buscalioni and Coveney [39]. The optimal values for the parameter for the LJ-fluid have also been taken from there. Due to the randomly chosen starting position, $\boldsymbol{r}_0$, it is very likely that the position overlaps with an existing atom. The overlap is detected in the first line of Equation (7.4.13) if the potential energy, $\mathcal{V}^{(0)}$, is higher than the threshold $\mathcal{V}_{ovlp}$, which is chosen to be sufficiently high: $\mathcal{V}_{ovlp} \approx 10^4$. The initial overlapping can be resolved in one iteration step by choosing the appropriate displacement $\delta s_{ovlp} = r_\sigma - (4\mathcal{V}^{(n)})^{1/12}$. The value of r_σ should be slightly larger than the characteristic distance between two atoms. For LJ fluids, a value of $r_\sigma = 0.9$ has been used.

After resolving the initial overlap ($\mathcal{V}^{(n)} < \mathcal{V}_{ovlp}$), the atom is moved downhill towards the target energy, $\mathcal{V}_0$. The second line of Equation (7.4.13) ensures that the energy $\mathcal{V}^{(n)}$ decrease linearly as the atom is moved along the steepest-descent direction. When the gradient of the energy slope becomes flat ($f^{(n)} \approx 0$), the maximum displacement is limited to Δs. The maximum displacement is important for the rate of convergence. For optimal convergence rates, the value should be matched to the width of the low-energy tubes in the landscape of the potential energy field. The value used here is $\Delta s = 0.1n^{-3/2}$.

The search terminates successfully if a position with the desired energy has been found. Obviously, it is not necessary that the energy, $\mathcal{V}^{(n+1)}$, at the position, $\boldsymbol{r}^{(n+1)}$ matches with extreme precision to the target energy $\mathcal{V}_0$. Therefore the following criteria is used to determine whether $\mathcal{V}^{(n+1)}$ is acceptably close to $\mathcal{V}_0$:

$$|\xi|^{(n+1)} < \xi_{max} \,, \quad \text{with } \xi^{(n+1)} = \frac{\mathcal{V}^{(n+1)} - \mathcal{V}_0}{|\mathcal{V}_0|} \,. \tag{7.4.14}$$

where ξ_{max} is the maximum deviation of $\mathcal{V}^{(n+1)}$ from $\mathcal{V}_0$ relative to $\mathcal{V}_0$. In this work a $\xi_{max} = 0.05$ was used. The difference $\mathcal{V}_0 - \mathcal{V}^{(n+1)}$ is added to the internal kinetic energy $E_{k,int}$ in order to really insert the prescribed amount of energy.

It has been shown that it is computationally more efficient to abort the search and to restart from a new randomly generated position [39], once the search leads uphill instead of downhill, i.e. when $\mathcal{V}^{(n+1)} > \mathcal{V}^{(n)}$. The reason is that it is very likely that the current energy well is not deep enough to find a position with the target energy level. The implementation of the USHER scheme as used in the MD-code is given in Algorithm 7.

Algorithm 7: USHER algorithm to insert an atom into a dense medium.

Generate random position $\boldsymbol{r}$;
Calculate potential energy $\mathcal{V}$ and force $\boldsymbol{f}$ for position $\boldsymbol{r}$;
Scale $\mathcal{V}$ and $\boldsymbol{f}$ with sign $\leftarrow (\mathcal{V} - \mathcal{V}_0)/|\mathcal{V} - \mathcal{V}_0|$;
repeat
  if $\mathcal{V} > \mathcal{V}_{ovlp}$ **then**
    $\delta s \leftarrow r_\sigma - (4\mathcal{V})^{1/12}$;
  else
    $\delta s \leftarrow (\mathcal{V} - \mathcal{V}_0)/|\boldsymbol{f}|$;
    if $\delta s > \Delta s$ **then** $\delta s \leftarrow \Delta s$;
  $\boldsymbol{r} \leftarrow \boldsymbol{r} + (\boldsymbol{f}/|\boldsymbol{f}|) \cdot \delta s$;
  Calculate potential energy $\mathcal{V}_{new}$ and force $\boldsymbol{f}_{new}$ for position $\boldsymbol{r}$;
  Scale $\mathcal{V}$ and $\boldsymbol{f}$ with sign $\leftarrow (\mathcal{V} - \mathcal{V}_0)/|\mathcal{V} - \mathcal{V}_0|$;
  if $\mathcal{V}_{new} > \mathcal{V}$ **then** Return and restart search;
  $\mathcal{V} \leftarrow \mathcal{V}_{new}$, $\boldsymbol{f} \leftarrow \boldsymbol{f}_{new}$;
  $\xi \leftarrow (\mathcal{V} - \mathcal{V}_0)/|\mathcal{V}_0|$;
until $|\xi| < \xi_{max}$;

Computational optimisations

The majority of computational time of the USHER algorithm is spent in the calculation of the potential energy of the new atom on the investigated position and the force that would act on it. Hence, it is worth generating a temporary neighbour list for this purpose that contains the indexes of all atoms that are within a radius of r_{neigh} around the initial position $\boldsymbol{r}_0$. The radius, r_{neigh}, is the cutoff distance, r_c, extended by r_{add} ($r_{neigh} = r_c + r_{add}$). Before each force and energy calculation it must be checked whether the position has moved further than r_{add}. If yes, the neighbour list is rebuild, using the current position, $r^{(n)}$, as centre.

Modified USHER scheme

The USHER scheme works especially well in situations where the density across the boundary cell gradually decreases. Here, the scheme has no difficulties finding a suitable place to insert the atom. If the density falls to zero by a sharp drop, the zone in which USHER can find a suitable position with little computational effort is very small. In solid state, materials have a specific structure, also near the boundary. Inserting atoms into the structure would alter it and cause defects that would otherwise not be there. In these cases it is more desirable to add the new atom onto the outer 'surface'. To this end, the USHER scheme was modified by the author to an algorithm that basically moves the new atom from outside to a suitable place on the surface. This is indicated in Figure 7.3b where the search path for the position of the green marked new atoms is drawn in. The figure has been prepared based on a solid LJ-system with a density of $\rho = 1.2$. For reasons of simplicity, only cases where the direction of the surface normal coincides with one dimension are considered in the modified USHER algorithm. The dimension in which the surface normal ($\boldsymbol{n}_S$) is di-

rected is labelled with α and the two other dimensions with β and γ ($n_{S\alpha} = 1$, $n_{S\beta} = 0$, $n_{S\gamma} = 0$). In principle the boundary face can point in positive or negative α direction. Both cases can be handled unified by using $n_{S\alpha}$ as directional factor.

The main difference of the modified USHER scheme to the original one, is that the position is moved primarily in dimension α towards the surface. Only if the move in dimension α direction would lead uphill, the position is moved within the two other dimensions β and γ. This can easily be achieved by scaling the force vector, $\boldsymbol{f}^{(0)}$, with

$$\boldsymbol{f}^{(n)} \leftarrow \left(\boldsymbol{f}^{(n)} \cdot \left(c_\alpha, c_\beta, c_\gamma\right)\right)^{\mathrm{T}} , \tag{7.4.15}$$

where the factors c_α, c_β and c_γ are:

$$c_\alpha = \begin{cases} 1 & f_\alpha n_{S\alpha} < 0 \\ 1 & f_\alpha n_{S\alpha} \geq 0 \end{cases} , \tag{7.4.16}$$

$$c_\beta = \begin{cases} 0 & f_\alpha n_{S\alpha} < 0 \\ 0 & f_\alpha n_{S\alpha} \geq 0 \end{cases} , \tag{7.4.17}$$

$$c_\gamma = c_\beta . \tag{7.4.18}$$

To start the position search, the components $r_\beta^{(0)}$ and $r_\gamma^{(0)}$ of the initial position $\boldsymbol{r}^{(0)}$ are generated randomly within the area of the boundary cell. Projecting a circle in the β,γ plane with the cutoff radius r_c and the centre $(r_\beta^{(0)}, r_\gamma^{(0)})$ along the surface vector $\boldsymbol{n}_S$ generates a cylinder. The contours of this cylinder are coloured green in Figure 7.3b. If one moved the position of the new atom along the central axis of the cylinder, it would interact with each of the atoms within at some position along the line. From all atoms within the cylinder, the algorithm determines the atom j which is furthest outside in $\boldsymbol{n}_S$ direction. The initial position $r_\alpha^{(0)}$ is assigned with $r_{j_\alpha} + r_{min} n_{S\alpha}$, where $r_{min} n_{S\alpha}$ is an additional shift in $n_{S\alpha}$ direction to exclude any initial overlapping. A value of $r_{min} = 1.22$, which is the position of the potential well of the LJ potential, has be used throughout this work. The pseudo code of the modified USHER scheme is given in Algorithm 8.

When using for solids, the target energy cannot be exactly the potential energy of the bulk system, but must be the average energy of a surface atom. This condition has further implication on the imposition of convective fluxes which will be addressed later.

Random velocity vector generation

After, the position for a new atom has been found, it needs to be assigned with a velocity vector that satisfies the conditions that have been introduced at the beginning of this section. Because of the unidirectionality of the internal energy [1], this can only be achieved by matching the averages of a number of atoms that are inserted during a period of time. For

[1] Here it is meant that the internal kinetic energy, which is the thermal motion of the atoms, is not bound to the movement in any direction, but rather has zero overall velocity by definition.

Algorithm 8: Modified USHER algorithm to insert an atom into a dense medium with a very sharp density profile.

Generate initial position components r_β and r_γ randomly;
Build list L_{tmp} of atoms within the circle (r_β, r_γ) at projected along $\boldsymbol{n}_S$;
Find atom j in L_{tmp} furthest in direction $\boldsymbol{n}_S$;
Set $r_\alpha \leftarrow r_{j_\alpha} + r_{min}$;
Calculate potential energy $\mathcal{V}$ and force $\boldsymbol{f}$ for position $\boldsymbol{r}$;
Scale $\mathcal{V}$ and $\boldsymbol{f}$ with sign $\leftarrow (\mathcal{V} - \mathcal{V}_0)/|\mathcal{V} - \mathcal{V}_0|$;
repeat
  if $(n_{S\alpha} \cdot f_\alpha) < 0$ **then**
    $c_\alpha \leftarrow 1$;
    $c_\beta, c_\gamma \leftarrow 0$;
  else
    $c_\alpha \leftarrow 0$;
    $c_\beta, c_\gamma \leftarrow 1$;
  $\boldsymbol{f} \leftarrow \left(\boldsymbol{f} \cdot \left(c_\alpha,\ c_\beta,\ c_\gamma\right)\right)^{\mathrm{T}}$;
  if $\mathcal{V} > \mathcal{V}_{ovlp}$ **then**
    $\delta s \leftarrow r_\sigma - (4\mathcal{V})^{1/12}$;
  else
    $\delta s \leftarrow (\mathcal{V} - \mathcal{V}_0)/|\boldsymbol{f}|$;
    if $\delta s > \Delta s$ **then** $\delta s \leftarrow \Delta s$;
  $\boldsymbol{r} \leftarrow \boldsymbol{r} + (\boldsymbol{f}/|\boldsymbol{f}|) \cdot \delta s$;
  Calculate potential energy $\mathcal{V}_{new}$ and force $\boldsymbol{f}_{new}$ for position $\boldsymbol{r}$;
  Scale $\mathcal{V}$ and $\boldsymbol{f}$ with sign $\leftarrow (\mathcal{V} - \mathcal{V}_0)/|\mathcal{V} - \mathcal{V}_0|$;
  if $\mathcal{V}_{new} > \mathcal{V}$ **then** Return and restart search;
  $\mathcal{V} \leftarrow \mathcal{V}_{new}$, $\boldsymbol{f} \leftarrow \boldsymbol{f}_{new}$;
  $\xi \leftarrow (\mathcal{V} - \mathcal{V}_0)/|\mathcal{V}_0|$;
until $|\xi| < \xi_{max}$;

a period of time with constant convective fluxes, the velocities of N_{ins} inserted atoms must fulfill the conditions:

$$\langle \boldsymbol{v} \rangle = \boldsymbol{u} = \frac{1}{N_{ins}} \sum_i^{N_{ins}} \boldsymbol{v}_i \ , \tag{7.4.19}$$

$$\langle E_{k,int} \rangle = \frac{1}{N_{ins}} \sum_i^{N_{ins}} \frac{1}{2} m_i (\boldsymbol{v}_{i,int})^2 \ , \tag{7.4.20}$$

where $\boldsymbol{v}_{i,int}$ is the internal velocity of the inserted atom i that is related to the absolute velocity $\boldsymbol{v}_i$ by $\boldsymbol{v}_i = \boldsymbol{v}_{i,ext} + \boldsymbol{v}_{i,int}$; the external velocity vector $\boldsymbol{v}_{i,ext}$ being equal for all inserted atoms: $\boldsymbol{v}_{i,ext} = \langle \boldsymbol{v} \rangle = \boldsymbol{u}$. In connection with Equation (7.4.19), this requires the sum of the internal velocities to be zero:

$$\boldsymbol{v}_{int}^{sum} = \sum_i^{N_{ins}} \boldsymbol{v}_{i,int} = \boldsymbol{0} \ . \tag{7.4.21}$$

Since the external velocities $\boldsymbol{v}_{i,ext}$ of the inserted atoms are determined, the remaining task is to generate the internal velocity vectors $\boldsymbol{v}_{i,int}$ in a way that Equation (7.4.20) and Equation (7.4.21) are satisfied. Being of macroscopic nature (four degrees of freedom), these conditions require the reconstruction of the microscopic information ($3N_{ins}$ degrees of freedom), which are the direction and speed of the individual velocities. Principally, this is identical to the initialisation of velocities at the beginning of a MD simulation (q.v. Section 2.2.8).

Maxwell-Boltzmann distribution: As in the velocity generation schemes for the initialisation of the molecular dynamics simulation, the internal velocity vectors can be sampled from a Maxwell-Boltzmann distribution and as far as known to the author, this has been done in all previous works [38, 40, 188, 193]. The Maxwell-Boltzmann distribution has already been defined in Equation (2.2.28), where the average energy is expressed in terms of temperature. Here, the average energy level is $\langle E_{k,int} \rangle$ and the probability of the velocity v_α in dimension $\alpha = \{x, y, z\}$ is given by

$$f(v_\alpha) = \sqrt{\frac{3m}{4\pi \langle E_{k,int} \rangle}}\, e^{\left(-\frac{3mv_\alpha^2}{4\langle E_{k,int}\rangle}\right)}, \tag{7.4.22}$$

where the individual dimensions, α, are independent from each other. Sampling from Equation (7.4.22), one obtains a set of isotropic velocity vectors with the average length of $\langle |\boldsymbol{v}| \rangle = \sqrt{2\langle E_{k,int} \rangle / m}$. As demonstrated in Section 2.2.8, a generator that draws random numbers from a gaussian distribution and a subsequent scaling $v_\alpha \leftarrow v_\alpha \sqrt{2\langle E_{k,int}\rangle / 3m}$ will deliver the desired velocity vectors.

Summed momentum: An unexpected effect occurs for free stream test cases when sampling the internal velocities of the inflowing atoms from a Maxwell-Boltzmann distribution. When direct convective flux boundary conditions (DC) are used, the momenta which are removed on the outlet, correspond exactly to the analytical value (Equation (6.3.2)). Therefore, any deviations of the inserted momenta on the inlet from the prescribed momentum fluxes remain within the molecular system.

Sampling the internal velocities from the Gaussian distribution, the average sampled velocity $\langle v_{int\alpha} \rangle$ in each dimension $\alpha = \{x, y, z\}$ approaches 0 for increasing numbers of inserted atoms. However, this is not true for sum of the internal velocities, $\boldsymbol{v}_{int}^{sum}$, as required by Equation (7.4.21). Rather, $\boldsymbol{v}_{int}^{sum}$ fluctuates stronger, the more atoms are inserted. This is easily realised when considering that the standard deviation of the sum of the inserted atoms grows with increasing number of samples: $\sigma(v_{int\ \alpha}^{sum}) \sim \sqrt{N_{ins}}$. Therefore, the longer the simulation runs, the more likely larger deviation of $\boldsymbol{v}_{int}^{sum}$ from $\mathbf{0}$ become. The logical effect, which has been observed in test simulation, are deviations in the velocities u_x, u_y and u_z of the overall system.

Figure 7.4 shows the development of the velocity of a system with 10,000 atoms at a temperature of $T = 2$ during the influx of 5,000 and 50,000 atoms. The figures have not

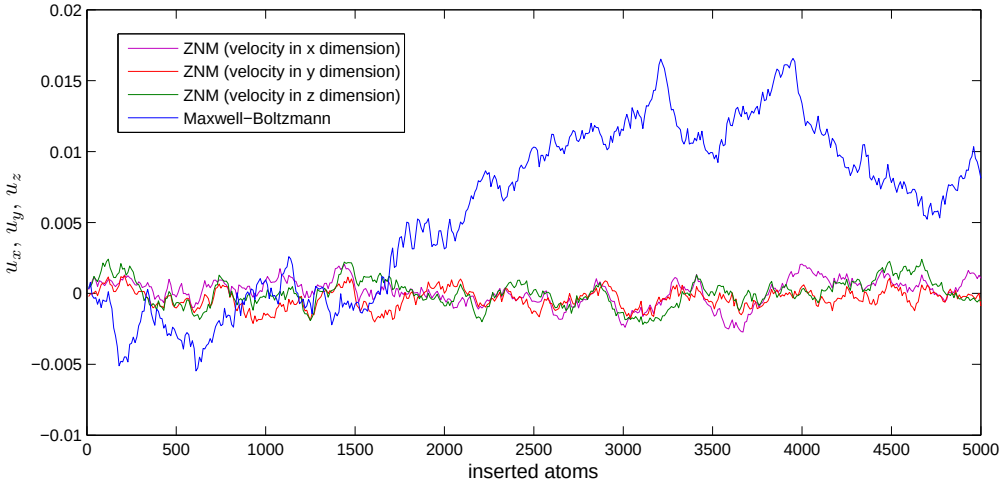
been recorded from an actual simulation, but to avoid any side effects, the velocity generation algorithms have been run independently and the generated internal velocity vectors, corresponding to a temperature of $T = 2$, have been summed up. The blue line marks the velocity, u_y, of the system for which the velocities of the inserted atoms have been sampled from the Maxwell-Boltzmann distribution (Equation (7.4.22)). The velocities in the other two dimensions are not shown because the curves are effectively similar and would not contribute any additional insight. It is obvious that the velocity, u_y, starts to deviate up to 0.015 for less than 5,000 inserted atoms. Continued influx causes u_y to rise nearly to 0.05 after 50,000 atoms have been inserted. How problematic this value is depends on the free stream velocity u_x of the specific test case.[2] For example in several test cases, a velocity of $u_x = 0.1$ was used. The deviation, in Figure 7.4b, is nearly half of this value, which is very significant.

For free stream test cases, deviation of the free stream velocity $\boldsymbol{u}$ from the analytic value have direct and indirect implications on stability of the MD flux boundary condition schemes. The decisive point is the outlet, which for direct convection (DC) schemes only function in a stable manner as long as the actual fluxes of density, momentum and energy on the interface correspond to the prescribed ones. Small deviation are buffered, but persisting deviations eventually lead to a breakdown of the MD flux boundary conditions, either because of a lack or excess of mass, momentum or energy. The flow of mass, momentum and energy on the outlet boundary is influenced directly by changes in u_x or indirectly by any velocities u_y or u_z different from 0. The latter effect is because any velocity in y or z dimension also binds a part of the total energy. Because no momentum ρu_y or ρu_z is removed by the MD flux boundary conditions on the outlet, the energy that is bound to this momentum cannot be extracted. Instead, more internal energy is removed until the internal energy near the outlet boundary is zero and the boundary conditions break down.

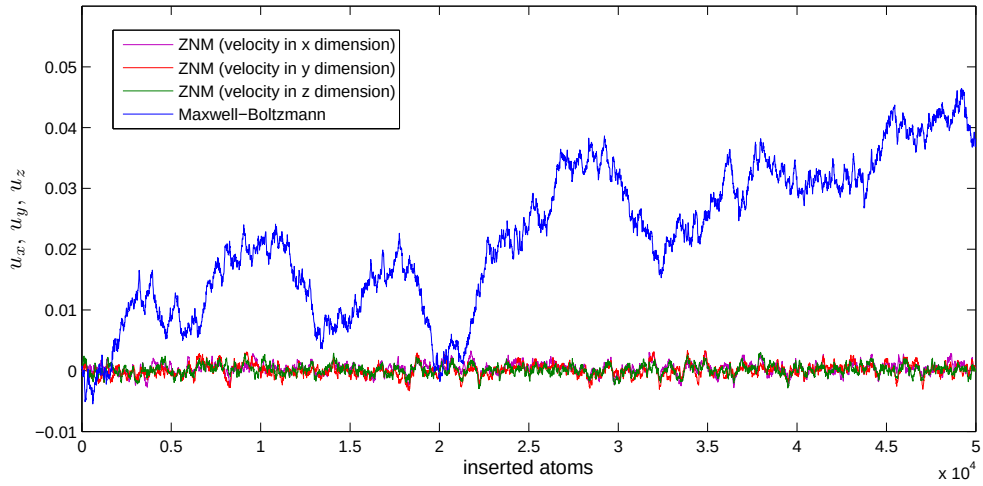
Obviously for a fully coupled system, any change of the velocity in the molecular system would also change the calculated fluxes on the outlet, so that the effect would probably not be noticed. However, the idea of the free stream test case is, that MD flux boundary conditions introduce and remove the prescribed flux. Also, the velocities of the inflowing atoms are not strictly independent and uncorrelated from each other, because any greater random fluctuation leading to a change in flow speed would be cause a ‘physical reaction’ of the system counteracting the disturbance.

Zero net momentum: To avoid the unwanted effects that arise by sampling velocities from a Maxwell-Boltzmann distribution, a *zero net momentum* (ZNM) algorithm for the generation of the internal velocities with was devised by the author. Instead of drawing the components of the velocity vectors v_x, v_y and v_z individually from a basically infinite pool of possible numbers, the underlying idea of the ZNM algorithm is to select the complete vector $\boldsymbol{v}$ randomly from a limited set of possible vectors. To this end, a set $\mathbb{S}_{pv}$ of N_{pv} possible unit velocity vectors is generated initially. When a new velocity vector is needed, one element $\boldsymbol{v}_k$ of $\mathbb{S}_{pv}$ is selected through a randomly generated index k, where k is an in-

[2] As defined in the description test case (Section 6.3), the free flow direction is in x direction.



(a) Inserting the first 5,000 atoms.



(b) Inserting 50,000 atoms.

Figure 7.4: Example for the change in velocity of a system with 10,000 atoms at $T = 2$ through the influx of atoms, with velocities sampled through Maxwell-Boltzmann distribution (blue lines) or ZNM algorithm (purple, red and green lines).

teger that obeys a discrete uniform distribution ($k \in [\![1, N_{pv}]\!], k \sim U(1, N_{pv})$. The selected random vector is removed from $\mathbb{S}_{pv}$ ($\mathbb{S}_{pv} \leftarrow \mathbb{S}_{pv} \backslash \boldsymbol{v}_k,\ N_{pv} \leftarrow N_{pv} - 1$). Once $\mathbb{S}_{pv}$ is empty ($\mathbb{S}_{pv} = \emptyset$), the complete set of possible velocity vectors is generated again. If the sum of the all vectors in the full set $\mathbb{S}_{pv}$ is $\mathbf{0}$, then the maximum deviation of the summed velocities is effectively restricted, because every time the entire set $\mathbb{S}_{pv}$ has been used, the introduced net momentum must be zero.

To achieve an isotropic distribution of the velocity vectors in $\mathbb{S}_{pv}$, a virtual sphere is used with centre at the point of origin and unity radius. The vectors are generated from the centre of the sphere to evenly distributed points at the surface. In order to ensure exact

symmetry in all dimensions the surface points are defined in the first octant ($v_x \geq 0$, $v_y \geq 0$, $v_z \geq 0$) and then mirrored into the seven other octants. The vectors in the first octant are calculated through

$$\begin{pmatrix} v_x \\ v_y \\ v_z \end{pmatrix} = \begin{pmatrix} \cos(\alpha) \\ \sin(\alpha)\cos(\beta) \\ \sin(\alpha)\sin(\beta) \end{pmatrix} \tag{7.4.23}$$

where the angles α and β are obtained by stepping through all combinations of the indexes a and b

$$\alpha = \frac{\pi a}{2R} \quad \text{and} \quad \beta = \begin{cases} \frac{b}{R\sin(\alpha)} & \alpha > 0 \\ 0 & \alpha = 0 \end{cases} . \tag{7.4.24}$$

The index a is an integer that rotates the vector within the xy-plane and runs from 0 to the selected resolution R: $a = \{0, 1, \ldots, R\}$. The index b rotates the vector in the yz-plane at $v_x = \cos(\alpha)$. The resolution of b depends on the circumference, $2\pi sin(\alpha)$, of the circle that forms the intersection line of the sphere and the yz-plane at $v_x = \cos(\alpha)$. Hence, the value b runs through $b = \{0, 1, \ldots, R\sin(\alpha)\}$.

In Figure 7.5, the first octant of the virtual sphere for the velocity vector generation is illustrated. The points marked, show some of the points on the surface, to which the velocity vectors are drawn. The first number in the denotation is the index a and the second is index b: (a, b). The blue shaded surface represents the yz plane for $a = 3$ and the drawn vector the velocity of the combination $a = 3$ and $b = 4$. The length of the vectors that are generated by Equation (7.4.23) are of unity length. To match the required energy level (Equation (7.4.7)), they are scaled with $\sqrt{2E_{k,int}/m}$. Obviously, all velocity vectors are of the same length and do not correspond to the natural Maxwell-Boltzmann distribution. However, as previously tested in Section 2.2.8, velocity vectors generated with a fixed length quickly approach a Maxwell-Boltzmann distribution. Also, the new atoms are not inserted all at once, but over a period of time, which reduces the effect of the fixed length. The implementation of the ZNM algorithm is given in Algorithm 9. The required input data is the internal kinetic energy $E_{k,int}$ and the atomic mass m. It returns a random velocity vector for the requested energy level.

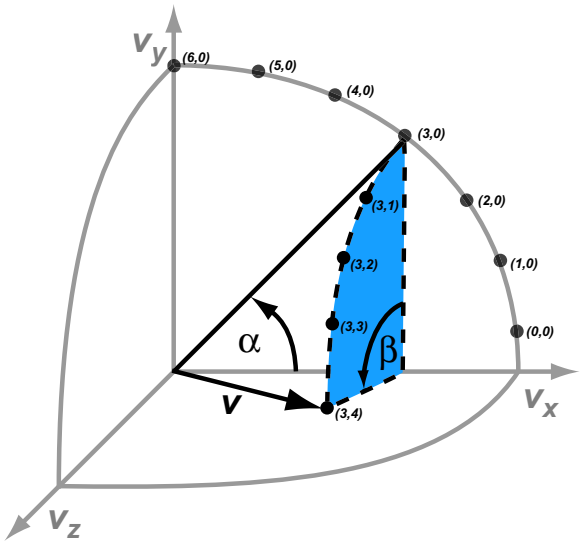


Figure 7.5: Illustration of a the first octant of the virtual sphere for the ZNM velocity vector generation.

Algorithm 9: Zero net momentum (ZNM) algorithm for generating the velocity vector for a new atom.

input : Energy level $E_{k,int}$ and mass of the new atom m
output: New velocity vector $\boldsymbol{v}_{out}$

if $N_{pv} = 0$ **then**
  for $a \leftarrow 0$ **to** R **do**
    $\alpha \leftarrow \pi a/(2R)$;
    $R_b \leftarrow \sin(\alpha)R$;
    for $b \leftarrow 0$ **to** R_b **do**
      if $R_b > 0$ **then** $\beta \leftarrow b/(\sin(\alpha)R)$;
      else $\beta \leftarrow 0$;
      $v_x \leftarrow \cos(\alpha)$;
      $v_y \leftarrow \sin(\alpha) \cdot \cos(\beta)$;
      $v_z \leftarrow \sin(\alpha) \cdot \sin(\beta)$;
      Add $\boldsymbol{v}$ to $\mathbb{S}_{pv}$ ($N_{pv} \leftarrow N_{pv} + 1$);
  foreach *Combination of* $\boldsymbol{o} = (\{-1, 1\}, \{-1, 1\}, \{-1, 1\})$ **do**
    foreach *vector* $\boldsymbol{v}$ *in* $\mathbb{S}_{pv}$ **do**
      if $\boldsymbol{o} \cdot \boldsymbol{v} <> \boldsymbol{v}$ **then**
        Add vector $\boldsymbol{o} \cdot \boldsymbol{v}$ to $\mathbb{S}_{pv}$ ($N_{pv} \leftarrow N_{pv} + 1$);
$s \leftarrow \mathrm{U}(0, N_{pv} - 1)$;
$\boldsymbol{v}_{out} \leftarrow$ element number s of $\mathbb{S}_{pv}$;
Remove element number s from $\mathbb{S}_{pv}$;
$\boldsymbol{v}_{out} \leftarrow \sqrt{2E_{k,int}/m} \cdot \boldsymbol{v}_{out}$;

The ZNM algorithm has been tested under the same conditions as the previously discussed Maxwell-Boltzmann distributed velocity vector generation. Firstly, 5,000 new velocity vectors, corresponding to $T = 2$, were generated by the ZNM algorithm and added to the velocity $\boldsymbol{u}$ of a 10,000 atom system. In a second step, the insertion was continued to 50,000 atoms. The resolution of $\mathbb{S}_{pv}$ was set to $R = 30$ for the test run. The individual velocity components u_x, u_y and u_z for the first 5,000 atoms have been plotted into Figure 7.4a. It is obvious that the deviations from 0 are much smaller than for the sum of the Maxwell-Boltzmann distributed velocity vectors. The continued insertion of up to $50,000$ atoms, shown in Figure 7.4b, demonstrates this even more clearly since the sum of velocity vectors generated with the ZNM algorithm stays close to zero, independent of the number of atoms inserted. The curves are similar in all three dimensions, with a mean of $u_x = u_y = u_z = 0$ and a standard deviation of $\sigma(u_x) = 9.1 \times 10^{-4}$, $\sigma(u_y) = 8.9 \times 10^{-4}$ and $\sigma(u_z) = 9.3 \times 10^{-4}$. These values were calculated from the run that is shown in Figure 7.4; they vary slightly between individual runs, because of the random number generation.

7.4.3 Removal of atoms

Any outflow across the boundary requires the physical removal of boundary atoms from the reservoir region in Figure 7.1. While for inflow boundary conditions it is possible to insert atoms that match exactly the constraints prescribed by the convective fluxes, the same is not true for the removal of atoms for outflow boundary conditions. Based on the integrated fluxes, any removed atom should ideally have the momentum $p_{ida} = F_{ec}^{int}/F_{mc}^{int}$ and the total energy $E_{tida} = F_{ec}^{int}/F_{mc}^{int}$. It is basically impossible to find a boundary atom whose momentum and energy match exactly p_{ida} and E_{tida}. Thus, outflow boundary conditions require a special treatment to ensure the agreement of the actual removed quantities to the prescribed integrated convective fluxes: $F_{pc}^{imp} = F_{pc}^{int}$ and $F_{ec}^{imp} = F_{ec}^{int}$.

The first step of an appropriate outflow scheme is to select the atom to be removed from the set of N_b boundary atoms in the reservoir region. A number of different selection schemes have been investigated for their applicability:

***Random*:** The atom is determined randomly from entire set of N_b boundary atoms.

***Position based*:** Probably the most intuitive scheme is to chose the atom which is furthest away from the boundary, i.e. highest value of d_{bi}. Conveniently, the boundary atom list, which is sorted by d_{bi}, can be used for this purpose; the first atom in the list being the one furthest away from the boundary.

***Momentum based*:** Both previous schemes do not guarantee $F_{pc}^{imp} = F_{pc}^{int}$. To ensure the correct convective momentum flux implementation, one can select the atom whose momentum matches the required momentum transfer closest. This is the boundary atom i with the smallest momentum difference Δp_i that is calculated by

$$\Delta p_i = |\boldsymbol{v}_i - p_{ida}| \; . \tag{7.4.25}$$

***Energy based*:** Similar to the momentum based scheme, the selection of the atom can be based on the convective energy flux. In that case the criteria is

$$\Delta e_i = |E_{p_i} + \frac{1}{2} m_i \boldsymbol{v}_i^{\,2} - E_{tida}| \; , \tag{7.4.26}$$

where the atom i with the smallest Δe_i is chosen to be removed.

***Best fit*:** The position, momentum and energy based schemes focus on only one aspect, neglecting the other two. Therefore, it is only logical to derive a scheme that compensates this deficiency by taking into account the position, momentum and energy fluxes. To this end, one can define the criteria c_{bf} by

$$c_{bf_i} = \frac{1}{d_{bi}^{\,2}} + \frac{\Delta p_i^{\,2}}{p_{ida}^{\,2}} + \frac{\Delta e_i^{\,2}}{e_{ida}^{\,2}} \; . \tag{7.4.27}$$

The atom i with the lowest c_{bf_i} is the one that is selected to be removed.

Which of the above selection schemes is used for the test cases, is stated at the respective sections.

Even when using the best fit scheme for the atom selection, a match of actual implemented ($F_{\alpha\beta}^{imp}$) and integrated ($F_{\alpha\beta}^{int}$) fluxes is not guaranteed. The main reason is that with the density in the reservoir region gradually approaching zero, the average atomic values of velocity, $\langle \boldsymbol{v} \rangle$ and energy $\langle E_t \rangle$ also differ from the values of the bulk materials in the boundary cell. The density profile across the interface, detailed in Section 7.8.3, is of lesser importance for the removal of atoms, but naturally influences the velocity and energy profiles. The exact shapes of velocity and energy profiles are complicated in many cases and subject of oscillations and fluctuations. These are mainly a result of the employed imposition scheme for the fluxes due to stress (MTRP, MTF, MTDVC or MTRV), but also subject to the interplay of these schemes with implementation of the convective fluxes. Therefore, the boundary atoms in the reservoir region may, sometimes only temporarily, not correspond to the convective flux constraints. In Figure 7.6, the profiles across the boundary are shown for the average values of density, velocity, temperature and atomic energy. Clearly, the atoms in the area of the boundary have different average velocities and energies than the atoms inside the bulk material. The four plotted cases also demonstrate that the profiles of the curves depend on the selected scheme and fluid properties. Hence, the expectation of finding an atom selection scheme that matches exactly the convective momentum and energy fluxes is unrealistic.

One possibility to overcome the mismatch between the implemented and the integrated convective fluxes is to correct the respective differences subsequently after the removal of an atom. This is achieved by coupling the convective fluxes with the fluxes by stress. The approach is suitable in conjunction with direct convective flux imposition; it is being discussed in Section 7.4.1.

Momentum and energy correction for removed atoms

When employing indirect convective flux imposition schemes (IC), the convective fluxes are not prescribed directly. The atoms removed under outflow conditions in these schemes are expected to have the same properties as the ones in the boundary cell adjacent to the reservoir. However, because the energy of an average atom in the reservoir region, i.e. behind the boundary, is higher (see Figure 7.6 or 7.4.3), the removal of atoms from the reservoir region without some kind of correction leads to a convective energy outflow that is too high. This is demonstrated for static test case in Section 7.10.1. Figure 7.45 shows that without correcting the energy of removed atoms, the average atomic energy in a static system decreases because of the mentioned effects.

A similar situation prevails for the average momentum of an atom in the reservoir region. Due to the static pressure that is applied onto the boundary atoms in the reservoir, the average momentum of an atom in the relaxation zone tends to be higher than the average momentum of atoms in the boundary cell under inflow conditions and lower under outflow condition (for static situations the deviation is zero). Removing atoms without correcting

the value of the removed momentum to the average atomic value of the boundary cell will therefore lead to a situation where too much or too less momentum is convected across the boundary.

The idea of the momentum and energy correction for removed atoms is to remove an atom whose momentum and energy level correspond to the average atomic momentum and energy level inside the first inner boundary cell. To this end, the average atomic momentum and energy of the atoms in the boundary cell (q.v. Figure 7.1) are calculated and used as reference values $\langle \boldsymbol{p} \rangle_{ref}$ and $\langle \boldsymbol{E}_t \rangle_{ref}$ for momentum and energy respectively. The value of $\langle \boldsymbol{p} \rangle_{ref}$ is obtained from the N_c atoms in the inner virtual cell by

$$\langle \boldsymbol{p} \rangle_{ref} = \frac{1}{N_c} \sum_{i}^{N_c} m_i \boldsymbol{v}_i \ . \tag{7.4.28}$$

$\langle \boldsymbol{E}_t \rangle_{ref}$ is a vector that contains the reference energy for the individual dimensions. The potential energy is split in three equal shares, so that $\langle \boldsymbol{E}_t \rangle_{ref}$ is calculated over the N_c atoms in the inner virtual cell by

$$\langle \boldsymbol{E}_t \rangle_{ref} = \frac{1}{N_c} \sum_{i}^{N_c} \left[\frac{1}{2} m_i \begin{pmatrix} v_{i_x}^2 \\ v_{i_y}^2 \\ v_{i_z}^2 \end{pmatrix} + \begin{pmatrix} E_{p_i}/3 \\ E_{p_i}/3 \\ E_{p_i}/3 \end{pmatrix} \right] \ . \tag{7.4.29}$$

After an atom i has been removed from the reservoir region, the difference between the energy of the atom i and the reference energy value:

$$\boldsymbol{E}_{diff} = \langle \boldsymbol{E}_t \rangle_{ref} - \boldsymbol{E}_{t_i} = \langle \boldsymbol{E}_t \rangle_{ref} - \left[\frac{1}{2} m_i \begin{pmatrix} v_{i_x}^2 \\ v_{i_y}^2 \\ v_{i_z}^2 \end{pmatrix} + \begin{pmatrix} E_{p_i}/3 \\ E_{p_i}/3 \\ E_{p_i}/3 \end{pmatrix} \right] \tag{7.4.30}$$

is transferred as kinetic energy onto the remaining reservoir atoms. The energy transfer scheme (Algorithm 3), as described in Section 2.2.10, has been employed for transferring the energy at each dimension separately.

The difference in momentum, $\boldsymbol{p}_{diff} = \langle \boldsymbol{p} \rangle_{ref} - m_i \boldsymbol{v}_i$, is added to the integrated momentum fluxes by stress, F_{ps}^{int}, and transferred along with the momentum flux transfer scheme. The transfer of pure momentum without transfer of energy is, however, only guaranteed when using the reverse velocity scheme (MTRV), which is described in Section 7.9.1, for the momentum transfer. Thus, momentum correction of removed atoms is only possible in conjunction with this scheme. When using other momentum transfer schemes, the transfer of momentum will, by construction, always transfer some energy as well.[3]

7.5 Breakdown of flux boundary conditions

Before continuing, it is useful at this point to define the conditions under which the flux boundary conditions are considered to have broken down. These comprise a number of

[3]Except for the unlikely case, where the velocity of the atoms onto which the momentum is transferred is zero.

situations in which the simulation must be aborted because it has become impossible to impose the prescribed flux boundary conditions. In detail, those situations are:

No atom that can be removed: The outflow of mass requires the actual presence of an atom that can be removed, i.e. there must be material within the reservoir region. If no boundary atom is left then the convective fluxes can not be imposed and the boundary conditions are violated. This type of break down only applies for direct convective flux imposition. For indirect convective flux imposition, the condition can not occur by construction.

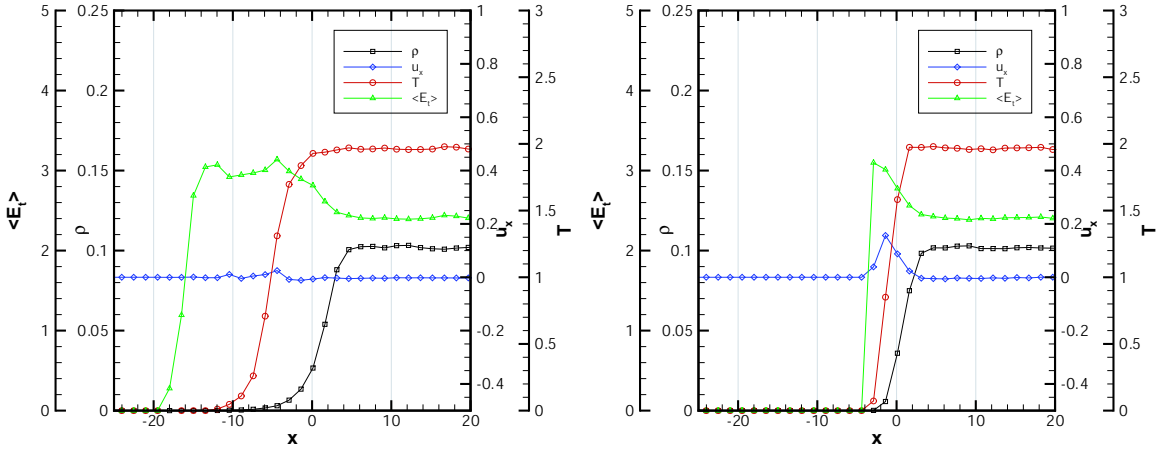
Not enough internal energy: For the outflow of energy, Algorithm 3 is used. Since the energy is removed from the internal kinetic energy, a situation can occur where more energy needs to be removed than is actually available as internal kinetic energy within the reservoir. In this case the flux boundary conditions cannot be fulfilled and are considered to have broken down.

Unable to find an insertion site: For the inflow boundary conditions, situations can occur where the algorithm to find an insertion site fails. These lead to a failure of the boundary conditions as well.

No atom in the reservoir: This situation is similar to the first point, but applies to the fluxes by stress. If no atom is in the reservoir, fluxes by stress can not be applied and their imposition must fail, i.e. flux boundary conditions have broken down.

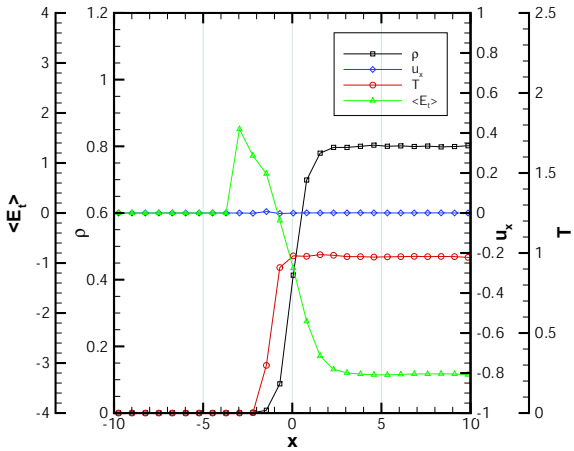
7.6 Further remarks

From the previous description of the MD flux boundary conditions it should be clear that there are a large number of parameters that determine the way in which the fluxes are imposed onto the boundary of the molecular system. In fact, the boundary conditions, as they are implemented into the MD code, have twenty four parameters, some of them containing several sub parameters. It is obvious that the investigation needs to be restricted to a few combination of these parameters and that there is no guarantee that any of the non-tested combinations provides better results than the ones presented. During the investigation a large number of different combinations have been tested, which can not be included into the discussion here due to the restriction in available space and time. However, none of the excluded versions gave better results than the ones presented here. Most of the excluded ones, proved to be unstable for numerous reasons.

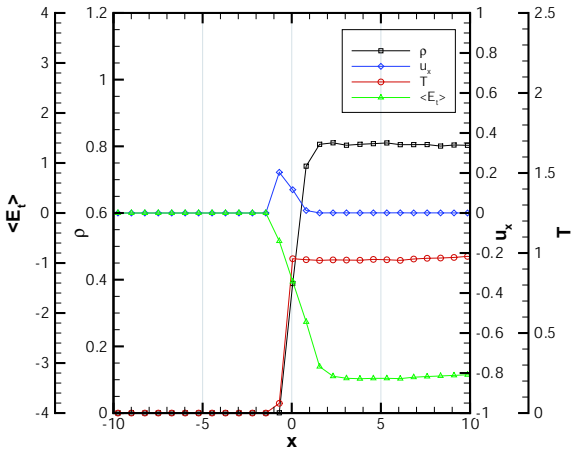


(a) MTF, gaseous state ($\rho = 0.1, T = 2$)

(b) MTRV, gaseous state ($\rho = 0.1, T = 2$)



(c) MTF, liquid state ($\rho = 0.8, T = 1$)



(d) MTRV, liquid state ($\rho = 0.8, T = 1$)

Figure 7.6: Profiles of density (ρ), velocity (u_x), temperature (T) and average total atomic energy ($\langle E_t \rangle$) across the boundary for two different momentum transfer schemes: momentum transfer by force (MTF) and momentum transfer by reverse velocity (MTRV). The upper and lower figures show the a gaseous and liquid state respectively.

7.7 Direct convective flux and reflective planes (DC-MTRP)

This section starts with a discussion on reflective planes for equilibrium situations, followed by a description on how these are used in conjunction with the flux boundary condition.

Reflective plane

A reflective plane is a simple and passive means to confine a molecular system as well as to insert momentum into it. The idea of using a reflective wall in conjunction with MD flux boundaries is that in a local equilibrium situation, in average, for each atom that crosses the boundary, another atom, with opposite velocity vector, will enter the molecular domain. Instead of removing the atom and inserting another one, it is simply reflected. Hence, applying a reflective plane, the part of the momentum flux by stress $\boldsymbol{F}_{ps}$ which arises from the static pressure p is automatically transferred to the system. For each reflected atom i a momentum of $-2m_i\boldsymbol{v}_i \cdot \boldsymbol{n}$ is introduced into the system.

Reflective planes have been regularly used as boundary conditions in MD simulations such as for channel walls [153, chap 7], rising bubbles [129] or lubricants on surfaces [143]. A reflective plane is commonly understood as an artificially defined surface that reflects all atoms whose paths cross the surface. The velocity vector $\boldsymbol{v}_o$ of an atom that is to be reflected is inverted. The reflected velocity vector $\boldsymbol{v}_n$ is calculated by

$$\boldsymbol{v}_n = \boldsymbol{v}_o - 2\boldsymbol{n}(\boldsymbol{v}_o \cdot \boldsymbol{n}) \,, \tag{7.7.1}$$

where $\boldsymbol{n}$ is the unity normal surface vector. In case, the surface is moving with a speed of u into the direction of $\boldsymbol{n}$, the atom's new velocity is

$$\boldsymbol{v}_n = \boldsymbol{v}_o + 2\boldsymbol{n}(u - \boldsymbol{v}_o \cdot \boldsymbol{n}) \,. \tag{7.7.2}$$

The energy in and out flow can be controlled through scaling the velocity vector by a factor f. In the most general case the new velocity vector is then

$$\boldsymbol{v}_n = f(\boldsymbol{v}_o + 2\boldsymbol{n}(u - \boldsymbol{v}_o \cdot \boldsymbol{n})) \,. \tag{7.7.3}$$

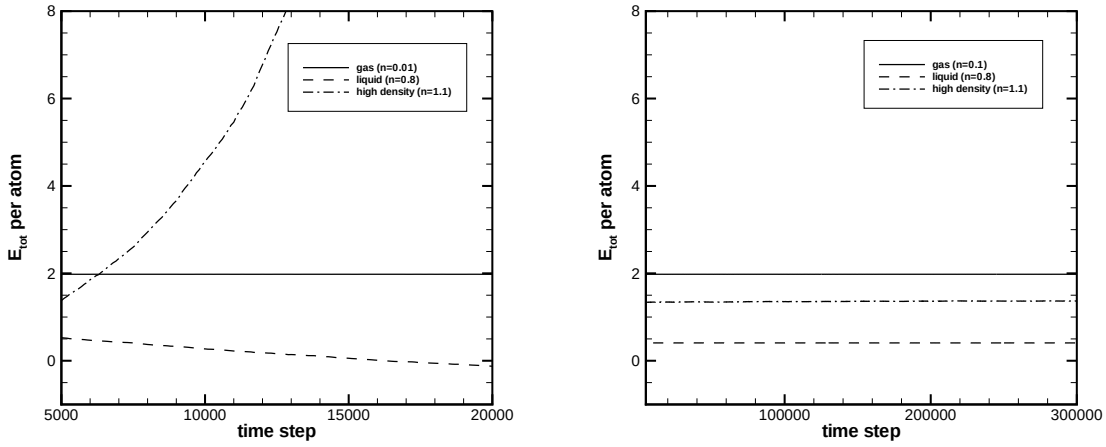
However, manipulating the energy in that manner also involve momentum changes that need to be considered.

In all works known to the author that have used reflective planes as boundary condition, the algorithm to enforce the reflective plane is similar to Algorithm 10, in which $\boldsymbol{r}_p$ is a point anywhere in the plane with the normal vector $\boldsymbol{n}$. If the distance d to the plane is negative, the atom has crossed the plane and is reflected. Thereby, its position is shifted back behind the plane by a small distance, ϵ [153, chap 7]. Other options involve the calculation of the point of impact and compute the new position of the atom from this point.

Regardless of the precise way, changing the position of the reflected atom from the position that has been obtained by the numerical integration algorithm has consequences for the potential energy of the atom. To demonstrate this, Algorithm 10 was tested for

Algorithm 10: Standard reflective plane algorithm.

foreach *atom i* **do**
  $d = (\boldsymbol{r}_p - \boldsymbol{r}_i) \cdot \boldsymbol{n}$;
  if $d < 0$ **then**
    $\boldsymbol{r}_i \leftarrow \boldsymbol{r}_i + \boldsymbol{n}(d + \epsilon)$;
    $\boldsymbol{v}_i \leftarrow f(\boldsymbol{v}_i + 2\boldsymbol{n}(u - \boldsymbol{v}_i \cdot \boldsymbol{n}))$;



(a) Reflective plane using Algorithm 10

(b) Reflective plane using Algorithm 11

Figure 7.7: Total energy per atom is time for a box enclosed in x direction by two reflective walls.

equilibrium simulations of a box filled with 3200 LJ atoms in gaseous ($n = 0.001$), liquid ($n = 0.8$) or high density ($n = 1.1$) states. Periodic boundary conditions have been applied in y and z direction. In x direction the box was enclosed by two reflective planes based on Algorithm 10. The simulations were run through an NVE ensemble. Hence, the energy was expected to be constant in time. From Figure 7.7a it is obvious that the energy is only conserved for the gaseous state. The simulation for the liquid state shows a drop in energy, while in the high density case the energy increases rapidly.

The reason for this unwanted behaviour is the shift of the reflected atoms back in front of the wall. By doing so, the potential energy of the shifted atom is changed as it is moved closer to other atoms near the reflective plane. In the liquid case the average atomic distance is just far enough that when moving two average atoms closer together by a small distance, the pair potential energy between them becomes lower due to the potential well of the LJ potential (see Section 2.1.5). For higher densities, the situation is different. Moving two average atoms towards each other will bring them in the region of strong repulsion of the LJ potential, where the energy increases rapidly for smaller distances. Hence, for these densities, a reflective plane, using Algorithm 10, introduces additional energy into the system. In the gaseous state, atoms are separated so far from each other that a slight shift of the atomic positions has negligible influence on its potential energy.

In theory, one could measure the change in potential energy, E_{p_i}, of the shifted atom, i, and correct it by rescaling its velocity vector $\boldsymbol{v}_i$. This procedure is impractical for two reasons. Firstly, the energy to be subtracted may be higher than the kinetic energy of the atom; the procedure would fail in that case. Secondly, it would involve additional computational and programming efforts to compute the potential energy of the reflected atom outside the standard force calculation procedures.

Another approach, much simpler and practical, is therefore suggested: Reflected atoms are not shifted back behind the wall. Only the velocity vector is reversed. The reflected atom will then travel back, by itself, behind the reflective plane in the next time step. The pseudo-code of the new algorithm is given in Algorithm 11. An identical simulation as the one compared for Algorithm 10 was carried out for Algorithm 11. The total energies per time for each are plotted in Figure 7.7b. The energy is clearly conserved for the gaseous, liquid and high density state.

Algorithm 11: None shifted reflective plane algorithm.

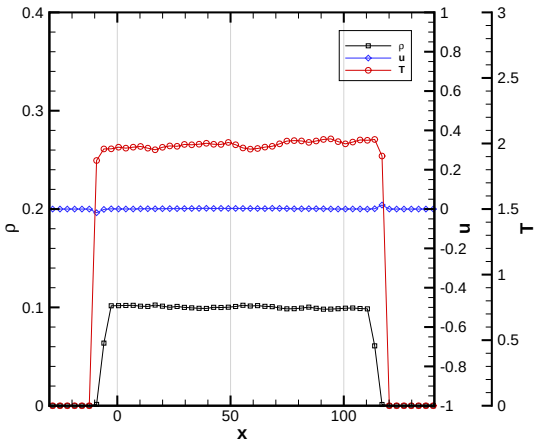
```
foreach atom i do
    d = (r_p − r_i) · n;
    if d < 0 then
        v_i ← f(v_i + 2n(u − v_i · n));
```

7.7.1 Static Case

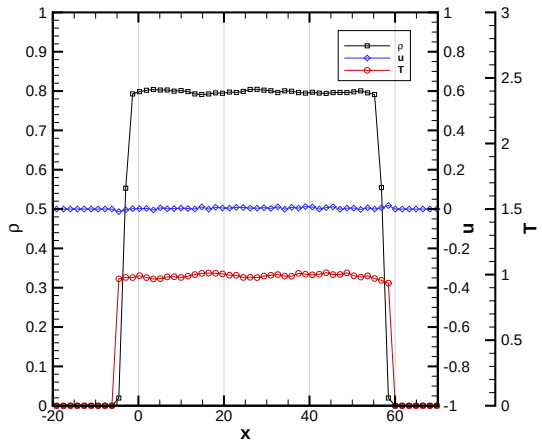
Using the (DC-MTRP) flux boundary imposing scheme for the static case, the momentum fluxes, which for that case only arise from the static pressure, are transferred automatically through the presence of the reflective plane. Thus, the momentum flux does not need to be measured or calculated in advance for the chosen state. For other schemes, the precise knowledge of the equation of state for the pressure, $p(\rho, e_i)$, to apply the correct equilibrium momentum flux across the boundary is necessary. The pressure on the interface can be measured by using Equation 3.2.14 and has already been used for obtaining the results in Section 3.4.

The static case was tested for gaseous (MDS-G: $\rho = 0.1$, $T = 2$), liquid (MDS-L: $\rho = 0.8$, $T = 1$) and solid (MDS-S: $\rho = 1.2$, $T = 1$) states. The setup, preparation and investigation procedures were described in Section 6.1. Not unexpectedly, all cases gave stable results with the correct quantities for density, velocity and temperature in total as well as for the profiles across the x dimension. For reasons of completeness, the profiles have been plotted for the three considered states in the Figures 7.8a – 7.8c. The plots have been averaged over $N_m = 10,000$ measured every $\tau_{mes} = 10$ time steps.

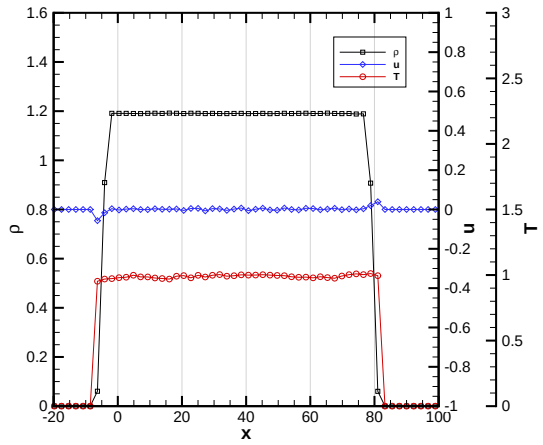
Through its nature, the reflective plane gives a very sharp transition of the density from the respective value to zero, on the boundary. This is the ideal situation; it will later be



(a) Gaseous state ($\rho = 0.1, T = 2$), MDS-G system



(b) Liquid state ($\rho = 0.8, T = 1$), MDS-L system



(c) Solid state ($\rho = 1.2, T = 1$), MDS-S system

Figure 7.8: Profiles of density, velocity and temperature across the x dimension for the different states.

shown that for other methods of momentum transfer the transition is more gradual and, therefore, involves a zone where density and material properties change.

The small discrepancies in velocity on the boundaries, (negative on the western side and positive on the eastern side) are only a measuring artefact. It arises from atoms that enter the measuring cell only briefly and the whose contribution is mostly negative or positive, depending on the side. The very low density magnifies the contribution of single atoms enormously.

7.7.2 Flow parallel to the boundary

The Couette flow test case, as described in Section 6.2, was investigated for the molecular systems MDCF-G, MDCF-L1 and MDCF-L2. The dimensions of the test cases are described in Table B.1 (Appendix B). Different velocities, u_e, at the interface were considered. The selected combination of test cases and velocities are listed in Table B.2 (Appendix B). It also contains the expected temperatures at the interface.

Like in the static test case, the flux of momentum perpendicular to the surface (ρv) is automatically transferred through the reflective plane. The parallel fraction of the momentum flux can be transferred by the MTF, MTDTV or MTRV scheme. Here, the MTF scheme was selected. It is discussed in detail in the next section, so that at this point only a few important parameters will be given (Their meanings are described in the next section). The external force was applied using a weighting function $g(\boldsymbol{r}, \boldsymbol{v}) = 1$ and an interaction depth of $NP_b = 1$.

The most important simulation parameters are given in Table 7.2. For the liquid case, the equilibration was performed in two stages. In the first equilibration run over 10,000 time steps, the temperature control was applied to the entire system for obtaining $T = 1$. A second equilibration run, over 30,000 time steps, was performed this kind of temperature control only applied to the wall atoms. It should be noted that temperature control applied directly on the wall during the simulation is artificial. In reality the increased wall temperature would cause a heat flux which would keep the temperature of the wall nearly constant, provided the heat is transported away fast enough. A fully realistic simulation has to model these effects, but for the purpose of investigating MD boundary conditions, a constant wall temperature is sufficiently accurate. The simulations had to be run for a long time to allow the temperature profile to develop. It was considered to have reached an equilibrium situation when the last three calculated profiles did not show significant differences.

Parameter	**System**	
	MDCF-G	MDCF-L1 / MDCF-L2
Momentum flux by stress (perpendicular)	MTRP	
Momentum flux by stress (parallel)	MTF	
NP_b	1.0	
Crossed atoms	reflect	
Equilibrium time steps	10,000	40,000
Simulation time steps	4,000,000	400,000
Measuring frequency (τ_m)	100	100
Averaged over (N_m)	10 000	1,000

Table 7.2: Parameter settings for the DC-MTRP Couette flow test cases.

For obtaining the minimum number of time steps for sufficient averaging, the following assumptions have been made. For the gaseous case at the slowest velocity of $u_e = 1$, the temperature range of interest is ≈ 0.1. To obtain a standard deviation of less than 5 %

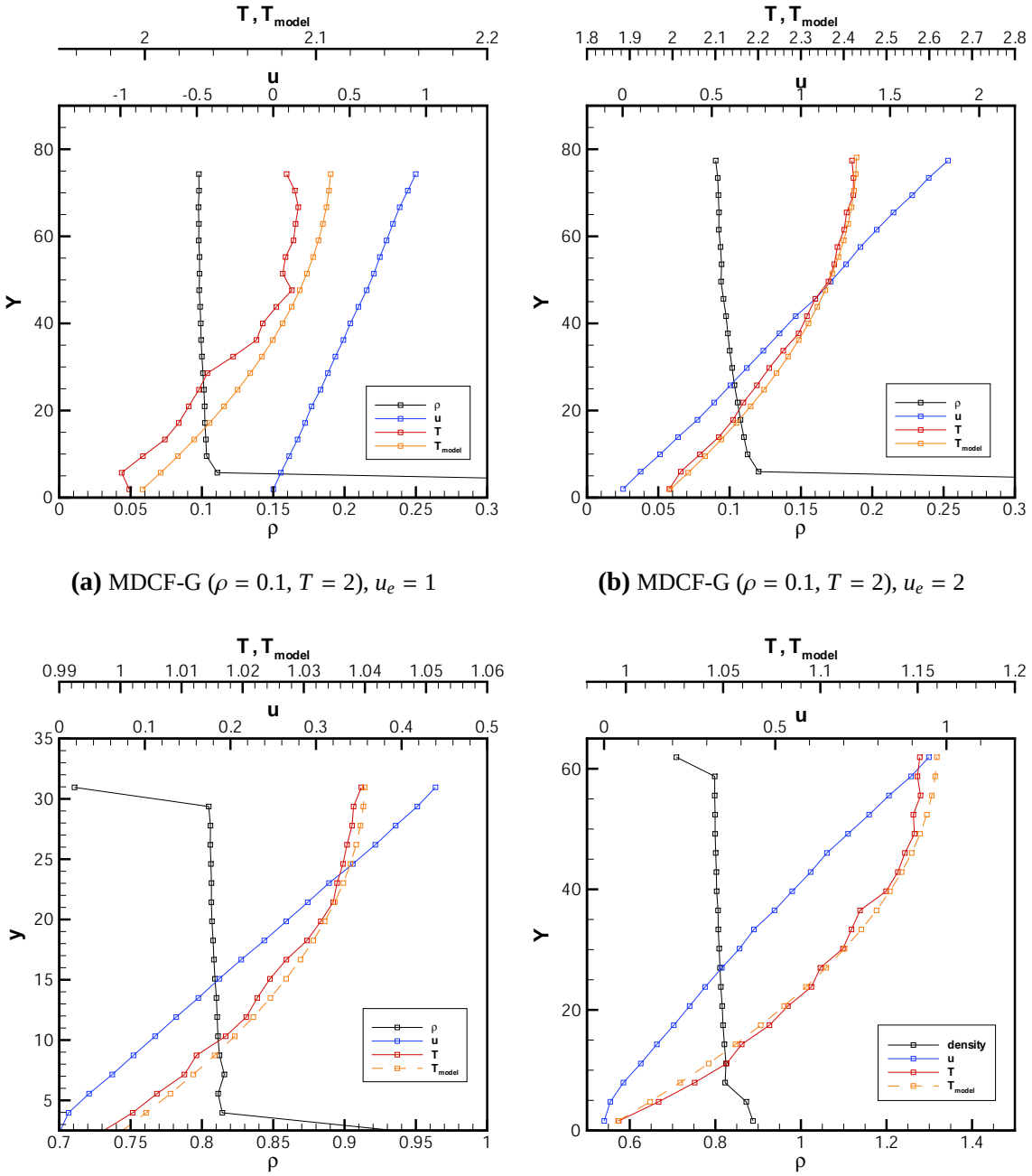
relative to this interval, one needs to keep the fluctuations relative to $T = 2$ below 0.003 ($0.1 \cdot 0.05/2$). For this level, Equation (3.3.25) gives a minimum number of samples of $N^T_{m,min} \approx 1200$. With an estimated correlation time of 600 time steps, the minimum number of time steps is approximately 72,000. The number of simulated time steps that results from the chosen combination of $\tau_m = 100$ and $N_m = 10,000$ is larger; therefore, the size of the fluctuations is expected to be below the desired level. The fluctuations of density and velocity are below the one of the temperature and since all quantities are evaluated together their fluctuations are lower than the one for the temperature.

For the liquid cases, the number of samples from which the average needs to be chosen to limit the fluctuations below 5 % of the considered ranges of 0.8 for the density, 0.5 for the velocity and 0.1 for the temperature. Again, the calculation of the temperature requires the highest number of values to average: $N_{m,min} \approx 250$. Therefore, the chosen combination of $\tau_m = 100$ and $N_m = 1000$ is sufficient to fulfill these conditions. For all following simulations, the minimum number of samples needed to calculate the macroscopic quantities have been obtained in the same manner. For reasons of brevity, the precise figures are not given for every single case.

The Figures 7.9a – 7.9d show the profiles of density, velocity and temperature in y direction for the Couette flow cases. As expected, the simulations were stable for all tested cases and the results obtained correspond to the analytical solution. In all cases, the density is nearly constant. The slight decrease towards the interface is because of the higher temperature away from the wall. The velocity is, in all cases, linear and at the upper end (y_e) very close to the expected velocity, u_e. The slight discrepancies are causes by several contributions. The error in the measured value for viscosity causes a relative error of $u_{eerr} \approx 5\,\%$ in u_e [25, 131]. Also, the finite size of the measuring cells ($\delta l = 3.2$) does not allow a perfectly exact solution. The largest contribution, however, can be expected from the variation in temperature and density. These in turn cause changes in the viscosity and, hence, influence the velocity at the HSI. The analytical model, as described before, has been plotted as orange dashed line in all figures. Clearly, the simulation results match the theoretical temperature profiles very well apart from the gaseous test case with $u_e = 1$. There, the measured temperatures are lower than the prediction from the model. The error in the temperature profile have similar reasons to those discussed regarding velocity. However, considering that the reported error in the thermal conductivity is 40 % [137], the coincide of the other cases with the theoretical temperature profile is actually surprising. Also, the temperature profile depends on the viscosity and thermal conductivity. It is therefore subjected to a greater extent to changes in density and temperature. Considering these facts, the results can be considered in good agreement with the analytical model.

7.7.3 Flow perpendicular to the boundary

As mentioned before, the flow across the boundary poses the most difficult situation for the MD flux boundary conditions since it requires the insertion and removal of atoms into and out of the MD system. For the insertion of new atoms on the inlet, the USHER algorithm was used in conjunction with the ZNM scheme for generating the velocities of the new atoms. The number of boundary atoms was set to 100, which is less important because the



(a) MDCF-G ($\rho = 0.1$, $T = 2$), $u_e = 1$

(b) MDCF-G ($\rho = 0.1$, $T = 2$), $u_e = 2$

(c) MDCF-L ($\rho = 0.8$, $T = 1$), $u_e = 0.5$

(d) MDCF-L ($\rho = 0.8$, $T = 1$), $u_e = 1$

Figure 7.9: Profiles for density, velocity and temperature of the Couette flow using the reflective plane scheme for gaseous (upper figures) and liquid state (lower figures). T_{model} is the analytical solution for the temperature profile.

only fluxes imposed on the boundary atoms in the reservoir are F_{es} (energy flux by stress). The results show that these fluxes are of no importance for the cases where the DC-MTRP scheme is working. For all cases, where these fluxes play a significant role, the DC-MTRP scheme was unstable for any tested configuration with or without imposition of energy flux

by stress. For removing atoms on the outlet, it made no difference, which atom selection scheme was used. The same applies to any FCCS scheme for correcting the convective momentum and/or energy flux. Hence, no energy correction scheme was applied. A summary for the employed parameters is given in Table 7.3.

Parameter	System	
	MDFF-G	MDFF-L
Momentum flux by stress (perpendicular)	MTRP	
insert atoms	USHER	
velocities of inserted atoms	ZNM	
NP_b	1.0	
Crossed atoms	reflect	
Equilibrium time steps	25,000	25,000
Simulation time steps	200,000	200,000
Measuring frequency (τ_m)	100	100
Averaged over (N_m)	100	100

Table 7.3: Parameter settings for the DC-MTRP free flow test cases.

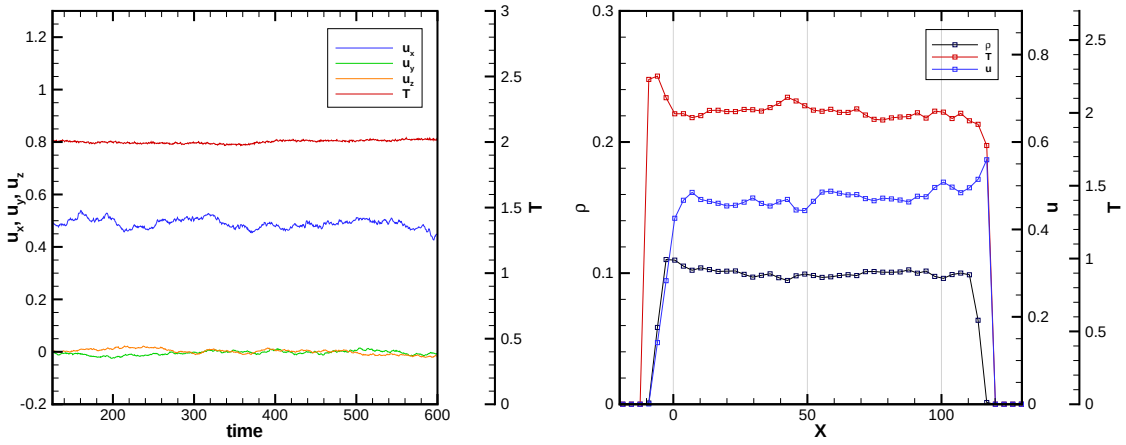
The DC-MTRP scheme was tested for the free flow test case for the gaseous state (system MDFF-G1, $n = 0.1$, $T = 2$) with the velocities $u_x = \{0.5, 1.0, 2.0, 3.0\}$ and for the liquid state (system MDFF-L1, $n = 0.8$, $T = 1$) with the velocities $u_x = \{0.1, 0.25, 0.5, 1.0\}$. The simulations were stable for $u_x \leq 1.0$ in the gaseous state and for $u_x \leq 0.25$ in the liquid case. For higher velocities, the simulations led to physically incorrect conditions causing the breakdown of the flux boundary conditions. All simulation were started with the system at the respective velocity. This is easily achieved through an equilibration run of the system at rest with subsequently adding the respective velocity $\boldsymbol{u} = (u_x, 0, 0)$ to the velocity of every atom in the system.

The analysis is started with the gaseous state at the velocity $u_x = 0.5$. On the left side of Figure 7.10, the overall velocities and temperature of the system are plotted over time. On the right side, the profiles in x dimension are shown for density, velocity in flow direction, u_x, and temperature. As shown, the overall velocity u_x stays at the correct level. The velocity profile shows small deviation on the in- and outlets. The same applies for the temperature and density. While on the inlet, the density and temperature are slightly too high and the velocity is too low, the situation is reversed on the outlet. In between the profiles are flat and at the correct levels. Because the DC-MTRP scheme is a direct convection scheme, i.e. the number of inserted atoms is equal to the number of removed atoms, the overall density matches at all times exactly at the correct value of $\rho = 0.1$.

For a higher velocity of $u_x = 1.0$ the graphs are shown in Figure 7.11. Here, the situation is similar with an amplified distortion in the profiles. On the inlet, the temperature is significantly too high and velocity is too low; the density is slightly increased also. The overall velocity and temperature are within an acceptable range.

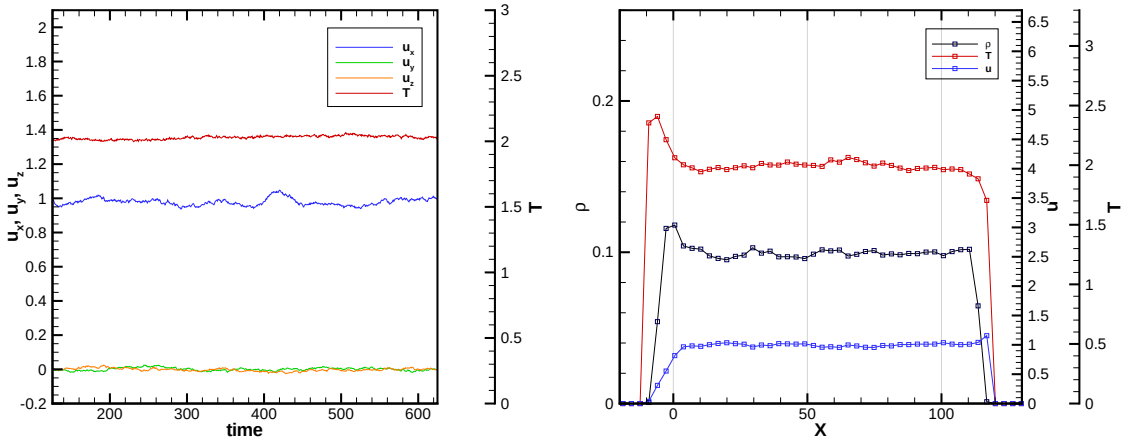
For the velocities of $u_x = 2.0$, the DC-MTRP scheme was not stable. This can be seen

on the left hand side of Figure 7.12, where the temperature starts to increase and fluctuate rapidly. Also the profiles on the right side show severe distortions. The profiles have been taken shortly before the breakdown of the flux boundary conditions. It was not possible to achieve a stable simulation with any of the tested parameter combinations for the DC-MTRP flux boundary conditions.



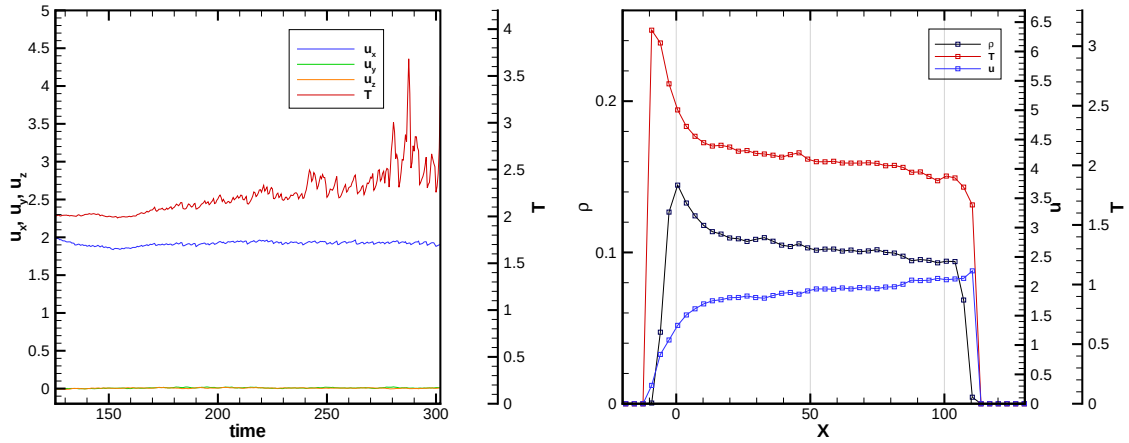
(a) Density, velocity and temperature over time. **(b)** Density, velocity and temperature profiles.

Figure 7.10: Free flow test case of the gaseous system (MDFF-G1, $n = 0.1$, $T = 2$) at $u_x = 0.5$ with the DC-MTRP scheme for flux boundary conditions.



(a) Density, velocity and temperature over time. **(b)** Density, velocity and temperature profiles.

Figure 7.11: Free flow test case of the gaseous system (MDFF-G1, $n = 0.1$, $T = 2$) at $u_x = 1.0$ with the DC-MTRP scheme for flux boundary conditions.



(a) Density, velocity and temperature over time. **(b)** Density, velocity and temperature profiles.

Figure 7.12: Free flow test case of the gaseous system (MDFF-G1, $n = 0.1$, $T = 2$) at $u_x = 2.0$ with the DC-MTRP scheme for flux boundary conditions.

For the liquid case, the situation is similar to the gaseous one. At the relatively low velocity of $u_x = 0.1$, the flux boundary conditions (using the DC-MTRP scheme) generate a stable simulation. Figure 7.13 shows that the systems velocity and temperature over time as well as the profiles of density, velocity and temperature match the physically expected levels. However, minor oscillations of the overall velocity, u_x, are observable. These oscillations must be accounted for waves, which are travelling in x direction through the system being reflected on the walls. For $u_x = 0.1$ the oscillations are reasonable damped and a stable simulation is possible.

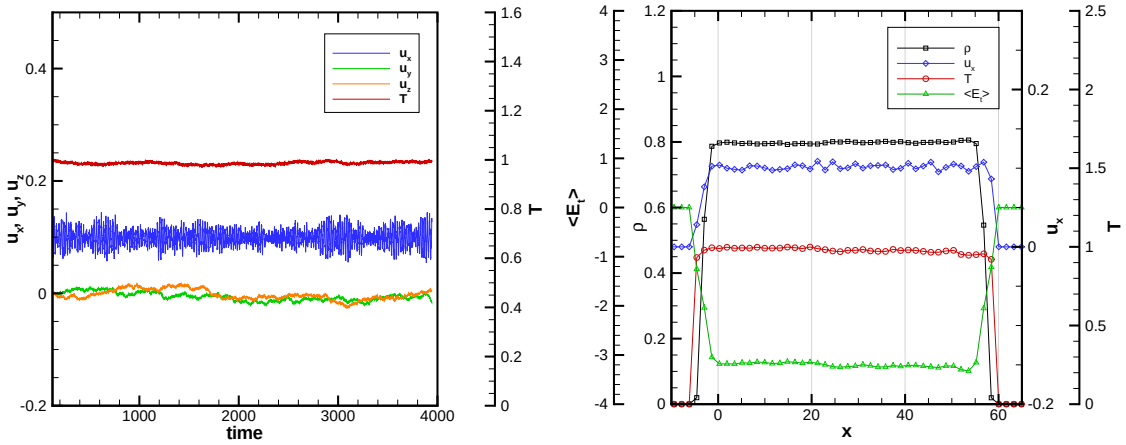
For the higher velocity of $u_x = 0.25$, one can observe that the waves overlaying the flow in x direction are stronger than for the previous case with lower velocity. The stronger oscillations herald the unstable regime for even higher values of u_x. However, the profiles in Figure 7.14b still show good results with a plateau across the entire domain.

Increasing u_x to 0.5 leads the simulation into the unstable regime where the oscillations in velocity, caused by the DC-MTRP flux boundary conditions, are unstable. This can be seen in Figure 7.15a, where the velocity oscillation grows until the flux boundary conditions break down. Figure 7.15b on the right side shows the profiles shortly before.

7.7.4 Summary of results of the DC-MTRP

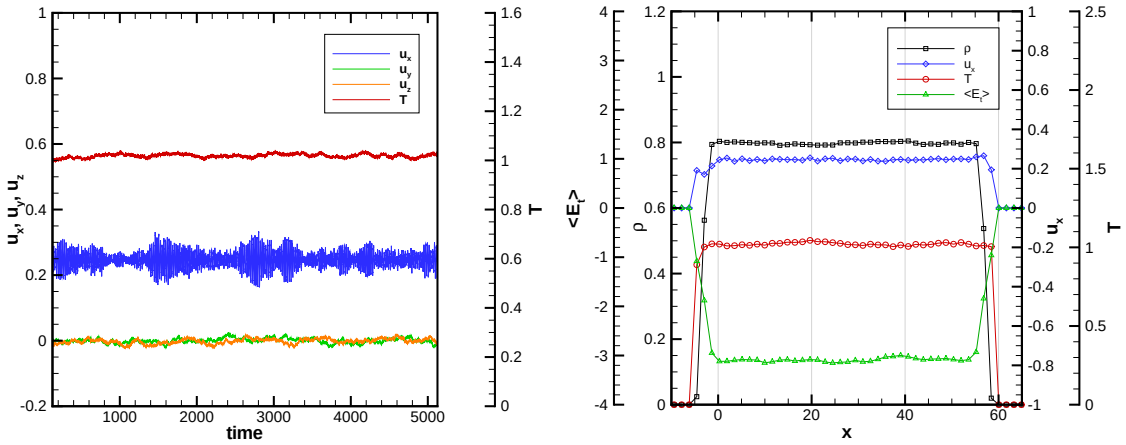
From the performed test cases, a number of advantages and disadvantages of imposing MD flux boundary conditions by the direct convective flux and reflective plane scheme (DC-MTRP) can be deduced. These apply especially for the use of the reflective plane. The advantages are:

- No prior knowledge of the exact equation of states (Equation (2.3.13f)) are required.
- The density profile on the interface is very sharp.



(a) Density, velocity and temperature over time. **(b)** Density, velocity and temperature profiles.

Figure 7.13: Free flow test case of the liquid system (MDFF-L1, $n = 0.8$, $T = 1$) at $u_x = 0.1$ with the DC-MTRP scheme for flux boundary conditions.



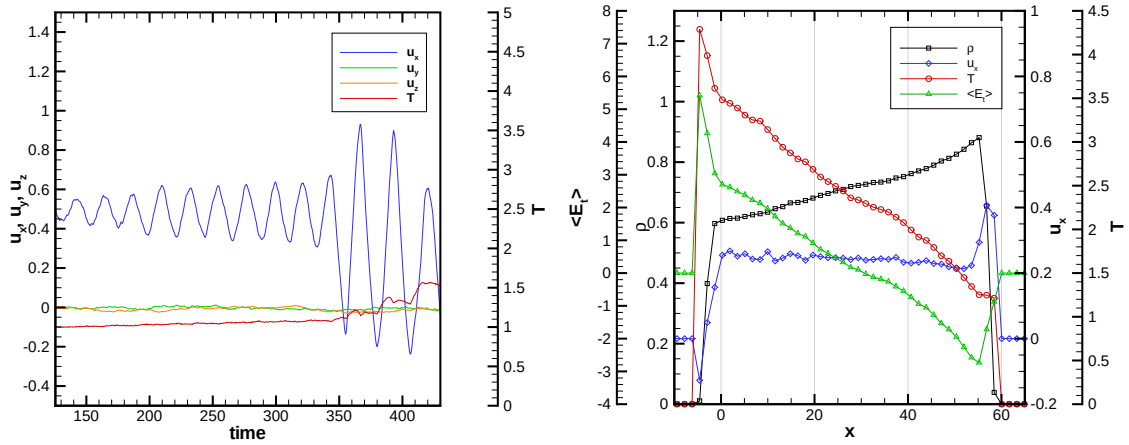
(a) Density, velocity and temperature over time. **(b)** Density, velocity and temperature profiles.

Figure 7.14: Free flow test case of the liquid system (MDFF-L1, $n = 0.8$, $T = 1$) at $u_x = 0.25$ with the DC-MTRP scheme for flux boundary conditions.

- The scheme is simply designed and easy to implement into an MD-solver.

The disadvantages of the applying the static pressure via a reflective plane are:

- The reflective plane does not actively choose how many atoms are reflected. Thus, the transferred momentum and energy depends on the number of atoms that actually try to cross the plane.
- The scheme is completely inflexible and deviations of the molecular mass and mo-



(a) Velocities and temperature over time.

(b) Density, velocity and temperature profiles.

Figure 7.15: Free flow test case of the liquid system (MDFF-L1, $n = 0.8$, $T = 1$) at $u_x = 0.5$ with the DC-MTRP scheme for flux boundary conditions.

mentum fluxes from the expected values lead to physically incorrect solutions that cause the breakdown of the flux boundary conditions.

- The reflective plane schemes cause the generation of waves travelling through the system. While these waves seem to be damped for lower velocities, they appear to grow for higher velocities and cause the MD flux boundary conditions to break down.

To summarise, the performed test cases show that the (DC-MTRP) flux imposing scheme is suitable for the simulation of static cases, flows parallel to the boundary and for slow flows across the boundary, i.e. where the external kinetic energies are insignificant compared to the internal kinetic energies. In the case of the tested LJ-system the scheme was only stable for low convective momentum fluxes across the boundary, i.e. $\rho \boldsymbol{un} < 0.1$. For gaseous Argon ($\rho = 0.1$, $T = 2$) this value corresponds to a velocity of approximately 157 m/s. In the liquid state ($\rho = 0.8$, $T = 1$), the respective velocity is 20 m/s. In terms of micro- and nanoscale flows these velocities are still high and, in many cases, the velocities of interest will be much lower. For these, the DC-MTRP flux imposing scheme is an attractive scheme because of its straightforward implementation.

7.8 Direct convective flux and momentum transfer by force (DC-MTF)

The combination discussed in this section is the direct convective flux imposition and the MTF momentum transfer scheme, which imposes the fluxes due to stress by an external force. The alternative and very similar MTDVC scheme (momentum transfer through direct velocity change) is treated as well.

7.8.1 The external force

To the knowledge of the author, the usage of an external force, $\boldsymbol{F}^{ext}$, has been the means to impose the momentum and energy fluxes due to stress for all previous flux imposition schemes. The external force is applied onto the N_b outermost boundary atoms in the reservoir and related to the momentum flux by stress through

$$\boldsymbol{F}_{ps} = \frac{1}{g_t} \sum_{i=1}^{N_b} g(\boldsymbol{r}_i, \boldsymbol{v}_i) \boldsymbol{F}^{ext} \quad \text{or} \tag{7.8.1}$$

$$\frac{\boldsymbol{F}_{ps}^{int}}{\delta t} = \frac{1}{g_t} \sum_{i=1}^{N_b} g(\boldsymbol{r}_i, \boldsymbol{v}_i) \boldsymbol{F}^{ext} \quad \text{with} \tag{7.8.2}$$

$$g_t = \sum_{i=1}^{N_b} g(\boldsymbol{r}_i, \boldsymbol{v}_i) \ , \tag{7.8.3}$$

where the weighting function, $g(\boldsymbol{r}_i, \boldsymbol{v}_i)$, distributes the total force, $\boldsymbol{F}^{ext}$, onto the N_b boundary atoms. In the first equation (Equation 7.8.1), the external force is related directly to the momentum flux, $\boldsymbol{F}_{ps}$. The alternative, Equation (7.8.2), relates the external force to the integrated momentum flux in order to impose the full value of $\boldsymbol{F}_{ps}^{int}$ in one time step δt. The advantage of the latter version is that it allows additional momenta to be applied. For example, it is used for the coupling of integrated fluxes by convection and by stress (Section 7.4.1). Thus Equation (7.8.2) has been used in all test cases. Applying the external force automatically introduces the energy flux $\boldsymbol{F}_{es}$ through

$$F_{es} = \frac{1}{g_t} \sum_{i=1}^{N_b} g(\boldsymbol{r}_i, \boldsymbol{v}_i) \ \boldsymbol{F}^{ext} \boldsymbol{v}_i \tag{7.8.4}$$

or for individual dimensions α

$$F_{es\alpha} = \frac{1}{g_t} \sum_{i=1}^{N_b} g(\boldsymbol{r}_i, \boldsymbol{v}_i) \ F_{\alpha}^{ext} v_{i\alpha} \ , \tag{7.8.5}$$

where F_{α}^{ext} is the force, $v_{i\alpha}$ the velocity and $F_{es\alpha}$ the energy flux in dimension α.

Different weighting functions have been proposed, however it is difficult to find one that satisfies both Equations (7.8.2) and (7.8.4). Delgado-Buscalioni and Coveney [40] pointed

out that this is a formidable task and that a solution is likely to by computationally very demanding. They suggested using the simplest function, $g(\boldsymbol{r}_i, \boldsymbol{v}_i) = 1$, because it automatically transfers the correct energy fluxes, $F^{es} = \boldsymbol{\Pi} \cdot \boldsymbol{u}$, if the velocity of the boundary atoms ($\boldsymbol{v}_b = (1/M) \sum_i^{N_{bf}} m_i \boldsymbol{v}_i$ with $M = \sum_i^{N_{bf}} m_i$) is equal to $\boldsymbol{u}$. In this work, this route was followed and the weighting function $g(\boldsymbol{r}_i, \boldsymbol{v}_i) = 1$ was used in all cases.

When using the simple weighting function $g(\boldsymbol{r}_i, \boldsymbol{v}_i) = 1$, only the parameter on how many atoms the external force $\boldsymbol{F}^{ext}$ is acting on needs to be chosen. Initially, one may be tempted to simply apply $\boldsymbol{F}^{ext}$ onto all N_b atoms within the reservoir. However, this can lead to situations where the external force is distributed to a such a large number of atoms, that the force felt by the outermost atoms is so weak that almost no pressure is exerted on them and these atoms literally evaporate into the vacuum beyond the boundary.

Even if the described scenario does not occur, the number of boundary atoms onto which the external force is applied determines the characteristics of the material in the boundary zone. This is because the external force does not produce a sharp discontinuous transition in density on the boundary. Rather, it creates a relaxation zone in which the density slowly falls from its bulk value to zero. For $g(\boldsymbol{r}_i, \boldsymbol{v}_i) = 1$, the number of atoms onto which the external force is applied is the only parameter to influence the characteristic of the boundary zone. This relationship is investigated in detail in the subsequent sections.

For the direct convection (DC) scheme the number of inserted and removed atoms corresponds strictly to the convective fluxes. For that reason, the number of atoms in the relaxation zone is not constant but varies and is subject to fluctuations. It is, therefore, not possible to choose the size of the relaxation zone in a way that gives exactly the desired number of N_b boundary atoms. Rather, the external force can only be exerted onto the outermost boundary atoms. In the course of the text these will be still be referred to as N_b boundary atoms, even if there can be more atoms within the relaxation zone, i.e. beyond the boundary. By using the sorted boundary atom list, these atoms are easily identified.

The external force is integrated into the MD-code by simply adding it to force on the respective atoms after the usual interatomic forces have been calculated (q.v. flow chart in Figure 7.2). The momentum and energy transfer takes place automatically through the normal MD time integration algorithm.

7.8.2 Direct velocity change (MTDVC)

Alternatively, the momentum fluxes by stress can also be imposed by changing the velocity vectors directly. The advantage of this way of imposing the momentum flux, is that it is based on the integrated momentum fluxes F_{ps}^{int}. Using integrated fluxes makes it easier to include any additional fluxes, which may need to be added or subtracted through an employed scheme for coupling the convective fluxes and fluxes by stress (q.v. FCCS in Section 5.3). When these corrections are simply added to the integrated fluxes their imposition is done automatically.

For MTDVC, the integrated momentum fluxes, F_{ps}^{int}, are transferred every time step by

adding $\Delta \boldsymbol{v}$ to the N_b boundary atoms:

$$\boldsymbol{v}_i^{'} = \boldsymbol{v}_i + \Delta \boldsymbol{v} \quad \text{with} \quad \Delta \boldsymbol{v} = \frac{F_{ps}^{int}}{N_{bf}} \,, \tag{7.8.6}$$

where $\boldsymbol{v}_i^{'}$ are the new velocities. The transferred energy $\boldsymbol{E}_{es}$ can be calculated precisely as

$$E_{es} = \sum_{i=1}^{N_b} \frac{1}{2} m_i \left(\boldsymbol{v}_i^{'2} - \boldsymbol{v}_i^{2} \right) = \sum_{i=1}^{N_b} m_i \boldsymbol{v}_i \Delta \boldsymbol{v} + \frac{1}{2} \sum_{i=1}^{N_b} \Delta \boldsymbol{v}^2 \tag{7.8.7}$$

or for individual dimensions α

$$E_{es\alpha} = \sum_{i=1}^{N_b} m_i v_{i\alpha} \Delta v_\alpha + \frac{1}{2} \sum_{i=1}^{N_b} \Delta v_\alpha^2 \,. \tag{7.8.8}$$

Similar to the usage of the external force, the transferred energy E_{es} corresponds to the prescribed energy transfer of $F_{es}^{int} = \Pi \cdot \boldsymbol{u} \delta t$ under the condition that $\sum_{i=1}^{N_{bf}} m_i v_i = \sum_{i=1}^{N_{bf}} m_i \, \boldsymbol{u}$. However, additional contributions to the integrated energy mean that the integrated energy fluxes may not correspond necessarily to $\Pi \cdot \boldsymbol{u} \delta t$. Thus, after the integrated momentum flux F_{ps}^{int} has been transferred by Equation (7.8.6), the transferred energy is subtracted from the integrated energy fluxes, F_{es}^{int}. In addition, the remaining integrated energy in F_{es}^{int} is transferred to the boundary atoms using Algorithm 3.

The complete algorithm for the transfer of the integrated momentum and energy fluxes by stress (F_{ps}^{int}, F_{es}^{int}) through direct velocity change (MTDVC) is given in Algorithm 12. In the first part, the integrated momentum flux is transferred and in the second part the integrated energy flux is transferred. The algorithm is executed before the first step of the MD time integrator (q.v. flow chart in Figure 7.2).

7.8.3 Relaxation profile for static case

Any design of the MD flux boundary conditions, which does not employ solid or artificial walls, will have a relaxation zone. In this zone the density relaxes from the value inside the molecular zone down to zero. The characteristic of the relaxation zone depends on the transfer scheme of the momentum and energy fluxes by stress to the boundary atoms. When constructing the HSI, it is important to know the characteristics of the relaxation zone, particulary its width which puts constraints on the selection of cell sizes on the HSI. For this reason the density profile across the relaxation zone, specifically the width and gradient, have been investigated. When using an external force or direct velocity change as transfer scheme in conjunction with the weighting function $g(\boldsymbol{r}_i, \boldsymbol{v}_i) = 1$, these characteristics depend on:

a) the type of the fluid and specific properties, for example, the molecular size, surface tension, etc.;

Algorithm 12: Imposition of fluxes by stress through MTDVC

$\Delta \boldsymbol{v} \leftarrow \boldsymbol{F}_{ps}^{int}/N_{bf}$;
$M \leftarrow 0$;
$\boldsymbol{p}_b \leftarrow \boldsymbol{0}$;
$\boldsymbol{E}_k \leftarrow \boldsymbol{0}$;

Impose momentum flux by direct velocity change;
foreach *boundary atom i* **do**
  $\boldsymbol{v}_i \leftarrow \boldsymbol{v}_i + \Delta \boldsymbol{v}$;
  $M \leftarrow M + m_i$;
  $\boldsymbol{p}_b \leftarrow \boldsymbol{p}_b + m_i \boldsymbol{v}_i$;
  for $\alpha = \{x, y, z\}$ **do**
    $F_{es\,\alpha}^{int} \leftarrow F_{es\,\alpha}^{int} + m_i v_{ia} lpha \Delta v_\alpha$;
    $E_{k\alpha} \leftarrow E_{k\alpha} + (1/2) m_i v_{i\alpha}^2$;
$\boldsymbol{F}_{ps}^{int} \leftarrow \boldsymbol{0}$;
$\boldsymbol{v}_b \leftarrow \boldsymbol{p}_b/M$;
for $\alpha = \{x, y, z\}$ **do**
  $F_{es\,\alpha}^{int} \leftarrow F_{es\,\alpha}^{int} + (1/2) M \Delta v_\alpha^2$;
  $E_{i,k_\alpha} \leftarrow E_{k\alpha} - (1/2) M v_{b\alpha}^2$;

Impose energy flux by scaling internal velocities;
if $-\boldsymbol{E}_{i,k} < \boldsymbol{F}_{es}^{int}$ **then**
  Output error message that not enough internal energy is available;
  return;
for $\alpha = x, y, z$ **do**
  $f_\alpha = \sqrt{1 - \boldsymbol{F}_{es\,\alpha}^{int}/E_{i,k_\alpha}}$;
  $c_\alpha = v_{b\alpha}(f_\alpha - 1)$;
foreach *boundary atom i* **do**
  $v_{ix} \leftarrow v_{ix} f_x + c_x$;
  $v_{iy} \leftarrow v_{iy} f_y + c_y$;
  $v_{iz} \leftarrow v_{iz} f_z + c_z$;

b) the state of the material, for example, the density and temperature at the boundary;

c) the number of atoms onto which the momentum is transferred;

This work focuses on the dependencies b) and c). The dependency a) will of course depend on the specific material and needs to be analysed for each material. Here the general Lennard-Jones potential has been employed and it is reasonable to expect that discovered characteristics also hold true for other simple fluids.

An illustration of the relaxation zone is displayed in Figure 7.16. The static pressure that keeps the atoms inside the molecular domain is realised by the external force $\boldsymbol{F}^{ext}$ that transfers the momentum flux to the right. The width or thickness of the relaxation zone is designated by δl_b. As indicated on density profile at the top of the figure, the density

function across the zone shows an asymptotic behaviour, meaning that the point where it reaches full density and zero density cannot be determined precisely. For this reason a criteria was employed that is also used for the definition of the thickness of shockwaves. A threshold value of 5% defines the thickness of the density transition zone from the lower value to the upper value. Hence, per definition, the relaxation zone ranges from $n = 0.05n_0$ to $n = 0.95n_0$, where n_0 is the number density inside of the bulk material.

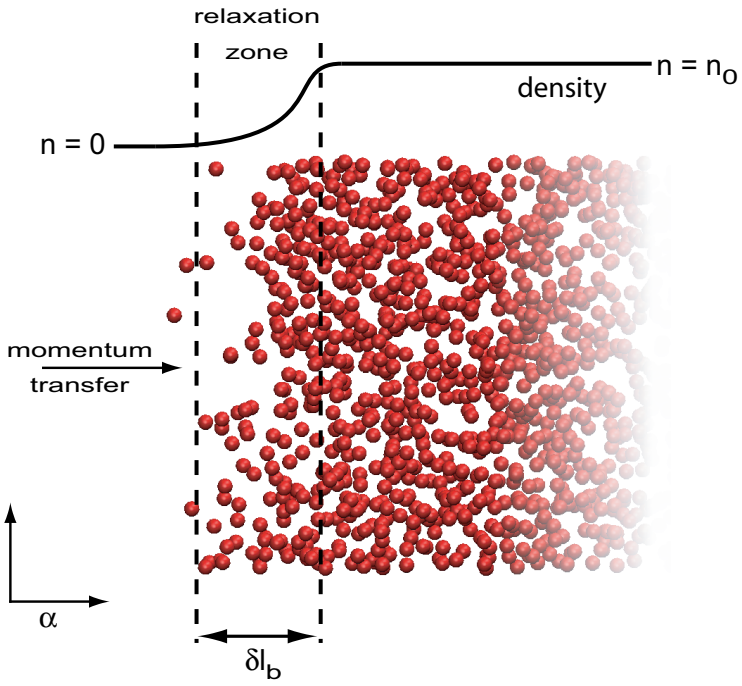


Figure 7.16: Illustration of the relaxation zone.

The number of atoms onto which the momentum is transferred is designated by N_b. Relating this value to the surface area of the boundary, A_b, one obtains the number of atoms per surface area:

$$n_{bA} = \frac{N_b}{A_b} \,. \tag{7.8.9}$$

Assuming a discontinuous density profile ($n = 0$ for $x < x_b$, $n = n_0$ for $x > x_b$) for a single atomic material, the interaction depth can be defined as

$$id_b = \frac{n_{bA}}{n_0} \,. \tag{7.8.10}$$

Imagining that the atoms are arranged in a crystal structure, the interaction depth can also be expressed in atomic layers (atomic planes):

$$NP_b = \frac{n_{bA}}{(n_0)^{2/3}} \,, \tag{7.8.11}$$

$$NP_b = id_b \, (n_0)^{1/3} \,, \tag{7.8.12}$$

where NP_b is the interaction depth in atomic layers.

Another way of characterising the density profile across the relaxation zone is to use the normalised inverse maximum gradient, img_b, which is obtained by relating the number density to the maximum gradient:

$$img_b = \frac{n_0}{\max\left(\frac{\mathrm{d}n}{\mathrm{d}\alpha}\right)} , \tag{7.8.13}$$

with α being the distance perpendicular to the interface.

For low density gaseous mediums, one can derive a theoretical expression for the relaxation zone width by balancing the kinetic energy of the atoms in one dimension with the energy field that is generated by the external force F^{ext}. The Boltzmann distribution [198],

$$f(E) = Ae^{-\frac{E}{k_B T}} , \tag{7.8.14}$$

gives the probability for an atom to have the energy E in one degree of freedom, which, for the purpose here, is the velocity in x direction (perpendicular to the boundary). The factor A normalises the integral of Equation (7.8.14) to one.

The external force generates a potential energy field within the relaxation zone. In the normal direction to the boundary surface (here the x direction) the potential energy is $E_p(x) = (F_x^{ext}/N_b)x$, where x is the distance from the boundary (located at x_b) and F_x^{ext} is the component of the external force, $\boldsymbol{F}^{ext}$, acting in x direction. The distance an atom will travel against the external force depends on its kinetic energy in x direction. Thus the potential energy can be inserted into Equation (7.8.14) to obtain the density distribution within the relaxation zone:

$$n_{model}(E) = Ae^{-\frac{F^{ext}}{N_b k_B T}} , \tag{7.8.15}$$

The normalisation factor A is determined by the condition that at $x = 0$ the number density is $n_{model}(x) = 0.95n_0$ (This point is defined above as start point of the relaxation zone). Since for static cases, F_x^{ext} arises purely from the static pressure it can be replaced by $F_x^{ext} = PA_b$. The density profile on the boundary can then be written as

$$n_{model}(E) = 0.95n_0\, e^{-\frac{PA_b}{N_b k_B T}} , \tag{7.8.16}$$

From Equation (7.8.15), the width of the relaxation zone can be derived:

$$\delta l_b = \frac{NP_b\, n_0^{\frac{2}{3}}\, k_B T}{P} \ln\frac{0.05}{0.95} , \tag{7.8.17}$$

which is in the given form based on the interaction depth in atomic layers, NP_b. The constant ($\ln(0.05/0.95)$) emerges from above's definition of the relaxation zone width (from $n = 0.05n_0$ to $n = 0.95n_0$).

To investigate the dependency of the relaxation zone width, δl_b, on the interaction depth, a series of simulations were performed to calculate δl_b for the range of interaction depths

($NP_b = \{0.1, \ldots, 10\}$) for both low-density (gaseous state: $n_0 = 0.1$, $T = 2$) and high-density (liquid state: $n_0 = 0.8$, $T = 1$) cases. The temperature dependence of δl_b was investigated by repeating the simulations for the high-density case with a higher temperature (supercritical state: $n_0 = 0.8$, $T = 2$). Finally, the dependence of δl_b on the number density was investigated for $n = \{0.001, \ldots, 1\}$ and $N_b = 100$.

As illustrated in Figure 7.16, the flux boundary conditions were applied from the negative x direction ($\alpha = x$). In positive x direction, the system was confined by a reflecting wall. For all simulations, periodic boundary conditions were applied in y and z direction. The equilibration was performed over $50,000$ time steps with $\delta t = 0.005$. Velocity scaling was applied to settle the temperature at the desired value, i.e. $T = 1$ or $T = 2$. After the density profile at the boundary had stabilised, the density profiles were calculated for each case over 100,000 time steps by using a one-dimensional mesh in x direction. The calculation procedure is explained in Chapter 3; the employed weighting scheme was CIC and the calculations were performed every $\tau_{mes} = 100$ time steps and averaged over $M = 1,000$ values.

In Figure 7.17, the density profiles for the gaseous case, with $n_0 = 0.1$, have been plotted. To achieve a better comparison, the profiles have been shifted so that the density value of $n = 0.95n_0$ is at $x_b = 0$. The measured density profiles fit well to the theoretical prediction of Equation (7.8.16), showing a sharp decrease from the bulk value followed by an asymptotic approach to zero. Obviously for higher N_b, the width of the relaxation zone increases, because the density function drops more slowly.

The corresponding density plots for a liquid with $n_0 = 0.8$ and temperature of $T = 1$ are presented in Figure 7.18. Comparing these with the previous results for a gas, it is obvious that there is no sharp density drop from $n \approx n_0$. Instead, one can see a smooth transition to a linear section, followed by an asymptote to zero. For small N_b, the liquid density profiles shows an oscillatory behaviour near x_b that resembles the density plot of solid materials. In fact, very low interaction depths tend to resemble a reflecting wall and therefore have an 'ordering' effect on the liquid, which imposes a crystal like structure onto the liquid at the boundary. Obviously, Equation (7.8.16) does not model the density profile for the liquid case because it predicts a sharp decrease at x_b instead of the smooth transition observed in Figure 7.18. The origin of the smooth transition lays in the attractive interatomic forces between the individual atoms, which cause energy transfers that are not accounted for in the model. Therefore, the number density profiles predicted by the model are not included in the plot.

For the supercritical state ($n_0 = 0.8$, $T = 2$), the situation is similar to the liquid one. The profiles of number density are plotted in Figure 7.19. Here too, a smooth transition near $x_b = 0$ can be seen for higher values of NP_b and oscillations for small values of NP_b.

One interesting question is: how does the width of the relaxation zone change depending on the interaction depth? From Equation (7.8.17), it is expected that the relaxation zone width δl_b depends linearly on the interaction depth, at least for gaseous states. In Figure 7.20 δl_b is plotted over the interaction depth in atomic layers, NP_b, for different states. The left side shows the complete investigated space and the right side contains a magnified

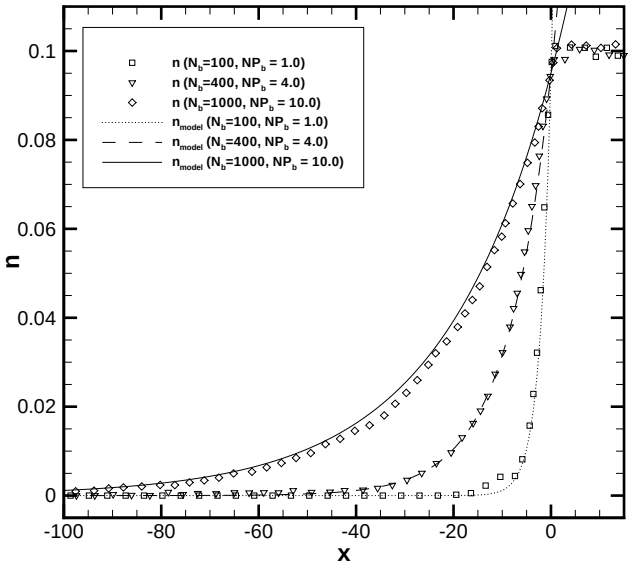


Figure 7.17: Number density profile across the relaxation zone for a set of interaction depths: $NP_b = \{1, 4, 1\}$. The bulk material is in gaseous state with a number density of $n_0 = 0.1$ and temperature of 2. All graphs have been shifted so that $n(0) = 0.95n_0$. The lines are the curves according to the theoretical model.

view of the smaller values of NP_b.

The complete plot shows a good agreement between Equation (7.8.17) and measured relaxation zone widths for the three investigated states. In the gaseous case ($n = 0.1$, $T = 2$), the prediction only fails for very low values of NP_b ($NP_b < 0.5$), where the measured values are higher than predicted by the linear trend (magnified view).

The relaxation widths are also well predicted for the liquid ($n = 0.8$, $T = 1$) and supercritical state ($n = 0.8$, $T = 2$). This is particulary surprising considering that the theoretical density profiles do not correspond with the model at all. The curve for the supercritical state fits very well over the entire investigated range. In the liquid case the measured values are higher than the predicted ones for $NP_b < 6$ and lower for $NP_b < 6$.

Another interesting question is: how does the relaxation zone width change for different densities. Based on Equation (7.8.17), δl_b must be proportional to $n_0^{2/3}/P$. In Figure 7.21 one can see that this is indeed true for the series of calculations which has been performed with $N_b = 100$ for a range of number densities, $n = \{0.001, \ldots, 1\}$. The compliance is within the predicted values, but measured values are slightly higher than anticipated.

Next, the results of all performed simulations are compared in the Diagrams 7.22, where δl_b is plotted over the term $NP_b n^{2/3} T/P$. For reasons of clarity, the data is plotted on a linear scale (left sided figure) and logarithmic scale (right sided figure). According to Equation (7.8.17), the measured values for δl_b should lay along the theoretical line, which is

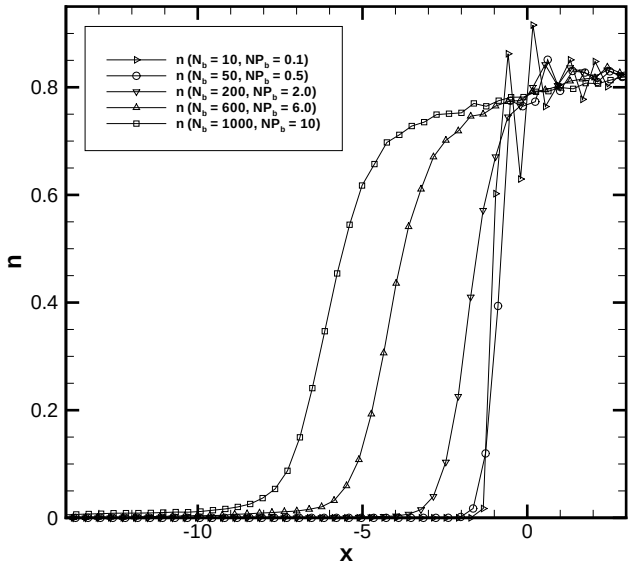


Figure 7.18: Number density profile across the relaxation zone for a set of interaction depths: $NP_b = \{0.1, 0.5, 2.0, 6.0, 10.0\}$. The bulk material is in liquid state with a number density of $n_0 = 0.8$ and temperature of 1. All graphs have been shifted so that $n(0) = 0.95n_0$.

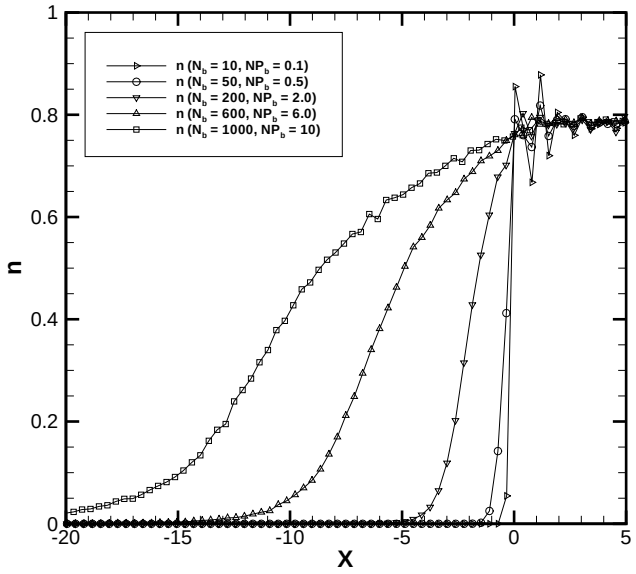
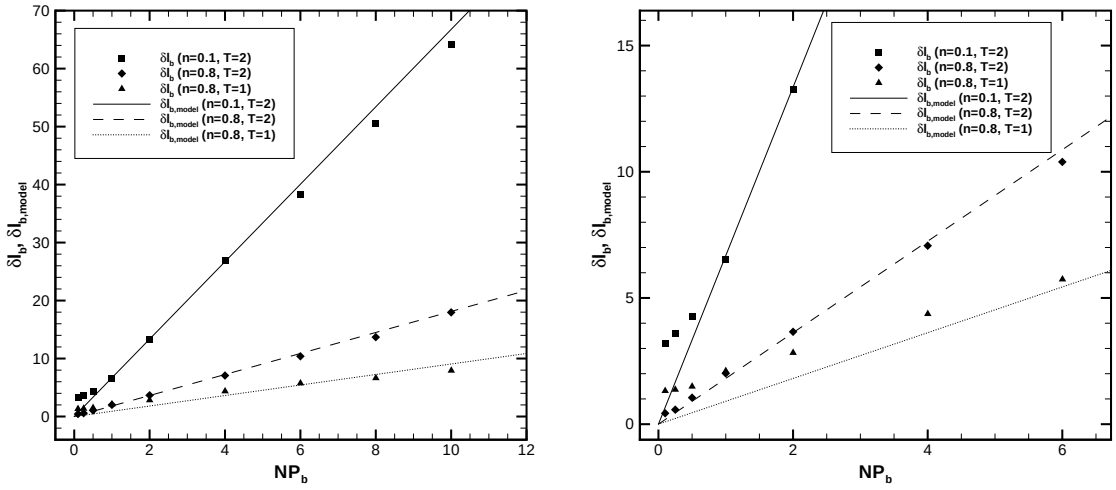


Figure 7.19: Number density profile across the relaxation zone for a set of interaction depths: $NP_b = \{0.1, 0.5, 2.0, 6.0, 10.0\}$. The bulk material is in supercritical state with a number density of $n_0 = 0.8$ and temperature of 2. All graphs have been shifted so that $n(0) = 0.95n_0$.



(a) Complete **(b)** Magnified

Figure 7.20: Relaxation zone width, δl_b, over interaction depth in atomic layers, NP_b.

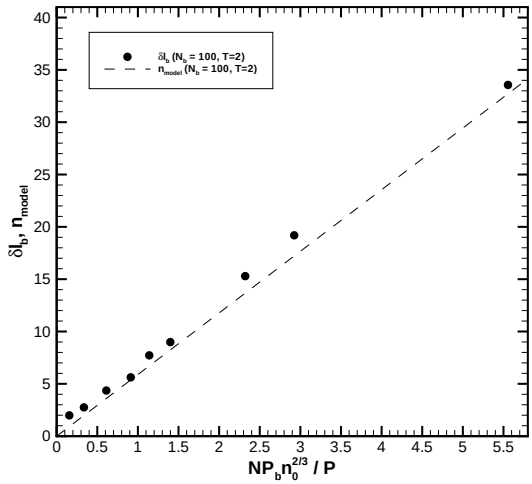
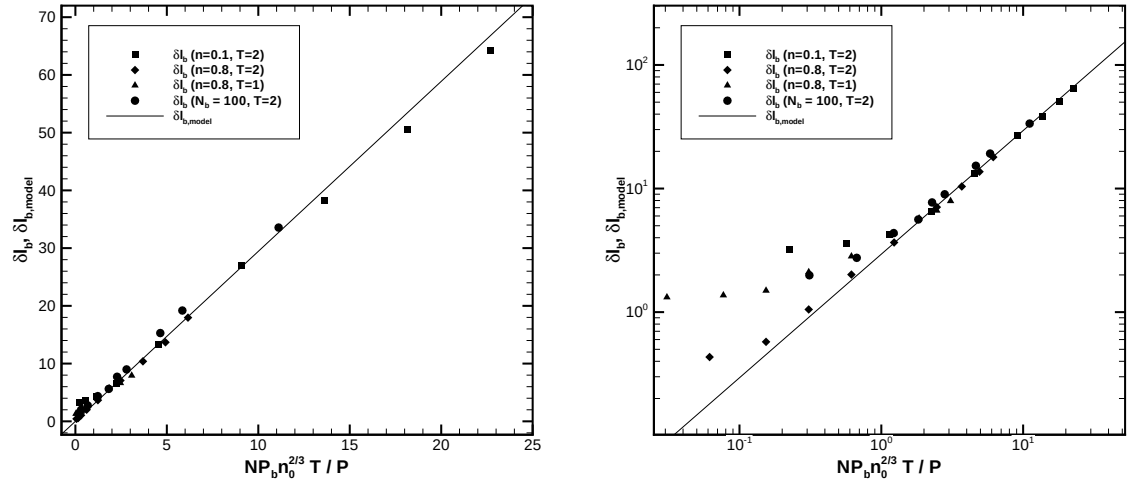


Figure 7.21: Relaxation zone width, δl_b, over $n_0^{2/3}/P$ for $N_b = 100$.

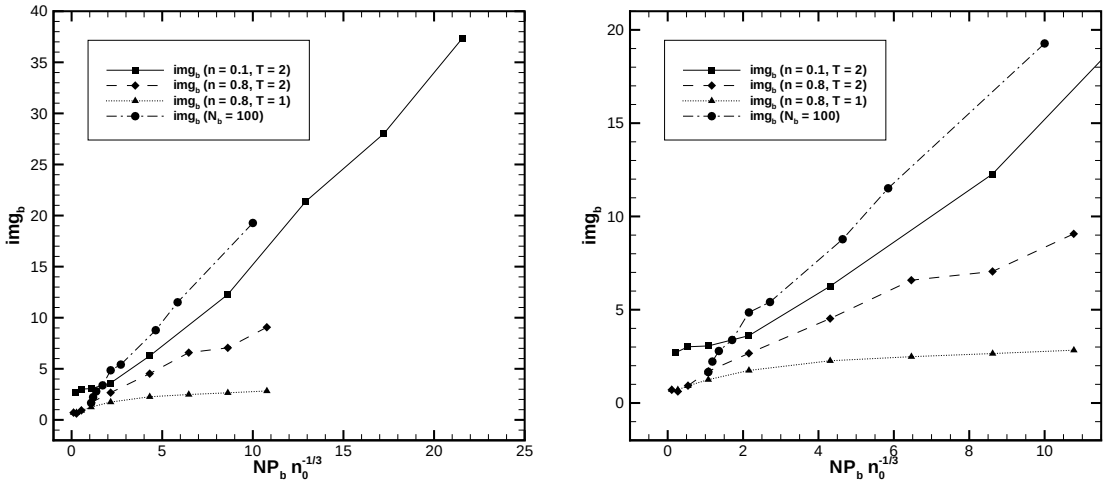
printed as a solid line in both figures. This is true for all investigated series for values of $NP_b n^{2/3} T/P$ larger than 1. The series that aligns best with the predictions is the one of the supercritical state, for which the measured values of δl_b follow the theoretical line even for values of $NP_b n^{2/3} T/P$ smaller than 1. The other series show more or less strong deviations for low $NP_b n^{2/3} T/P$. That especially concerns the liquid state, which has also the largest deviation in Figure 7.18. One should, however, note that the deviations in Figure 7.22b appear especially large due to the logarithmic scale, where they can be considered as relative deviations. On the linear scale, which shows absolute values, the deviation are almost unnoticeable.



(a) linear scale **(b)** logarithmic scale

Figure 7.22: Relaxation zone width, δl_b, over $NP_b n^{2/3} T/P$.

Finally, the gradients of the density profile within the relaxation zone are analysed. As described above, one way to analyse this is to use the inverse maximum gradient img_b, normalised relative to the density n_0. In Figure 7.23, img_b has been plotted over $NP_b n_0^{-1/3}$ for the performed series of cases. One could say, that parts of these curves also obey a linear law. However, the available data is not enough to firmly support this hypothesis and no theoretical model has been found.



(a) Overall view **(b)** Magnified view

Figure 7.23: Normalised inverse maximum gradient over $NP_b n_0^{-1/3}$.

Summarising the investigation on the relaxation zone width for LJ-fluids, it has been found that:

- The interaction depth in atomic layers, NP_b, is the most suitable parameter to describe the number of atoms, onto which the external force is applied.
- The relaxation zone width, δl_b, increases with increasing interaction depth.
- The density profile for gaseous states are well described by the theoretical model represented by Equation (7.8.16).
- The relaxation zone width, δl_b, is well predicted by Equation (7.8.17) for the investigated gaseous and supercritical states and partly for the performed liquid states.
- A linear relationship is suspected between the inverse maximum gradient, img_b, and $NP_b n_0^{1/3}$.

It should be noticed that the analysis has been performed for situations with no flow across the interface, i.e. for velocity zero or parallel to the interface. For fast flow velocities perpendicular to the interface, the density profile is distorted.

7.8.4 Confinement of atoms

The previous section gave a detailed description of the relaxation zone and made clear that the usage of an external force (MTF), or direct velocity change (MTDVC), to transfer the momentum flux by stress creates a relaxation zone within which the density falls from its bulk value to zero. The profile of energy has an even more complex shape (q.v. Figures 7.6). It is obvious that the different values of density and energy have a significant effect on the properties of the material and its transfer coefficients inside the relaxation zone. Altered transfer coefficients, such as viscosity, thermal conduction coefficient or speed of sound, mean that the transfer rates of momenta and energies differ as well. This may lead to unphysical profiles across the HSI for hybrid MD-CFD simulations. In addition, the asymptotic character of the density profiles implies that some atoms with higher kinetic energy travel very far into the vacuum of the relaxation zone, thereby completely losing contact with the other atoms in the bulk material. This behaviour is particularly dominant for low densities.

For the reasons stated, it has been suggested to confine the atoms close to the boundary by restricting their movement to within a distance of δx_{max} from the boundary [40, 54, 55]. Atoms that have moved further away from the boundary than δx_{max}, are called *crossed atoms*. The confinement can be achieved by different measures to handle crossed atoms:

Remove and reinsert crossed atoms: One way is to remove atoms that have crossed the threshold of δx_{max} and to reinsert them a distance δx further inside the relaxation zone. This requires finding a new position for the respective atom with exactly the same potential energy as it had in the old position. Basically, this approach is restricted to single atoms only.

Move atoms back: A crossed atom would be moved back towards the boundary by a small $\delta \boldsymbol{x} = \delta x \boldsymbol{n}_S$ at every time step. The changes in potential energy, $\delta E_{p_i} = E_{p_i}(\boldsymbol{x}_i + \delta \boldsymbol{x}) - E_{p_i}(\boldsymbol{x}_i)$, are added to the integrated conductive fluxes: $F_{eq}^{int} \leftarrow F_{eq}^{int} + \delta E_{p_i}$.

Adapting the integration algorithm: Another possibility is to adapt the time integration algorithm in a way that the positions of crossed atoms are not updated if they would proceed beyond the boundary. However, velocity changes due to the external forces are still integrated. In this case, the atoms at the boundary still experience the external force, which eventually reverses the velocity vector. Once the velocity vector is pointing into the molecular domain ($\boldsymbol{v}_i \cdot \boldsymbol{n}_S > 0$) the atom will move back towards the boundary. When adapting the integrator, one has to make sure that the velocity of the crossed atoms is only updated through external forces. The internal forces must be zeroed, otherwise the integration algorithm is changed, and will generate kinetic energy without decreasing the potential energy accordingly.

If the boundary is parallel to the cartesian coordinates, the rules can be implemented easily by introducing a test before updating the position and velocity of each atom i. The position and velocity in the dimension α that is perpendicular to the boundary are not updated if the condition

$$(r_b - r_{i\alpha}) \cdot n_{S\,\alpha} > \delta x_{max} \ \wedge \ v_{i\alpha} \cdot n_{S\,\alpha} < 0 \tag{7.8.18}$$

is fulfilled, where x_b is the position of the boundary.

Testing the flux boundary conditions with MTF momentum transfer in conjunction with the confinement schemes for the static cases (MDS-G and MDS-L), one expects the simulation to give a stable solution and a confined relaxation zone. However, the simulations showed a drop in the total energy of the system for all described confinement schemes. This is shown clearly in Figure 7.24 (dashed line), where the total energy of the MDS-G system is plotted over time when using the adapted integration algorithm to confine the atoms within the relaxation zone (Equation (7.8.18)). Clearly, the total energy density, e_t, is not conserved.

The (unexpected) energy drain has its origin in the component F_S^{ext} of the external force $\boldsymbol{F}^{ext}$ that is acting perpendicular to the boundary surface: $F_S^{ext} = \boldsymbol{F}^{ext} \cdot \boldsymbol{n}_S$. It generates a potential energy field and every atom that experiences the external force has an additional external potential energy that is:

$$E_p^{ext}(\boldsymbol{r}_i) = \frac{1}{N_b} F_S^{ext} \left[(\boldsymbol{r}_{N_b} - \boldsymbol{r}_i) \cdot \boldsymbol{n}_S \right] , \tag{7.8.19}$$

where $\boldsymbol{r}_{N_b}$ is the position of the N_bth atom of the atom boundary list that marks the border of the external force field. Shifting the position, $\boldsymbol{r}_i$, of an atom towards the boundary reduces the potential energy $E_p^{ext}(\boldsymbol{r}_i)$ as well.

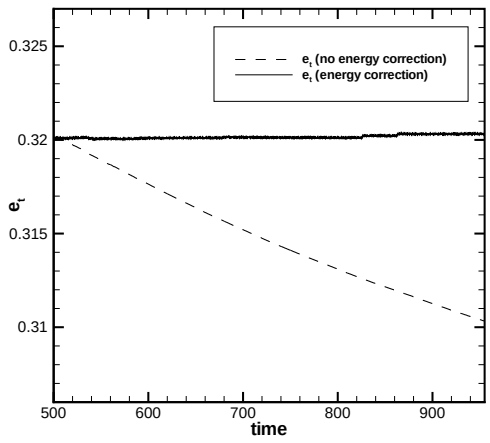


Figure 7.24: Equilibrium state for $u = 0$ for test case (MDS-G) of the adapted integration algorithm. The figure shows the conservation of energy density e_t with and without energy correction.

Consider the unconfined trajectory of a single atom, as illustrated in Figure 7.25, that moves towards the boundary. For this thought experiment, all other atoms are considered to be frozen and the moving atom is not influenced by them. As indicated in Figure 7.25, the external force is acting on all atoms that are further away from the boundary than the N_bth atom of the atom boundary list. After the atom crosses the virtual line, marked by the N_bth atom, at time $t = 1$, it starts to experience the external force $\boldsymbol{F}_S^{ext}$ which decelerates the atom. At some point, at time $t = 2$, its velocity component perpendicular to the boundary surface will be zero ($\boldsymbol{v}_{t=2} \cdot \boldsymbol{n}_S = 0$) and it starts moving back towards the boundary. When it leaves the zone of the external force at time $t = 3$, the velocity component respective to the boundary surface is simply reversed ($\boldsymbol{v}_{t=3} \cdot \boldsymbol{n}_S = -\boldsymbol{v}_{t=1} \cdot \boldsymbol{n}_S$). The velocity vector $\boldsymbol{v}_{t=3}$ has the same magnitude as the initial velocity vector $\boldsymbol{v}_{t=1}$ ($|\boldsymbol{v}_{t=1}| = |\boldsymbol{v}_{t=3}|$); no energy has been added or removed. However, confining the atoms by any of the suggested schemes has the effect that the atom is moved closer to the boundary. This means that the distance the atom is travelling from time $t = 2$ to time $t = 3$ while it is being accelerated becomes shorter than in the uncon-

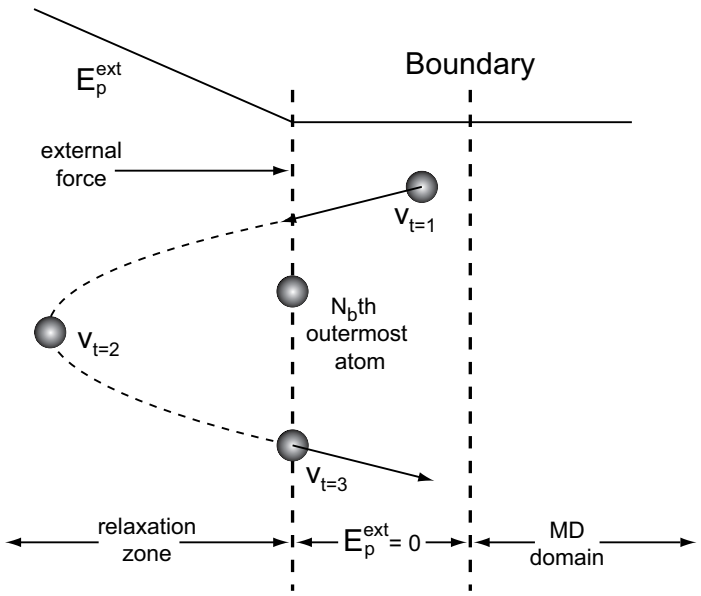


Figure 7.25: Trajectory of a single atom within the relaxation zone.

fined case. Therefore, it picks up less velocity ($|\boldsymbol{v}_{t=1}| > |\boldsymbol{v}_{t=3}|$) and its energy at $t = 3$ is less than at $t = 1$. This is what causes the decrease in energy that is shown in Figure 7.24. For indirect convective flux imposition, the reference line, i.e. where the external force starts acting, coincides with the boundary surface because the number of atoms inside the relaxation zone is kept constant.

For constant momentum fluxes by stress, the energy decrease that is entailed when shifting an atom by $\delta\boldsymbol{r}$ is

$$\delta E_p^{ext} = \frac{1}{N_b} F_S^{ext} \cdot \delta\boldsymbol{r} \,. \tag{7.8.20}$$

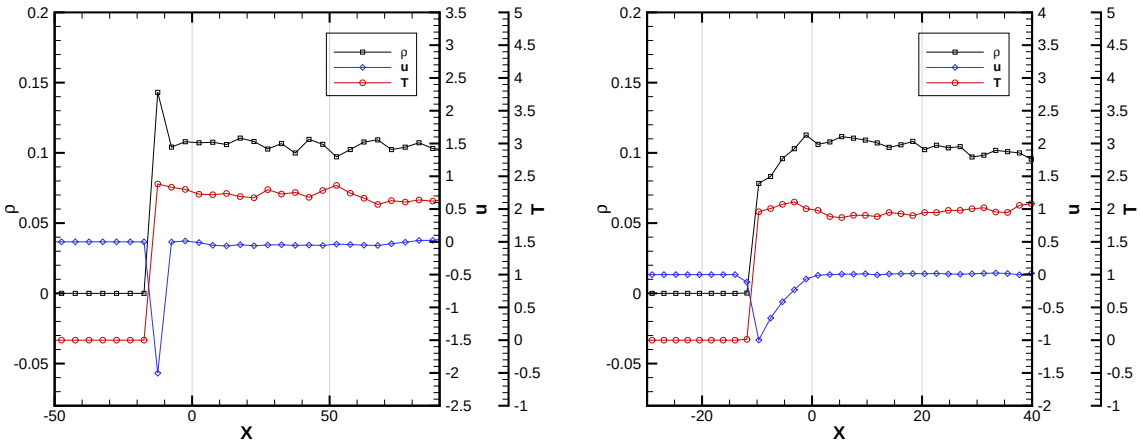
The energy difference δE_p^{ext} can be re-introduced using Algorithm 3 or by adding it to the integrated conductive fluxes: $F_{eq}^{int} \leftarrow F_{eq}^{int} + \delta E_p^{ext}$. It is then introduced automatically along with the transfer of F_{eq}^{int}. This *energy correction* has been tested in conjunction with the adapted integration algorithm to confine the atoms for the MDS-G system. The total energy, e_t, over time is plotted as solid line in Figure 7.24. It kept its original level throughout the simulation, meaning that simulation was stable when using the energy correction scheme. The simulation was also stable for the tested liquid case MDS-L.

The profiles of density, velocity and temperature, that correspond to the investigated case (system MDS-G), are plotted for the left boundary in Figure 7.26a. The atoms were confined to a minimum position of $\delta x_{max} = -10$, using the adapted integration algorithm. At this position, $x = -10$, the density profile shows a positive and the velocity a negative peak. This is a logical consequence of the employed confinement scheme, because atoms which travel fast in a negative x direction stay in the area close to $x = -10$ until their velocity vector becomes positive again before they move back into the system. Even if it is only within the relaxation zone, where the molecular description is not completely valid, the peaks in density and velocity still constitute an unwanted artefact. In Figure 7.26b, the profiles for the same test case are shown, when using the remove and reinsert scheme to confine the atoms at $x = -10$. There is no peak in density, because the atoms that have crossed the lower boundary at $x = -10$ are distributed equally in the relaxation zone. This is also the reason why the velocity throughout the relaxation zone is negative.

Using the confinement scheme of removing and reinserting atoms gives better results for the profiles, however, the computational effort is much greater, because it requires the continuous removal and insertion of atoms. Especially in the liquid case, the insertion algorithm requires significant computing time to find new insertion sites.

7.8.5 Further implications of momentum transfer by force

Beside the complications that arise from confining atoms, the external force field E^{ext} generated by the external force $\boldsymbol{F}^{ext}$ has further implications on the insertion and removal of atoms from the relaxation zone for flows across the boundary. This is because, inserting or removing an atom into or out of the relaxation zone also alters the potential energy field itself and thus the external potential energy of the other boundary atoms. Even if the energy



(a) Adapted integration algorithm

(b) Remove and reinsert atoms

Figure 7.26: Density, velocity and temperature profiles for the left boundary for gaseous static ($u = 0$) test case (MDS-G) using confinement schemes.

change can be calculated theoretically, this is elaborate in a practical sense. Alternatively, the atoms can be inserted or removed from the behind the zone in which the external force is acting, i.e. between the boundary surface and the N_bth boundary atom. In Figure 7.25 the area is marked with $E_p^{ext} = 0$. Inserting or removing atoms from this area with $E_p^{ext} = 0$ leaves the external potential energy of the other boundary atoms unchanged. The applicability of this solution is investigated in Section 7.8.7.

For indirect convective flux imposition this problem does not occur, because the number of atoms inside the relaxation zone is constant and the external potential energy is exactly $E_p^{ext}(\boldsymbol{r}_i) = (1/N_b)F_S^{ext}\left[(\boldsymbol{r}_b - \boldsymbol{r}_i) \cdot \boldsymbol{n}_S\right]$, where $\boldsymbol{r}_b$ is a point on the boundary surface. One, only has to take into account the external energy $E_p^{ext}(\boldsymbol{r}_i)$ when inserting or removing an atom.

7.8.6 Flow parallel to the boundary

The previous sections have concentrated on the static case of the momentum transfer by force (MTF) and the characteristics of the arising relaxation zone. In this section, the momentum transfer by force is analysed for the flow parallel to the boundary. The general setup is explained in Section 6.2. The used test cases were MDCF-G and MDCF-L1 and MDCF-L2 (q.v. Table B.1). The gaseous case, MDCF-G, was simulated with the velocities $u_e = \{1.0, 2.0\}$, the liquid case MDCF-L1 with $u_e = 0.5$ and the larger liquid case MDCF-L2 with $u_e = 1.0$. This corresponds to the velocities and systems that have already been used to investigate the reflective plane momentum transfer scheme in Section 7.7.2.

The most important simulation parameters are given in Table 7.4. For the liquid case, the equilibration was performed in two stages. In the first equilibration run over 10,000

time steps, the temperature control was applied to the entire system to obtain $T = 1$. A second equilibration run over 30,000 time was performed with the temperature control applied to the wall atoms only. The simulations had to be run for a long time to allow the temperature profile to develop. It was considered to have reached an equilibrium situation when the last three calculated profiles did not show significant differences.

Parameter	System	
	MDS-G	MDS-L1/MDS-L2
Momentum flux by stress	MTF	
Crossed atoms	no confinement	
NP_b	1.0	
Equilibrium time steps	10,000	40,000
Simulation time steps	4,000,000	400,000
Measuring frequency (τ_m)	100	100
Averaged over (N_m)	10,000	1,000

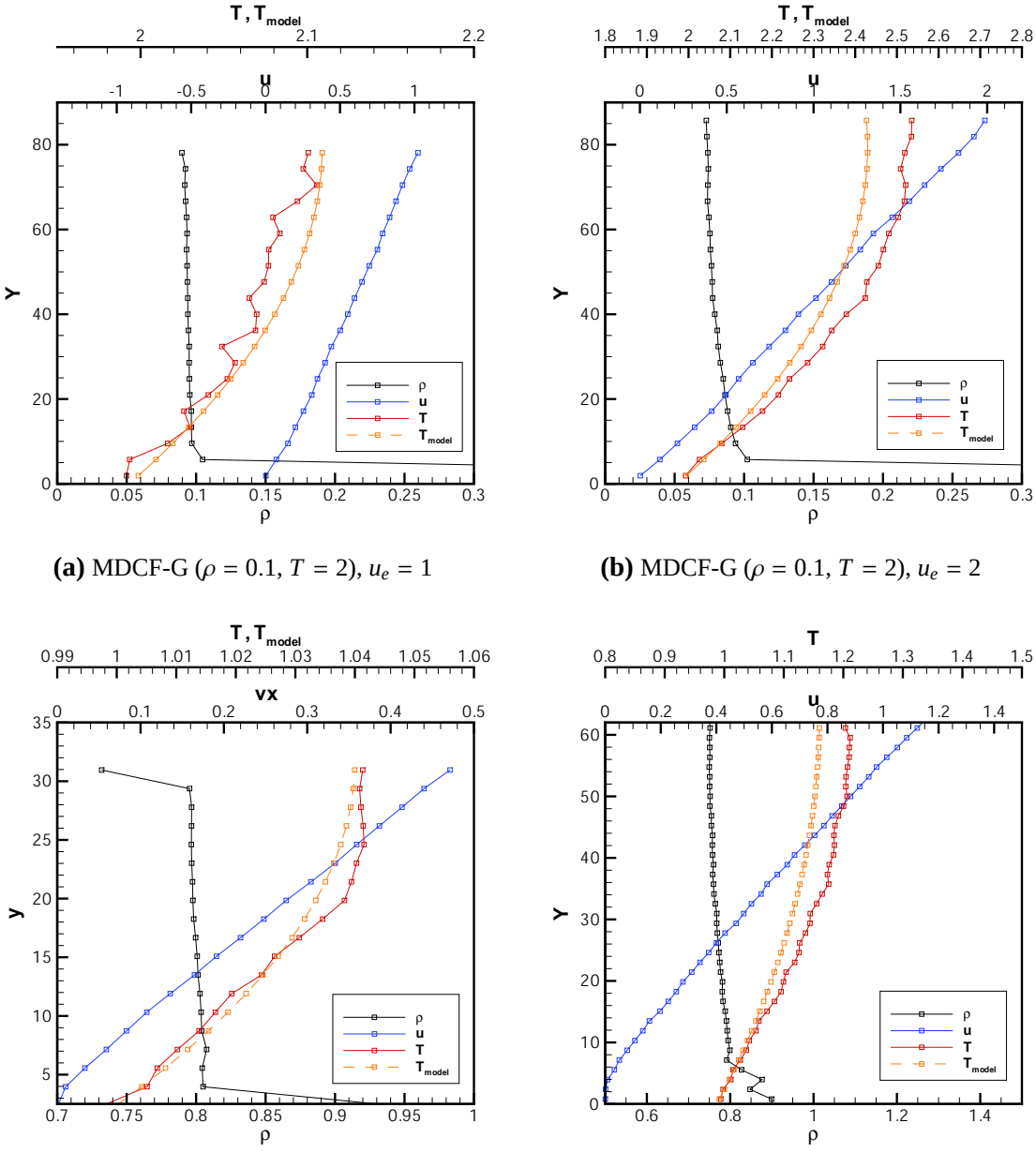
Table 7.4: Parameter settings for the DC-MTF Couette flow test cases.

In Figure 7.27 the profiles of the performed simulations for the gaseous and liquid systems are gathered for the selected velocities. Each figure shows the profiles of density, velocity in x direction and temperature in y direction for the respective case. The theoretical temperature profiles, which are based on Equation (6.2.2), are plotted as orange lines in all figures. The sharp jump in density at the lower sides marks the beginning of the wall, which has a higher density ($\rho_{wall} = 1.0$) than the fluid.

As can be seen in all cases, the velocity profiles are linear as expected and the velocities at y_e match well to the theoretical values, u_e. Only for the liquid case with $u_e = 1$ is the velocity slightly higher, because of the stronger effect of the decreased viscosity due to increased temperature and lower density for higher values of y. The temperatures match well to the theoretical predictions, considering the uncertainties of the values of the transport coefficients and their slight variations across the fluid in y direction due to the viscous heating of the material.

7.8.7 Flow perpendicular to the boundary

For the free flow test case with the flow perpendicular to the boundary, three parameter combinations have been tested for the gaseous and liquid cases. A complete overview is given in Table 7.5. In all cases, the momentum flux by stress is transferred through the external force that is calculated on the base of the integrated momentum flux, F_{ps}^{int}. Three parameter combinations have been tested, for the gaseous (MDFF-G) and the liquid (MDFF-L) system. These are named in Table 7.5 as MTF-G1, MTF-G2, MTF-G3 for the gaseous and MTF-L1, MTF-L2, MTF-L3 for the liquid system respectively. The difference between the parameter combinations is the employed scheme of the coupling of integrated fluxes by convection and by stress (FCCS). For the combinations MTF-G1 and MTF-L1, no flux



(a) MDCF-G ($\rho = 0.1$, $T = 2$), $u_e = 1$

(b) MDCF-G ($\rho = 0.1$, $T = 2$), $u_e = 2$

(c) MDCF-L1 ($\rho = 0.8$, $T = 1$), $u_e = 0.5$

(d) MDCF-L2 ($\rho = 0.8$, $T = 1$), $u_e = 1$

Figure 7.27: Profiles for density, velocity and temperature of the Couette flow using the momentum transfer by force scheme (DC-MTF) for gaseous (upper figures) and liquid state (lower figures). T_{model} is the analytical solution for the temperature profile.

coupling was used. In the MTF-G2 and MTF-L2 combination, the FCCS has been applied to the momentum only, i.e. the integrated convective fluxes of momentum F_{pc}^{int} are coupled with the integrated momentum flux by stress F_{ps}^{int}. For the combination MTF-G3 and MTF-L3, the energy fluxes F_{ec}^{int} and F_{es}^{int} were coupled additionally. The gaseous combinations MTF-G1, MTF-G2 and MTF-G3 were tested for the velocities $u = \{0.5, 1.0, 2.0\}$, the liquid combinations MTF-L1, MTF-L2 and MTF-L3 for velocities $u = \{0.5, 1.0\}$ in order to

investigate the behaviour of the combinations at different velocities.

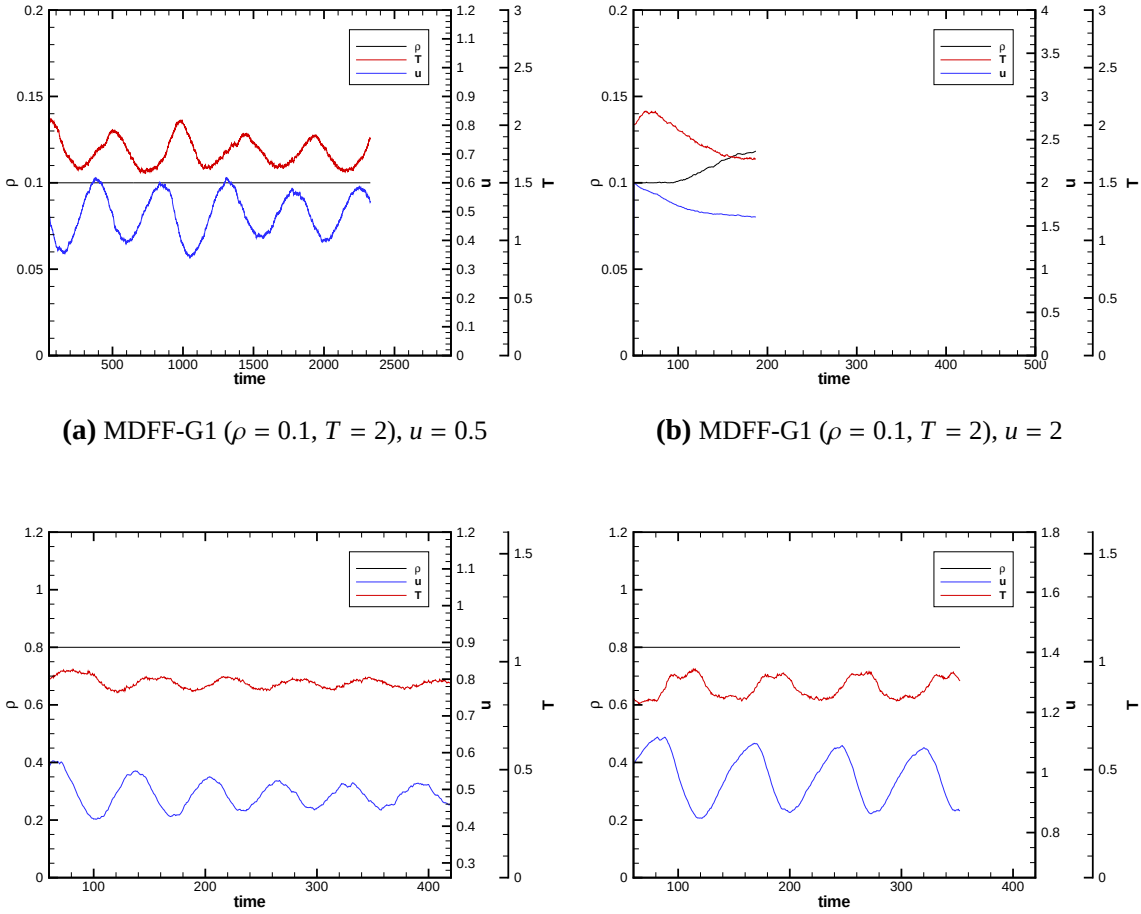
Parameter	Combination					
	MTF-G1	MTF-G2	MTF-G3	MTF-L1	MTF-L2	MTF-L3
System	MDFF-G			MDFF-L		
Momentum flux by stress	MTF of F_{ps}^{int}					
FCCS momentum coupling	no	yes	yes	no	yes	yes
FCCS energy coupling	no	yes	yes	no	no	yes
Crossed atoms	no confinement					
NP_b	1.0					
Insertion algorithm	USHER					
Removal of atoms	outermost one					
Equilibrium time steps	10,000			40,000		
Simulation time steps	4,000,000			400,000		
Measuring frequency (τ_m)	100			100		
Averaged over (N_m)	10,000			1,000		

Table 7.5: Parameter settings for the DC-MTF free flow test cases.

Because of the issues connected with the external force field that is generated by the external force F^{ext}, the areas for inserting and removing atoms must be carefully selected. As pointed out in Section 7.8.5, the change in potential energy of the boundary atoms can not be determined when adding or removing an atom for situations with a non-constant number of boundary atoms. Thus, initially the insertion area was assigned between the boundary at $\boldsymbol{r_b}$ and the position of the innermost boundary atom, onto which the external force is applied, the Nth atom in the boundary atom list. The average position of the Nth atom was calculated in advance from the simulation of the static test cases. For removing atoms, several selection schemes were tested for their applicability. Theoretically, the atoms should be removed from an area equivalent to the insertion area. However, it was not possible to achieve any stable simulation when using this selection scheme. The results showed that best choice is to base the selection on the position and to remove the outermost atom. Other schemes, such as selecting the best fit or random selection, led to a breakdown of the MD flux boundary conditions.

First, the test cases MDFF-G1 and MDFF-L1 were investigated. For those, no FCCS scheme is applied, i.e. the momentum fluxes by convection were not coupled with the fluxes by stress nor the respective energy fluxes. This parameter combination proved to be unstable for all tested simulations because it caused oscillations in the overall values of velocity and temperature of the system. The development in time of the overall values for density, velocity and temperature are plotted in Figure 7.28 for selected velocities. Clearly noticeable are the oscillations of velocity and temperature for all shown cases, except MDFF-G1 at $u = 2$, where the boundary conditions break down before the end of the simulation. The reason for the oscillations is that the external force acts like an harmonic oscillator, causing the position of the boundary atoms to move in and out. During this motion, energy is alternatingly added and removed from the system. This causes the fluctuations which can

be seen in Figure 7.28. In Figure 7.29, the position in the x dimension of the boundary atoms, which is simply the centre of mass of the N_b boundary atoms ($r_x = (1/N_b) \sum_{i=1}^{N_b} r_{ix}$) is plotted over time together with overall temperature and velocity. The position r_x of the boundary atoms oscillates with the same frequency as the overall velocity and temperature, indicating that these oscillation are connected to each other. The oscillating motion of the boundary can actually also be seen when visualising the trajectories of the system as an animation in time. For obvious reasons, this animation can not be included here.



(a) MDFF-G1 ($\rho = 0.1$, $T = 2$), $u = 0.5$

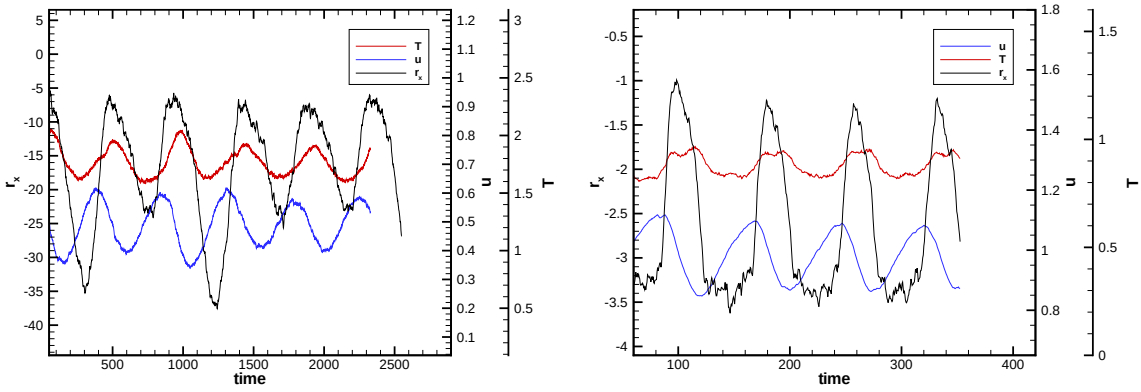
(b) MDFF-G1 ($\rho = 0.1$, $T = 2$), $u = 2$

(c) MDFF-L1 ($\rho = 0.8$, $T = 1$), $u = 0.5$

(d) MDFF-L1 ($\rho = 0.8$, $T = 1$), $u_e = 1$

Figure 7.28: Overall values for density, velocity and temperature of free flow test case for the combination MDFF-G1 and MDFF-L1.

Obviously, the parameter combinations in MDFF-G1 and MDFF-L1 are not suitable for flows perpendicular across interface. One reason is that the properties of the atoms removed on the outlet do not correspond with the convective fluxes. This causes the flow velocity of the boundary atoms on the outlet to differ from the constant flow velocity, u. Thus, the amount of energy by stress that is removed on the outlet is incorrect as well. This is what triggers the onset of the oscillations in Figure 7.29. It is therefore necessary to correct the



(a) MDFF-G1 ($\rho = 0.1$, $T = 2$), $u = 0.5$ **(b)** MDFF-G1 ($\rho = 0.8$, $T = 1$), $u = 1.0$

Figure 7.29: Overall velocity and temperature compared to the position r_x of the boundary atoms of the western boundary over time.

convective fluxes to its precalculated value by using the FCCS (Section 7.4.1) scheme.

The second attempt was to only couple the convective momentum fluxes convection and by stress for outlet conditions. This corresponds to the parameter combinations MDFF-G2 and MDFF-L2 for gaseous and liquid material states respectively. The results, shown in Figure 7.30 for the gaseous and in Figure 7.31 for the liquid test cases, confirm that this parameter combination provides stable flux boundary conditions. The overall values of density, velocity and temperature are plotted in the diagrams on the right column. These confirm that in all performed simulations a stable constant flow was established with density, velocity and temperature being constant in time at the correct value. In the diagrams on the left side in figures 7.30 and 7.31, the profiles in x direction have been plotted. For all cases, the conditions on the inlet (left side) appear to give good results. However, on the outlet (right side), all profiles, except the gaseous case with $u = 2$, show a peak in velocity while the temperature is decreasing within the relaxation zone.

A closer analysis of the conditions on the inflowing boundary (left side) revealed that actually the boundary atoms moved slightly closer to the boundary, so that the insertion area was partly inside the external potential energy field. This was observed for all performed simulations. The inserted atoms, therefore, had a slightly higher energy than actually prescribed by the convective fluxes. It was not possible to measure the amount of additional energy that is inserted or removed.

The actual energy transferred through the external force is not exactly equal to the integrated energy flux by stress F_{es}^{int}, which is calculated on basis of the analytical solution. This is caused by the velocity of the boundary atoms being different from the respective analytical flow velocity u. For the gaseous test case MDFF-G2, too much energy was transferred on the inlet and removed on the outlet: $\approx 1\,\%$ for $u = 0.5$, $\approx 1\,\%$ for $u = 1.0$ and $\approx 0.4\,\%$

for $u = 2.0$. In case of the liquid, i.e. MDFF-G2, too little energy was transferred on the inlet $\approx 9\,\%$ for $u = 0.5$ and $\approx 7\,\%$ for $u = 1.0$. While for the gaseous cases the differences are small, they are significantly higher for the liquid cases. The solutions are only stable because the deviations are counterbalanced through the convective momentum and energy fluxes on the outlet, which exhibit the inverse deviations from the analytical solution.

The MDFF-G2 and MDFF-L2 test cases were also tested with the random and best fit selection schemes for the removal of atoms. Neither of them gave a stable solution; the flux boundary condition broke down relatively quickly.

The last tested combination was to couple the momentum and energy fluxes by convection and by stress to ensure that the transferred momentum and energy fluxes correspond exactly to the analytical solution. This corresponds to the test combinations MDFF-G3 and MDFF-L3. Two different position-based selection schemes for removing atoms have been tested. First, the outermost atom, that does not experience the external force, which is the $(N_b + 1)$th atom, was selected. Removing this atom, avoids the problem encountered with the potential energy field that is generated by the external force. However, using this selection scheme led to a fast breakdown of the flux boundary conditions on the outlet in all tested cases. The reason is that if no atoms are removed from the outer boundary atoms, their average velocity must be zero. Thus, the energy F_{es}^{int} is removed on expanse of the internal kinetic energy in flow direction of the boundary atoms. This leads rapidly to a situation where no internal energy is available, i.e. temperature is zero. At this point, the flux boundary conditions on the outlet break down. Next, the selection scheme for removing the atom that is furthest away from the interface was tested. As expected, removing the atoms from the external potential energy field caused by F^{ext} lead to a decrease in the total energy. This is shown in Figure 7.32 for the gaseous test case MDFF-G3 with $u = 2.0$. The green curve is the average energy per atom, $\langle E_t \rangle$, which decreased constantly until the flux boundary condition broke down on the outlet, because no energy that could be removed was available. The same applies for the liquid test case MDFF-L3, which was investigated for $u = 0.5$ and $u = 1.0$. The corresponding results are shown in Figure 7.33, where the same energy decrease was observed as in the gaseous case.

The momentum transfer by external force is also very sensitive to the region where the atoms are inserted. If the atoms are inserted within the region where the external force is acting, then the added energy by stress, F_{es}^{int}, is higher than it should be. This was tested for all parameter combinations, i.e. MDFF-G1, MDFF-G2, MDFF-G3, MDFF-L1, MDFF-L2 and MDFF-L3, and led to a breakdown of the flux boundary conditions in each case.

In Section 7.8.3, the relaxation profile was investigated in detail for the static cases. For flows parallel to the boundary, the relaxation profile is expected to be very similar to the static case. However, for flow perpendicular to the boundary, some distortions were expected, due to the effects of the external force and the insertion and removal of atoms. The effects of different interaction depths was tested for the MDFF-L2 case with $u = 1.0$. In Figure 7.34, the profiles of density, velocity and temperature are plotted on the inlet for the two interaction depths, $NP_b = 0.5$ and $NP_b = 3.0$. The respective profiles on the outlet

are plotted in Figure 7.35. On the inlet, as well on the outlet, there are no big differences in the profiles between two interaction depths, although they exhibit sharper transitions and smaller deviation from the desired values for the $NP_b = 0.5$ than for $NP_b = 3.0$. In terms of stability, no difference could be registered between $NP_b = 0.5$ and $NP_b = 3.0$. Clearly apparent is the difference between the inlet and outlet profiles, where those on the outlet do not correspond to the ideal characteristics.

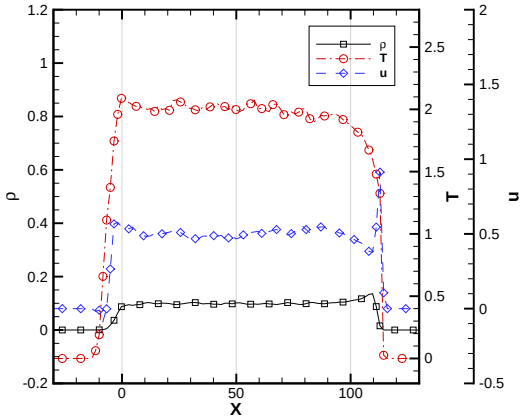
7.8.8 Summary of results of the DC-MTF

From the performed test cases, a number of conclusions can be drawn. Using momentum transfer by force, one can obtain stable boundary conditions in the standard situations for static fluids, flows parallel to the boundary and flows across the boundary.

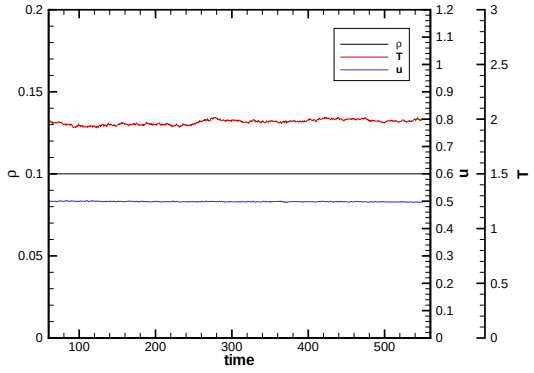
The external force, $\boldsymbol{F}^{ext}$, should be distributed on the boundary atoms by the simplest weighting function, $g(\boldsymbol{r}_i, \boldsymbol{v}_i) = 1$. In the relaxation zone that is generated by the external force, the density falls from its maximum value to zero. For gaseous states, the density profile can be described by Equation (7.8.16). The relaxation zone width, δl_b is well predicted by Equation (7.8.17) for the investigated gaseous and supercritical states and partly for the performed liquid states. An undesirable side effect is that some atoms travel far into the vacuum beyond the interface. Attempts to confine such atoms lead to a decrease in energy and, therefore, should not be used.

Static cases and flows parallel to the boundary posed no difficulties for the momentum transfer by force. For flow across the boundary, the only stable combinations were MDFF-G2 and MDFF-L2. The main feature of these are to include the coupling of the momentum fluxes by convection and by stress. It is not possible to insert the atoms into an area in which the external force is not acting. Similarly, for the removal of atoms, the only working scheme is to select the respective outermost atom. Thus, when using direct convection in conjunction with momentum transfer by force, it is not possible to transfer the prescribed convective energy fluxes, because of the changes in the external potential energy field when inserting or removing atoms. Additionally, the actually transferred energy can not be measured accurately and can thus not be reported back to the continuum solver for correcting the discrepancies. The density profiles show good characteristics for both inflow and outflow conditions. The velocity and temperature profiles are satisfactory for inflowing situations, but are distorted for outflows. It was found that smaller values of NP_b, i.e. smaller interaction depth, give more ideal profiles than larger ones. Any other parameter combination was unsuitable for the simulation of flow across the interface. The absence of FCCS led to oscillations of the system and a full coupling of momentum and energy fluxes by convection and by stress caused the flux boundary conditions to break down.

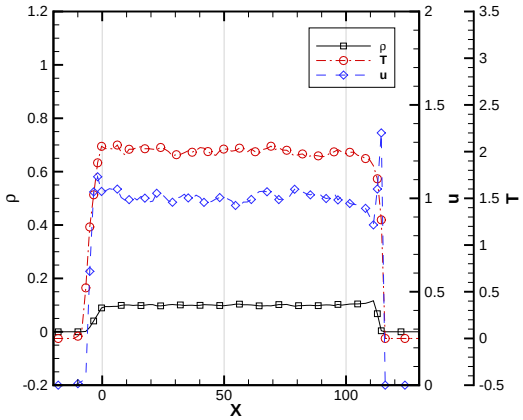
The conclusion is that momentum transfer can create a stable system, but the precise value of the actually transferred energy F_{es} is unknown and does not correspond to the analytical fluxes. This circumstance makes the MD flux boundary conditions scheme of direct convection and momentum transfer by force not the ideal scheme for the usage within the HSI for coupling of MD and CFD based on mass, momentum and energy fluxes.



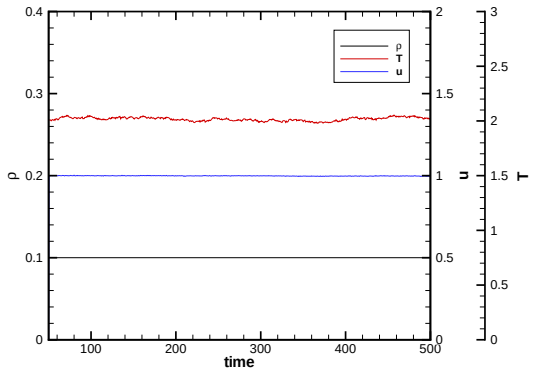
(a) Profile of ρ, u and T for $u = 0.5$



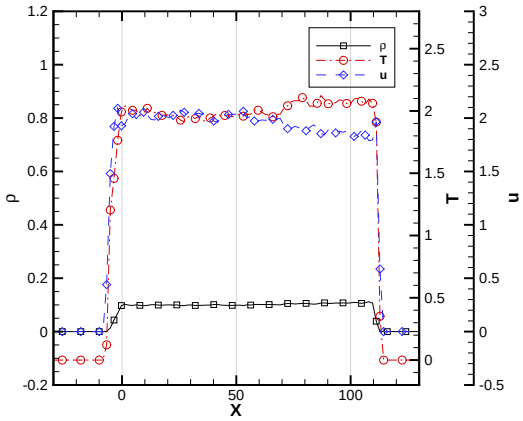
(b) Overall values of ρ, u and T over time for $u = 0.5$



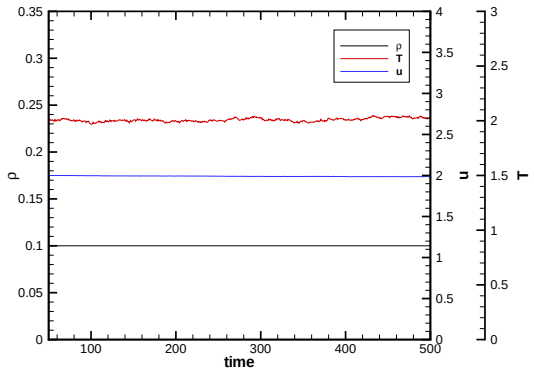
(c) Profile of ρ, u and T for $u = 1.0$



(d) Overall values of ρ, u and T over time for $u = 1.0$

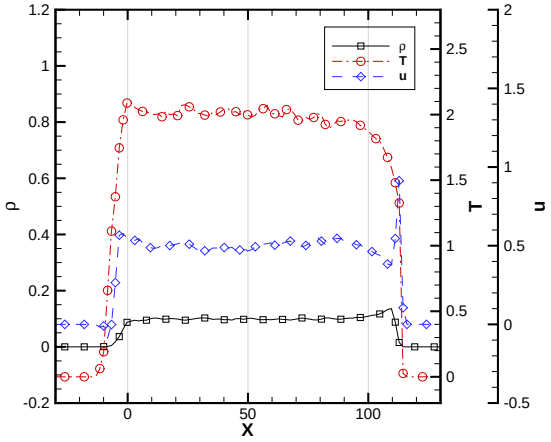


(e) Profile of ρ, u and T for $u = 2.0$

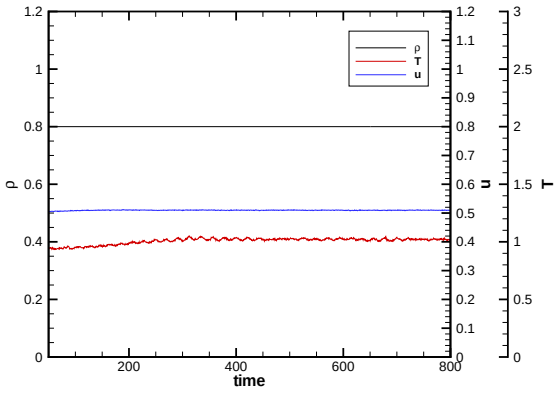


(f) Overall values of ρ, u and T over time for $u = 2.0$

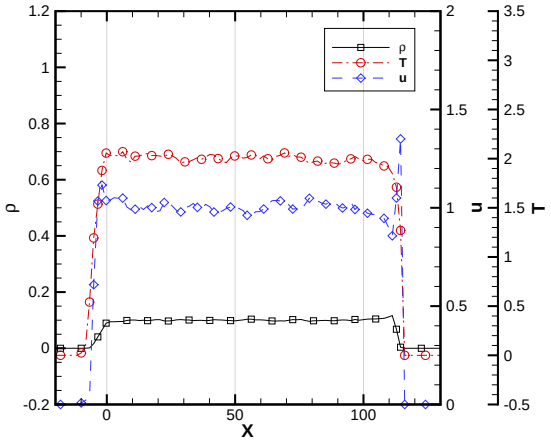
Figure 7.30: Profiles in x direction and overall values of density, velocity and temperature of gaseous free flow test case MDFF-G2 ($\rho = 0.1$, $T = 2$) (FCCS coupling of momentum fluxes by convection and by stress).



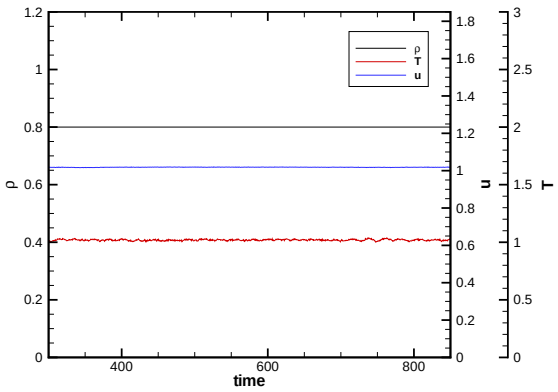
(a) Profile of ρ, u and T for $u = 0.5$



(b) Overall values of ρ, u and T over time for $u = 0.5$



(c) Profile of ρ, u and T for $u = 1.0$



(d) Overall values of ρ, u and T over time for $u = 1.0$

Figure 7.31: Profiles in x direction and overall values of density, velocity and temperature of liquid free flow test case MDFF-L2 ($\rho = 0.8$, $T = 1$) (FCCS coupling of momentum fluxes by convection and by stress).

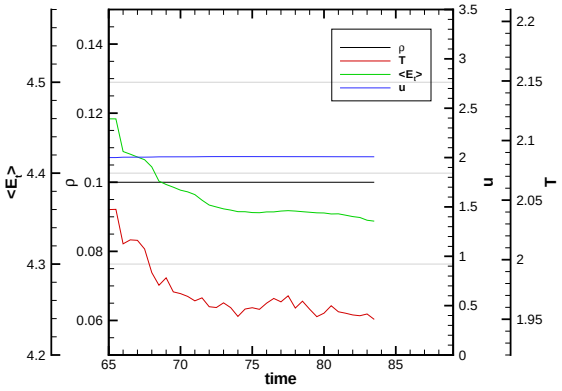
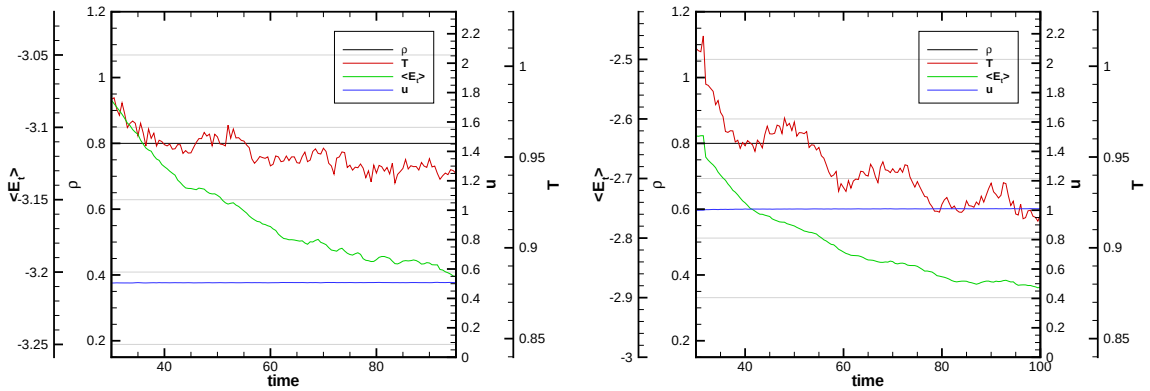
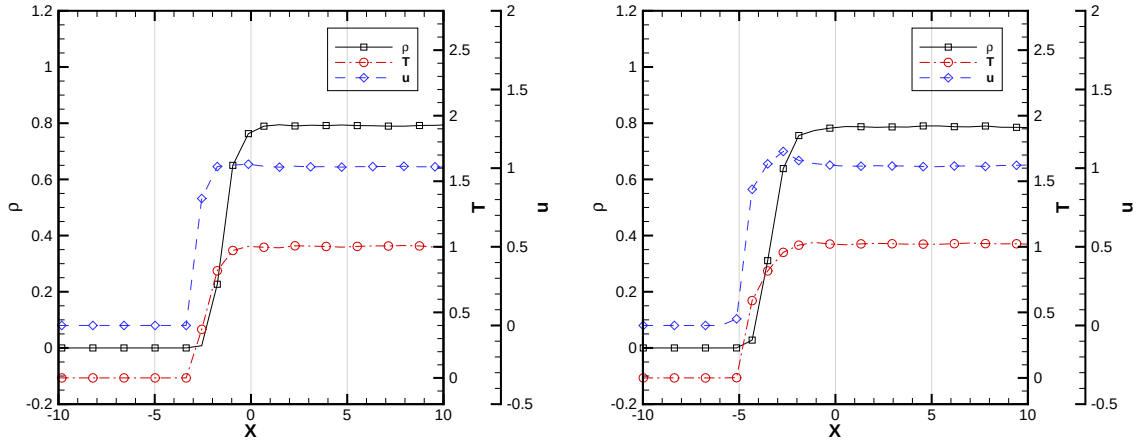


Figure 7.32: Overall values of density, velocity and temperature of gaseous free flow test case MDFF-G3 ($\rho = 0.1$, $T = 2$) for $u = 2$ (FCCS coupling of momentum and energy fluxes by convection and by stress).



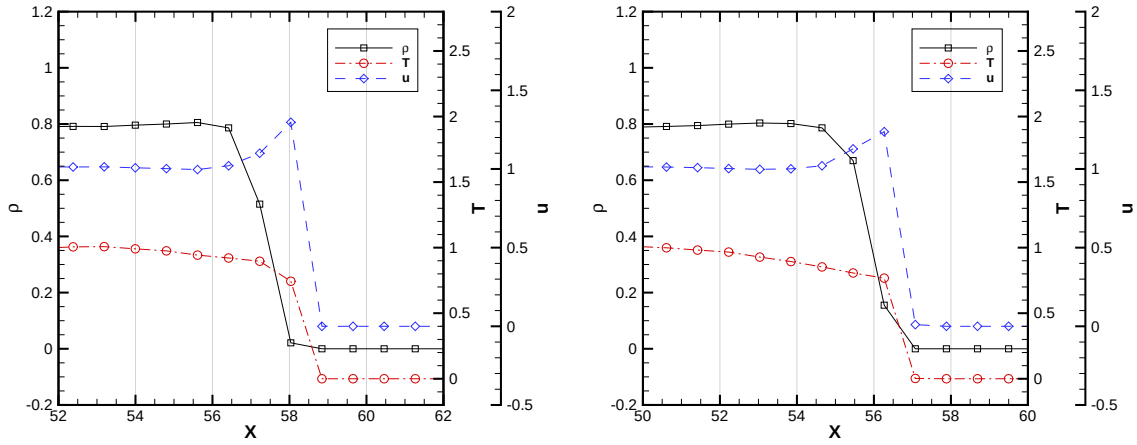
(a) $u = 0.5$ **(b)** $u = 1.0$

Figure 7.33: Overall values of density, velocity and temperature of liquid free flow test case MDFF-L3 ($\rho = 0.8$, $T = 1$) (FCCS coupling of momentum and energy fluxes by convection and by stress).



(a) $NP_b = 0.5$ **(b)** $NP_b = 3.0$

Figure 7.34: Profiles of density, velocity and temperature on the inlet for different interaction depths of liquid free flow test case MDFF-L2 ($\rho = 0.8$, $T = 1$) (FCCS coupling of momentum and energy fluxes by convection and by stress).



(a) $NP_b = 0.5$ **(b)** $NP_b = 3.0$

Figure 7.35: Profiles of density, velocity and temperature on the outlet for different interaction depths of liquid free flow test case MDFF-L2 ($\rho = 0.8$, $T = 1$) (FCCS coupling of momentum and energy fluxes by convection and by stress).

7.9 Direct convective flux and reverse velocity (DC-MTRV)

7.9.1 The reverse velocity scheme

Both previously investigated momentum transfer schemes, MTRP and MTF, have serious limitations in their application. The problems connected with the MTF scheme (this includes MTDVC), have their origin in the interwoven transfer of momentum and energy fluxes by stress, $\boldsymbol{F}_{ps}$ and F_{es}. The energy transfer can not be controlled directly and any fluctuation of velocity in the relaxation zone leads to the transfer of the incorrect amount of energy, causing oscillations and, in many cases, the breakdown of the flux boundary conditions. The issue could be overcome if one could separate the transfer of the momentum flux by stress from the transfer of energy flux by stress. This is exactly what the reverse velocity scheme, which has been invented by the author, does. It is based on the idea that reversing one component, $v_{i\alpha}$, of the velocity vector, $\boldsymbol{v}_i$, of an atom i transfers a momentum of $-2m_i v_{i\alpha}$ to the atom without changing its energy.

The implemented scheme is given in Algorithm 13. For each dimension x,y and z, the algorithm checks whether the integrated momentum fluxes is 'pointing' in the same direction as the momentum flux ($F^{int}_{ps_\alpha} \cdot F_{ps_\alpha} > 0$). If this is true, it finds the outermost atom that is moving in the opposite direction of the momentum flux F_{ps_α} and reverses the direction of its velocity component $v_{i\alpha}$ to transfer momentum to the system. The process is repeated until the integrated momentum flux is 'pointing' in the opposite direction of the momentum flux. If this is true, the algorithm has transferred more momentum than actually was necessary at this time step. However, in the next time steps the integration of the momentum flux will eventually correct this.

Algorithm 13: Reverse velocity momentum transfer scheme.

```
foreach α = {x, y, z} do
    Get first atomic index i from S_AIL;
    while F^int_{ps_α} · F_{ps_α} > 0 do
        if v_{iα} · F_{ps_α} then
            if atom i is last in S_AIL then return;
            Get next atomic index i from S_AIL;
        v_{iα} ← −v_{iα};
        F^int_{ps_α} ← F^int_{ps_α} + 2 m_i v_{iα};
```

Because Algorithm 13 introduces only momentum, the energy $\boldsymbol{F}^{int}_{es}$ has to be transferred separately. To this end, the dimensional version of Algorithm 3 is used to transfer the kinetic energy $F^{int}_{es\,\alpha}$ of each dimension α individually.

7.9.2 The static case

As for the previously investigated momentum transfer schemes, the first analysed test cases were for the static fluid. Compared to the momentum transfer by force, the velocity reverse scheme does not have any parameters that need to be chosen for static case. The employed settings for measuring and the number of time steps used for the simulations are summarised in Table 7.6.

Parameter	System	
	MDS-G	MDS-L
Momentum flux by stress	MTRV	
NP_b	1.0	
Equilibrium time steps	10,000	40,000
Simulation time steps	4,000,000	400,000
Measuring frequency (τ_m)	100	100
Averaged over (N_m)	10 000	1,000

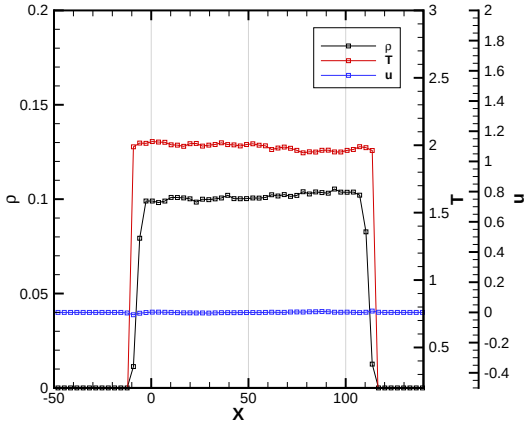
Table 7.6: Parameter settings for the DC-MTRV static test cases.

The reverse velocity scheme exhibits absolutely stable boundary conditions for static gases and liquids. Figure 7.36 shows the profiles and overall values of density, velocity and temperature for the gaseous and liquid static test cases. The overall values kept constant and the profiles were flat with sharp transitions on the boundary for density, velocity and temperature. No unwanted deviations were observed.

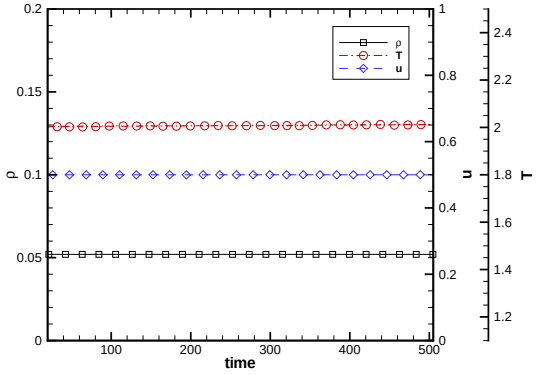
In order to asses the properties of the momentum transfer by the reverse velocity scheme (MTRV), it is useful to compare it to the momentum transfer by force scheme (MTF). To this end the density profiles of the left boundary are plotted for both methods into one diagram. Figure 7.37 shows the respective curves for the gaseous state on the left side and for the liquid case on the right side. The applied interaction depth of the MTF scheme was $NP_b = 1.0$ for the liquid case. For the gaseous state, two profiles with $NP_b = 1$ and $NP_b = 4$ are shown. In both diagrams, the profiles of the MTF scheme appear in black and the ones of the MTRV scheme in red. In the liquid case, both profiles are very similar, though the reverse velocity scheme generates a slightly sharper gradient. The difference is much more developed for the gaseous state; no asymptotic behaviour is perceivable for the reverse velocity scheme. This is a decisive advantage because it makes any need to confine atoms within the relaxation zone obsolete, even if confining would not encounter the same problems caused by the application of an external force.

7.9.3 Flow parallel to the boundary

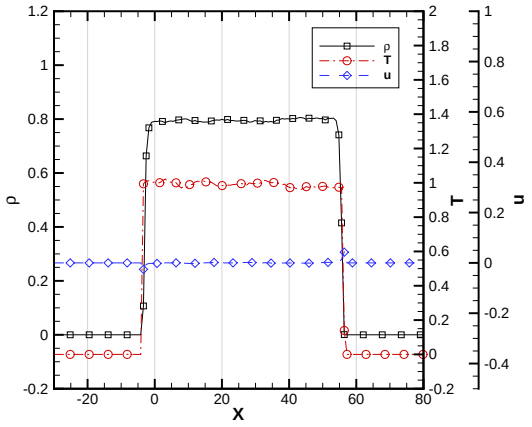
The Couette flow has been investigated in the same manner as for the previous flux imposition schemes. The general setup is explained in Section 6.2 and the employed test cases were again MDCF-G and MDCF-L1 and MDCF-L2 (q.v. Table B.1). The gaseous case



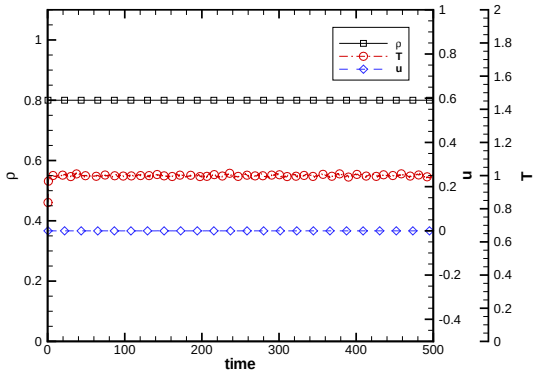
(a) Profiles for gaseous state (MDS-G, $\rho = 0.1$, $T = 2$).



(b) Overall values for gaseous state (MDS-G, $\rho = 0.1$, $T = 2$).



(c) Profiles for liquid state (MDS-L, $\rho = 0.8, T = 1$)

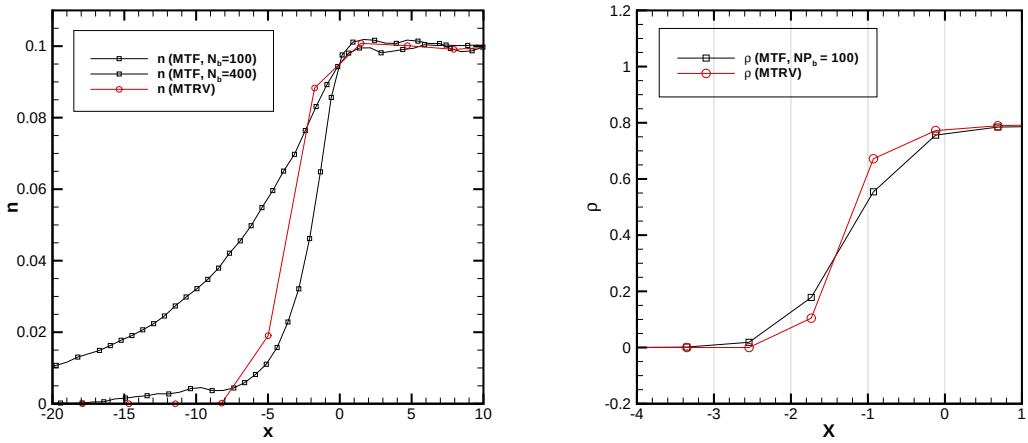


(d) Overall values for liquid state (MDS-L, $\rho = 0.8$, $T = 1$)

Figure 7.36: Profiles in x direction and overall values of density, velocity and temperature of the static fluid test case, applying the momentum transfer by the reverse velocity scheme (DC-MTRV).

MDCF-G was simulated with the velocities $u_e = \{1.0, 2.0\}$, the liquid case MDCF-L1 with $u_e = 0.5$ and the second, larger, liquid case MDCF-L2 with $u_e = 1.0$. An overview about the applied parameters is given in Table 7.7. The energy flux by stress was applied to the number of boundary atoms corresponding to $NP_b = 1.0$. The results were almost identical, when the energy flux was applied to values of up to $NP_b = 3.0$.

The profiles of density, velocity and temperature in the y dimension are plotted in Figure 7.38 for the performed test cases. The diagrams also include the analytical prediction of the temperature profile based on Equation (6.2.2), which is the orange coloured curve.



(a) Gaseous state ($\rho = 0.1, T = 2$) **(b)** Liquid state ($\rho = 0.1, T = 2$)

Figure 7.37: Comparison of density profiles between imposition of momentum flux by stress an through external force (MTF) and the reverse velocity scheme (MTRV).

Parameter	**System**	
	MDS-G	MDS-L1/MDS-L2
Momentum flux by stress	MTRV	
Crossed atoms	no confinement	
NP_b (for application of F_{es}^{int})	1.0	
Equilibrium time steps	10,000	100,000
Simulation time steps	5,000,000	500,000
Measuring frequency (τ_m)	100	100
Averaged over (N_m)	10,000	1,000

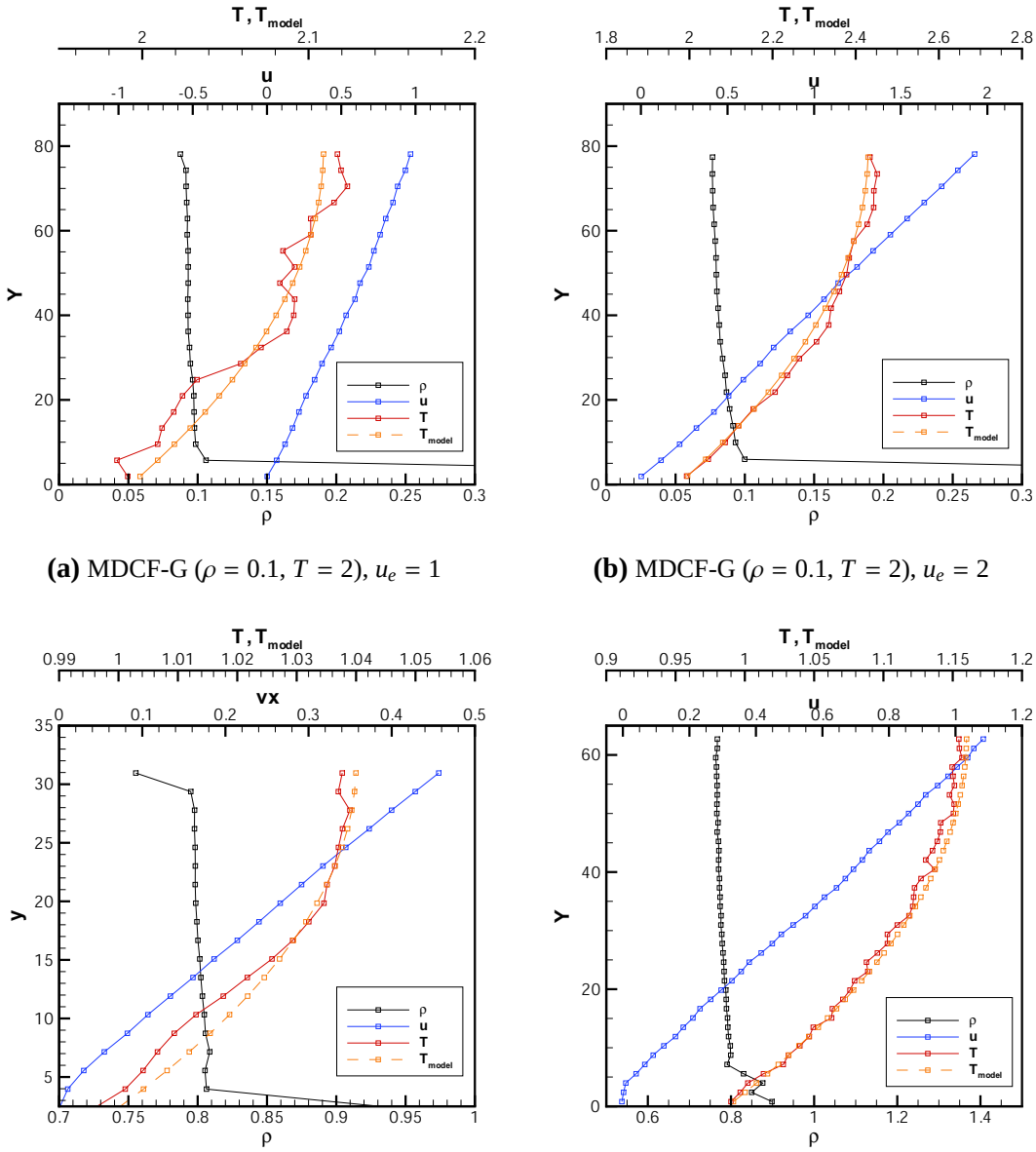
Table 7.7: Parameter settings for the DC-MTRV Couette flow test cases.

In all cases, the velocity profiles show the expected linear gradient[4] from $u = 0$ at $y = 0$ to the respective velocity $u = u_e$ at y_e. Also, the measured temperature profiles fit well to the analytical predictions. The density decreases slightly with increasing y due to the viscous heating. From the results it can be reasoned that the MTRV scheme is suitable for the standard situation of flow parallel to the boundary without any restrictions.

7.9.4 Flow perpendicular to the boundary

Both previously discussed schemes for the transfer of momentum by stress had major problems in handling the mass flux across the boundary. For this reason the reverse velocity scheme has been suggested. Using the reverse velocity scheme in conjunction with direct convective flux transfer, the values of the integrated fluxes, F_{mc}^{int}, $\boldsymbol{F}_{pc}^{int}$, F_{ec}^{int}, $\boldsymbol{F}_{ps}^{int}$ and F_{es}^{int},

[4]This is satisfied, because $Kn < 10^{-3}$ for the investigated cases.



(a) MDCF-G ($\rho = 0.1$, $T = 2$), $u_e = 1$

(b) MDCF-G ($\rho = 0.1$, $T = 2$), $u_e = 2$

(c) MDCF-L1 ($\rho = 0.8$, $T = 1$), $u_e = 0.5$

(d) MDCF-L2 ($\rho = 0.8$, $T = 1$), $u_e = 1$

Figure 7.38: Profiles for density, velocity and temperature of the Couette flow transferring the momentum by stress through the reverse velocity scheme (MTRV) for gaseous (upper figures) and liquid state (lower figures). T_{model} is the analytical solution for the temperature profile.

are imposed exactly. This is achieved by using the FCCS scheme for coupling both, the momentum and energy, fluxes by convection and by stress. The integrated energy flux by stress, F_{es}^{int}, was imposed onto the N_b outermost boundary atoms, with N_b corresponding to $NP_b = 1.0$. For the gaseous state, the positions of new atoms were generated randomly. For the liquid and solid material states, the modified USHER scheme was used. While for the liquid state, the basic USHER scheme could be applied as well, it cannot be used in the case of solids. This is because the implemented version of the USHER scheme is not

able to find a new position for an atom to be inserted into solid state material. In all cases, the respective outermost atom was selected for removal. The discrepancies in convective momentum and energy fluxes were automatically balanced through the FCCS scheme. A summary of the employed parameters is given in Table 7.8.

Parameter	System		
	MDFF-G	MDFF-L	MDFF-S
Momentum flux by stress	MTRV		
Crossed atoms	no confinement		
NP_b	1.0		
FCCS momentum coupling	yes		
FCCS energy coupling	yes		
Insertion algorithm	randomly	modified USHER	
Removal of atoms	outermost one		
Equilibrium time steps	10,000	10,000	30,000
Simulation time steps	100,000	200,000	50,000
Measuring frequency (τ_m)	10	10	10
Averaged over (N_m)	1,000	1,000	100

Table 7.8: Parameter settings for the DC-MTRV free flow test cases.

The free flow test cases were carried out for gaseous, liquid and solid material states. The gaseous state (system MDFF-G) was tested for the velocities $u = \{0.5, 1.0, 2.0\}$, the liquid state (system MDFF-L) for the velocities $u = \{0.5, 1.0\}$ and the solid state (system MDFF-S) for the velocity $u = \{0.1\}$. The equilibration times and measuring parameters are also listed in Table 7.8.

The flux boundary conditions, using the reverse velocity momentum transfer, produced stable simulations for all performed test cases. The overall values and profiles in x dimension of density, velocity and temperature are plotted in Figure 7.39 for the gaseous system MDFF-G. The overall values show a constant flow and the profiles give the desired rectangular shape without any major deviations. The small deviations observable at the lowest velocity of $u = 0.5$ are because the number of samples required to reduce the fluctuations even further was too high. Comparing the profiles with those obtained from the simulations using the MTF scheme (external force) for the momentum transfer (Figure 7.30), it can be seen that the MTRV scheme delivers more ideal results than the MTF scheme for flows perpendicular to the boundary.

In Figure 7.40, the overall values and profiles in x direction are shown for the liquid test case. As for the gaseous cases, the overall values indicate a constant flow. The density, velocity and temperature profiles have the rectangular shape. Only the velocity profiles show a small peak on the inlet (left side). This probably results from the insertion of new atoms and may be corrected by further improvements of the modified USHER scheme. As for the gaseous state, the profiles that have been obtained by using the MTRV scheme are qualitatively better than the profiles obtained by using the MTF scheme (compare Figure 7.31).

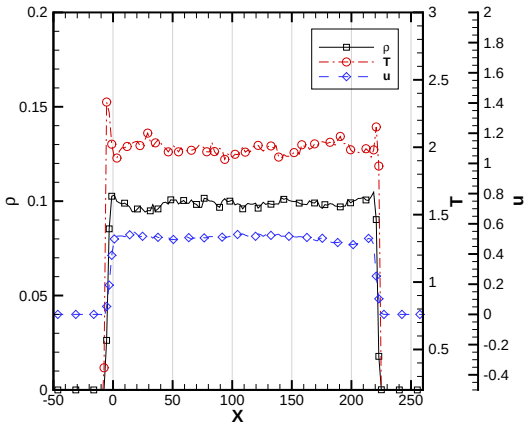
For applying the flux boundary conditions to solid material under the conditions of mass transfer across the boundary, the modified USHER scheme is essential to find the insertion site for new atoms. Using the original USHER scheme, it was not possible to find a single new site for inserting a new atom. The modified USHER scheme, on the other hand, performed very well for the solid and for the liquid states. For the solid state material ($\rho = 1.2$, $T = 1$), it required averagely 28.7 trials with 4.2 iterations per trial to find a new position with the desired energy; these are about 120 total iterations. For the liquid state ($\rho = 0.8$, $T = 1$), the modified USHER scheme required averagely 9.8 trials with 5 iterations per trial; hence about 49 total iterations to find a new insertion site.

In Figure 7.41, the overall values and profiles in x direction for the solid material moving with $u = 0.1$ are plotted. As for the previously presented gaseous and liquid cases, the flux boundary conditions produce a stable simulation with the profiles showing the desired rectangular shape. Contrary to the gaseous and liquid states, the solid material has an ordered crystal structure. For the LJ-material this is the face centred cubic (FCC) lattice, which has been used to initialised the system MDFF-S (compare Section 2.2.8). The initial structure after the equilibration is shown in Figure 7.42. The left and right boundaries are drawn as dashed lines and the relaxation zones are coloured green on both sides. To visualise the movement of the material, all atoms, whose position is initially $r_x < 10$, are coloured red while all other atoms are coloured blue. Figures 7.43 and 7.44 show the system at the later times $t = 81$ and $t = 150$. During this time, the red coloured part has moved from the left to the right side of the system. All atoms left of the red coloured ones have been inserted into the system according to the prescribed convective fluxes F_{mc}, $\boldsymbol{F}_{pc}$ and F_{ec}. It is noticeable that the structure of the newly generated material corresponds basically to the initial FCC structure; with the orientation being identical as well. This is remarkable because the modified USHER scheme does not insert atoms according to a preferred structure. Rather, the atoms that are inserted on the left side ‘condense’ on the prevailing structure, thus maintaining the initial structure in the system. Some small deviations from the ideal lattice can be observed in Figures 7.43 and 7.44, but these have not continued in the ‘upstream’ direction.

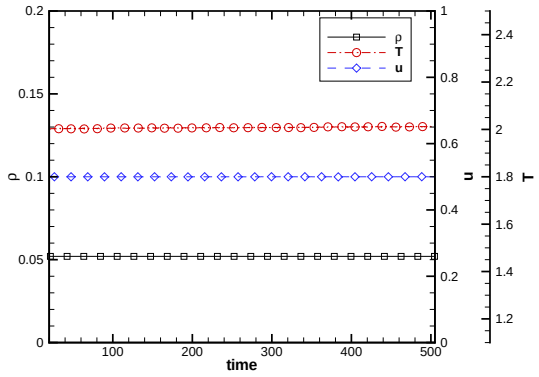
7.9.5 Summary of the results of DC-MTRV flux boundary conditions

The combination of the flux imposition schemes DC and MTRV was investigated for the standard flow situations of a static fluid, flow parallel to the boundary and flow across the boundary. The DC-MTRV flux boundary conditions enabled a stable simulation for all investigated cases. Comparing the results of momentum transfer by reverse velocity to the other investigated schemes shows that the MTRV scheme gives the best results for all tested standard situations. This applies in particularly to the flow across the boundary, where the MTRP and MTF momentum transfer schemes have major difficulties. The free flow test case was also tested for the solid material state. For this case the modified USHER scheme is essential in finding the insertion sites of new atoms, because the USHER scheme failed for solids.

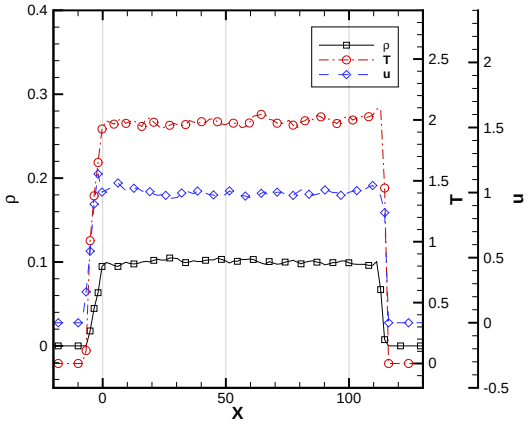
It should be noted that the tested velocities are relatively high and probably beyond the speeds that will occur in most simulation scenarios. For example, assuming the solid material in the last simulation to be copper, the velocity of $u = 0.1$ corresponds to $79\,{}^{\mathrm{m}}/_{\mathrm{s}}$. For the gaseous cases, the speed of $u = 2$ is just above speed of sound of the LJ material at that state ($a = 1.8$ at $\rho = 0.1$ and $T = 2$). However, the purpose of this work is to test different flux boundary condition scheme in order to discover their characteristics, application range and limitations.



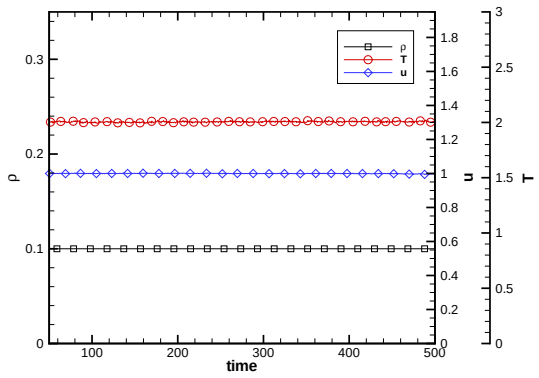
(a) Profile of ρ, u and T for $u = 0.5$



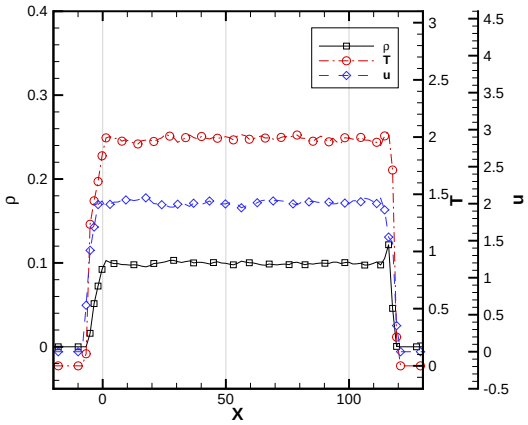
(b) Overall values of ρ, u and T over time for $u = 0.5$



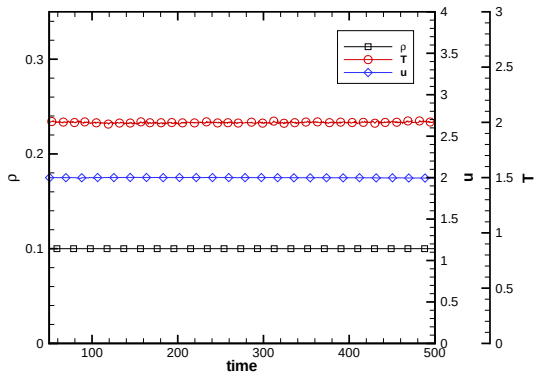
(c) Profile of ρ, u and T for $u = 1.0$



(d) Overall values of ρ, u and T over time for $u = 1.0$

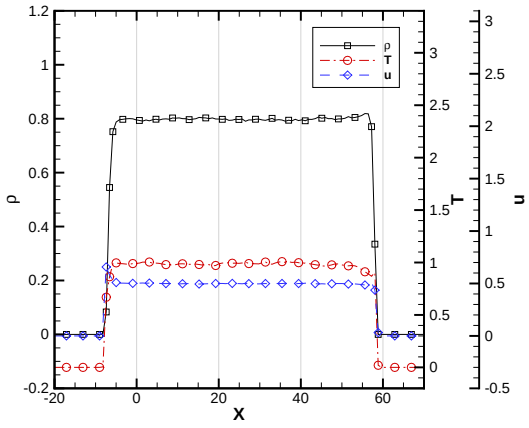


(e) Profile of ρ, u and T for $u = 2.0$

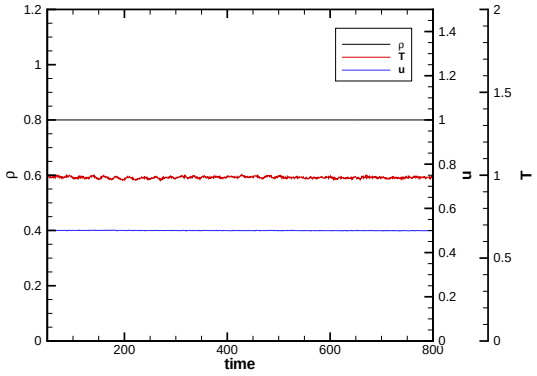


(f) Overall values of ρ, u and T over time for $u = 2.0$

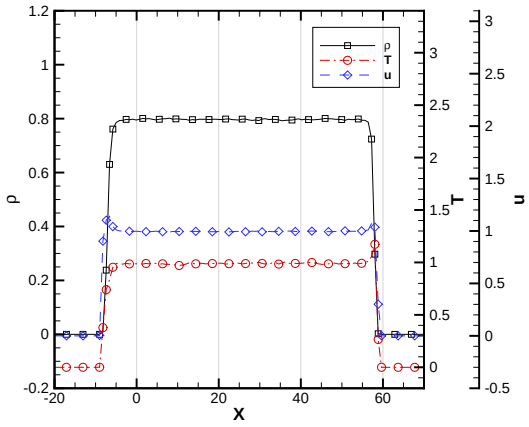
Figure 7.39: Profiles in x direction and overall values of density, velocity and temperature of gaseous free flow test case MDFF-G ($\rho = 0.1$, $T = 2$) for direct convection and momentum transfer by reverse velocity (MTRV).



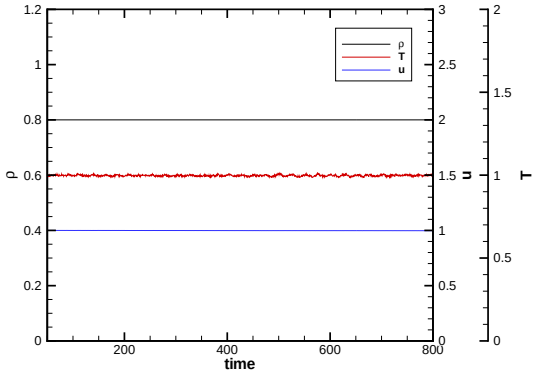
(a) Profile of ρ, u and T for $u = 0.5$



(b) Overall values of ρ, u and T over time for $u = 0.5$

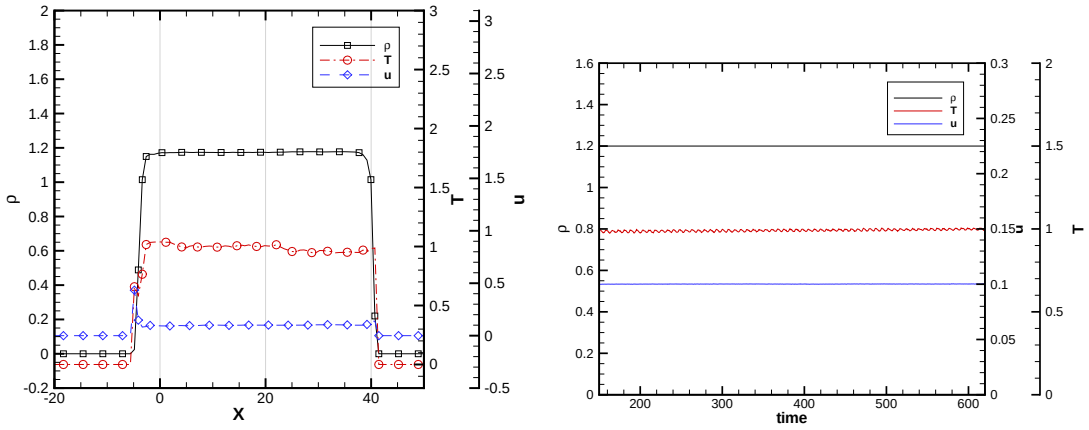


(c) Profile of ρ, u and T for $u = 1.0$



(d) Overall values of ρ, u and T over time for $u = 1.0$

Figure 7.40: Profiles in x direction and overall values of density, velocity and temperature of liquid free flow test case MDFF-L ($\rho = 0.8$, $T = 1$) for direct convection and momentum transfer by reverse velocity (MTRV).



(a) Profile of ρ, u and T for $u = 0.1$ **(b)** Overall values of ρ, u and T over time for $u = 0.1$

Figure 7.41: Profiles in x direction and overall values of density, velocity and temperature of solid free flow test case MDFF-S ($\rho = 1.2$, $T = 1$) for direct convection and momentum transfer by reverse velocity (MTRV).

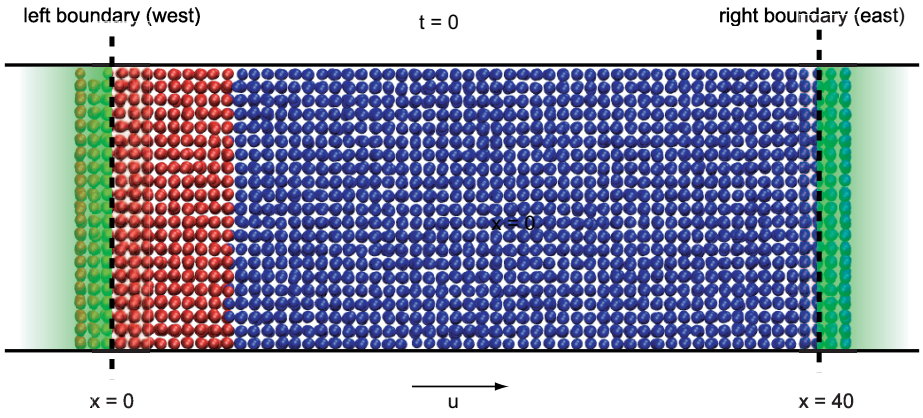


Figure 7.42: Visualisation of the free flow test case with solid material at time $t = 0$.

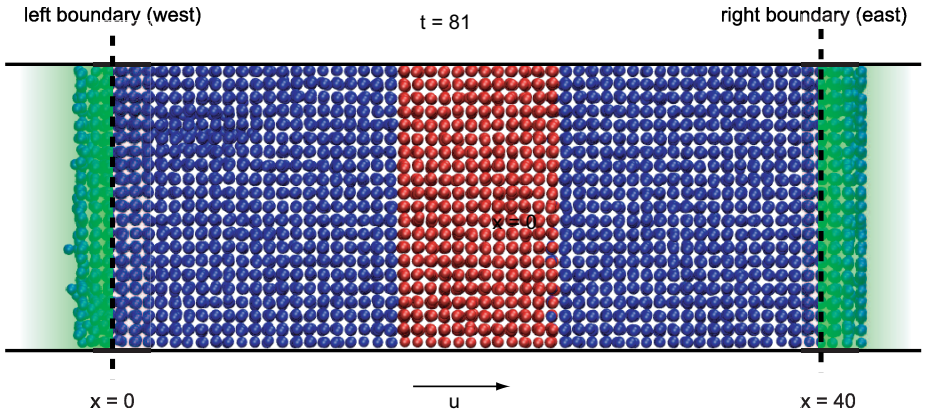


Figure 7.43: Visualisation of the free flow test case with solid material at time $t = 81$.

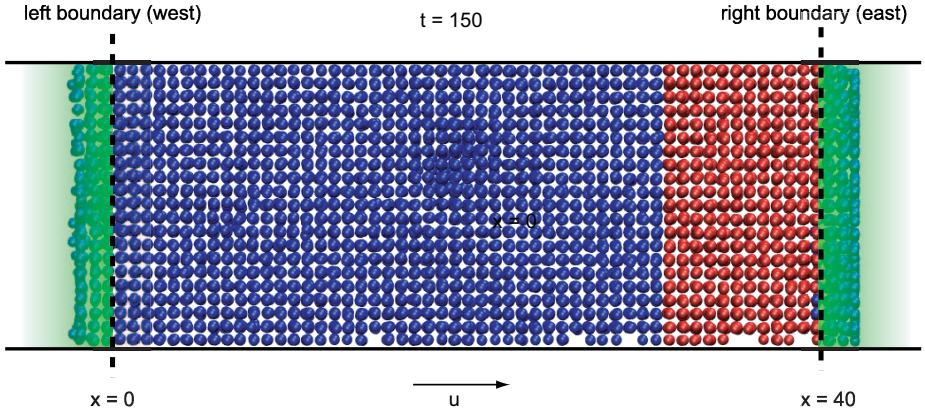


Figure 7.44: Visualisation of the free flow test case with solid material at time $t = 150$.

7.10 Indirect convection and reverse velocity (IC-MTRV)

One of the issues related to direct convective flux imposition is that all the fluxes of mass, momentum and energy are strictly prescribed. If these do not correspond to the actual situation of the molecular system on the boundary, then the fluxes cannot be imposed and the boundary conditions break down. This problem emerges especially when the situation in the molecular domain changes rapidly and the fluxes are not adapted timely to fit to the new situation. Indirect convection circumvents this problem elegantly by only prescribing the momentum fluxes by stress and the properties of the convecting atoms. The idea and function of the indirect convective flux scheme has been described in Section 7.4, where two schemes have been proposed: (1) Separated momentum fluxes (ICS) and (2) Combined momentum fluxes (ICC).

During the investigation of the indirect convective flux schemes, many possible variations have been tested. However, only the two mentioned versions have actually produced stable flux boundary conditions. The discussion will start with version 1 (ICS), where the momentum fluxes by convection and by stress are separated. This is followed by the investigation of the version 2 (ICC), where the momentum fluxes by convection and by stress are combined. For reasons of time, the indirect convective flux imposition has only be investigated in conjunction with the reverse velocity scheme (MTRV) for transferring the momentum fluxes due to stress. Also, the indirect convective flux schemes have only been tested for static and the free flow test cases. However, it can be assumed that if these are applicable in both situations, they will also provide good results for flows parallel to the boundary. Proof of this assumption must be left to further investigations on MD flux boundary conditions.

7.10.1 Static case (ICS-MTRV) with separated momentum fluxes

As for the previously investigated schemes, the first and least demanding test case is the static fluid. Whereas no convective flux occurred when using the direct convective flux imposition scheme, for the indirect convective flux imposition there will be a continuous in- and outflow at the boundary due to thermal fluctuations. However, the net flux must be zero, which is one of the points that is considered for testing the IC scheme for a static fluid. The static fluid test case has been performed for the gaseous and liquid systems MDS-G and MDS-L. The complete set of parameters is given in Table 7.9.

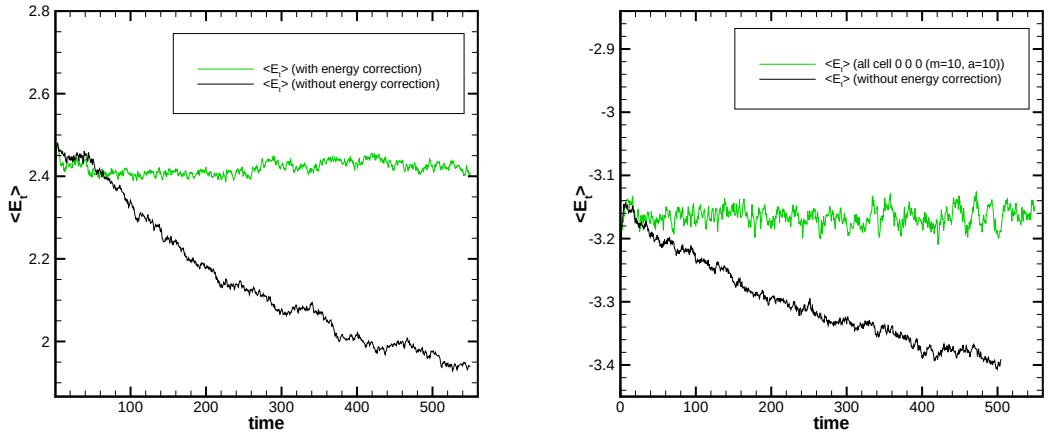
For indirect convection with separated momentum fluxes by convection and by stress (ICS), the imposed energy flux by stress is prescribed by the continuum solver, F_{es}^{int}, and FCCS coupling is disabled for both momentum and energy. The velocity vector of inserted atoms is generated based on the convective fluxes F_{mc}, $\boldsymbol{F}_{pc}$ and F_{ec}. Effectively, this means that the transferred momentum and energy fluxes by stress correspond to those prescribed by the continuum solver. Only, the convective fluxes are subject to the flow conditions and are not previously defined.

Parameter	System	
	MDS-G	MDS-L
Convective flux imposition	indirect	
Insertion algorithm	randomly	modified USHER
New atoms properties	ZNM based on F_{mc}, $\boldsymbol{F}_{pc}$ and F_{ec}	
Removal of atoms	outermost one	
Momentum correction for removed atoms	no	
Energy correction for removed atoms	yes	
Momentum flux by stress	MTRV	
Transferred energy flux by stress	F_{es}^{int}	
FCCS momentum coupling	no	
FCCS energy coupling	no	
NP_b	1.0	
Equilibrium time steps	10,000	10,000
Simulation time steps	100,000	100,000
Measuring frequency (τ_m)	10	10
Averaged over (N_m)	10,000	10,000

Table 7.9: Parameter settings for the ICS-MTRV (separated momentum flux) test cases.

As described in Section 7.4.3, IC-schemes require a correction of the energy that is subtracted when atoms are removed from the reservoir region, because the energy of the outermost atoms is in average higher than the energy of atoms inside the virtual boundary cell. To demonstrate the need of the energy correction, the static gaseous and liquid systems (MDS-G and MDS-L) were simulated with and without energy correction. The evolution of the average atomic energy $\langle E_t \rangle$ is plotted in Figure 7.45. Because the fluids are static, the energy level must stay constant over time, with the expected average energy of $\langle E_t \rangle = e_t n = 2.445$ in the gaseous and $\langle E_t \rangle = e_t n = -3.15$ in the liquid case. Both plots show that the energy levels with energy correction fluctuate around the expected level. Without energy correction, the average atomic energy starts to decrease immediately. The decrease stops at a certain energy level, because the energy of inserted atoms corresponds to the correct energy level (in the static case the atoms inserted correspond approximately to the removed atoms). This prevents a further drop in the energy level. Because even a small drop in energy would violate the stipulations for the flux boundary conditions, it is necessary to use the energy correction scheme for removed atoms.

In Figure 7.46, the profiles in x direction and overall values of density, velocity and temperature are plotted for the gaseous (MDS-G) and liquid system (MDS-L). The first noticeable difference to direct convective flux schemes is the fluctuation of the overall density. Whereas for direct convection, the number of atoms is constant, it is allowed to vary for indirect convective schemes. In both cases, the density fluctuates around the desired value of $\rho = 0.1$ and $\rho = 0.8$ for the gaseous and liquid cases respectively. The overall velocity in x direction is close to zero and temperature stays at the desired levels. That the overall levels of overall density, velocity and temperature are correct, is also a confirmation that the values of the static pressure applied through the momentum fluxes by stress are correct. This refers also to the properties, i.e. $\langle E_p \rangle$ and $\langle E_k \rangle$, of the inserted atoms.



(a) gaseous state, ($\rho = 0.1, T = 2$) **(b)** liquid state, ($\rho = 0.8, T = 1$)

Figure 7.45: Energy per atom $\langle E_t \rangle$ over time for the gaseous and liquid static cases with ICS-MTRV flux boundary conditions, with and without energy correction for removed atoms.

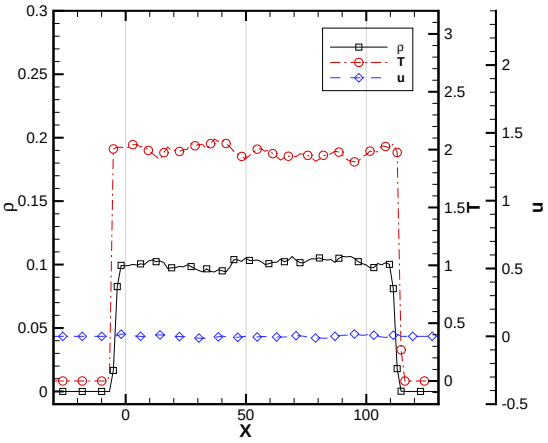
For indirect convective fluxes, the values of the actually implemented mass, momentum and energy fluxes by convection are contained within the variables F_{mc}^{int}, $\boldsymbol{F}_{pc}^{int}$ and F_{ec}^{int}. These have been checked after the simulation for any net flux. In both cases, gaseous and liquid, these were close to zero, confirming that the state is really static and no creeping flow is present.

The profiles in the left hand side diagrams of Figure 7.46 show in the gaseous, as well as in the liquid, state the desired rectangular shape with the sharp discontinuous transitions at the boundaries.

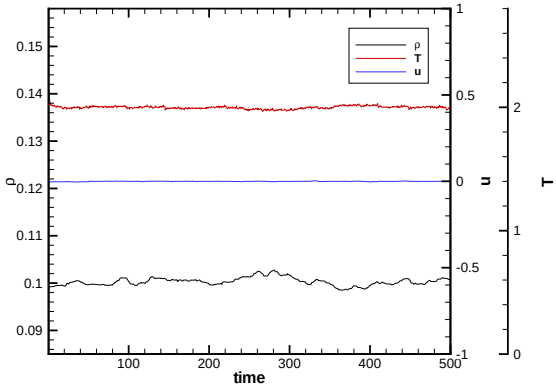
7.10.2 Flow perpendicular to the boundary with separated momentum fluxes (ICS-MTRV)

For the free flow test cases using indirect convective boundary conditions with separated momentum fluxes (ICS-MTRV), the same parameters settings as for the static cases have been used (q.v. Table 7.9). The test case was performed for the gaseous state with a velocity of $u = 1.0$. The liquid case, which poses more difficulties, the ICS-MTRV flux boundary conditions were tested for three velocities, $u = \{0.5, 1.0, 2.0\}$.

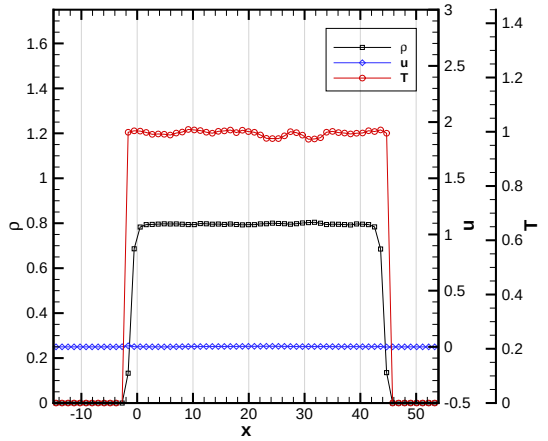
In Figure 7.47, the results for the gaseous case are plotted. The profiles are given for two different points in time, $t = 500$ and $t = 1300$. Due to the time averaging, these measurements do not correspond exactly to the given time, but are representative of the time interval of 50 time units before the given time. The overall values of density, velocity and temperature correspond to the expected levels and indicate a stable behaviour. However,



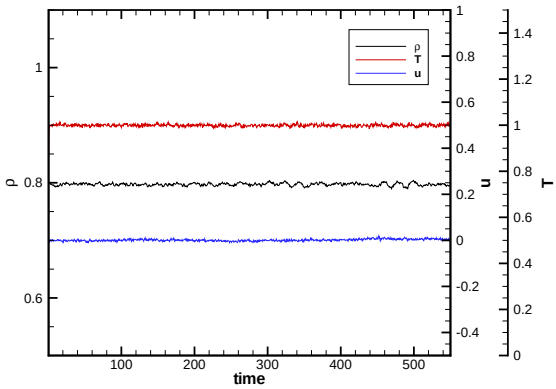
(a) Profiles for gaseous state (MDS-G).



(b) Overall values for gaseous state (MDS-G)



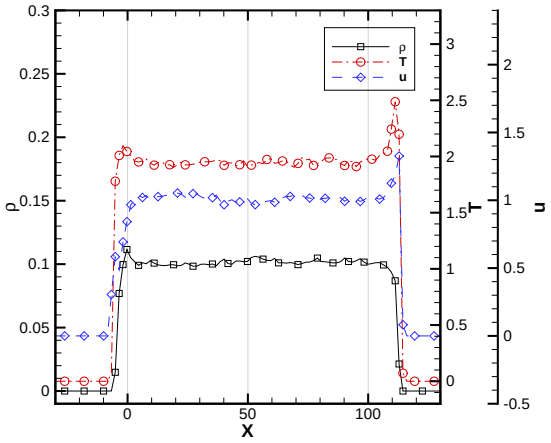
(c) Profiles for liquid state (MDS-L).



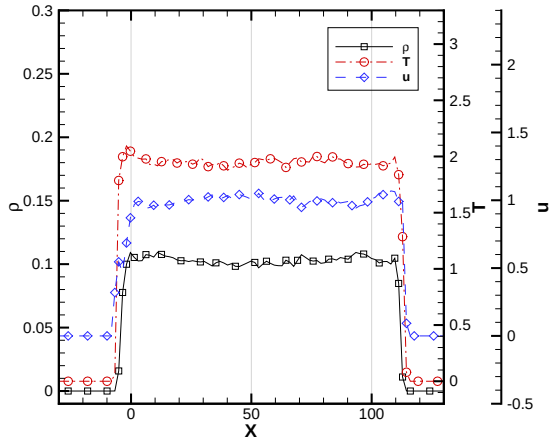
(d) Overall values for liquid state (MDS-L)

Figure 7.46: Profiles in x direction and overall values of density, velocity and temperature of the static fluid test case when applying the ICS-MTRV scheme for the flux boundary conditions.

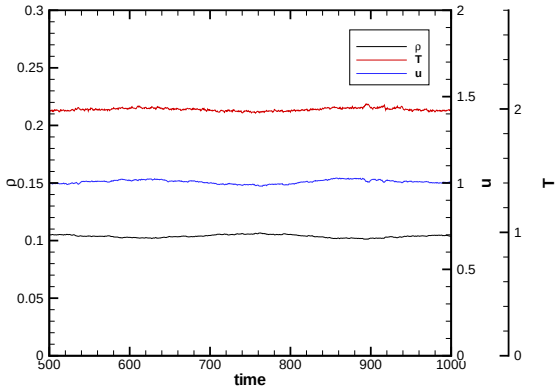
the profiles for the time $t = 500$ show undesired peaks in velocity and temperature on the boundary that is subject to outflow conditions (right side of Figure 7.47a). This is accompanied by a drop in density. To investigate this behaviour further, the simulation was extended to 1,000,000 time steps. The system continued to show a stable flow with $\rho = 0.1$, $u = 1.0$ and $T = 2$. Measuring the profiles in x direction on several successive time intervals revealed an alternating behaviour of the observed peaks in the velocity and temperature. In Figure 7.47b, no peaks are observable. During the simulated period, profiles changed repeatedly from states with and without peaks. It should also be noted that the peaks occur at the outermost site of the relaxation zone. The density is very low at this site, thus the peaks represent only a small number of atoms.



(a) Profiles of ρ, u and T with $u = 1.0$ at $t = 500$.



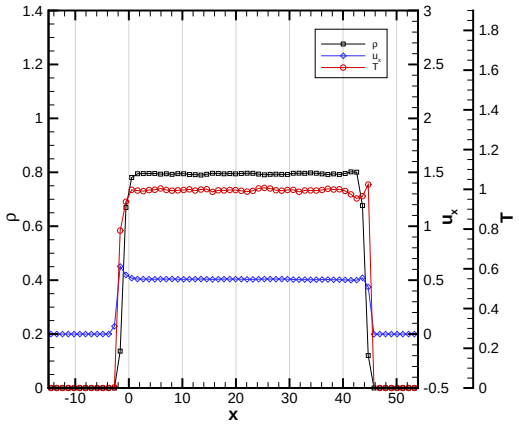
(b) Profiles of ρ, u and T with $u = 1.0$ at $t = 1300$.



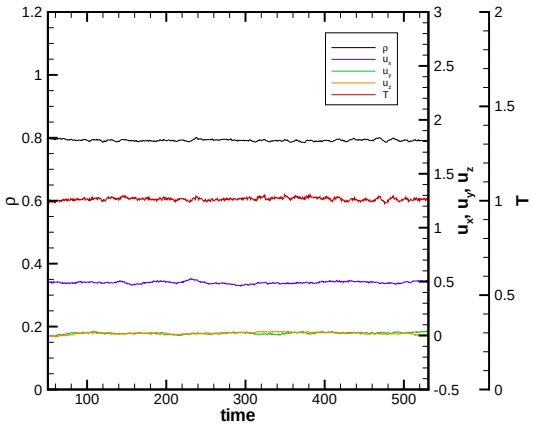
(c) Overall values of ρ, u and T with $u = 1.0$

Figure 7.47: Profiles in x direction and overall values of density, velocity and temperature of the free gaseous test case MDS-G ($\rho = 0.1$, $T = 2$) when applying the ICS-MTRV scheme for the flux boundary conditions.

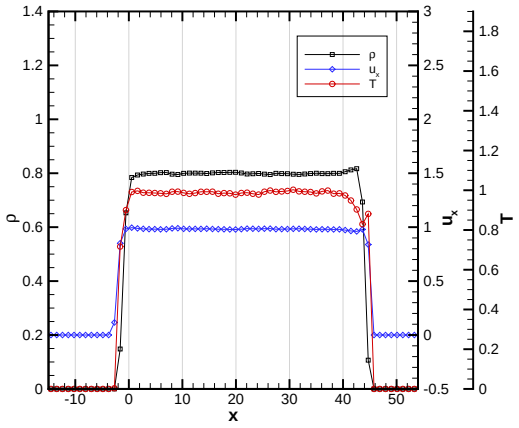
The results for the liquid cases are summarised in Figure 7.48. The ICS-MTRV flux boundary conditions enabled a stable simulation for the three tested velocities. Again, the profiles are slightly distorted under outflow conditions; otherwise, they show the desired rectangular profile. The profile plots also include the velocities in the other two dimensions, y and z. These have been included here to show that the flow is only in the x direction.



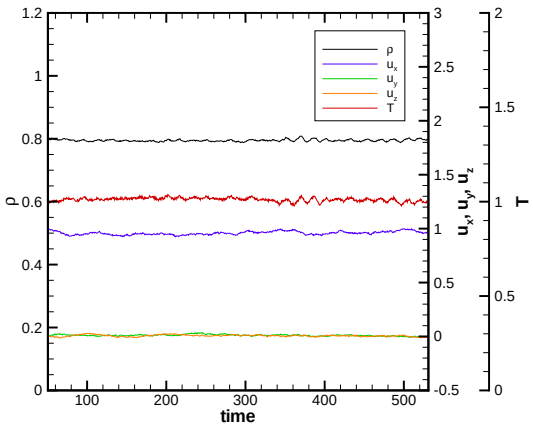
(a) Profiles of ρ, u and T with $u = 0.5$.



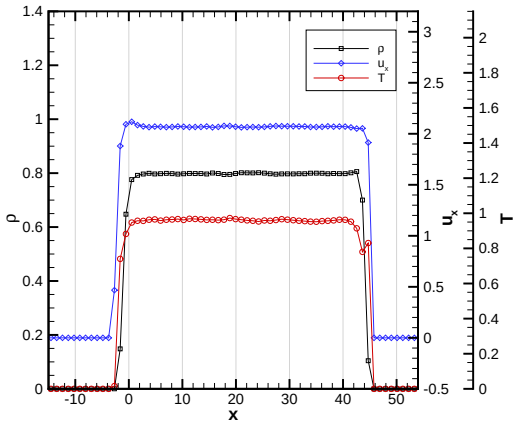
(b) Overall values of ρ, u and T with $u = 0.5$



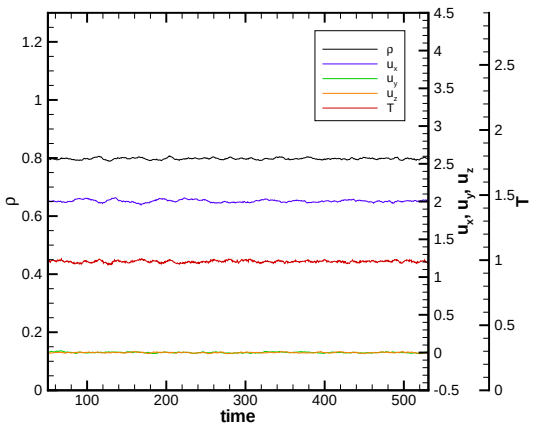
(c) Profiles of ρ, u and T with $u = 1.0$.



(d) Overall values of ρ, u and T with $u = 1.0$



(e) Profiles of ρ, u and T with $u = 2.0$.



(f) Overall values of ρ, u and T with $u = 2.0$

Figure 7.48: Profiles in x direction and overall values of density, velocity and temperature of the free liquid test case MDS-L ($\rho = 0.8$, $T = 1$) when applying the ICS-MTRV scheme for the flux boundary conditions.

7.10.3 Static case with combined momentum fluxes (ICC-MTRV)

For indirect convection using combined momentum fluxes (ICC-MTRV), the FCCS momentum coupling is activated for inflow conditions. Inflow conditions are considered when $F_{mc} > 0$. No coupling is activated for outflow conditions, i.e. if $F_{mc} < 0$. The momentum and energy correction for removed atoms is activated too. The energy that is transferred through energy flux by stress is not precalculated. It is computed every time step based on the momentum flux by stress, $\boldsymbol{F}_{ps}$ and the velocity $\boldsymbol{u}_C$ of the boundary cell C. The amount of energy is considered separately for each dimension $\alpha = \{x, y, z\}$ and calculated by

$$E_{ps_\alpha} = F_{ps_\alpha} u_{C\alpha} \delta t \,, \tag{7.10.1}$$

where δt is the length of the time step. This mimics what would happen if the momentum flux by stress was applied through an external force. Thus, one may try to use the MTF scheme to transfer the momentum and energy fluxes $\boldsymbol{F}_{ps}$ and F_{es}. However, in doing so, the momentum correction of the removed atoms could no longer be transferred energy neutrally. Therefore, the MTRV scheme seems to be the better choice in conjunction with the ICC scheme.

The transferred energy by stress must also be reported back to the continuum solver along with the implemented convective fluxes. In this respect, the ICC scheme is the most extreme scheme. It enforces flux boundary conditions, based only on the momentum fluxes, $\boldsymbol{F}_{pc}$ and $\boldsymbol{F}_{ps}$ and the properties of the inflowing material, which are calculated from the convective fluxes (q.v. Section 7.4.2). This is the minimum amount of information needed to establish flux boundary conditions. All other fluxes are the result of the simulation and are generated by the flux boundary conditions, depending on the situation in the relaxation zone and boundary cell.

The remaining parameters are identical to the previously discussed scheme with separated momentum fluxes (ICS-MTRV). The summary of the employed parameters is given in Table 7.10.

In Figure 7.49, the profiles in x direction and overall values of density, velocity and temperature are plotted for the static test case for the gaseous system MDS-L and the liquid system MDS-L. For both states, the simulation is stable and the profiles are close to the ideal rectangular shape. Similar to the other investigated indirect convection scheme, the overall values for density show small fluctuations around the expected values of $\rho = 0.1$ for the gaseous and $\rho = 0.8$ for the liquid case. The overall velocities are zero and the temperatures are at the correct level.

7.10.4 Flow perpendicular to the boundary with combined momentum fluxes (ICC-MTRV)

An important advantage of the ICC-MTRV scheme is that the simulations do not have to be initialised with the expected flow velocity. Rather, the test system can be equilibrated

Parameter	System	
	MDS-G	MDS-L
Convective flux imposition	indirect	
Insertion algorithm	randomly	modified USHER
New atoms properties	based on F_{mc}, $\boldsymbol{F}_{pc}$ and F_{ec}	
Removal of atoms	outermost one	
Momentum correction for removed atoms	yes	
Energy correction for removed atoms	yes	
Momentum flux by stress	MTRV	
Transferred energy flux by stress	based on $\boldsymbol{F}_{ps}$ and $\boldsymbol{u}_C$	
FCCS momentum coupling ($F_{mc} < 0$)	yes	
FCCS momentum coupling ($F_{mc} > 0$)	no	
FCCS energy coupling	no	
NP_b	1.0	
Equilibrium time steps	10,000	10,000
Simulation time steps	100,000	100,000
Measuring frequency (τ_m)	100	100
Averaged over (N_m)	10 000	1,000

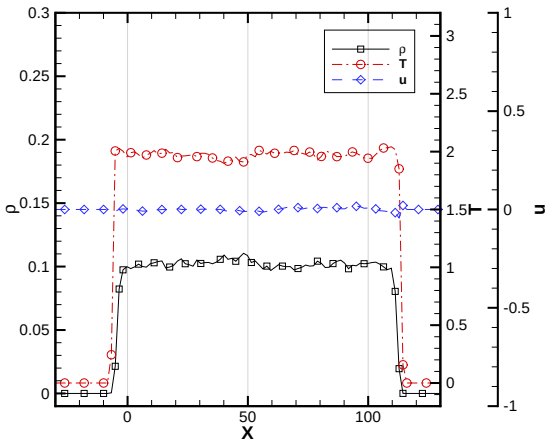
Table 7.10: Parameter settings for the ICC-MTRV (combined momentum flux) test cases.

to zero velocity, i.e. static fluid, and the applied ICC-MTRV flux boundary conditions accelerate the system to the velocity which the fluxes have been calculated for. Thus, all free flow simulations for the ICC-MTRV scheme have started from zero overall velocity. The parameter settings used for the free flow test cases are identical to those in Table 7.10, which have been used for the static test case.

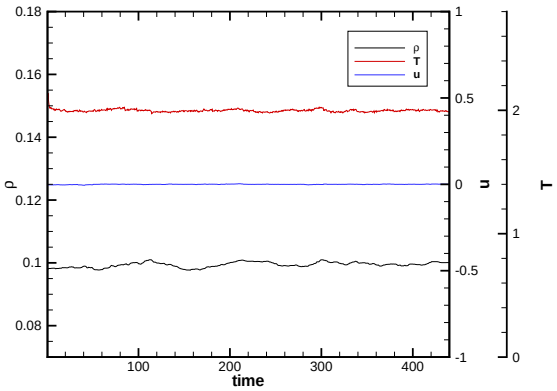
For the gaseous state, the system MDFF-G1 has been investigated for $u = 1$. The profiles in x direction at the end of the simulation and the overall values of density, velocities and temperature are plotted in Figure 7.50. The overall values in the right diagram show that the velocity starts at $u_x = 0$, rises to the expected value of $u_x = 1$ and stays constant at this level for the rest of the simulation. The velocities in the other two dimensions, u_y and u_z are zero for the entire simulation. The density and temperature start at $\rho = 0.1$ and $T = 2$. While the velocity is growing, the inflow of material from on the left boundary causes the density and temperature to rise temporarily until the constant flow in the system is established.

The profiles in Figure 7.50a have been measured during the last part of the simulation, where the constant flow had been established. The profiles of velocity and temperature show the desired rectangular profile. Only the density profile has an unwanted peak in the reservoir region under outflow conditions (see right boundary).

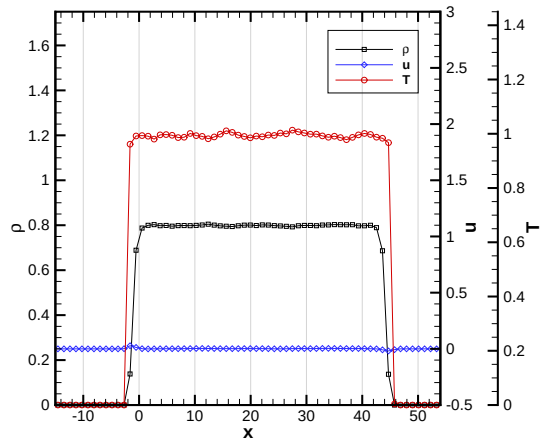
The free flow test cases for the liquid system MDFF-L1 have been performed for the velocities $u_x = \{0.5, 1.0, 2.0\}$. The profiles in x direction and overall values of density, velocity and temperature are given in Figure 7.51. As for the gaseous test case, the systems have been started at $u_x = 0.0$. For all velocities, the systems reached the desired velocities.



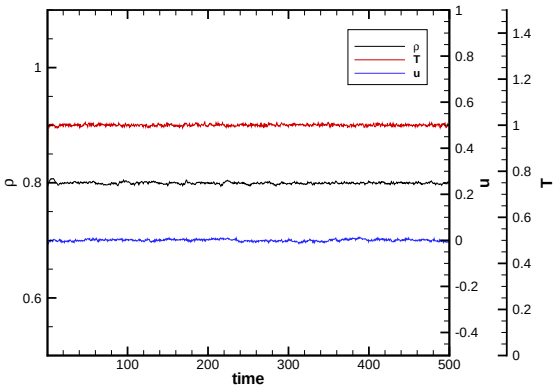
(a) Profiles for gaseous state (MDS-G).



(b) Overall values for gaseous state (MDS-G)



(c) Profiles for liquid state (MDS-L).

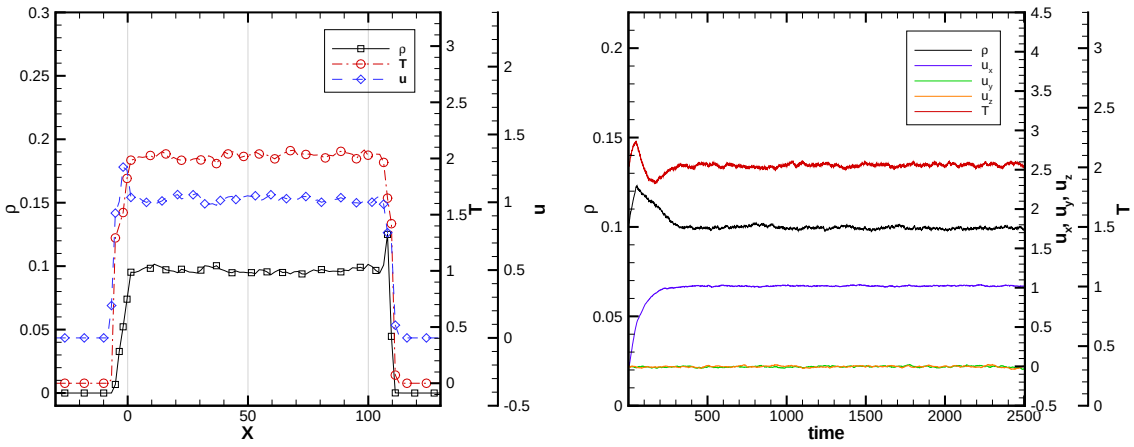


(d) Overall values for liquid state (MDS-L)

Figure 7.49: Profiles in x direction and overall values of density, velocity and temperature of the static fluid test case when applying the ICC-MTRV scheme for the flux boundary conditions.

Only for the case with $u_x = 1.0$, is the velocity slightly too low ($u_x \approx 0.96$), by 5 %. All profiles show the expected rectangular shape for each of the three quantities: density, velocity and temperature.

It should be noted that the correction of momentum when removing atoms seems not strictly necessary for the liquid cases. The results were almost identical when the correction of the momentum transferred along with the removed atoms was deactivated. This is probably because of the high density where the high interactivity between atoms produces a much smoother distribution of momentum. Thus, average momentum in the relaxation zone is very similar to the one in the boundary cell.



(a) Profiles of ρ, u_x, u_y, u_z and T with $u = 1.0$. **(b)** Overall values of ρ, u_x and T with $u = 1.0$

Figure 7.50: Profiles in x direction and overall values of density, velocity and temperature of the gaseous free flow test case MDFF-G1 ($\rho = 0.1$, $T = 2$) when applying the ICC-MTRV scheme for the flux boundary conditions.

7.10.5 Summary of and additional points on indirect convective flux imposition

The indirect convective flux imposition has been investigated in conjunction with the reverse velocity scheme for separated momentum fluxes (ICS-MTRV) and combined momentum fluxes (ICC-MTRV). The schemes have been tested for static and free flow (flow normal to the boundary) situations. The two schemes performed well for the tested velocities in both, gaseous and liquid, states. Both schemes, the ICS-MTRV and ICC-MTRV can therefore be used as flux boundary conditions.

Even if indirect flux imposition schemes are flux coupling schemes in the C → M direction, they cannot be used as pure state coupling schemes in the M → C direction. This is because not the complete set of fluxes, which is provided by the continuum solver, is actually enforced, but only a selected part of the fluxes. The remaining fluxes are imposed depending on the molecular situation on the boundary. The values of these fluxes are integrated over the cell face and in time.[5] They must be transported back to the continuum solver along with the states of the boundary cells.

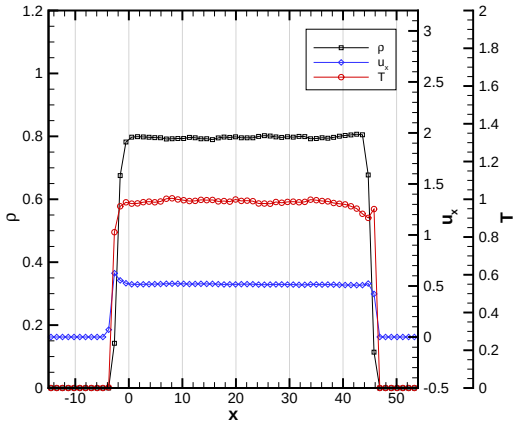
Table 7.11 gives a comparison of the three investigated convective imposition schemes: DC, ICS and ICC, in terms of which fluxes is used as in or output variable for the flux boundary conditions. Obviously, the DC uses the complete set of fluxes for input and no integrated fluxes are delivered back. Neglecting the heat fluxes, the ICS scheme uses only the fluxes by stress as input data. The ICC scheme can be considered as most 'extreme'

[5] In the implemented version of the flux boundary conditions, this is done automatically and the fluxes actually imposed are stored in the integrated flux variables $F^{int}_{...}$.

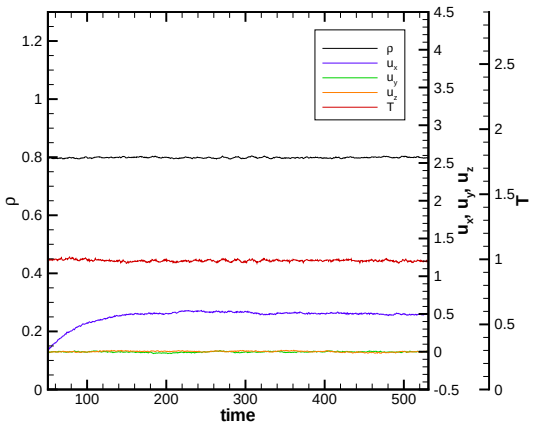
Convective Scheme		DC	ICS	ICC
input fluxes (C → M)	convective	F_{mc}, $\boldsymbol{F}_{pc}$, F_{ec}		
	by stress	$\boldsymbol{F}_{ps}$, F_{es}	$\boldsymbol{F}_{ps}$, F_{es}	$\boldsymbol{F}_{pc} + \boldsymbol{F}_{ps}$
	heat	F_q	F_q	F_q
output fluxes (M → C)	convective		F^{int}_{mc}, $\boldsymbol{F}^{int}_{pc}$, F^{int}_{ec}	F^{int}_{mc}, F^{int}_{ec}
	by stress			F^{int}_{es}
	heat			

Table 7.11: Overview over the in- and output fluxes of different convective schemes.

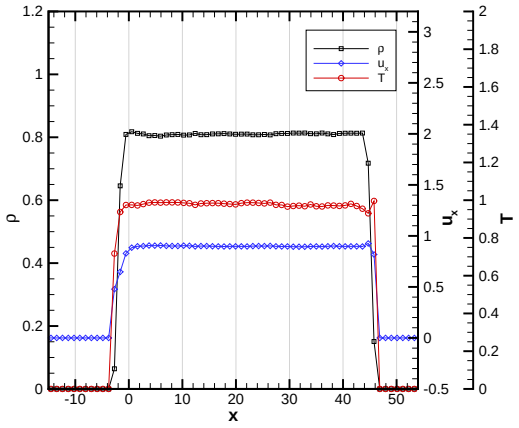
scheme, as it only imposes the sum of the momentum fluxes directly. These can already be summed up by the continuum solver ($\boldsymbol{F}_{pc+ps} = \boldsymbol{F}_{pc} + \boldsymbol{F}_{ps}$); the ICC scheme then imposes only a single flux, the combined momentum flux $\boldsymbol{F}_{pc+ps}$. This has a decisive advantage of making it the most robust scheme, because it relies on the least assumptions about the molecular situation on the boundary. All other schemes require the state of the molecular system on the boundary to correspond to the calculated values. This may cause numerical instabilities if the prescribed fluxes are not adapted fast enough to the changing molecular state on the boundary. Because the ICC scheme makes no assumption about the molecular situation on the boundary, rather it only applies the prescribed pressure, it is stable by construction. Therefore, the ICC-MTRV scheme appears to be the most suitable scheme for the usage in a HSI for coupling of MD and CFD.



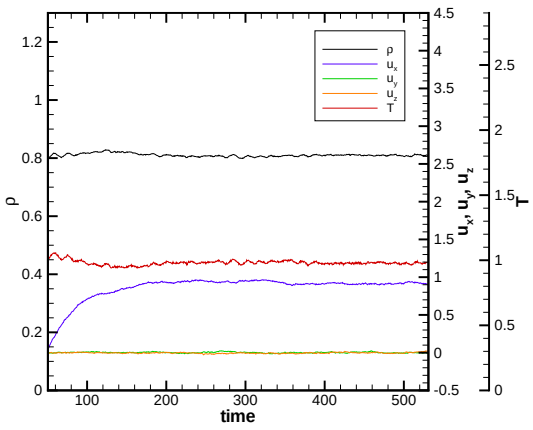
(a) Profiles of ρ, u, u_y, u_z and T with $u = 0.5$.



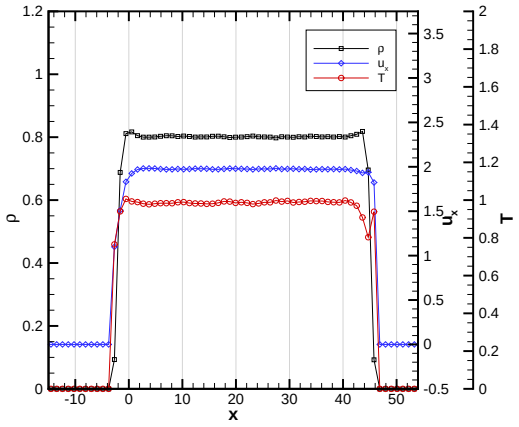
(b) Overall values of ρ, u and T with $u = 0.5$



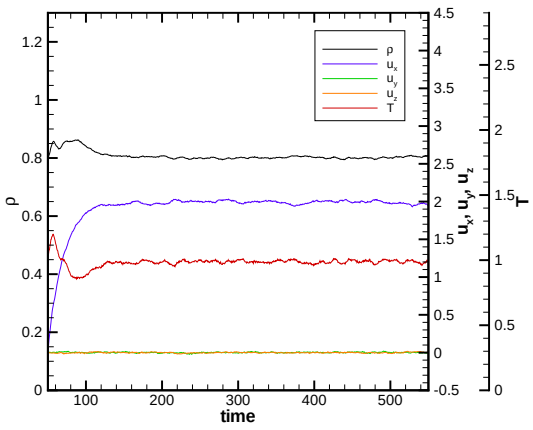
(c) Profiles of ρ, u, u_y, u_z and T with $u = 1.0$.



(d) Overall values of ρ, u and T with $u = 1.0$



(e) Profiles of ρ, u, u_y, u_z and T with $u = 2.0$.



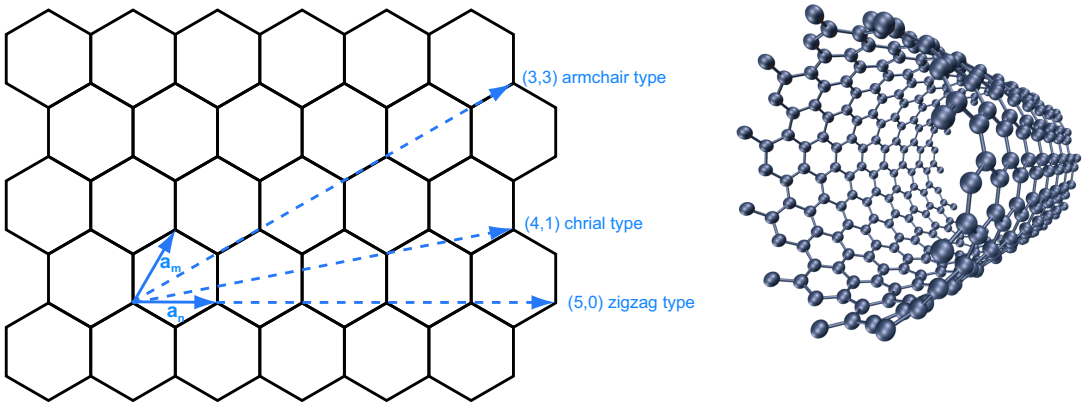
(f) Overall values of ρ, u and T with $u = 2.0$

Figure 7.51: Profiles in x direction and overall values of density, velocity and temperature of the liquid free flow test case MDS-L ($\rho = 0.8$, $T = 1$) when applying the ICC-MTRV scheme for the flux boundary conditions.

7.11 Pressure boundary conditions using ICC-MTRV

The ICC-MTRV flux imposition scheme can also be useful for boundary conditions for pure MD simulation. So far, most MD simulations are carried out using periodic boundary conditions or walls, made of fixed or tethered atoms, or reflecting planes. The ICC-MTRV flux boundary conditions can be applied equivalently to pressure boundary conditions in a CFD simulation. One defines a pressure difference across the molecular domain of $\delta P = F_{pc+ps_r} - F_{pc+ps_l}$, where F_{pc+ps_l} is for example the momentum flux on the left boundary and F_{pc+ps_r} the momentum flux on the right boundary. The flow that develops between the two sides depends on the pressure drop, δP, and the properties of the atoms that are inserted under inflow conditions. Such pressure boundary conditions are expected to be very useful for MD simulations.

To demonstrate the application of the ICC-MTRV flux imposition scheme for pressure boundary conditions the flow field around carbon nanotubes (CNTs) has been simulated. A CNT is a graphene sheet rolled up to a cylindrical tube. It usually has a diameter of few nanometers, but can reach a length of several micrometers [163]. Because of their strength and flexibility, CNTs are potentially useful for a large number of applications. One is to weave a fabric or net, made of CNTs, which could be used for filtering purposes [170]. The design of such a CNT-filter would require the understanding of the flow past an array of CNTs. To investigate these flows, the pressure boundary conditions with ICC-MTRV could be used; and to demonstrate this, the flow of Argon (Ar) past two parallel aligned CNTs, which is one of the basic structures of a CNT-filter, was simulated.



(a) Graphene sheet with three chiral vectors **(b)** Armchair CNT (10,10)

Figure 7.52: Carbon nanotube structure

The basic structural unit of a CNT is a hexagon of carbon (C) atoms. A single layer graphene sheet is formed by arranging the carbon hexagons in a two dimensional lattice. This is shown in Figure 7.52a, where the vectors $\boldsymbol{a}_n$ and $\boldsymbol{a}_m$ are the basis vectors of the lattice by which the carbon hexagon must be shifted for creating the sheet. The CNT is

formed by rolling up the sheet into a tube. Depending on the angle between the rolling direction and the orientation of the hexagon, different types of CNTs emerge. Usually, the type is designated by the chiral vector, (n, m), that defines the rolling direction based on the choice of the integers n and m. (Figure 7.52a). Three type are possible; the structure can be zigzag ($m = 0$), armchair ($n = m$) or chiral ($n \neq m$). The length of the chiral vector determines the diameter of the CNT. For the simulation here, a (10,10) CNT was used. One section of the (10,10) CNT is shown in Figure 7.52b, where the carbon atoms are rendered as spheres and the covalent bonds as sticks between them.

To simulate the flow of Argon through a CNT-filter, two infinite (10,10) CNTs were aligned parallel to each other in z direction. The setup is shown in Figure 7.53. The carbon atoms of the CNTs are rendered in black and Argon atoms in red. The case is very similar to the free flow test case. The difference is that, here, two CNTs are placed within the flow. The size of the simulation box was 255 Å × 82 Å × 10.7 Å. Periodic boundary conditions are applied in the y and in the z dimension. The ICS-MTRV flux boundary conditions were applied in the x dimension on the eastern and western sides. Because the CNTs are aligned in the z dimension, which is periodical, the beginning of the each CNT is connected to its end, making it practically infinite. Within the x,y plane, the CNTs are placed at (68 Å, 27 Å) and (68 Å, 55 Å). The rest of the simulation domain was filled with Argon atoms at a density of 1.344 $^{\text{gram}}/_{\text{cm}^3}$ using the random filling scheme (q.v. Section 2.2.8).

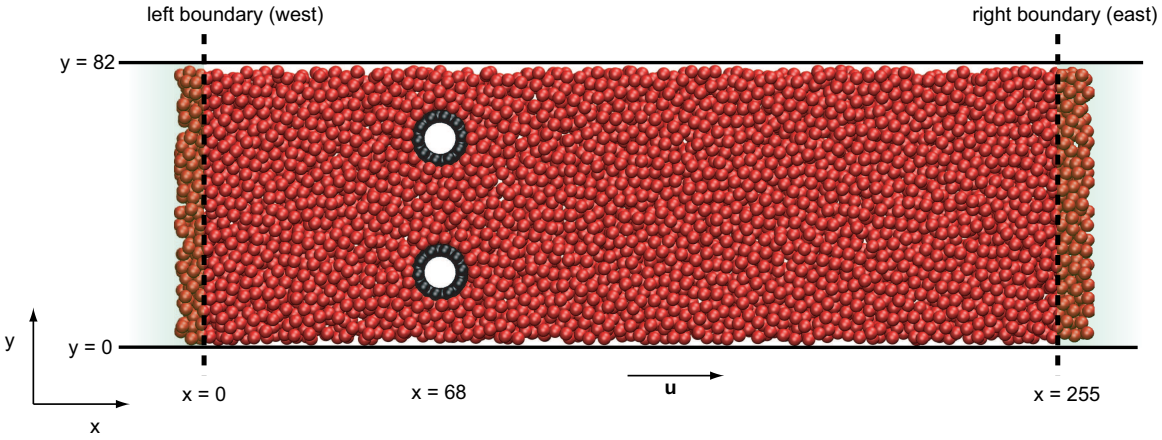


Figure 7.53: Setup of the simulation for across two parallel carbon nanotubes

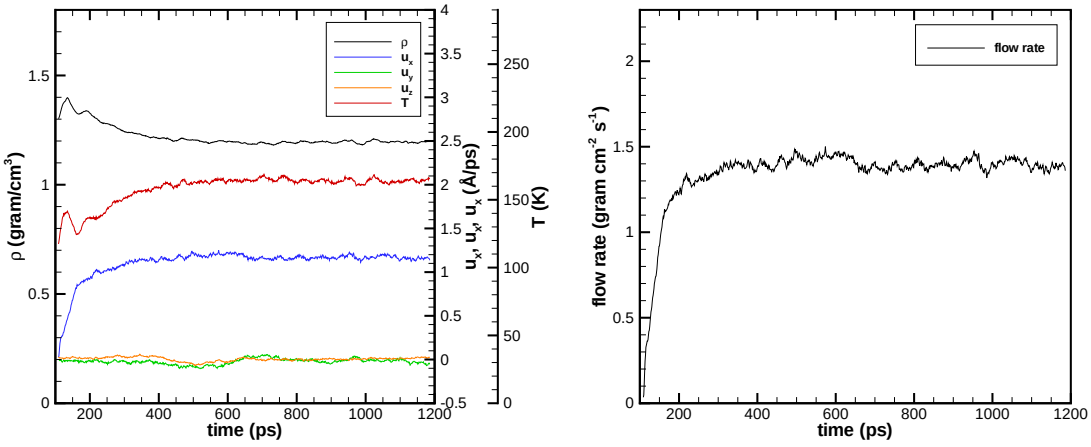
The CNTs were modelled through bond stretching and bond bending potentials. For C=C bond stretching, the Morse potential (Equation 2.1.31) and for C=C=C bond bending the harmonic potential (Equation 2.1.33) was used. Interactions of Ar-Ar and C-Ar were modelled by the LJ potential (Equation 2.1.23). The potential coefficients have been taken from Reference [181] and are compiled in Table 7.12. A time step of $\delta t = 10$ fs was used. All other parameters and settings were identical to the free flow test case.

After the initialisation, the system was equilibrated for 10,000 time steps to a temperature of 120 K using velocity scaling. During the equilibration phase a pressure of 68 $^{\text{N}}/_{\text{mm}^2}$,

Potential	Type	Coefficients
C=C bond stretching	Morse	$r_{C=C0} = 1.418\,\text{Å}$ $\beta_{C=C} = 2.187\,{}^{1}/_{\text{Å}}$ $D_{C=C} = 114.38\,{}^{\text{kcal}}/_{\text{mol}}$
C=C=C bond bending	harmonic	$\theta_{C=C=C0} = 50.35\,{}^{\text{kcal}}/_{(\text{mol rad})}$ $k^{bend}_{C=C} = 120\,\text{deg}$
Ar-Ar interaction	LJ potential	$\epsilon_{Ar-Ar} = 0.238\,{}^{\text{kcal}}/_{\text{mol}}$ $\sigma_{Ar-Ar} = 3.4\,\text{Å}$
Ar-Ar interaction	LJ potential	$\epsilon_{C-Ar} = 0.136\,{}^{\text{kcal}}/_{\text{mol}}$ $\sigma_{C-Ar} = 3.4\,\text{Å}$

Table 7.12: Potentials and coefficients for the simulation of Argon flow around CNTs.

which is the static pressure of Argon at the chosen temperature and density, was applied on the western and eastern boundaries. To keep the CNTs at their initial positions, the overall momentum of their atoms was set to zero at each time step. Following the equilibration, the simulation was run for 100,000 time steps. The pressure on the western side (inlet) was increased to $160\,{}^{\text{N}}/_{\text{mm}^2}$. Because of the pressure difference a flow in positive x direction developed. This is shown in Figure 7.54a; the velocity rises from $0\,{}^{\text{Å}}/_{\text{ps}}$ to saturate at $2.156\,{}^{\text{Å}}/_{\text{ps}}$. The temperature increases slightly, while the density decreases due to the viscous heating. Multiplying velocity and density, the total flow rate can be obtained (ρu_x), which is plotted in Figure 7.54b.



(a) Density, velocities and temperature.

(b) Flow rate.

Figure 7.54: Systems overall values in time.

Contour plots of the density and velocity in x direction were generated, using the

particle-to-mesh method. To this end, a cartesian mesh of 500×100 grid points was defined within the simulation domain. The CIC scheme was used to assign the atoms to the grid points. The calculations were performed every 10th time step and averaged over 5,000 samples. Figures 7.55 and 7.56 show the fields of density and velocity in x direction. The plots give the situation within the last 50,000 time steps, where the flow had already converged to a steady state. Streamlines have been included in both plots. From the density contours, it can be seen that the fluid atoms are aligned in rings that form several layers around the CNTs. Relating this to the velocity contours, one can see that the fluid atoms move fastest along the lines of the highest densities.

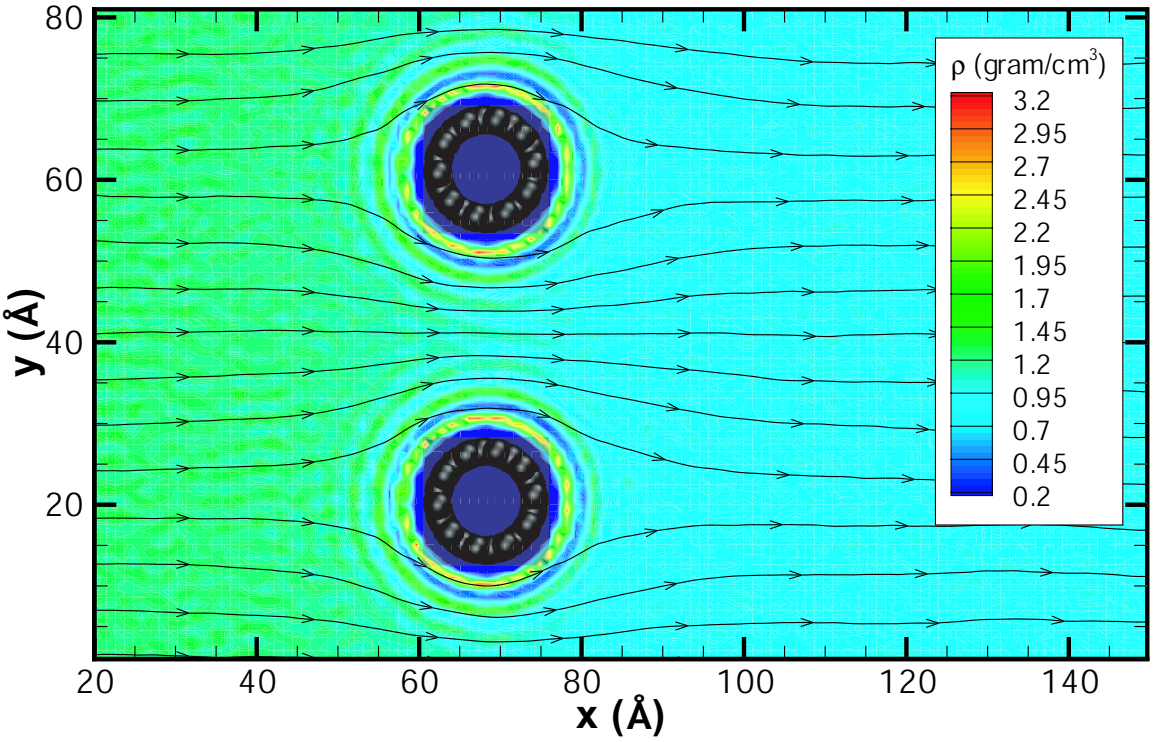


Figure 7.55: Density field of the Argon flow past two CNTs.

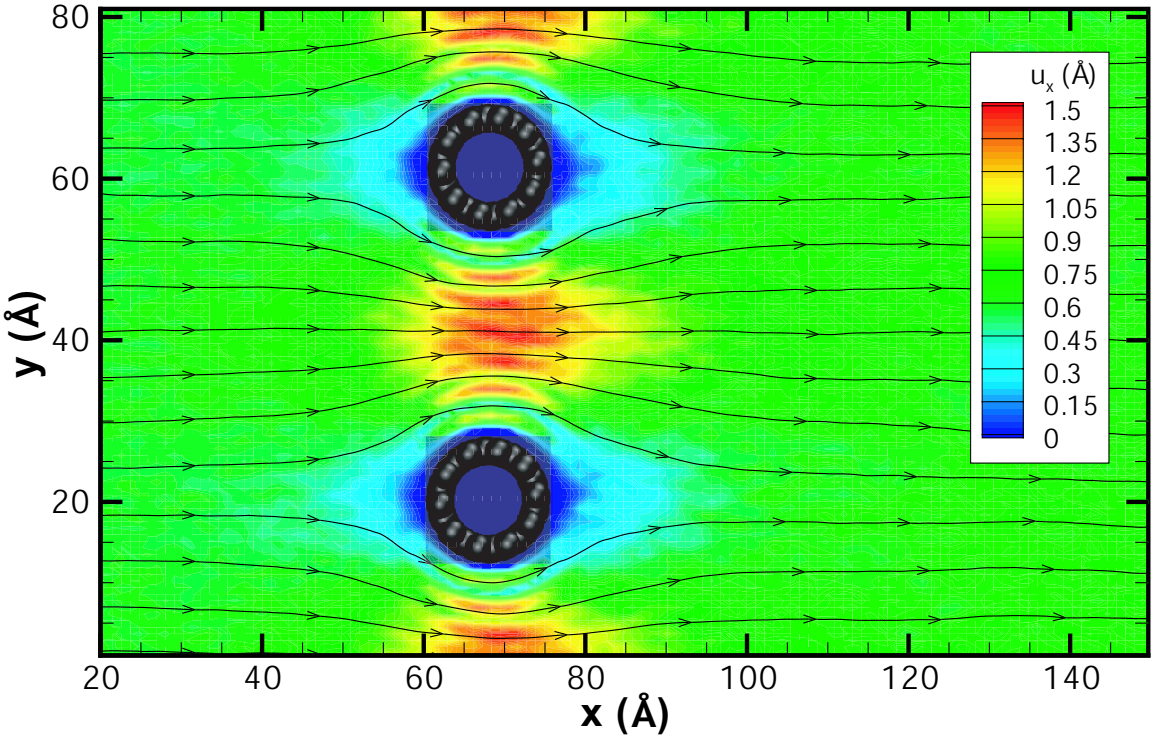


Figure 7.56: Field of velocity in x direction, u_x, of the Argon flow past two CNTs.

CHAPTER 8

Conclusions and Future Work

8.1 Conclusions

The MD flux boundary conditions for the usage within a MD-CFD hybrid method were investigated in detail. The heart of the MD flux boundary is the imposition of the convective mass, momentum and energy fluxes as well as the momentum and energy fluxes by stress. This PhD is the first structured approach for a direct comparison between different flux imposition schemes. Three schemes for convective flux imposition were considered: the direct convective flux imposition (DC), indirect convective flux imposition with separated momentum fluxes (ICS) and indirect convective flux imposition with combined momentum fluxes (ICC). For imposing the momentum fluxes by stress, four schemes were considered: momentum flux imposition by reflective plane (MTRP), momentum flux imposition by external force (MTF), momentum flux imposition by direct velocity change (MTDVC) and momentum flux imposition by reverse velocity (MTRV). The MTDVC scheme has identical characteristics as MTF. Energy fluxes by stress are determined by the employed momentum flux imposition scheme.

Several combination of convective flux imposition and momentum flux by stress imposition schemes were investigated. The combinations were DC-MTRP, DC-MTF, DC-MTRV, ICS-MTRV and ICC-MTRV were used as flux boundary conditions and tested for standard flow situations at different speeds in gas and liquid phase. Since the motivation for this work is in investigating the functionality in principle, the generic Lennard-Jones 12-6 was used for all performed simulations. It is reasonable to assume that conclusions also apply for other material models. The employed standard flow situations were: static fluid, flow parallel to the boundary and flow normal to the boundary. For testing the standard situation of flow parallel to the boundary, a Couette flow was chosen. To simulate the flow normal to the boundary, a free flow test case was used, where in and outflow conditions were tested at the same time.

Static situations

It has been found that all investigated combinations of flux imposing schemes were suitable for the static situation, for gaseous and liquid phases. The combination DC-MTRV was also successfully tested for the solid phase. The momentum flux imposition by an external force (MTF), which has been used in all previously published hybrid MD-CFD

methods, was investigated in more detail. The application of the static pressure by an external force generates a relaxation zone in which the density falls from its bulk value to zero. Because fluid properties change with the density, a sharp transition is preferred. The employed weighting function for the distribution of the external force $\boldsymbol{F}^{ext}$ was the most simplest one: $g(\boldsymbol{r}_i, \boldsymbol{v}_i) = 1$, where the characteristics of the relaxation zone only depend on the number of atoms onto which $\boldsymbol{F}^{ext}$ is applied.

A number of conclusions for the MD flux boundary conditions can be drawn from the investigation carried out:

- It has been found that the most suitable parameter to describe the number of atoms onto which the fluxes are applied is the interaction depth given in atomic layers, NP_b, which is a non-dimensional parameter.
- The relaxation zone width, δl_b, increases with increasing interaction depth, NP_b. Based on an energy balance, a theoretical model for the characteristics of the relaxation zone has been derived. Using the model, δl_b can be obtained from NP_b by Equation (7.8.17):

$$\delta l_b = \frac{NP_b\, n_0^{\frac{2}{3}}\, k_B T}{P} \ln \frac{0.05}{0.95}.$$

  Though the model does not take into account interatomic interactions and should therefore strictly only be applicable for gases, the relaxation zone width, δl_b, predicted by Equation (7.8.17) agrees well with the measured values for gaseous ($\rho = 0.1$, $T = 2$), supercritical ($\rho = 0.8$, $T = 2$) and partly liquid ($\rho = 0.8$, $T = 1$) states. Equation (7.8.17) can be used to calculate the expected relaxation zone depth when applying an external force onto the boundary. In fact, this is not only useful in the context of hybrid MD-CFD methods, but in all situations where an external force field is applied to an MD simulation.

  The model also predicts the relaxation density profile on the boundary by Equation (7.8.16):

$$n_{model}(x) = 0.95 n_0\, e^{-\frac{PA_b}{2N_b k_B T} x}\,, \tag{8.1.1}$$

  Excellent agreement of Equation (7.8.16) with measured density profile was found for the gaseous state ($\rho = 0.1$, $T = 2$). No agreement was found for the liquid and supercritical state. A linear relationship is suspected between the inverse maximum gradient, img_b, and $NP_b n_0^{1/3}$.
- Theoretical considerations and test simulations carried out in this work have shown that any attempt to reduce the size of the relaxation zone by using confinement schemes are not energy neutral when used in conjunction with the momentum transfer through an external force (MTF). This is caused by the potential energy field that is generated by the external force, $\boldsymbol{F}^{ext}$. Thus, such confinement schemes should not be used, which poses a deficiency of the MTF scheme, since some higher energetic atoms can travel 'far' into vacuum beyond the boundary.

- Using the MTRV scheme for transferring the momentum flux by stress, the density profile on the boundary is very sharp; thus, the MTRV scheme should be preferred.

Flow parallel to the boundary

For the situation of flow parallel to the boundary, no net mass transfer takes place. This flow situation has been investigated on the example of the Couette flow, with the boundary aligned in flow direction, for all direct convective flux combinations DC-MTRP, DC-MTF and DC-MTRV. For the simulation of Couette flows, all direct convective schemes gave good results. The measured profiles of density, velocity and temperature agreed well with theoretical models in all performed test cases.

Flow normal to the boundary

The situation of flow normal to the boundary is the most difficult to handle for flux imposition schemes. Based on the simulations that have been carried out, a number of conclusion can be made:

- The combination DC-MTRP was only stable for slow flows normal to the boundary, where the external kinetic energies are insignificant compared to the internal kinetic energies.
- The combination DC-MTF can produce stable flux boundary conditions. However, it is not possible to transfer the prescribed energy flux F_{es} nor can the actual transferred energy be measured. The reason is that the external force F^{ext} used in the MTF momentum transfer scheme generates an energy field. When inserting or removing atoms through inflow or outflow conditions the energy field is changed and, therefore, the energy of each atom onto which the external force acts is changed too. For this reason the DC-MTF scheme is not the ideal combination for MD flux boundary conditions.
- The momentum transfer by reverse velocity was developed as part of this PhD to overcome the problem encountered in the application of the external force. The DC-MTRV combination gave the best results for free flow conditions of all direct convective schemes.
- Both investigated combinations for indirect convective flux imposition schemes, ICS-MTRV and ISS-MTRV, showed a good performance for all free flow tested cases. The ICC-MTRV combination is the only flux scheme for which the free flow test case could be started from a system at rest.
- The ICC-MTRV flux imposition scheme can be used to generate pressure boundary conditions for MD simulations.
- For solid phases, the USHER scheme [39] was not able to find insertion sites with the desired energy level. The modified USHER, which has been developed as part of this work, finds new insertion sites on the surface of a molecular system's boundary. The new scheme proved its effectiveness and efficiency in the performed test cases for the solid phase of the DC-MTRV flux boundary conditions.

Most suitable combination of flux imposition schemes

Finally, the question of which scheme is the most suitable for MD flux boundary conditions can be answered. After having tested different combinations it can be reasoned that the flux imposition scheme ICC-MTRV is the most suitable scheme for a number of reasons:

- Because of its construction, the scheme is the most stable one and will give in any case a solution based on the applied pressure and properties of the inserted atoms.
- The indirect convective flux imposition avoids elegantly any problems that may arise from small differences between the prescribed fluxes and the actual flow situation on the boundary.
- The position of the interface is always fixed through the indirect mass flux imposition.
- The momentum flux imposition by reverse velocity gives the sharpest thus best density profiles on the boundary.
- The modified USHER scheme inserts efficiently atoms on the surface of the boundary.

Using the MTRV scheme, the relaxation zone is sharper than the ones obtained by the MTF scheme.

8.2 Future research

1. Investigate further potential combinations of flux imposition schemes, such as indirect convective fluxes with momentum transfer by force (ICS-MTF).
2. Perform further tests with the flux boundary conditions, to investigate their performance when subjected to dynamic flow situations, such as transversal or longitudinal waves.
3. The currently implemented version of the MD-flux boundary conditions is only sequential. However, practical multiscale simulations require the use of massive parallel computing power. Therefore, it is necessary to develop a parallelised version of the MD flux boundary conditions. Unfortunately, there are some fundamental problems. One is that the MD codes, like LAMMPS, divide the entire simulation domain into smaller cubes, which are each computed by one processor. Using flux boundary conditions, these cubes will overlap with the cells of the continuum grid on the boundary. A number of complicated arrangements between the two grids are conceivable, which will cause problems for imposing the fluxes while still maintaining the speed advantage of the parallel code. Additionally, inserting or removing atoms near the boundary of the processor cubes must be concerted carefully with adjacent processor cubes because of potential situations in which two atoms are inserted independently by two processors within interaction distance. This would cause, at least, inaccurate energy transfers or, at worst, a crash of the simulation through extremely high energies of very close inserted atoms.

4. This PhD is part of a research program on multiscale modelling using hybrid MD-CFD methods. After, the MD flux boundary conditions have been successfully verified for standard situations with constant continuum fluxes, the next logical step is performing a fully coupled MD-CFD simulation. The envisaged test case is the dynamic friction problem. Here, two blocks of solid material slide along each other with such a high velocity that the material at the friction interface melts. This causes the appearance of several phenomena such as the dynamic variance of the friction coefficient, material mixing, shear waves and plastic deformation. No continuum formulation is available to model the physical processes at the interface. At the same time, due to the heating of the interface which results in a shallow temperature profile, one would require a very large physical domain size normal to the interface in a pure MD simulation. Thus, the dynamic friction problem is a very good test case for hybrid MD-CFD methods. To achieve a fully coupled simulation, a number of issues must be resolved. These include methodological ones, like the synchronisation of the MD and CFD time integration schemes to obtain a stable coupling, and practical implementation issues, such as the information transfer between the MD and CFD solver, the implementation of the correct constitutive equations in the CFD solver and the parallelisation of the MD boundary condition.
5. Perform fully coupled MD-CFD simulations for channel flows, such as Poiseuille flow.
6. Extend the MD flux boundary conditions to the convective flux of molecules. This would require the adaption of the modified USHER scheme to find insertion sites of molecules.

Bibliography

[1] R. K. Agarwal and K. Y. Yun. *MEMS Introduction and Fundamentals*, chapter Burnett Simulations of Flows in Microdevices, pages 7–1 – 7–33. Taylor & Francis, Boca Raton, FL, USA, 2002.

[2] M. P. Allen and D. J. Tildesley. *Computer simulation of liquids*. Oxford University Press, Oxford, 1987.

[3] N. L. Allinger. A hydrocarbon force field utilizing V1 and V2 torsional terms. *J. Am. Chem. Soc.*, 99:8127–8134, 1977.

[4] N. L. Allinger and Y. Fan. Molecular mechanics studies (MM4) of sulfides and mercaptans. *J. Comput. Chem.*, 18:1827–1847, 1997.

[5] N. L. Allinger, M. T. T., M. A. Miller, and D. H. Wertz. Conformational analysis. LXIX. Improved force field for the calculation of the structures and energies of hydrocarbons. *J. Am. Chem. Soc.*, 93:1637–1648, 1971.

[6] N. L. Allinger, Y. H. Yuh, and J. H. Lii. Molecular mechanics. The MM3 force field for hydrocarbons. 1. *J. Am. Chem. Soc.*, 111:8551–8566, 1989.

[7] N. L. Allinger, F. Li, and L. Yan. Molecular mechanics. The MM3 force field for alkenes. *J. Comput. Chem.*, 11:848–867, 1990.

[8] N. L. Allinger, K. Chen, J. A. Katzenellenbogen, S. R. Wilson, and G. M. Anstead. Hyperconjugative effects on carbon-carbon bond lengths in molecular mechanics (MM4). *J. Comput. Chem.*, 17:747–755, 1996.

[9] N. L. Allinger, K. Chen, and J. H. Lii. An improved force field (MM4) for saturated hydrocarbons. *J. Comput. Chem.*, 17:642–668, 1996.

[10] J. D. JR. Anderson. *Fundamentals of Aerodynamics*. McGraw-Hill, Inc., New York, 2nd edition, 1991.

[11] J. D. JR. Anderson. *Computational Fluid Dynamics: The Basics with Applications*. 2nd Ed. McGraw-Hill, New York, 1995.

[12] S. Arcidiacono, N. R. Bieri, D. Poulikakos, and C. P. Grigoropoulos. On the coalescence of gold nanoparticles. *Int. J. Multiphase Flow*, 30:979–994, 2004.

[13] V. A. Arkhipov, I. M. Vasenin, and V. F. Trofimov. Stability of colliding drops of ideal liquid. *Zh. Prikl. Mekh. Tekh. Fiz.*, 3, 1983.

[14] N. Ashgriz and J. Y. POO. Coalescence and seperation in binary collision of liquid drops. *J. Fluid Mech.*, 221:183–204, 1990.

[15] W. T. Ashurst and W. G. Hoover. Dense-fluid shear viscosity via nonequilibrium molecular dynamics. *Phys. Rev. A*, 11:658–678, 1975.

[16] P.W. Atkins and R. S. Friedman. *Molecular quantum mechanics*. Oxford University Press, New York, 1996.

[17] S. Barsky, R. Delgado-Buscalioni, and P. V. Coveney. Comparison of molecular dynamics with hybrid continuummolecular dynamics for a single tethered polymer in a solvent. *J. Chem. Phys.*, 121:2403–2411, 2004.

[18] K. J. Bathe. *Finite Element Procedures*. Prentice Hall, 1995.

[19] N. R. Bimalendu. *Fundamentals of classical and statistical thermodynamics*. John Wiley & Sons, West Sussex, England, 2002.

[20] G. A. Bird. *Molecular gas dynamics and the direct simulation of gas flows*. Clarendon Press, Oxford, 1998.

[21] E. Blaisten-Barojas and M. R. Zachariah. Molecular-dynamics study of cluster growth by cluster-cluster collisions. *Phys. Rev. B*, 45:4403–4409, 1992.

[22] K. Bolton, M. Svanberg, and J. B. C. Petterson. Classical trajectory study of argon-ice collision dynamics. *J. Chem. Phys.*, 110(11):5380–5391, 1999.

[23] P. R. Brazier-Smith, S. G. Jennings, and J. Latham. The interaction of falling water drops: coalescence. *Proc. R. Soc. Lond. A*, 326:393–408, 1972.

[24] G. Brenn, D. Valkovska, and K. D. Danov. The formation of satellite droplets by unstable binary drop collisions. *Physics of Fluids*, 13:2463–2477, 2001.

[25] I. N. Bronshtein, K. A. Semendyayev, G. Musiol, and H. Mühlig. *Handbook of Mathematics*. Springer, Berlin, 4th edition, 2004.

[26] B.R. Brooks, R.E. Bruccoleri, B.D. Olafson, D.J. States, S. Swaminathan, and M. Karplus. A program for macromolecular energy, minmimization, and dynamics calculations. *J. Comput. Chem.*, 4:187–217, 1983.

[27] U. Burkert and N. L. Allinger. *Molecular mechanics*, volume 177 of *ACS Monograph Series*. American Chemical Society, Washington, DC, 1982.

[28] C. Caccamo, P. V. Giaquinta, and G. Giunta. Phase diagram of simple fluids: a comprehensive theoretical approach. *J. Phys. Cond.*, 5:B75–B82, 1993.

[29] S. Chen and G. D. Doolen. Lattice boltzmann method for fluid flows. *Annu. Rev. Fluid Mech.*, 30:329–364, 1998.

[30] D. L. G. Cheung. *Structures and properties of liquid crystals and related molecules from computer simulation*. PhD thesis, University of Durham, Chemistry Department, 2002.

[31] G. Ciccotti and J. P. Ryckaert. Molecular dynamics simulation of rigid molecules. *Comput. Phys. Rep.*, 4:345–392, 1986.

[32] W. D. Cornell, P. Cieplak, C. I. Bayly, I. R. Gould, K. M. Jr. Merz, D. M. Ferguson, D. C. Spellmeyer, T. Fox, J. W. Caldwell, and P. A. Kollman. A second generation force field for the simulation of proteins, nucleic acids and organic molecules. *J. Am. Chem. Soc.*, 117:5179–5197, 1986.

[33] G. Csanyi, T. Albaret, M. C. Payne, and A. D. Vita. "learn on the fly": A hybrid classical and quantum-mechanical molecular dynamics simulation. *Phys. Rev. Lett.*, 93:175503, 2004.

[34] W. A. Curtin and R. E. Miller. Atomistic/continuum coupling in computational materials science. *Mod. Simul. Mater. Sci. Eng.*, 11:R33–R68, 2003.

[35] G. Dahlquist and A. Bjoerk. *Numerical methods*. Prentice-Hall, Englewood Cliffs, NJ, 1974.

[36] W. Damm, A. Frontera, J. Tirado-Rives, and W. L. Jorgensen. OPLS all-atom force field for carbohydrates. *J. Comput. Chem.*, 18:1955–1970, 1997.

[37] G. De Fabritiis, R. Delgado-Buscalioni, and P. V. Coveney. Energy controlled insertion of polar molecules in dense fluids. *J. Chem. Phys.*, 121:12139–12142, 2004.

[38] R. Delgado-Buscalioni and P. Coveney. Hybrid molecular-continuum fluid dynamics. *Phil. Trans. R. Soc. Lond. A*, 362:1639–1654, 2004.

[39] R. Delgado-Buscalioni and P. V. Coveney. Usher: An algorithm for particle insertion in dense fluids. *J. Chem. Phys.*, 119:978–987, 2003.

[40] R. Delgado-Buscalioni and P. V. Coveney. Continuum-particle hybrid coupling for mass, momentum and energy transfers. *Phys. Rev. E*, 67:046704, 2003.

[41] R. Delgado-Buscalioni, E. G. Flekkoy, and P. V. Coveney. Fluctuations and continuity in particle-continuum hybrid simulations of unsteady flows based on flux-exchange. *Europhys. Lett.*, 69:959–965, 2005.

[42] K. E. Drexler. *Nanosystems: Molecular machinery, manufacturing, and computation*. Wiley, New York, 1992.

[43] D. Drikakis and W. Rider. *High-Resolution Methods for Incompressible and Low-Speed Flows*. Computational Fluid and Solid Mechanics. Springer, Berlin, 2004.

[44] J. Eggers and A. Beylich. New algorithms for application in the direct simulation monte carlo method. *Prog. Astro. Aero.*, 159:166–173, 1994.

[45] J. Eggers, J. R. Lister, and H. A. Stone. Coalescence of liquid drops. *J. Fluid Mech.*, 401:293–310, 1999.

[46] E. M. Engler, J. D. Andose, and P. v. R. Schleyer. Critical evaluation of molecular mechanics. *J. Am. Chem. Soc.*, 95:8005–8025, 1973.

[47] G. De Fabritiis, R. Delgado-Buscalioni, and P. V. Coveney. Modelling the mesoscale with molecular specificity. *Phys. Rev. Lett.*, 97:134501, 2006.

[48] G. De Fabritiis, M. Serrano, R. Delgado-Buscalioni, and P. V. Coveney. Fluctuating hydrodynamic modelling of fluids at the nanoscale. *Phys. Rev. E*, page in press, 2007.

[49] R. P. Feynman. There's plenty of room at the bottom. *Engineering and Science (California Institute of Technology)*, 123:22–36, 1960.

[50] R. P. Feynman, Leighton R. B., and Sands M. *The Feynman Lectures on Physics vol. 1*. Addison-Wesley, 1963.

[51] D. Fincham and B. J. Ralston. Molecular dynamics simulation using the CRAY-1 vector processing computer. *Comput. Phys. Commun.*, 23:127–134, 1981.

[52] R. C. Flagan and M. M. Lunden. Particle structure control in nanoparticle synthesis from the vapor phase. *Mat. Sci. Eng. A*, 204:113–124, 1995.

[53] E. G. Flekkoy, J. Feder, and G. Wagner. Coupling particles and fields in a diffusive hybrid model. *Phys. Rev. E*, 64:066302, 2001.

[54] E. G. Flekkoy, G. Wagner, and J. Feder. Hybrid model for combined particle and continuum dynamics. *Europhys. Lett.*, 52:271–276, 2001.

[55] E. G. Flekkoy, R. Delgado-Buscalioni, and P. V. Coveney. Flux boundary conditions in particle simulations. *Phys. Rev. E*, 72:026703, 2005.

[56] A. Franks. Nanotechnology. *J. Phys. E: Sci. Instrum.*, 20:1442–1451, 1987.

[57] D. Frenkel and B. Smit. *Understanding molecular dynamics simulation*. Academic Press, San Diego, 1996.

[58] M. Gad-el Hak. *MEMS Introduction and Fundamentals*, chapter Flow Physics, pages 7–7 – 4–35. Taylor & Francis, Boca Raton, FL, USA, 2006.

[59] A. L. Garcia, J. B. Bell, W. Y. Crutchfield, and B. J. Alder. Adaptive mesh and algorithm refinement using direct simulation monte carlo. *J. Comput. Phys.*, 54: 134–155, 1999.

[60] C. W. Gear. The numerical integration of ordinary differential equations of various orders, report ANL 7126. Technical report, Argonne National Laboratory, 1966.

[61] C. W. Gear. *Numerical initial value problems for ordinary differential equations*. Prentice-Hall, Englewood Cliffs NJ, 1971.

[62] H. Goldstein, C. P. Poole, and J. L. Safko. *Classical mechanics*. Addison Wesley, San Francisco, 3rd edition, 2002.

[63] A. Gopinath and R. L. Koch. Collision and rebound of small droplets in an incompressible continuum gas. *J. Fluid Mech.*, 454:145–201, 2002.

[64] D. Greenspan. Supercomputer simulation of colliding microdrops of water. *Computers Math. Applic*, 19:91–97, 1990.

[65] D. Greenspan and F. L. Heath. Supercomputer simulation of the modes of colliding microdrops of water. *J. Phys. D: Appl. Phys*, 24:2121–2123, 1991.

[66] R. D. Groot and P. B. Warren. Dissipative particle dynamics: Bridging the gap between atomistic and mesoscopic simulation. *J. Chem. Phys.*, 107:4423–4435, 1997.

[67] K. Gundertofte, J. Palm, I. Pettersson, and A. Stamvik. A comparison of conformational energies calculated by molecular mechanics (MM2(85), Sybyl 5.1, Sybyl 5.21, ChemX) and semiempirical (AM1 and PM3) methods. *J. Comput. Chem.*, 12: 200–208, 1991.

[68] K. Gundertofte, T. Liljefors, P. O. Norrby, and I. Pattesson. A comparison of conformational energies calculated by several molecular mechanics methods. *J. Comput. Chem.*, 17:429–449, 1996.

[69] N. Hadjiconstantinou and A. T. Patera. Heterogeneous atomistic-continuum representations for dense fluid systems. *Int. J. Mod. Phys. C*, 8(4):967–976, 1997.

[70] N. G. Hadjiconstantinou. Discussion of recent developments in hybrid atomistic-continuum methods for multiscale hydrodynamics. *Bulletin of the Polish Acadamy of Sciences*, 53:335–342, 2005.

[71] N. G. Hadjiconstantinou. Hybrid atomistic-continuum formulations and the moving contact-line problem. *J. Comput. Phys.*, 154:245–265, 1999.

[72] N. G. Hadjiconstantinou, A. L. Garcia, M. Z. Bazant, and G. He. Statistical error in particle simulations of hydrodynamic phenomena. *J. Comput. Phys.*, 187:274–297, 2003.

[73] T. A. Halgren. The representation of van der Waals (vdW) interactions in molecular mechanics force fields: potential form, combination rules, and vdw parameters. *J. Am. Chem. Soc.*, 114:7827–7843, 1992.

[74] T. A. Halgren. Merck molecular force field. I. Basis, form, scope, parameterization and performance of MMFF94. *J. Comput. Chem.*, 17:490–519, 1996.

[75] T. A. Halgren. Merck molecular force field. II. MMFF94 van der Waals and electrostatic parameters for intermolecular interactions. *J. Comput. Chem.*, 17:520–552, 1996.

[76] T. A. Halgren. Merck molecular force field. III. Molecular geometrics and vibrational frequencies for MMFF94. *J. Comput. Chem.*, 17:553–586, 1996.

[77] T. A. Halgren. Merck molecular force field. V. Extension of MMFF94 using experimental data, additional computational data and empirical rules. *J. Comput. Chem.*, 17:616–641, 1996.

[78] T. A. Halgren. MMFF VII. Characterization of MMFF94, MMFF94s, and other widely available force fields for conformational energies and for intermolecular-interaction energies and geometries. *J. Comput. Chem.*, 20:730–748, 1999.

[79] T. A. Halgren and R. B. Nachbar. Merck molecular force field. IV. Conformational energies and geometries. *J. Comput. Chem.*, 17:587–615, 1996.

[80] D. Hall and N. Pavitt. A appraisal of molecular force fields for the representation of polypeptides. *J. Comput. Chem.*, 5:441–450, 1984.

[81] J. E. Hammerberg, T. C. Germann, B. L. Holian, and R. Ravelo. Nonequilibrium molecular dynamics simulations of metallic friction at ta/al and cu/ag interfaces. *Metallurg. Mat. Trans. A*, 35:2741–2745, 2004.

[82] B. L. Hammond, W. A. Lester, and P. J. Reynolds. *Monte Carlo methods in ab initio quantum chemistry*. World Scientific, Singapore, River Edge, NJ, 1994.

[83] J.P. Hansen, D. Levesque, and J. J. Weiss. Self diffusion in the two-dimensional classical electron gas. *Phys. Rev. Lett.*, 43:979–982, 1979.

[84] F. H. Harlow. The particle-in-cell computing method in fluid dynamics. *Meth. in Comput. Phys.*, 3:319–343, 1964.

[85] T. Hawa and M. R. Zachariah. Molecular dynamics study of particle-particle collisions between hydrogen-passivated silicon nanoparticles. *Phys. Rev. B*, 69:035417, 2004.

[86] T. Hawa and M. R. Zachariah. Coalescence kinetics of bare and hydrogen-coated silicon nanoparticles: A molecular dynamics study. *Phys. Rev. B*, 71:165434, 2005.

[87] T. Hawa and M. R. Zachariah. Coalescence kinetics of unequal sized nanoparticles. *J. Aerosol Sci.*, 37:1–15, 2006.

[88] H. Heinz, W. Paul, and K. Binder. Calculation of local pressure tensors in systems with many-body interactions. *Phys. Rev. E*, 72:066704, 2005.

[89] W. Heisenberg. Über quantentheoretische Umdeutung kinematischer und mechanischer Beziehungen. *Z. Phys.*, 33:879–893, 1925.

[90] D. M. Heyes. Thermal conductivity and bulk viscosity of simple fluids: A molecular-dynamics study. *J. Chem. Soc. Faraday Trans. 2*, 80:1363–1394, 1984.

[91] P. Hobza, M. Kabelac, J. Sponer, P. Mejzlik, and J. Vondrasek. Performance of empirical potentials (AMBER, CFF95, CVFF, CHARMM, OPS, POLTEV), semiemprical quantum chemical methods (AM1, MNDO/M, PM3) and ab initio Hartree-Fock method for interaction of DNA bases: Comparison of nonempirical beyond Hartree-Fock results. *J. Comput. Chem.*, 18:1136–1150, 1997.

[92] R. W. Hockney. The potential calculation and some applications. *Meth. of Comput. Phys.*, 9:136–211, 1970.

[93] R. W. Hockney and J. W. Eastwood. *Computer simulation using particles*. McGraw-Hill, New York, 1981.

[94] R. W. Hockney and J. W. Eastwood. *Computer Simulation Using Particles*. The Institute of Physics, 1988.

[95] W. Humphrey, A. Dalke, and K. Schulten. VMD - visual molecular dynamics. *J. Molec. Graphics*, 14(1):33–38, 1996.

[96] H. Ibach and H. Lüth. *Solid-State Physics*. Springer, Berlin, 2003.

[97] T. Ikeshoji, G. Torchet, and K. Koga. Surface tensions and stress tensors of liquid and solid clusters by molecular dynamics. *Eur. Phys. J. D*, 24:211–214, 2003.

[98] Tecplot Inc. Tecplot 10. Available at www.tecplot.com, 2004.

[99] J. H. Irving and J. G. Kirkwood. The statistical mechanical theory of transport processes. iv. the equations of hydrodynamics. *J. Chem. Phys.*, 18:817–829, 1950.

[100] R. L. Johnston. *Atomic and molecular cluster*. Taylor & Francis, London, 2002.

[101] W. L. Jorgensen and J. Tirado-Rives. The OPLS potential functions for proteins. Energy minimization for crystals of cyclic peptides and crambin. *J. Am. Chem. Soc.*, 110:1657–1666, 1988.

[102] W. L. Jorgensen, D. S. Maxwell, and J. Tirado-Rives. Development and testing of the OPLS all-atom force field on conformational energetics and properties of organic liquids. *J. Am. Chem. Soc.*, 118:11225–11236, 1996.

[103] K. Kadau. Molecular dynamics comes of age: 320 billion atom simulation on bluegene/l. *Int. J. Mod. Phys. C*, 17:1755–1761, 2006.

[104] M. Kalweit and D. Drikakis. Molecular dynamics of colliding nanoclusters. *J. Comput. Theo. Nano Sci.*, 1:367–377, 2004.

[105] M. Kalweit and D. Drikakis. Collision dynamics of nanoscale lennard-jones clusters. *Phys. Rev. B*, 74:235415, 2006.

[106] G. Kaminski, E. M. Duffy, T. Matsui, and W. L. Jorgensen. Free energies of hydration and pure liquid properties of hydrocarbons from the OPLS all-atom model. *J. Phys. Chem.*, 98:13077–13082, 1994.

[107] R. M. Kini and H. J. Evans. Comparison of protein models minimized by the all-atom and united atom models in the AMBER force field. *J. Bio. Struc. Dyn.*, 10: 265–279, 1992.

[108] G. H. Ko and H. S. Ryou. Droplet collision processes in an inter-spray impingement system. *J. Aerosol Sci.*, 36:1300–1321, 2004.

[109] S. Kohlhoff and S. Schmauder. A new method for coupled elastic-atomistic modelling. In V. Vitek and D.J. Srolovitz, editors, *Atomistic Simulation of Materials: Beyond Pair Potentials*, pages 411–418. Plenum Press, New York, 1989.

[110] J. Koplik and J. R. Banavar. Continuum deductions from molecular hydrodynamics. *Annu. Rev. Fluid Mech.*, 27:257–292, 1995.

[111] P. Koumoutsakos. Multiscale flow simulations using particles. *Annu. Rev. Fluid Mech.*, 37:457–487, 2005.

[112] K. S. Krane. *Modern physics*. John Wiley & Sons. Inc., New York, 2nd edition, 1996.

[113] K. W. Kratky and W. Schreiner. Computational techniques for spherical boundary conditions. *J. Comput. Phys.*, 47:313–320, 1982.

[114] E. Kreyszig. *Advanced Engineering Mathematics*. John Wiley & Sons Inc., New York, 8th edition, 1999.

[115] A. Leonard. Vortex method for flow simulation. *J. Comput. Phys.*, 37:289–335, 1980.

[116] W. A. Lester, S.M. Rothstein, and S. Tanaka. *Recent advances in quantum Monte Carlo methods - part II*. Word Scientific Publishing, Singapore, 2002.

[117] J.-H. Lii and N. L. Allinger. Molecular mechanics. The MM3 force field for hydrocarbons. 2. Vibration frequencies and thermodynamics. *J. Am. Chem. Soc.*, 111: 8566–8575, 1989.

[118] J.-H. Lii and N. L. Allinger. Molecular mechanics. The MM3 force field for hydrocarbons. 3. The van der Waals potentials and crystal data for aliphatic and aromatic hydrocarbons. *J. Am. Chem. Soc.*, 111:8576–8582, 1989.

[119] MathWorks Limited. Matlab 7. Available at www.mathworks.co.uk, 2005.

[120] S. T. Lin, M. Blanco, and W. A. Goddard. The two-phase model for calculating thermodynamic properties of liquids from molecular dynamics: Validation for the phase diagram of lennard-jones fluids. *J. Chem. Phys.*, 119:11792–11805, 2003.

[121] W. K. Liu, H. S. Park, D. Qian, E. G. Karpov, H. Kadowaki, and G. J. Wagner. Bridging scale methods for nanomechanics and materials. *Comput. Methods Appl. Mech. Eng.*, 195:1407–1421, 2006.

[122] D. A. Lockerby and J. Reese. High-resolution burnett simulations of micro couette flow and heat transfer. *J. Comput. Phys.*, 188:333–347, 2003.

[123] D. A. Lockerby, J. M. Reese, and M. A. Gallis. The usefulness of higher-order constitutive relations for describing the knudsen layer. *Physics of Fluids*, 17:100609, 2005.

[124] S. E. Lyshevski. *Nano- and Micro-Electromechanical Systems: Fundamentals of Nano-and-Micro-Engineering*. CRC Press LLC, Florida, USA, 2001.

[125] A. D. Mackerell, J. Wiorkiewiczkuczera, and M. Karplus. An all-atom empirical energy function for the simulation of nucleic acids. *J. Am. Chem. Soc.*, 117:11946–11975, 1995.

[126] G. C. Maitland, M. Rigby, E. B. Smith, and W. A. Wakeham. *Intermolecular forces: Their origin and determination*. International Series of Monographs on Chemistry. Clarendon Press, Oxford, 1987.

[127] G. Marc and W. G. McMillan. The virial theorem. *Adv. Chem. Phys*, 58:209–361, 1985.

[128] F. Mashayek, N. Ashgriz, W. J. Minkowycz, and B. Shotorban. Coalescence collision of liquid drops. *Int. J. Heat Mass Transfer*, 46:77–89, 2003.

[129] M. Matsumoto and T. Matsuura. Molecular dynamics simulation of a rising bubble. *Mol. Simulat.*, 30:13–15, 2004.

[130] D. A. McQuarrie. *Statistical mechanics*. Harper and Row, New York, 1976.

[131] K. Meier. *Computer Simulation and Interpretation of the Transport Coefficients of the Lennard-Jones Model Fluid*. PhD thesis, Department of Mechanical Engineering, University of the Federal Armed Forces Hamburg, 2002.

[132] Y. Meleán and L. Di G. Sigalotti. Coalescence of colliding van der waals liquid drops. *Int. J. Heat Mass Transfer*, 48:4041–4061, 2005.

[133] L. Ming, N. Markovic, M. Svanberg, and J. B. C. Petterson. Collision dynamics of large argon clusters. *J. Phys. Chem. A*, 101:4011–4018, 1997.

[134] F. A. Momany and R. Rone. Validation of the general purpose QUANTA 3.2/CHARMM force field. *J. Comput. Chem.*, 13:888–900, 1992.

[135] P. M. Morse. Diatomic molecules according to the wave mechanics. II. Vibrational levels. *Phys. Rev.*, 34:57–64, 1929.

[136] M. Moseler. On the origin of surface smoothing by energetic cluster impact: molecular dynamics simulation and mesoscopic modeling. *Nucl. Inst. & Meth. B*, 164:522, 2000.

[137] A. E. Nasrabad, R. Laghaei, and B. C. Eu. Molecular theory of thermal conductivity of the lennard-jones fluid. *J. Chem. Phys.*, 124:084506, 2006.

[138] N. Nevens and N. L. Allinger. Molecular mechanics (MM4) vibrational frequency calculations for alkenes an conjugated hydrocarbons. *J. Comput. Chem.*, 17:730–746, 1996.

[139] N. Nevens, K. Chen, and N. L. Allinger. Molecular mechanics (MM4) calculations on alkenes. *J. Comput. Chem.*, 17:669–694, 1996.

[140] N. Nevens, J. H. Lii, and N. L. Allinger. Molecular mechanics (MM4) calculations on conjugated hydrocarbons. *J. Comput. Chem.*, 17:695–729, 1996.

[141] N. T. Nyugen and S. T. Werely. *Fundamentals and Applications of Microfluidics*. Artech House Inc., Norwood, MA, USA, 2006.

[142] S. T. O'Connel and P. A. Thompson. Molecular dynamics-continuum hybrid computations: A tool for studying complex fluid flows. *Phys. Rev. E*, 52:5792, 1995.

[143] S. Ogata, Y. Mitsuya, H. Zhang, and K. Fukuzawa. Molecular dynamics simulation for analysis of surface morphology of lubricant films with functional end groups. *IEEE Trans. Magn.*, 41:3013–3015, 2005.

[144] Y. Pan and K. Suga. Numerical simulation of binary liquid droplet collision. *Physics of Fluids*, 17:082105–082119, 2005.

[145] R. Panico, W. H. Powell, and J. C. Richer. *A guide to IUPAC nomenclature of organic compounds (recommendations 1993)*. Blackwell Scientific publications, Ljubljana, 1993.

[146] H. S. Park, E. G. Karpov, P. A. Klein, and W. K. Liu. Three-dimensional bridging scale analysis of dynamic fracture. *J. Comput. Phys.*, 207:588–609, 2005.

[147] D. A. Pearlman, D. A. Case, J. C. Caldwell, G. L. Seibel, U. C. Singh, P. Weiner, and P. A. Kollman. *AMBER 4.0*. San Francisco, 1991.

[148] S. J. Plimpton. Fast parallel algorithms for short-range molecular dynamics. *J. Comput. Phys.*, 117:1–19, 1995. lammps.sandia.gov.

[149] S. L. Post and J. Abraham. Modeling the outcome of dropdrop collisions in diesel sprays. *Int. J. Multiphase Flow*, 28:997–1019, 2002.

[150] S. Prasalovich. *Energetic cluster-surface impact: study of crater and hillock formation*. PhD thesis, Department of Experimental Physics, Göteborg University, 2003.

[151] J. Qian and C. K. Law. Regimes of coalescence and separation in droplet collision. *J. Fluid Mech.*, 331:59–80, 1997.

[152] E. L. Quentrec and C. Brot. New method for searching for neighbors in molecular dynamics computations. *J. Comput. Phys.*, 13:430–432, 1973.

[153] D. C. Rapaport. *The art of molecular dynamics simulation*. Cambridge University Press, Cambridge, New York, 1995.

[154] K. Refson. Molecular dynamics simulation of solid n-butane. *Physica*, 13B:256–266, 1985.

[155] W. Ren and E. Weinan. Boundary conditions for the moving contact line problem. *Physics of Fluids*, 19:022101, 2007.

[156] M. Rieber and A. Frohn. Three-dimensional navier-stokes simulations of binary collisions between droplets of equal size. *J. Aerosol Sci.*, 26:929–930, 1995.

[157] M. Rieth. *Nano-Engineering in science and technology*. World Scientific Publishing, London, 2003.

[158] J. Rigaudy and S. P. Klesney. Nomenclature of organic chemistry, 1979.

[159] J. Rogan, R. Ramirez, A. H. Romero, and M. Kiwi. Rearrangement collisions between gold clusters. *Eur. Phys. J. D*, 28:219–228, 2003.

[160] R. E. Rudd and J. Q. Broughton. Coarse-grained molecular dynamics and the atomic limit of finite elements. *Phys. Rev. B*, 58:R5893, 1998.

[161] J. P. Ryckaert, G. Ciccotti, and H. J. C. Berendsen. Numerical integration of the cartesian equations of motion of a system with constraints: Molecular dynamics of n-alkanes. *J. Comput. Phys.*, 23:327–341, 1977.

[162] R. J. Sadus. *Molecular simulation of fluids*. ELSEVIER, Amsterdam, 2002.

[163] R. Saito, G. Dresselhaus, and M. S. Dresselhaus. *Physical properties of carbon nanotubes*. Imperial College Press, London, 1998.

[164] M. Schelke and A. Frohn. Three-dimensional lattice boltzmann simulations of binary collisions between equal droplets. *J. Aerosol Sci.*, 26:145–146, 1995.

[165] T. Schmatko, H. Hervet, and L. Leger. Friction and slip at simple fluid-solid interfaces: The roles of the molecular shape and the solid-liquid interaction. *Phys. Rev. Lett.*, 94:275501, 2005.

[166] R. Schmidt, G. Seifert, and H. O. Lutz. Cluster-cluster collisions. i. reaction channels - fusion, depp inelastic and quasielastic collisions. *Phys. Lett. A*, 158:231–236, 1991.

[167] W. Schommers. *Structures and dynamics of surfaces I*, volume 41, chapter Topics in current physics. Springer-Verlag, Berlin, 1986.

[168] S. Sinha, V. K. Dhir, and B. Shi. Surface tension evaluation in lennard-jones fluid system with untruncated potentials. In *2003 ASME Summer Heat Tranfer Conference July 21-28,2003, Las Vegas, Nevada USA*, pages HT2003–47164, Las Vegas, USA, 2003. ASME.

[169] J. M. Soler, N. Garcí, O. Echt, K. Sattler, and E. Recknagel. Microcluster growth: Transition from successive monomer addition to coagulation. *Phys. Rev. Lett.*, 49: 1857–1860, 1982.

[170] A. Srivastava, O. N. Srivastava, S. Talapatra, R. Vajtai, and P. M. Ajayan. Carbon nanotube filters. *Nature Materials*, 3:610–614, 2004.

[171] S. D. Stoddard. Identifying clusters in computer experiments on systems of particles. *J. Comput. Chem.*, 27:291–293, 1978.

[172] H. A. Stone. Dynamics of drop deformation and breakup in viscous fluids. *Annu. Rev. Fluid Mech*, 26:65–102, 1994.

[173] Q. Sun and D. Boyd. Modeling gas nucleation and condensation using the direct simulation monte carlo method. In *38th AIAA Thermophysics Conference 6-9 June 2005*, pages AIAA–2005–4831, Toronto, Canada, 2005. AIAA.

[174] G. Sutmann. *Quantum Simulations of complex many-body systems: From theory to algorithms, lecture notes*, volume 10, chapter Classical molecular dynamics, pages 211–254. John von Neumann Institute for Computing, Juelich, NIC Series, 2002.

[175] M. Svanberg, L. Ming, N. Markovic, and J. B. C. Pettersson. Molecular dynamics simulations of binary collisions between water clusters. *J. Chem. Phys.*, 108:5888–5897, 1998.

[176] W. C. Swope, H. C. Anderson, P. H. Berens, and K. R. Wilson. A computer simulation method for the calculation of equilibrium constants for the formation of physical clusters of molecules: Application to small water clusters. *J. Chem. Phys.*, 76:637–49, 1982.

[177] N. Taniguchi. On the basic concept of nanotechnology. In *Proc. Int. Conf. Prod. Eng. Tokyo*, volume 2, pages 18–23, Tokyo, 1974. JSPE.

[178] B. D. Todd, D. J. Evans, and P. J. Daivis. Pressure tensor for inhomogeneous fluids. *Phys. Rev. E*, 52:1627–1638, 1995.

[179] A. Tomsic, H. Schröder, K. L. Kompa, and C. R. Gebhardt. Impact dynamics of molecular clusters on surfaces: Fragmentation patterns and ansitropic effects. *J. Chem. Phys.*, 119(12):6314–6323, 2003.

[180] C. Trozzi and G. Ciccotti. Stationary nonequilibrium states by molecular dynamics. II. newton's law. *Phys. Rev. A*, 29:916–925, 1984.

[181] R. E. Tuzun, D. W. Noidy, B. G. Sumptery, and R. C Merkle. Dynamics of fluid flow inside carbon nanotubes. *Nanotechnology*, 7:241–246, 1996.

[182] D. van der Spoel, A. R. van Buuren, E. Apol, P. J. Meulenhoff, D. P. Tieleman, A. L. T. M. Sijbers, B. Hess, K. A. Feenstra, E. Lindahl, R. van Drunen, and H. J. C.

Berendsen. *GROMACS user manual*. Department of Biophysical Chemistry, University of Groningen., Nijenborgh 4, 9747 AG Groningen, The Netherlands, 3.1.1 edition, 2002. www.gromacs.org.

[183] L. Verlet. Computer 'experiments' on classical fluids. I. Thermodynamical properties of Lennard-Jones molecules. *Phys. Rev.*, 159:98–103, 1967.

[184] H. K. Versteeg and W. Malalasekera. *An introduction to computational fluid dynamics: The finite volume method*. Longman Scientific & Technical, Essex, England, 1995.

[185] J. von Neumann. Various technique used in connection with random digits. *US Nat. Bur. Stand. Appl. Math.*, 12:36–38, 1951.

[186] D. C. Wadsworth and D. A. Erwin. One-dimensional hybrid continuum/particle simulation approach for rarefied hypersonic flows. *AIAA Paper*, pages 90–1690, 1990.

[187] D. C. Wadsworth and D. A. Erwin. Two-dimensional hybrid continuum/particle simulation approach for rarefied hypersonic flows. *AIAA Paper*, pages 92–2975, 1992.

[188] G. Wagner, E. Flekkoy, J. Feder, and T. Jossang. Coupling molecular dynamics and continuum dynamics. *Computer Physics Communications*, 147:670–673, 2002.

[189] G.J. Wagner and W. K. Liu. Coupling of atomistic and continuum simulations using a bridging scale decomposition. *J. Comput. Phys.*, 190:249–274, 2004.

[190] P. K. Weiner and P. A. Kollman. AMBER: Assisted model building with energy refinement. A general program for modeling molecules and their interactions. *J. Comput. Chem.*, 2:287–303, 1981.

[191] S. J. Weiner, P. A. Kollman, and U. Case. A new force field for molecular mechanical simulation of nucleic acids and proteins. *J. Am. Chem. Soc.*, 106:765–784, 1984.

[192] S. J. Weiner, Peter. A. Kollman, D. T. Nguyen, and D. A. Case. An all atom force field for simulations of proteins and nucleic acids. *J. Comput. Chem.*, 7:230–252, 1986.

[193] T. Werder, J. Walther, and P. Koumoutsakos. Hybrid atomistic-continuum method for the simulation of dense fluid flows. *J. Comput. Phys.*, 205:373–390, 2005.

[194] T. U. Werder. *Multiscale Simulations of Carbon Nanotubes in Aqueous Environments*. PhD thesis, Swiss Federal Institute of Technology Zürich, 2005.

[195] H. S. Wijesinghe and N. G. Hadjiconstantinou. Discussion of hybrid atomistic-continuum methods for multiscale hydrodynamics. *Int. J. Multiscale Comput. Eng.*, 2:189–202, 2004.

[196] K. D. Willis and M. E. Orme. Experiments on the dynamics of droplet collisions in a vacuum. *Experiments in Fluids*, 29:347–358, 2001.

[197] K. D. Willis and M. E. Orme. Binary droplet collisions in a vacuum environment: An experimental investigation of the role of viscosity. *Experiments in Fluids*, 34: 28–41, 2003.

[198] G. Woan. *The Cambridge Handbook of Physics Formulas*. Press syndicate of the University of Cambridge, Cambridge, UK, 2000.

[199] B. Wyatt. Collisions of microdrops of water. *Computers Math. Applic*, 28:175–208, 1994.

[200] Y. Yamaguchi. Large-scale molecular dynamics simulations of cluster impact and erosion processes on a diamond surface. *Phys. Rev. B*, 66:155408, 2002.

[201] J. L. Yarnell, M. J. Katz, and R. G. Wenzel. Structure factor and radial distribution function for liquid argon at 85k. *Phys. Rev. A*, pages 2130–2144, 1973.

[202] R. Zachariah, M. and M. J. Carrier. Molecular dynamics computation of gas-phase nanoparticle sintering: A comparision with phenomenological models. *J. Aerosol Sci.*, 30:1139–1151, 1999.

[203] J. A. Zimmerman, E. B. Webb, J. J. Hoyt, R. E. Jones, P. A. Klein, and D. J. Bammann. Calculation of stress in atomistic simulation. *Mod. Simul. Mater. Sci. Eng.*, 12:S319–S332, 2004.

APPENDIX **A**

Mathematical Notation

In this section, the mathematical which are used in throughout this document are defined for the sake of completeness.

Dot product of two vectors

$$\boldsymbol{a} \cdot \boldsymbol{b} = \boldsymbol{a}^{\mathrm{T}}\boldsymbol{b} == \begin{bmatrix} a_x & a_y & a_z \end{bmatrix} \begin{bmatrix} b_x \\ b_y \\ b_z \end{bmatrix} = a_x b_x + a_y b_y + a_z b_z$$

Cross product of two vectors

$$\boldsymbol{a} \times \boldsymbol{b} = \begin{vmatrix} \boldsymbol{i} & \boldsymbol{j} & \boldsymbol{k} \\ a_x & a_y & a_z \\ b_x & b_y & b_z \end{vmatrix} = \begin{bmatrix} a_y b_z - a_z b_y \\ a_z b_x - a_x b_z \\ a_x b_y - a_y b_x \end{bmatrix}$$

Tensor (dyadic) product of two vectors

$$\boldsymbol{a} \otimes \boldsymbol{b} = \boldsymbol{a}\boldsymbol{b}^{\mathrm{T}} = \begin{bmatrix} a_x \\ a_y \\ a_z \end{bmatrix} \begin{bmatrix} b_x & b_y & b_z \end{bmatrix} = \begin{bmatrix} a_x b_x & a_x b_y & a_x b_z \\ a_y b_x & a_y b_y & a_y b_z \\ a_z b_x & a_z b_y & a_z b_z \end{bmatrix}$$

Nabla operator

$$\nabla = \boldsymbol{i}\frac{\partial}{\partial x} + \boldsymbol{j}\frac{\partial}{\partial y} + \boldsymbol{k}\frac{\partial}{\partial z}$$

Gradient of a scalar

$$\mathrm{grad}\, f = \nabla f = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \\ \frac{\partial f}{\partial z} \end{bmatrix}$$

Gradient of a vector

$$\begin{aligned}\operatorname{grad} \boldsymbol{a} &= \nabla \boldsymbol{a} = \boldsymbol{a} \otimes \nabla = \boldsymbol{a} \nabla^{\mathrm{T}} \\ &= \begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix} \begin{bmatrix} \frac{\partial}{\partial x} & \frac{\partial}{\partial y} & \frac{\partial}{\partial z} \end{bmatrix} = \begin{bmatrix} \frac{\partial a_x}{\partial x} & \frac{\partial a_x}{\partial y} & \frac{\partial a_x}{\partial z} \\ \frac{\partial a_y}{\partial x} & \frac{\partial a_y}{\partial y} & \frac{\partial a_y}{\partial z} \\ \frac{\partial a_z}{\partial x} & \frac{\partial a_z}{\partial y} & \frac{\partial a_z}{\partial z} \end{bmatrix}\end{aligned}$$

Divergence of a vector

$$\begin{aligned}\operatorname{div} \boldsymbol{a} &= \nabla \cdot \boldsymbol{a} = \nabla^{\mathrm{T}} \boldsymbol{a} \\ &= \begin{bmatrix} \frac{\partial}{\partial x} & \frac{\partial}{\partial y} & \frac{\partial}{\partial z} \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix} = \frac{\partial a_x}{\partial x} + \frac{\partial a_y}{\partial y} + \frac{\partial a_z}{\partial z}\end{aligned}$$

Divergence of a 2nd order tensor

$$\nabla \cdot \boldsymbol{A} \equiv \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \begin{bmatrix} \frac{\partial}{\partial x} \\ \frac{\partial}{\partial y} \\ \frac{\partial}{\partial z} \end{bmatrix} = \begin{bmatrix} \frac{\partial a_{11}}{\partial x} + \frac{\partial a_{12}}{\partial y} + \frac{\partial a_{13}}{\partial z} \\ \frac{\partial a_{21}}{\partial x} + \frac{\partial a_{22}}{\partial y} + \frac{\partial a_{23}}{\partial z} \\ \frac{\partial a_{31}}{\partial x} + \frac{\partial a_{32}}{\partial y} + \frac{\partial a_{33}}{\partial z} \end{bmatrix}$$

APPENDIX B

Coupling Test Cases

Molecular system	type	ρ	T	Phase	L_x	L_y	L_z	N_{tot}
MDS-G	static	0.1	2	dense gas	120.6	21.5	21.5	5600
MDS-L	static	0.8	1	liquid	60.3	10.8	10.8	5600
MDS-S	static	1.2	1	solid	52.7	7.5	7.5	5600
MDCF-G	Couette flow	0.1	2	dense gas	120.6	79.37	4.8	1644
MDCF-L1	Couette flow	0.8	1	liquid	31.7	31.7	4.8	7871
MDCF-L2	Couette flow	0.8	1	liquid	31.7	63.5	4.8	7871
MDFF-G1	free flow	0.1	2	dense gas	120.6	21.5	21.5	5600
MDFF-L1	free flow	0.8	1	liquid	60.3	10.8	10.8	5600

Table B.1: Dimensions and general parameters of the molecular domains that have been used for coupling test cases: density ρ, temperature T, simulation box size in three dimensions $\boldsymbol{L}$ and total number of atoms N_{tot}. The first column contains the identification name of the test cases. All values are given in LJ-units (compare Table (6.1)), i.e. density in $\frac{m_a}{\sigma^3}$, temperature in ϵ/k_B and length in σ.

Molecular system	ρ	T	D	u_e	T_e
MDCF-G	0.1	2	77	1.0	2.11
MDCF-G	0.1	2	77	2.0	2.44
MDCF-L1	0.8	1	29.4	0.5	1.04
MDCF-L2	0.8	1	61.1	1.0	1.16
MDCF-L2	0.8	1	61.1	2.0	1.64

Table B.2: Velocity u_e and temperature T_e on the HSI, calculated by the analytical model for the Couette flow test case based on the thickness of the fluid layer, D. All values are given in LJ-units (compare Table (6.1)), i.e. density in $\frac{m_a}{\sigma^3}$, temperature in ϵ/k_B, length in σ and velocity in $\sqrt{\frac{\epsilon}{m_a}}$.