**Politecnico di Milano**
School of Industrial and Information Engineering

Master of Science in Mathematical Engineering: Computational Science and Engineering

# A Direct Simulation Monte Carlo Parallel Method for the Enskog-Vlasov Equation Applied to Two Phases Microflows

Supervisor: Prof. Paolo BARBANTE

Michele PELLEGRINO

student ID: 876344

Academic Year 2018-2019

## Abstract

Fluid flows for which the continuum Navier-Stokes formulation becomes inadequate due to a high Knudsen number can be found in the study of rarefied gas flows, computational materials science or microfluidics. The Enskog-Vlasov kinetic equation is employed in order to reproduce multi-phase flows and capillarity effects in microfluidics, as it is capable of describing multi-species dense gas flows undergoing phase transitions. Direct Simulation Monte Carlo Method is employed to perform numerical simulations. The central idea behind DSMC lies in the integro-differential structure of the equations of kinetic theory of gas: the streaming of an ensemble of particles representing a probability density function is propagated deterministically, while collisions between particles are performed stochastically. Particles schemes are generally easy to parallelize; however, the version of DSMC for EV equation poses issues due to the non-local nature of the mean-field interactions and the collisions between particles. A C++ code with MPI parallelization for distributed memory architectures is developed.

**Keywords:** Enskog-Vlasov equation, DSMC, MPI

## Sommario

Problemi di fluidodinamica per i quali la formulazione continua delle equazioni di Navier-Stokes diviene inadeguata, a causa di un elevato numero di Knudsen, possono essere riscontrati nello studio di flussi di gas rarefatti, della scienza dei materiali o della microfluidica. L'equazione cinetica di Enskog-Vlasov viene impiegata per studiare flussi multifase ed effetti di capillarità nella microfluidica, in quanto in grado di descrivere flussi di gas densi multi-specie in coesistenza o transizione di fase. Il metodo Direct Simulation Monte Carlo viene utilizzato per eseguire simulazioni numeriche; l'idea principale alla base del DSMC risiede nella struttura integro-differenziale delle equazioni della teoria cinetica del gas: un ensemble di particelle, che rappresenta una funzione di densità di probabilità, viene propagato deterministicamente, mentre le collisioni tra particelle vengono eseguite in maniera stocastica. I metodi computazionali a particelle sono generalmente facili da parallelizzare; tuttavia, la versione di DSMC per l'equazione di Enskog Vlasov pone alcuni problemi a causa della natura non locale delle interazioni di campo medio e delle collisioni tra particelle. Viene pertanto sviluppato un codice C++ ad hoc, con routines di parallelizzazione MPI per architetture a memoria distribuita.

**Parole chiave:** Equazione di Enskog-Vlasov, DSMC, MPI

# Ringraziamenti

Ringrazio il mio relatore, il Professor Paolo Barbante, per aver supervisionato il lavoro di tesi. Quando ho iniziato ad interessarmi alla fisica computazionale avevo poche conoscenze di teoria cinetica e di simulazioni particellari: è stato grazie alle sue indicazioni e ai sui spunti di riflessione, sempre capaci di catturare il mio interesse, che ho avuto modo di approfondire questi argomenti, ai quali mi sono appassionato. Sono infinitamente grato per la costante attenzione e per la disponibilità mostrata in questi mesi.

Rivolgo inoltre un sentito ringraziamento al Professor Carlo de Falco e a Luigi Urbinati, per avermi guidato nell'utilizzo delle risorse computazionali necessarie per testare il codice sviluppato per la tesi.

Indipendentemente dal punto di partenza e dagli obiettivi raggiunti, il percorso è ciò che ha avuto realmente importanza; quello che mi ha portato a questo traguardo è stato discontinuo, caotico: tutto fuorché liscio e lineare. Ho provato sconforto e frustrazione, per non riuscire ad ottenere i risultati sperati, ma infine anche orgoglio e soddisfazione, per aver finalmente prodotto qualcosa in cui posso immedesimarmi.

Voglio ringraziare tutti coloro che in questi sei anni hanno saputo infondermi passione, coraggio e autostima. Ringrazio innanzitutto i miei genitori, che in questi mesi hanno cercato costantemente di essermi d'aiuto, sostenendomi nelle situazioni di difficoltà: li ringrazio per avermi spinto a cogliere ogni occasione, mai ponendomi limiti e sempre standomi vicini, anche se geograficamente lontani. Ringrazio poi i miei colleghi Ing. Mat. Giuseppe, Paolo, Michele e Giovanni, coi quali ho condiviso momenti di profonda crescita culturale (ascoltando le lezioni del Professor Vianello) e di sconfinata disperazione (preparando APC), intramezzati da interessanti gite fuori porta a tema matematico. Un ringraziamento speciale va agli amici della Croce Bianca di Magenta, in particolare alla mitica squadra 3, che in questi ultimi 5 anni è stata per me come una seconda famiglia. Infine, voglio ringraziare tutte le persone fantastiche che ho avuto modo di conoscere durante il mio Erasmus a Delft, soprattutto i miei scatenati ex-coinquilini: apartment 65 rules!

Come dice un vecchio adagio: non si smette mai di imparare. Tuttavia ci viene concesso il lusso di essere studenti solo una volta nella vita, tanto che laurearmi mi sembra quasi un peccato. Oggi si chiude un importante capitolo: spero la curiosità e il desiderio di esplorare nuovi orizzonti possano guidarmi verso quello che si sta aprendo.

# Contents

# List of Symbols

| | |
|---|---|
| $\nabla_{\boldsymbol{x}}$ | Nabla with respect of the set of variables $\boldsymbol{x}$ |
| $\mathbb{1}_A$ | Indicator function of set $A$ |
| $f$ | Particles distribution function |
| $\boldsymbol{F}$ | Force field |
| $C(\cdot,\cdot)$ | Collisional integral |
| $\star$ | $N$-dimensional convolution in the spacial coordinates |
| $\int^{(k)}$ | $k$-dimensional integral |
| $:$ | Standard inner product between tensors |
| $\boldsymbol{a}_x$ | Component of vector $\boldsymbol{a}$ in direction $x$ |
| $n$ | Number density |
| $\eta$ | Reduced number density |
| $\sigma$ | Particle diameter |
| $\chi$ | Two-particles correlation function |
| $m$ | Particle mass |
| $\rho$ | Inter-particles distance |
| $k_b$ | Boltzmann's constant |
| $\boldsymbol{v}_r$ | Particles relative velocity |
| $p$ | Hydrostatic pressure |
| $\lambda$ | Mean free path |
| $\tau$ | Mean free time |
| $N_p$ | Number of particles |
| $N_{p,ij}$ | No. of particles in cell $(i,j)$ |
| $N_C$ | Number of collisions |
| $N_{cells}$ | Total number of cells |

# List of Figures

# List of Tables

# Chapter 1

# Motivation

The vast majority of numerical methods seek to provide solutions of the mathematical equations that describe the physical processes of interest, rather than provide a direct physical simulation of these processes. However, some cases represent an exception to the preceding observation either because the underlying mathematical model presents overwhelming difficulties to conventional computational methods or because the very structure of the process of interest readily lends itself to direct physical simulations [10]. One of such cases is represented by fluid flows for which the continuum Navier-Stokes formulation becomes inadequate due to a high Knudsen number. Notable examples can be encountered in the study of rarefied gas flows, computational materials science or microfluidics. Various particle methods find their natural collocation in these situations, according to the time and space scale of the phenomena of interest; Lattice-Boltzmann methods, Molecular Dynamics and Monte Carlo methods are notable examples: each one of these approaches is tailored to work on a particular space-time scale, being it nanoscopic, microscopic or mesoscopic. The present work will focus on yet another method, that is Direct Simulation Monte Carlo.

The Direct Simulation Monte Carlo method (DSMC) was introduced in the 1960s by Bird [11] to compute re-entry flow fields between the free-molecular and continuum regimes, for which results could not be obtained from more traditional approaches based on solving partial differential equations. Since then the fundamental physics-based nature of the DSMC algorithm in conjunction with ever-increasing computational power has allowed DSMC to tackle complicated non-equilibrium problems outside its original regime of applicability, even in the near-continuum regime. The essential idea behind this technique lies behind the integro-differential structure of the mathematics of kinetic theory of gas, exemplified by the Boltzmann equation: an ensemble of particles representing a probability density function is first propagated through time and space, according to a differential streaming operator; consequently, collisions between particles are simulated in a Monte Carlo fashion, so that the collision integral is computed stochastically.

Micro-electromechanical systems have found an increased application in a variety of industrial and medical fields; in many of these MEMS applications, the operation and performance of the microdevice depend directly or indirectly on gas dynamics [15]: rarefied flow phenomena can form the basis of important systems in the micromechanical domain [3]. As an example, the formation of liquid menisci may lead to spontaneous adhesion between mechanical parts which should maintain the capability of relative motion [17]. A proper treatment of multiphase flows and capillarity is needed in order to resolve and simulate these phenomena: this is commonly investigated on the basis of continuum theories; however, the validity of the macroscopic approach is not obvious when applied to physical processes at the typical scale of the Knudsen layers developing in these microdevices. DSMC has the advantage of being in principle able to automatically capture the micro/nanoscale features of flows in microchannels; this advantage is counterbalanced by a lower computational efficiency per volume with respect to continuous approaches. The original DSMC method was devised for rarefied flows, which can be effectively modelled by the Boltzmann kinetic equation; however, dense gas flows demand a modification of the aforementioned equation, especially regarding the collision integral.

The Enskog-Vlasov equation is a kinetic model capable of describing dense gas flows undergoing phase transitions and with liquid-vapor and fluid-walls interactions. It essentially consists in a modification of the Boltzmann equation, in which particle interactions are modelled by a self-consistent force field, at long distances, and by a non-local collision integral, at short distances. Enskog-Vlasov equation poses challenging questions regarding both modelling and analysis; some of these problems (such as the derivation of a H-theorem, the expression of the short-range correlation functions and the formulation of criteria for phase transition) have been successfully tackled in the past years, but in general the discussion regarding well-posedness remains far from being complete, from a mathematical perspective; however, it is possible to obtain significant insights which prove to be useful in the following numerical procedure. In this work a possible DSMC scheme consistent with Enskog-Vlasov equation will be illustrated.

In the past years numerous DSMC codes for rarefied gas flow have been developed both for industrial and research purposes (SPARTA, MONACO, DS2V); the fundamental components such a code needs to incorporate have been therefore thoroughly established. The case of dense gas flows does not represent an exception, but and augmentation. The main goal of this thesis work is to design a code implementing DSMC for the Enskog-Vlasov model; the language adopted for this purpose is C++, as it offers good encapsulation and modularity features, retaining a satisfactory computational efficiency. The code will be based on and inspired by an already existing Fortran code for dense fluids DSMC simulations.

DSMC is inherently computationally intensive, being a *direct* method. Parallelization can be envisioned as an effective way to speed-up simulations, especially since it usually results easy to implement for particle methods. Parallelization of the standard DSMC method is fairly straightforward: domain decomposition is employed to subdivide particles between processes, and collisions are performed locally in each sub-domain; communication occurs only when transferring particles from a process to another. In case of dense gas the computation of the self-consistent force field as well as the non-local nature of collisions poses a challenging obstacle, especially when considering distributed memory architectures. One of the primary aspects of our DSMC code will be the implementation of MPI routines to overcome the non-locality issue.

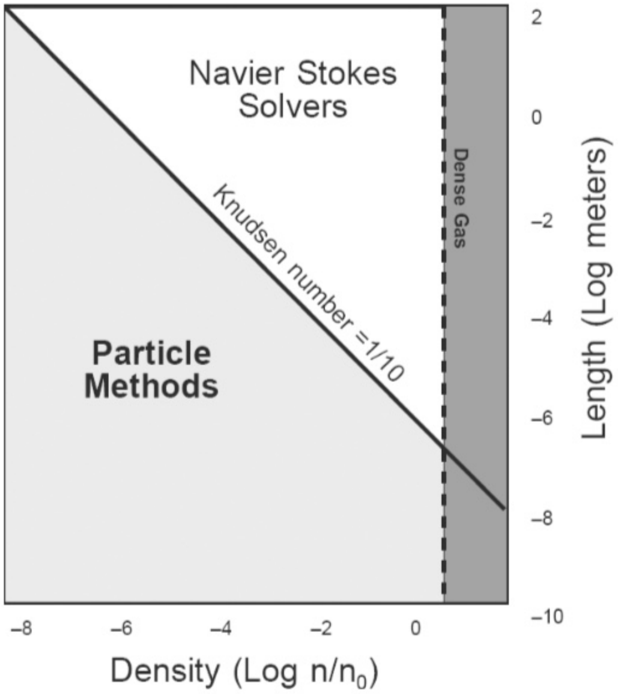


Figure 1.1: Validity regimes of a gas or fluid simulation method as a function of density relative to air and length scale [16]; the case of interest for this work situates in the bottom-right corner.

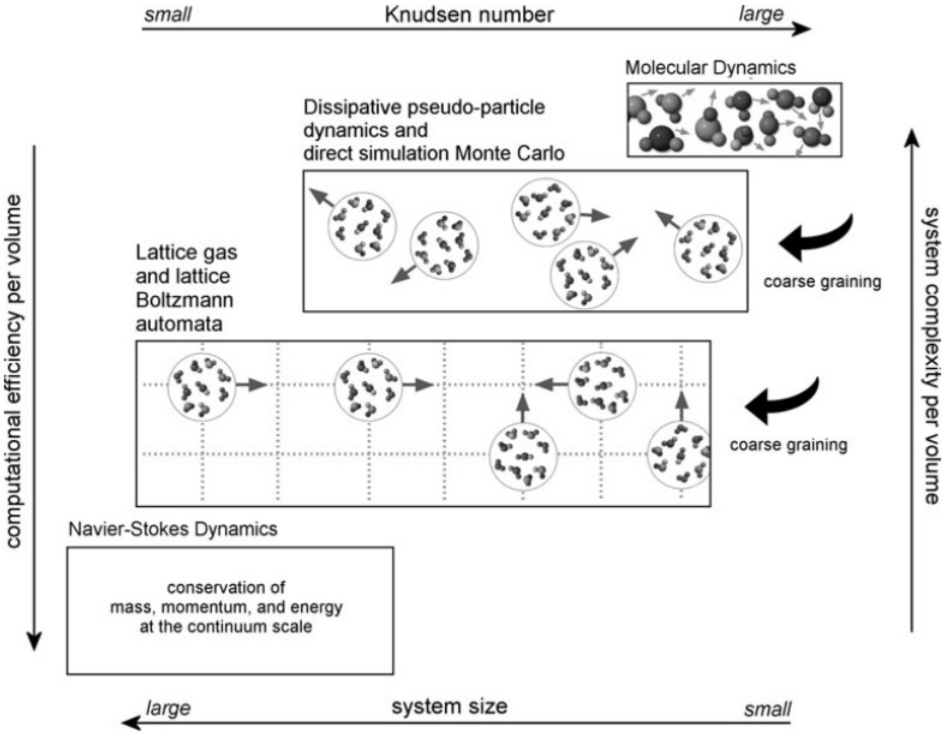

small
Knudsen number
large
Molecular Dynamics
Dissipative pseudo-particle dynamics and direct simulation Monte Carlo
coarse graining
Lattice gas and lattice Boltzmann automata
coarse graining
Navier-Stokes Dynamics
conservation of mass, momentum, and energy at the continuum scale
computational efficiency per volume
system complexity per volume
large
system size
small

Figure 1.2: Various approaches to computational fluid dynamics together with their preferred range of applicability [16]; all these methods have their respective strengths at different Knudsen numbers.

## 1.1 Validity of continuous and kinetic models

Rarefied gas dynamics can be defined as the study of gas flows in which the average value of the distance between two subsequent collisions of a molecule is not negligible in comparison with a length typical of the structure of the flow being considered [3]. The degree of rarefaction of a gas flow is normally expressed through the *Knudsen number*:

$$Kn = \frac{\lambda}{L} \,, \tag{1.1}$$

being $\lambda$ the particles' *average free path* and $L$ a typical dimension of the flow field. Inhomogeneous configurations are characterized by different values for the local Knudsen number; for example, it will be higher for the vapour phase with respect to the liquid one (due to a larger mean free path), and yet it will be different at interfaces (as the characteristic lengths would be on the same order of magnitude of the layer thickness). Definition (1.1) may be misleading, since both $\lambda$ and $L$ are not some *overall* dimensions of the flow [1]. The Knudsen number has therefore to be always intended as *local*; a possible, more precise, definition would define $L$ as the scale length of the macroscopic gradients [50]:

$$Kn(\boldsymbol{r}) = \left[\lambda \frac{||\nabla Q||}{Q}\right](\boldsymbol{r}) \,, \tag{1.2}$$

being $Q$ a physically relevant differentiable and positive scalar quantity, such as density, temperature or kinetic energy.

The Navier-Stokes model for continuum flows may be assumed as valid when $Kn \ll 1$ and the limit for which $Kn \to 0$ may be identified with the inviscid limit expressed through the Euler equations; the opposite limit, as $Kn \to \infty$, corresponds to the collisionless or free molecule flow in which intermolecular collisions may be neglected. The flow regime between free molecule flow and the limits of validity of of the Navier-Stokes equations is generally referred to as the transition flow regime; a Knudsen number about $Kn \simeq 0.1$ has traditionally been quoted as the boundary between continuum NS and transitional regime. For $0.01 < Kn \leq 0.1$ continuous hypothesis holds, but a free molecular flow is observed at the solid walls boundary, entailing the need for slip boundary conditions [10] [15].

The mean free path is dependent on density and one of its possible definition is, in case of only one specie [4]:

$$\lambda = \frac{1}{\pi\sqrt{2}n\sigma^2\chi} \,, \tag{1.3}$$

| **Knudsen number** | **Regime** | **Model** |
|---|---|---|
| $Kn \leq 0.01$ | Continuous flow | Navier-Stokes equations |
| $0.01 < Kn \leq 0.1$ | Slip flow | Navier-Stokes + slip b.c. |
| $0.1 < Kn \leq 10$ | Transition regime | Kinetic equations (Boltzmann, Enskog) |
| $Kn > 10$ | Free molecular flow | Collisionless Boltzmann equation |

Table 1.1: Summary of the validity ranges of continuous and kinetic formulations.

being $n$ the flow number density, $\sigma$ the particles' diameter and $\chi$ a correlation function (being $\chi = 1$ identically for an ideal gas). In order to clarify the relation between the Knudsen number and viscosity let us recall the *phenomenological* expressions dynamic viscosity, in case of a uniform gas of hard spheres [2]:

$$\mu = \nu\rho = \frac{1}{2}\rho\left\langle v\right\rangle\lambda = \sqrt{\frac{2k_bT}{\pi m}}\rho\lambda \ , \tag{1.4}$$

being $\rho$ the fluid density and $\langle v\rangle$ the average peculiar velocity (assumed computed from the Maxwell-Boltzmann distribution); therefore, the Knudsen number reads:

$$Kn = \frac{\lambda}{L} = \sqrt{\frac{\pi m}{2k_bT}}\frac{\mu}{\rho L} = \sqrt{\frac{\pi m}{2k_bT}}\frac{\nu}{L} \ . \tag{1.5}$$

It is then possible to relate the Knudsen number with the *Reynolds Number* and the *Mach number*:

$$\frac{Ma}{Re} = \frac{U_\infty/c}{\rho L U_\infty/\mu} = \sqrt{\frac{m}{\gamma k_b T}}\frac{\mu}{\rho L} \ ; \quad Kn = \sqrt{\frac{\gamma\pi}{2}}\frac{Ma}{Re} \quad , \tag{1.6}$$

being $U_\infty$ the free-stream speed, $c$ the speed of sound and $\gamma = C_p/C_V$ the ration of specific heats. In most microfluidics applications the flow regime is stated to be *high-Knudsen/low-Mach*, especially as the device dimensions shrink.

## 1.2 Structure of the thesis report

The goal of this thesis work is to study the features of the physical model (Enskog-Vlasov equation) and the computational method (DSMC), being thus able to develop a code capable to produce parallel simulations for two-phases microflows.

A *top-down* approach will be employed: in chapter 2 the mathematical model will be discussed, underlying its peculiarities and its characteristics of interests for the continuation; in chapter 3 the DSMC methods will be presented, focusing on its modifications with respect to the standard method in the context of dense gas flows; in chapter 4 the main structure of the code will be reported, as well as the parallel computing procedures; results concerning consistency and parallel speed-up will be presented in chapter 5, while chapter 6 will contain the conclusions, as well as hints for further developments.

# Chapter 2

# Mathematical model

The goal of this work is to propose a computational model which can simulate the flow of a dense gas interacting with solid walls and in which phase transitions may incur; it is therefore paramount to be consistent with a mathematical model capable of tackling all these aspects.

In order to be able to treat dense gases, the kinetic theory developed by David Enskog (1922) is considered: it consists in a modification of the classical theory of rarefied gas devised by Ludwig Boltzmann. The use of the Boltzmann equation must be indeed restricted to sufficiently diluted gas flows (i.e. in which molecular dimensions are small in comparison with the mean distance between particles), so that only binary collisions may be taken into account and molecular chaos may be assumed (*Stosszahlansatz*). Enskog's extension is a rigid sphere model with instantaneous collisions; the collisional operator is modified in order to account for correlations and non-local collisions (that arises from the particle diameter being no longer small compared to the average intermolecular distance) [2].

It is equally important to treat long-range interactions between particles, as they are related to the appearance of phase transition. These interactions are modelled through a soft-tail non-directional pair potential, which physically correspond to van der Waals forces in monoatomic non-ionic gasses. The self-consistent force field (also referred as *mean field*) arising from pair interactions is determined by a linear functional of the fluid number density. In case no collision between particles occurs the mean field model leads to an equation formally equivalent to what is referred as Vlasov equation in plasma physics [37].

The resulting equation, which combines both the collisional and the mean-field contributions, is referred as *Enskog-Vlasov* equation and provides a simplified description of the microscopic behaviour of the fluid but it has the capability of handling both the liquid and vapour phase, thus eliminating the necessity of postulating ad hoc models for boundary conditions at the vapour-liquid and solid-liquid interfaces [17], [19].

## 2.1 The form of EV equation

Let us consider a set of coordinates in a 6-dimensional phase space, parametrized by time $(\boldsymbol{r}, \boldsymbol{v}|t) \in \Omega \times \mathbb{R}^3$, $t \in [0, t_{max}]$; the density function $f$ at time $t$ is defined in such a way that its integral over a volume $V$ in the phase space is equal to the number of particles N having positions and velocities in $V$:

$$\int_V f(\boldsymbol{r}, \boldsymbol{v}|t) d\boldsymbol{r} d\boldsymbol{v} = N \quad \forall t \; ; \tag{2.1}$$

the fluid number density is therefore defined as:

$$n(\boldsymbol{r}|t) = \int_{\mathbb{R}^3} f(\boldsymbol{r}, \boldsymbol{v}|t) d\boldsymbol{v} \; . \tag{2.2}$$

The Enskog-Vlasov equation (that from now on will be referred as 'EV') may be written in a form that emphasizes a structure similar to the one of Boltzmann equation:

$$\frac{\partial f}{\partial t} + \boldsymbol{v} \cdot \nabla_{\boldsymbol{r}} f + \frac{\boldsymbol{F}}{m} \cdot \nabla_{\boldsymbol{v}} f = C(f, f) \; , \tag{2.3}$$

being $\boldsymbol{F}$ the mean field, $C(\cdot, \cdot)$ the collisional operator and $m$ the particles' mass; alternatively, the formulation proposed by Grmela [31] may be used instead:

$$\frac{\partial f}{\partial t} + \boldsymbol{v} \cdot \nabla_{\boldsymbol{r}} f = R_E f + R_V f = Rf \quad , \tag{2.4}$$

so that $R_E$ and $R_V$ are now respectively the non-linear Enskog and the Vlasov operators acting on $f$.

EV equation may be intuitively derived just by superimposing the dense gas collisional operator with the mean-field one; the structure of these operators will be discussed in the following sections. However, it is possible to derive this equation rigorously from BBGKY-hierarchy by introducing a proper Hamiltonian and using entropy maximization as closure principle [2], [29]; a brief overview of this method is displayed in the appendix.

## 2.2 Self-consistent force field

In this section the form of the self-consistent force field is discussed; the collision operator will be neglected for the moment. As previously stated, the Vlasov term models the van der Waals forces, similarly to the way electromagnetic forces are modelled in plasma physics. Let the particles exert on each other an attractive force with an isotropic pair potential $\phi$, and be $\boldsymbol{F}(\boldsymbol{r}|t)$ the force field generated collectively by all the molecules and evaluated at a point $\boldsymbol{r}$, at time $t$. The potential $\phi$ appears in the modelling of *long-range* interactions, while short-range interactions are instead modelled by the collision integral, as it will explained in the next section; this formally corresponds to the choice of a *Sutherland potential* of the kind:

$$\phi(\rho) = \begin{cases} +\infty & \rho < \sigma \\ \psi(\rho) & \rho \geq \sigma \end{cases} \; , \tag{2.5}$$

being $\rho$ the inter-particles distance, $\sigma$ the diameter of the particles (which determines the hard-sphere part of the interaction) and $\psi$ a function modelling the soft-tail part.

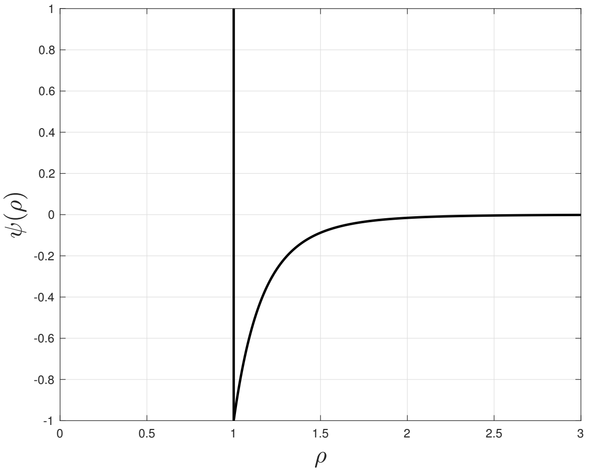


Figure 2.1: Example of Sutherland potential; in this case: $\psi(\rho) = -\epsilon(\sigma/\rho)^\gamma$, being $\epsilon = \sigma = 1$ and $\gamma = 6$ (non-dimensional quantities).

The Vlasov operator takes the following form [17]:

$$R_V f = \frac{1}{m}\nabla_{\boldsymbol{v}} \cdot \int_{\mathbb{R}}\int_{\rho>\sigma} f_2(\boldsymbol{r}, \boldsymbol{v}, \boldsymbol{r}_1, \boldsymbol{v}_1)\nabla\phi(\boldsymbol{r} - \boldsymbol{r}_1)d\boldsymbol{v}_1 d\boldsymbol{r}_1 \ , \tag{2.6}$$

where $\rho = ||\boldsymbol{r}_1 - \boldsymbol{r}||$ and the domain of integration is given by the form of the Sutherland potential. $f_2$ is the pair density function corresponding to a couple of particles respectively with position and velocity $(\boldsymbol{r}, \boldsymbol{v})$ and $(\boldsymbol{r}_1, \boldsymbol{v}_1)$. In order to express the mean field in terms of the one-particle density function it is assumed that the two-particles distribution can be factorized as:

$$f_2\big(\boldsymbol{r}, \boldsymbol{v}, \boldsymbol{r}_1, \boldsymbol{v}_1|t\big) = \mu\big(\boldsymbol{r}, \boldsymbol{r}_1|t\big) f\big(\boldsymbol{r}, \boldsymbol{v}|t\big) f\big(\boldsymbol{r}_1, \boldsymbol{v}_1|t\big) \ , \tag{2.7}$$

function $\mu$ corresponding to the *long-range* 2-particles correlation. By inserting the factorization of the pair density into the expression for the Vlasov operator, the following is obtained:

$$R_V f = \frac{1}{m}\left\{\int_{\mathbb{R}}\int_{\rho>\sigma} \nabla\phi(\boldsymbol{r} - \boldsymbol{r}_1)\mu\big(\boldsymbol{r}, \boldsymbol{r}_1|t\big) f(\boldsymbol{r}_1, \boldsymbol{v}_1|t) d\boldsymbol{v}_1 d\boldsymbol{r}_1\right\} \cdot \nabla_{\boldsymbol{v}} f(\boldsymbol{r}, \boldsymbol{v}|t) \ , \tag{2.8}$$

so that the force field is given by comparing equations (2.3) and (2.4):

$$\boldsymbol{F} = \int_{\mathbb{R}}\int_{\rho>\sigma} \nabla\phi(\boldsymbol{r}_1 - \boldsymbol{r})\mu\big(\boldsymbol{r}, \boldsymbol{r}_1|t\big) f(\boldsymbol{r}_1, \boldsymbol{v}_1|t) d\boldsymbol{v}_1 d\boldsymbol{r}_1 \ . \tag{2.9}$$

In conclusion, the final expression of the self-consistent force field is obtained by performing the integration over $\boldsymbol{v}_1$:

$$\boldsymbol{F}\big(\boldsymbol{r}|t\big) = \int_{\rho>\sigma} \nabla\phi\big(\boldsymbol{r}_1 - \boldsymbol{r}\big)\mu\big(\boldsymbol{r}, \boldsymbol{r}_1|t\big)n(\boldsymbol{r}_1|t)d\boldsymbol{r}_1 = \int_{\rho>\sigma} \frac{d\phi}{d\rho}\frac{\boldsymbol{r}_1 - \boldsymbol{r}}{||\boldsymbol{r}_1 - \boldsymbol{r}||}\mu\big(\boldsymbol{r}, \boldsymbol{r}_1|t\big)n(\boldsymbol{r}_1|t)d\boldsymbol{r}_1 \ . \tag{2.10}$$

The correlation function may be taken constant (basically $\mu = 1$, as a constant term may be absorbed into the potential): other than being a common choice performed by various authors (since spatial correlations are expected to decay rapidly, it is not unreasonable to ignore them completely [64]), it is also justified by that fact that setting $\mu = const$ is a sufficient condition to ensure total energy conservation, as stated by Benilov [37]. With this assumption, expression (2.10) becomes:

$$\boldsymbol{F}\big(\boldsymbol{r}|t\big) = \int_{\rho>\sigma} \frac{d\phi}{d\rho}\frac{\boldsymbol{r}_1 - \boldsymbol{r}}{||\boldsymbol{r}_1 - \boldsymbol{r}||}n(\boldsymbol{r}_1|t)d\boldsymbol{r}_1 \ . \tag{2.11}$$

The expression for the mean field can also be derived using an empirical approach [65]; let us consider the ensemble of particles described in section 2.1 being subject to Newtonian dynamics:

$$\begin{cases} \dot{\boldsymbol{r}}_i = \boldsymbol{v}_i \\ \dot{\boldsymbol{v}}_i = \dfrac{\boldsymbol{F}_i}{m} \end{cases} , \quad i = 1, ..., N \ , \tag{2.12}$$

being $\boldsymbol{F}_i$ the force exerted on the $i-$th particle, resulting from the interactions $\boldsymbol{\mathcal{F}}_{ij}$ with all other particles:

$$\boldsymbol{F}_i = \sum_{i,j} \boldsymbol{\mathcal{F}}_{ij} = \sum_{i,j} \nabla\phi(\boldsymbol{r}_i - \boldsymbol{r}_j) \ . \tag{2.13}$$

The following simplification is performed: the phase space is partitioned in subsets $V_j$, each centred at $\boldsymbol{r}_j$; a empirical density may be approximated as a piecewise-constant function on each patch:

$$f_j\Big|_{(\boldsymbol{r},\boldsymbol{v})\in V_j} = \frac{N_j}{V_j} \ , \quad \big(N_j = \#\text{particles in the j-th patch}\big) \ . \tag{2.14}$$

An approximation for the mean field at position $\boldsymbol{r}$ can be therefore computed as:

$$\overline{\boldsymbol{F}}(\boldsymbol{r}|t) = \sum_j N_j \nabla\phi(\boldsymbol{r}_j - \boldsymbol{r}) = \sum_j V_j f_j \nabla\phi(\boldsymbol{r}_j - \boldsymbol{r}) \ ; \tag{2.15}$$

$$\overline{\boldsymbol{F}}(\boldsymbol{r}|t) \overset{|V_j|\to 0}{\to} \iint f(\boldsymbol{r}, \boldsymbol{v}|t)\nabla\phi(\boldsymbol{r} - \boldsymbol{r_1})d\boldsymbol{v}d\boldsymbol{r}_1 \ . \tag{2.16}$$

(2.10) and (2.18) are hence obtained by integrating over the velocities and considering the shape of the Sutherland potential.

As a final remark, it may be interesting to note that equation (2.10) exhibit a convolutional structure; in fact, by defining:

$$\widetilde{\boldsymbol{\mathcal{K}}}(\boldsymbol{r}_1 - \boldsymbol{r}) = \frac{d\phi}{d\rho}\frac{\boldsymbol{r}_1 - \boldsymbol{r}}{||\boldsymbol{r}_1 - \boldsymbol{r}||}\mathbb{1}_{\rho>\sigma} \ , \tag{2.17}$$

it is possible to rewrite the force field as:

$$\boldsymbol{F}(\boldsymbol{r}|t) = \left[\widetilde{\boldsymbol{\mathcal{K}}} \star n\right](\boldsymbol{r}|t) \ , \tag{2.18}$$

'$\star$' representing the convolution in the spacial coordinate.

## 2.3 Enskog collisional operator

In this section the structure of the collisional operator will be discussed; staring from a suitable expression of the Boltzmann operator, modifications for a dense-gas case will be presented.

In the derivation of the Boltzmann collisional operator (see [2], chapter 3, section 1) it is found that for a gas at normal pressure the average number of collisions per unit time at time $t$ such that the centre of the first particle lies in in the volume element $d\boldsymbol{r}$, the velocities before the collision lie respectively in $d\boldsymbol{v}$ and $d\boldsymbol{v}_1$ and the geometrical collision variables lie in $db$ and $d\epsilon$ around $b$ (*impact parameter*, i.e. the relative distance between the two centres when particles are infinitely far away) and $\epsilon$ (*scattering angle*) is equal to:

$$f\left(\boldsymbol{r}, \boldsymbol{v}|t\right) f\left(\boldsymbol{r}, \boldsymbol{v}_1|t\right) v_r b db d\epsilon d\boldsymbol{v} d\boldsymbol{v}_1 d\boldsymbol{r} \quad , \tag{2.19}$$

being $v_r$ the magnitude of the relative velocity $\boldsymbol{v}_r = \boldsymbol{v}_1 - \boldsymbol{v}$. Before stating the Enskog's modification to the previous formula, let us introduce the unit vector joining the centres of the two particles $\hat{\boldsymbol{k}}$ (going from particle 2 to particle 1, as shown in fig. 2.3) and the angle $\psi$ between $\hat{\boldsymbol{k}}$ and $\boldsymbol{v}_r$; then, recalling that the impact parameter remains constant

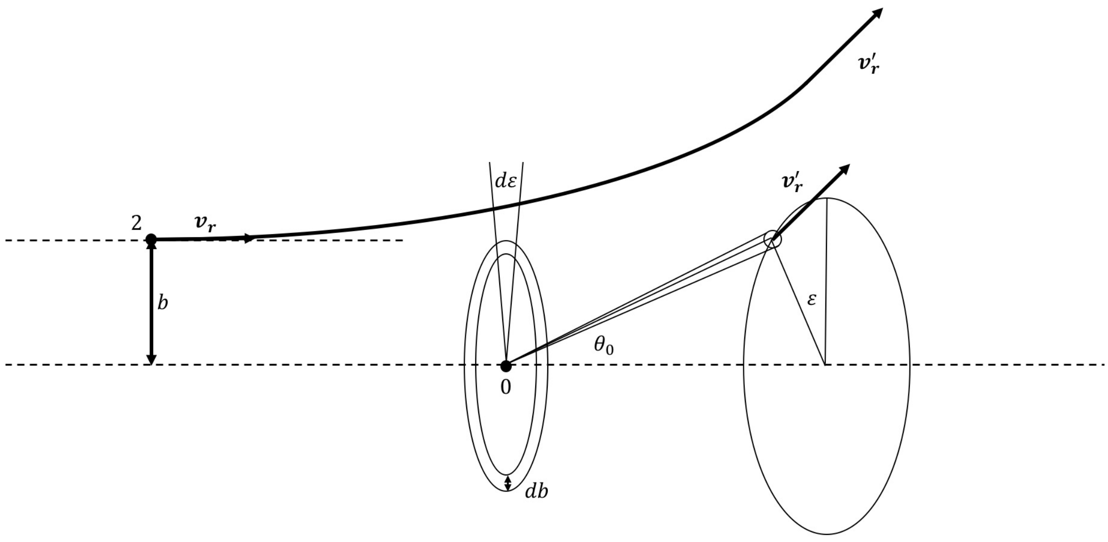


Figure 2.2: Geometry of a binary encounter; particle 1 is at rest with its centre in the origin. In addition to all the parameters in (2.19) the angle $\theta_0$ between the pre and post-collisional relative velocity is reported.

during the collision for a fixed $\psi$ in the case of hard spheres, the following is obtained:

$$b = \sigma \sin\psi \quad \Longrightarrow \quad d^2\hat{\boldsymbol{k}} = \sin\psi d\psi d\epsilon = \frac{bdbd\epsilon}{\sigma^2 \cos\psi} \quad , \tag{2.20}$$

being $\sigma$ the particle's diameter (that in case of the rigid spheres model with only one specie corresponds to the distance between particles' centres); since $\cos\psi = \hat{\boldsymbol{k}} \cdot \boldsymbol{v}_r$, we obtain that the infinitesimal solid angle $v_r bdbd\epsilon$ is equal to $\sigma^2(\hat{\boldsymbol{k}} \cdot \boldsymbol{v}_r)d^2\hat{\boldsymbol{k}}$; it follows that (2.19) is equal to:

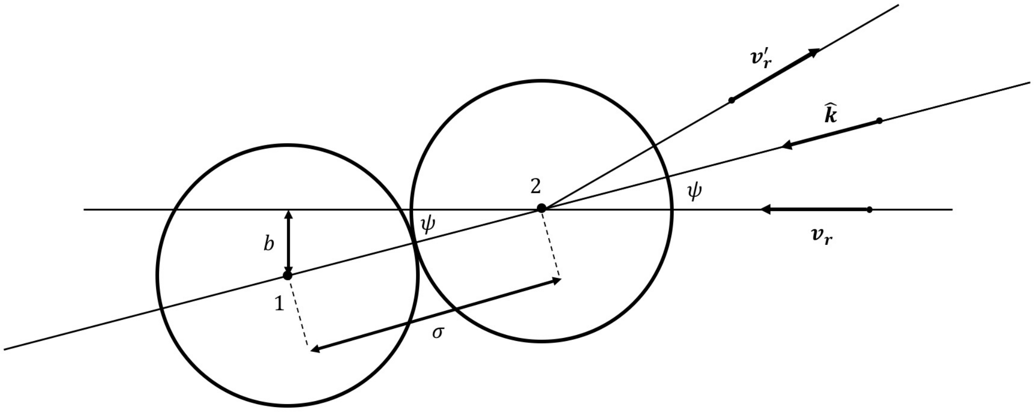


Figure 2.3: Geometry of a collision of two rigid spheres.

$$f\big(\boldsymbol{r}, \boldsymbol{v}|t\big) f\big(\boldsymbol{r}, \boldsymbol{v}_1|t\big)\sigma^2\big(\hat{\boldsymbol{k}} \cdot \boldsymbol{v}_r\big)d^2\hat{\boldsymbol{k}}d\boldsymbol{v}d\boldsymbol{v}_1 d\boldsymbol{r} \quad . \tag{2.21}$$

The above expression requires corrections when the gas is dense. First of all the centres of the colliding particles are not in the same point, because of their finite size; if at the instant of the collision the centre of the first particle is at $\boldsymbol{r}$, that of the second it at $\boldsymbol{r} - \sigma\hat{\boldsymbol{k}}$, so that we obtain:

$$f\big(\boldsymbol{r}, \boldsymbol{v}|t\big) f\big(\boldsymbol{r} - \sigma\hat{\boldsymbol{k}}, \boldsymbol{v}_1|t\big)\sigma^2\big(\hat{\boldsymbol{k}} \cdot \boldsymbol{v}_r\big)d^2\hat{\boldsymbol{k}}d\boldsymbol{v}d\boldsymbol{v}_1 d\boldsymbol{r} \quad . \tag{2.22}$$

Furthermore, in a dense gas the volume *per particle* becomes comparable with the volume *of a particle*, that is:

$$\frac{1}{n} \sim \frac{4}{3}\pi\Big(\frac{\sigma}{2}\Big)^3 \quad , \tag{2.23}$$

being $n$ the characteristic number density of the gas; hence, the particles' free volume is reduced, and the probability of a collision increases. Enskog's theory account for this effect by multiplying (2.22) by a function of the number density $\chi$ (referred as *short-range* pair correlation function), evaluated at the contact point $\boldsymbol{r} - \sigma\hat{\boldsymbol{k}}/2$ [2]:

$$\chi\big[n\big(\boldsymbol{r} - \sigma\hat{\boldsymbol{k}}/2\big)\big] f\big(\boldsymbol{r}, \boldsymbol{v}|t\big) f\big(\boldsymbol{r} - \sigma\hat{\boldsymbol{k}}, \boldsymbol{v}_1|t\big)\sigma^2\big(\hat{\boldsymbol{k}} \cdot \boldsymbol{v}_r\big)d^2\hat{\boldsymbol{k}}d\boldsymbol{v}d\boldsymbol{v}_1 d\boldsymbol{r} \quad . \tag{2.24}$$

Corresponding to any direct collision specified by $\boldsymbol{v}$, $\boldsymbol{v}_1$ and $\hat{\boldsymbol{k}}$ there exists an analogous inverse collision having $\boldsymbol{v}'$ and $\boldsymbol{v}_1'$ as post-collisional velocities and $-\hat{\boldsymbol{k}}$ is the direction of the apse-line; in such a collision the centre of the second molecule is at $\boldsymbol{r}+\sigma\hat{\boldsymbol{k}}$, while the point of contact is at $\boldsymbol{r}+\sigma\hat{\boldsymbol{k}}/2$, so that (2.22) reads:

$$\chi\big[n\big(\boldsymbol{r}+\sigma\hat{\boldsymbol{k}}/2\big)\big]f\big(\boldsymbol{r},\boldsymbol{v}'|t\big)f\big(\boldsymbol{r}+\sigma\hat{\boldsymbol{k}},\boldsymbol{v}_1'|t\big)\sigma^2\big(-\hat{\boldsymbol{k}}\cdot\boldsymbol{v}_r'\big)d^2\hat{\boldsymbol{k}}d\boldsymbol{v}d\boldsymbol{v}_1d\boldsymbol{r} \quad ; \tag{2.25}$$

Liouville's law for elastic collisions entails that $-\hat{\boldsymbol{k}}\cdot\boldsymbol{v}_r' = \hat{\boldsymbol{k}}\cdot\boldsymbol{v}_r$, so that the above expression can be written as:

$$\chi\big[n\big(\boldsymbol{r}+\sigma\hat{\boldsymbol{k}}/2\big)\big]f\big(\boldsymbol{r},\boldsymbol{v}'|t\big)f\big(\boldsymbol{r}+\sigma\hat{\boldsymbol{k}},\boldsymbol{v}_1'|t\big)\sigma^2\big(\hat{\boldsymbol{k}}\cdot\boldsymbol{v}_r\big)d^2\hat{\boldsymbol{k}}d\boldsymbol{v}d\boldsymbol{v}_1d\boldsymbol{r} \quad . \tag{2.26}$$

The Enskog collision operator has, therefore, the following expression:

$$\begin{aligned}C(f,f) = \sigma^2\int_{\mathbb{R}^3}\int_{\mathcal{S}_+}\Big\{&\chi\big[n\big(\boldsymbol{r}+\sigma\hat{\boldsymbol{k}}/2\big)\big]f\big(\boldsymbol{r},\boldsymbol{v}'|t\big)f\big(\boldsymbol{r}+\sigma\hat{\boldsymbol{k}},\boldsymbol{v}_1'|t\big)+\\ &-\chi\big[n\big(\boldsymbol{r}-\sigma\hat{\boldsymbol{k}}/2\big)\big]f\big(\boldsymbol{r},\boldsymbol{v}|t\big)f\big(\boldsymbol{r}-\sigma\hat{\boldsymbol{k}},\boldsymbol{v}_1|t\big)\Big\}\big(\boldsymbol{v}_r\cdot\hat{\boldsymbol{k}}\big)^+d\boldsymbol{v}_1d^2\hat{\boldsymbol{k}} \quad ;\end{aligned} \tag{2.27}$$

the unit vector $\hat{\boldsymbol{k}}$ lies on the portion of the unit sphere $\mathcal{S}_+$ where the condition $\boldsymbol{v}_r\cdot\hat{\boldsymbol{k}} > 0$ holds. The relation between pre- and post-collisional velocities is:

$$\begin{cases}\boldsymbol{v}' = \boldsymbol{v}+\hat{\boldsymbol{k}}\big(\boldsymbol{v}_r\cdot\hat{\boldsymbol{k}}\big)\\ \boldsymbol{v}_1' = \boldsymbol{v}_1-\hat{\boldsymbol{k}}\big(\boldsymbol{v}_r\cdot\hat{\boldsymbol{k}}\big)\end{cases} \quad . \tag{2.28}$$

## 2.4 Correlation function

In this section the form of the short-range correlation function will be briefly discussed. Let us first state its general definition:

$$\chi\big(\boldsymbol{r}_1,\boldsymbol{r}_2|t\big) = \frac{f_2\big(\boldsymbol{r}_1,\boldsymbol{r}_2,\boldsymbol{v}_1,\boldsymbol{v}_2|t\big)}{f\big(\boldsymbol{r}_1,\boldsymbol{v}_1|t\big)f\big(\boldsymbol{r}_2,\boldsymbol{v}_2|t\big)} \quad , \tag{2.29}$$

being $f_2$ the 2-particles density function. In the derivation of the Enskog operator $\chi$ has been described as a function of the density evaluated at the contact point, as it is stated by the *standard Enskog theory* (SET); however, the *revised Eskog theory* (RET, see Ernst, Van Beijeren [34], [35]) proposes to replace the 2-particles correlation in case of a non-uniform fluid with a *functional* of the whole density field $n(\boldsymbol{r}|t)$:

$$\chi_{\text{RET}}[n]\big(\boldsymbol{r}_1,\boldsymbol{r}_2\big) = 1+\int d\boldsymbol{r}_3 n(\boldsymbol{r}_3)\mathcal{V}(12|3)+\frac{1}{2}\int d\boldsymbol{r}_3 d\boldsymbol{r}_4 n(\boldsymbol{r}_3)n(\boldsymbol{r}_4)\mathcal{V}(12|34)+\dots\,, \tag{2.30}$$

being $\mathcal{V}(\cdot|\cdot)$ Husimi functions representing possible interactions of pair $(1,2)$ with all other particles (see [5], chapter 8, section 2, for an analogous formalism); the functional nature of the pair-correlation function can also be deducted by inspecting Ornstein-Zernike equation [36].

Although an expression can be obtained as a formal cluster expansion in the density, in practical applications simpler approximations are recommended; we opt therefore for a compromise proposed and justified by Fischer and Methfessel [24] and already exploited by Frezzotti et al. [21], namely taking $\chi$ locally as the pair correlation function of a homogeneous fluid at a density that correspond to the average over a spherical volume of radius $\sigma$:

$$\overline{n}\big(\boldsymbol{r}|t\big) = \frac{3}{4\pi\sigma^3}\int_{\rho<\sigma} n\big(\boldsymbol{r}_1|t\big)d\boldsymbol{r}_1 \quad , \tag{2.31}$$

$$\chi[n]\big(\boldsymbol{r},\boldsymbol{r}\pm\sigma\hat{\boldsymbol{k}}\big) \simeq \chi_{\text{SET}}\Big[\overline{n}\Big(\boldsymbol{r}\pm\frac{\sigma}{2}\hat{\boldsymbol{k}}\Big)\Big] \quad , \tag{2.32}$$

the averaging hence being done over a volume of molecular size (that is a spherical volume of radius $\sigma$).

An approximate but accurate expression for $\chi_{\text{SET}}$ as function of the local density can be obtained by comparing the equation of state of hard sphere fluids proposed by Carnahan and Starling [39] and the expression for the hydrostatic pressure of a Enskog fluid [2]:

$$\begin{cases} \dfrac{p}{nk_bT} = Z \overset{(CS)}{=} \dfrac{1+\eta+\eta^2-\eta^3}{(1-\eta)^3} \\ \\ p = \dfrac{1}{3}\boldsymbol{P}:\boldsymbol{I} \overset{(EV)}{=} \Big(1+\dfrac{2}{3}\pi n\sigma^3\chi\Big)nk_bT = (1+nb\chi)nk_bT \end{cases} , \tag{2.33}$$

being $Z$ the compressibility factor, $b$ the *specific* covolume and $\eta$ the reduced density:

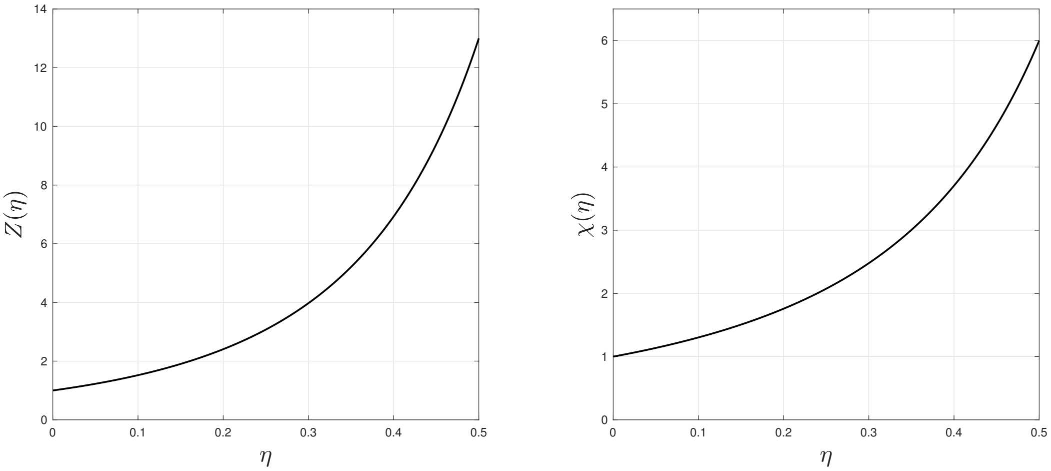


Figure 2.4: Plot of the compressibility factor and correlation from Carnahan-Starling equation of state over reduced density (also referred as packing fraction).

$$b = \frac{2\pi\sigma^3}{3} \ , \quad \eta = \frac{\pi\sigma^3 n}{6} \ . \tag{2.34}$$

Finally, we identify the density with its averaged value $\overline{n}$ and solve for $\chi$, thus obtaining:

$$\chi(\overline{\eta}) = \frac{1}{2}\frac{2-\overline{\eta}}{\left(1-\overline{\eta}\right)^3}\ , \quad \overline{\eta} = \frac{\pi\sigma^3\overline{n}}{6}\ . \tag{2.35}$$

## 2.5 Energy conservation and H-theorem

The complete expression of EV equation is obtained by combining (2.10) and (2.27) into (2.3):

$$\frac{\partial f}{\partial t}+\boldsymbol{v}\cdot\nabla_{\boldsymbol{r}} f+\frac{1}{m}\left[\int_{\rho>\sigma}\nabla\phi\big(\boldsymbol{r}_1-\boldsymbol{r}\big)\mu\big(\boldsymbol{r},\boldsymbol{r}_1|t\big)n\big(\boldsymbol{r}_1|t\big)d\boldsymbol{r}_1\right]\cdot\nabla_{\boldsymbol{v}} f=\sigma^2\int_{\mathbb{R}^3}\int_{\mathcal{S}_+}\Big\{\chi\big[\overline{\eta}\big(\boldsymbol{r}+\sigma\hat{\boldsymbol{k}}/2\big)\big]\cdot$$
$$\cdot f\big(\boldsymbol{r},\boldsymbol{v}'|t\big)f\big(\boldsymbol{r}+\sigma\hat{\boldsymbol{k}},\boldsymbol{v}_1'|t\big)-\chi\big[\overline{\eta}\big(\boldsymbol{r}-\sigma\hat{\boldsymbol{k}}/2\big)\big]f\big(\boldsymbol{r},\boldsymbol{v}|t\big)f\big(\boldsymbol{r}-\sigma\hat{\boldsymbol{k}},\boldsymbol{v}_1|t\big)\Big\}\big(\boldsymbol{v}_r\cdot\hat{\boldsymbol{k}}\big)^+d\boldsymbol{v}_1d^2\hat{\boldsymbol{k}}\ . \tag{2.36}$$

It is now essential it satisfies the fundamental principles of thermodynamics, such as energy conservation (corresponding macroscopically to the first law) and the H-theorem (corresponding macroscopically to the second law); for this purpose, the discussion by Benilov and Benilov [37] is briefly reported.

Net energy can be defined as the sum of kinetic and potential energy, as chemical effects or polyatomic molecules are not taken into consideration; recalling equation (2.7), and assuming that $f$ decays sufficiently fast as $||\boldsymbol{r}|| \to \infty$, the total energy can be written as:

$$U=\int_{\mathbb{R}^3}\int_{\Omega}\frac{m||\boldsymbol{v}||^2}{2}f\big(\boldsymbol{r},\boldsymbol{v}|t\big)d\boldsymbol{v}d\boldsymbol{r}+\frac{1}{2}\iint_{\rho>\sigma}\mu\big(\boldsymbol{r},\boldsymbol{r}_1|t\big)n\big(\boldsymbol{r}|t\big)n\big(\boldsymbol{r}_1|t\big)\phi\big(||\boldsymbol{r}-\boldsymbol{r}_1||\big)d\boldsymbol{r}_1d\boldsymbol{r}\ . \tag{2.37}$$

Time derivative of $U$ can be computed by combining (2.2), (2.10), (2.28) and (2.36):

$$\frac{dU}{dt}=\int^{(4)}f(\boldsymbol{r},\boldsymbol{v}|t)f(\boldsymbol{r}_1,\boldsymbol{v}_1|t)\phi(\boldsymbol{r}_1-\boldsymbol{r})\Big[\frac{\partial\mu}{\partial t}+(\boldsymbol{v}\cdot\nabla_{\boldsymbol{r}}+\boldsymbol{v}_1\cdot\nabla_{\boldsymbol{r}_1})\mu(\boldsymbol{r},\boldsymbol{r}_1|t)\Big]d\boldsymbol{v}d\boldsymbol{v}_1d\boldsymbol{r}d\boldsymbol{r}_1\ ; \tag{2.38}$$

hence, $U$ is conserved if the long-range correlation $\mu = const$, which justifies the choice made in section 2.2.

The H-theorem states the existence of a quantity $H = -S/k_b$ (negative entropy) such that $\partial_t H \leq 0$; in that it portrays a quantitative formulation for the fact that kinetic theory describes a process irreversible in time and represents a key result in the analysis of the trend to global equilibrium of spatially inhomogeneous kinetic systems. Resibois proved an H-theorem for the a modified version of the Enskog equation [38]; Benilov et al. propose a proof in the case of the Enskog-Vlasov equation for a entropy functional of the following form:

$$S=-k_b\int_{\mathbb{R}^3}\int_{\Omega}f\big(\boldsymbol{r},\boldsymbol{v}|t\big)\ln f\big(\boldsymbol{r},\boldsymbol{v}|t\big)d\boldsymbol{v}d\boldsymbol{r}+k_bQ[n]\ , \tag{2.39}$$

being $Q$ a functional of the number density, such that:

$$\nabla \frac{\delta Q}{\delta n} = \int_{\Omega} \chi(\boldsymbol{r}, \boldsymbol{r}_1|t) n(\boldsymbol{r}_1|t) \delta(||\boldsymbol{r} - \boldsymbol{r}_1|| - \sigma) \frac{\boldsymbol{r} - \boldsymbol{r}_1}{\sigma} d\boldsymbol{r}_1 \ . \tag{2.40}$$

Equation (2.40) is derived carrying out a similar algebra to the one used in the usual proof of the H-theorem for the Boltzmann equation. One interesting remark regards the non-local nature of the H-theorem for EV equation: the system is found to relax toward an distribution function

$$f_{eq}(\boldsymbol{r}, \boldsymbol{v}) = n(\boldsymbol{r}) \frac{\exp\left[\frac{-m(\boldsymbol{v}-\boldsymbol{u})^2}{2k_bT}\right]}{(2\pi k_b T)^{3/2}} \tag{2.41}$$

which is very different from the local equilibrium solution of the Boltzmann collision term, in that $\boldsymbol{u}$ and $T$ are *constant over the whole system*; this noteworthy feature can be traced back to the mechanism of collisional transfer: this latter allows the different points of the system to exchange momentum and energy through (non-local) collisions [38].

## 2.6 Kinetic equation approach to phase transition

As it is stated at the beginning of this chapter, EV equation offers the possibility of describing phase transition. Grmela analyses equation (2.4) in order to show there exists a set of critical parameter for which phase transition may occur [31]; his derivation is coherent with Kirkwood-Monroe theory [32]: *equilibrium KM theory* states that phase transition occurs as the thermodynamic parameters have such values that there is a discontinuity in the equilibrium distribution functions; *non-equilibrium KM theory* studies instead the approach to equilibrium by analysing the spectrum of the linearized dynamical operator and identifies phase transition with the presence of linear instabilities. Grmela analyze EV equation within the scope of both theories, proposing bifurcation conditions for the van Kampen equation [33] derived from a suitable equilibrium inhomogeneous solution on one hand, and studying the spectrum of the linearized EV operator on the other (exploiting mathematical tools similar to the ones used to analyse the spectral properties of the Boltzmann equation [3]).

Another viable approach is proposed by Frezzotti et al. [18], [19]; it is shown that the equation of state of a fluid described by (2.36) has the following generalized van der Waals form [5]:

$$p(n, T) = p_{hs}(n, T) - \alpha_{tail} n^2, \tag{2.42}$$

being $p_{hs}$ the contribution to hydrostatic pressure given by hard-sphere collisions, having the form proposed by Carhahan and Starling (2.33), and $\alpha_{tail}$ a constant depending on the soft-tail potential. The critical temperature $T_c$ corresponds to the value for which (2.42) has a point $\eta_c$ with null derivative; for $T < T_c$ two physically consistent values for $\eta$ coexists.

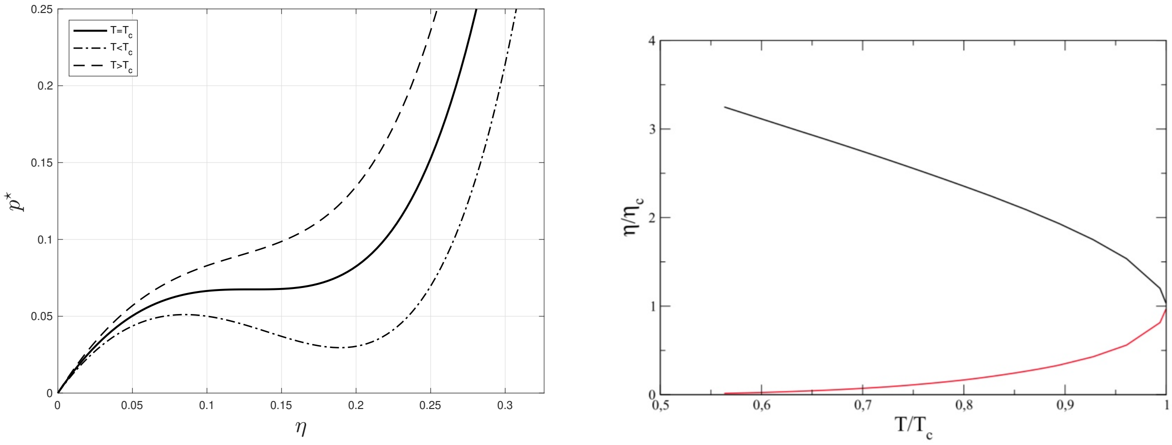


Figure 2.5: Left: Isothermal curves for temperatures above, equal and below $T_c$ in case of a Mie potential with $\gamma = 6$ (reduced pressure $p^\star = p\sigma^3/\overline{\phi}$ is plotted against reduced density $\eta$); right: vapor-liquid equilibrium coexistence curve for the same choice of potential (relative reduced density plotted against relative temperature).

$T_c$ ultimately depends on the type of potential chosen to model soft-tail attractions; a possible expression for function $\psi$ inside (2.5) may be given by a Mie-like algebraic potential:

$$\psi(\rho) = -\overline{\phi}\left[\frac{\sigma}{\rho}\right]^{\gamma} , \tag{2.43}$$

or by a Morse-like exponential potential, instead:

$$\psi(\rho) = -\overline{\phi}\exp\big[-\alpha(\rho-\sigma)\big] , \tag{2.44}$$

being $\overline{\phi}$ the depth of the potential well and $\gamma$, $\alpha$ parameters tuning the soft-tail decay. The critical temperature is obtained as:

$$T_c^{(\gamma)} = \frac{\overline{\phi}}{k_b\alpha_c}\frac{4\gamma}{\gamma-3} , \tag{2.45}$$

in case of (2.43), while in case of (2.44):

$$T_c^{(\alpha)} = \frac{4\overline{\phi}}{k_b\alpha_c}\Big(1+\frac{3}{\lambda_\alpha}+\frac{6}{\lambda_\alpha^2}+\frac{6}{\lambda_\alpha^3}\Big) , \tag{2.46}$$

being $\alpha_c \simeq 10.60122838879298$ and $\lambda_\alpha = \alpha\sigma$; the critical reduced density is $\eta_c \simeq 0.1304439008$, regardless of the tail potential. The equilibrium liquid and vapour bulk density values $n_v(T)$, $n_l(T)$ can be determined by Maxwells equal areas rule applied to isothermal curves given by (2.42).

## 2.7 Wall interaction

One key aspect of this kinetic model, as it is stated in the motivation section, regards the interaction between the liquid/gas phase and solid walls; in the following the fluid properties will be labelled with the subscript $g$ (or the subscript will be omitted), while walls properties with $w$.

Fluid-fluid and fluid-wall interaction forces are supposed to be given by the Sutherland potentials:

$$\phi_{gg}(\rho) = \begin{cases} +\infty & \rho < \sigma_g \\ \psi_{gg}(\rho) & \rho \geq \sigma_g \end{cases} , \quad \phi_{gw}(\rho) = \begin{cases} +\infty & \rho < \sigma_{gw} \\ \psi_{gw}(\rho) & \rho \geq \sigma_{gw} \end{cases} , \quad \text{being: } \sigma_{gw} = \frac{1}{2}(\sigma_g + \sigma_w) \ . \tag{2.47}$$

Different sets of parameters or even different forms for the tail function may be chosen in order to tune interaction ranges; the self-consistent force field generated by the tail potentials is a superposition of the fluid-fluid and the fluid-wall components:

$$\boldsymbol{F}(\boldsymbol{r}|t) = \boldsymbol{F}_{gg}(\boldsymbol{r}|t) + \boldsymbol{F}_{gw}(\boldsymbol{r}|t) \ , \tag{2.48}$$

being:

$$\begin{cases} \boldsymbol{F}_{gg}(\boldsymbol{r}|t) = \displaystyle\int_{r>\sigma_g} \frac{d\phi_{gg}}{dr} \frac{\boldsymbol{r}_1 - \boldsymbol{r}}{||\boldsymbol{r}_1 - \boldsymbol{r}||} n(\boldsymbol{r}_1|t) d\boldsymbol{r}_1 \\ \boldsymbol{F}_{gw}(\boldsymbol{r}|t) = \displaystyle\int_{r>\sigma_{gw}} \frac{d\phi_{gw}}{dr} \frac{\boldsymbol{r}_1 - \boldsymbol{r}}{||\boldsymbol{r}_1 - \boldsymbol{r}||} n_w(\boldsymbol{r}_1) d\boldsymbol{r}_1 \end{cases} . \tag{2.49}$$

Similarly, the collisional operator reads:

$$C(f,f) = C_{gg}(f,f) + C_{gw}(f,f_w) \ , \tag{2.50}$$

being:

$$\begin{cases} C_{gg}(f,f) = \sigma_g^2 \displaystyle\int_{\mathcal{S}_+} \Big\{ \chi_{gg}[n]\big(\boldsymbol{r}, \boldsymbol{r} + \sigma_g \hat{\boldsymbol{k}}\big) f\big(\boldsymbol{r}, \boldsymbol{r} + \sigma_g \hat{\boldsymbol{k}}, \boldsymbol{v}_1^*|t\big) f\big(\boldsymbol{r}, \boldsymbol{v}^*|t\big) + \\ \qquad -\chi_{gg}[n]\big(\boldsymbol{r}, \boldsymbol{r} - \sigma_g \hat{\boldsymbol{k}}\big) f\big(\boldsymbol{r}, \boldsymbol{r} - \sigma_g \hat{\boldsymbol{k}}, \boldsymbol{v}_1|t\big) f\big(\boldsymbol{r}, \boldsymbol{v}|t\big) \Big\} \big(\boldsymbol{v}_r \cdot \hat{\boldsymbol{k}}\big)^+ d\boldsymbol{v}_1 d^2\hat{\boldsymbol{k}} \\ C_{gw}(f,f) = \sigma_{gw}^2 \displaystyle\int_{\mathcal{S}_+} \Big\{ \chi_{gw}[n,n_w]\big(\boldsymbol{r}, \boldsymbol{r} + \sigma_{gw} \hat{\boldsymbol{k}}\big) f_w\big(\boldsymbol{r}, \boldsymbol{r} + \sigma_{gw} \hat{\boldsymbol{k}}, \boldsymbol{v}_1^*|t\big) f\big(\boldsymbol{r}, \boldsymbol{v}^*|t\big) + \\ \qquad -\chi_{gw}[n,n_w]\big(\boldsymbol{r}, \boldsymbol{r} - \sigma_{gw} \hat{\boldsymbol{k}}\big) f_w\big(\boldsymbol{r}, \boldsymbol{r} - \sigma_{gw} \hat{\boldsymbol{k}}, \boldsymbol{v}_1|t\big) f\big(\boldsymbol{r}, \boldsymbol{v}|t\big) \Big\} \big(\boldsymbol{v}_r \cdot \hat{\boldsymbol{k}}\big)^+ d\boldsymbol{v}_1 d^2\hat{\boldsymbol{k}} \end{cases} \tag{2.51}$$

No explicit assumption is made about the interaction among wall molecules, and it is simply assumed that walls are in a prescribed state of equilibrium which is not altered by the interaction with the fluid. Hence, the velocity distribution function will take the form of the *local equilibrium Maxwellian*:

$$f_w(\boldsymbol{r}, \boldsymbol{v}) = \frac{n_w(\boldsymbol{r})}{\big(2\pi R_w T_w(\boldsymbol{r})\big)^{3/2}} \exp\left[ - \frac{\big(\boldsymbol{v} - \boldsymbol{u}_w(\boldsymbol{r})\big)^2}{2 R_w T_w(\boldsymbol{r})} \right] \quad , \tag{2.52}$$

being $n_w(\boldsymbol{r})$, $T_w(\boldsymbol{r})$ and $\boldsymbol{u}_w(\boldsymbol{r})$ the prescribed wall molecules number density, temperature and mean velocity, respectively; the gas constant $R_w$ is defined as $k_b/m_w$. The calculation of the fluid-wall pair correlation function can be performed by assuming that excluded volume effects are determined solely by wall molecules through a modified number density $\tilde{n}_w$ [17]; the specific form of $\chi_{gw}(\tilde{\eta}_w)$ is given by (2.35), with:

$$\tilde{\eta}_w = \frac{\pi \tilde{n}_w \sigma_{gw}^3}{6} \; ; \tag{2.53}$$

this number density can be different from the value of $n_w$ that appears in equation (2.49). The rationale is to be able to tune independently the behaviour of the long-range attractive interaction and of the short-range repulsive interaction between fluid and walls [17].

In this section a specific example of two-species flow has been described; indeed, fluid and walls particles have different diameter and mass. The application of EV equation to more general multi-species fluids is briefly presented in the appendix.

# Chapter 3

# Numerical method

This chapter is devoted to the numerical procedure to solve EV equation. As it is stated in chapter 1, Direct Simulation Monte Carlo will be employed; even if this technique finds its natural collocation in the context of kinetic theory, its original formulation has been devised to solve the Boltzmann equation [11] and therefore some modifications have to be performed in order to be able to apply the same numerical scheme to dense gasses.

DSMC is a *particle* technique: the distribution function $f$ is represented by a number $N_p$ of particles (each one characterized by its position $\boldsymbol{r}_i$ and velocity $\boldsymbol{v}_i$) which move in the computational domain and collide according to stochastic rules derived from the kinetic equation. It is important to specify in this sense that particles are *mathematical* entities (as they represent a statistical ensemble of possible realizations of the density function), rather than physical, even if their identification with real molecules seems natural. Macroscopic flow properties are usually obtained by time averaging suitable particle properties, ultimately representing momenta of $f$.

The particular aspects in which the method proposed in this work differs from the standard DSMC regard the force field computation and the simulation of collisions [17]. The mean-field needs to be re-computed at each time step, being self-consistent, as it depends on the configuration of the ensemble. Moreover, collisions are non-local: while the original scheme searches for collision partners in the same computational cell, it is now paramount to extend the research to neighbouring cells [1].

In the following all quantities will be non-dimensional, unless otherwise specified; in particular, the reference quantities will be the gas particle's diameter, the depth of the gas-gas potential and the gas particle's mass; apices will be dropped to simplify notation.

[1] Here 'neighbouring' is not intended as 'directly attached'; collision partners will be searched in cells up to a distance of $\sigma/2$ (may be several, depending on the discretization).

| Distance | Temperature | Mass | Density | Pressure |
|---|---|---|---|---|
| $\boldsymbol{r}^* = \frac{\boldsymbol{r}}{\sigma_g}$ | $T^* = \frac{k_b T}{\bar{\phi}_{gg}}$ | $m^* = \frac{m}{m_g}$ | $n^* = \sigma^3 n$ | $p^* = \frac{p\sigma^3}{\bar{\phi}_{gg}}$ |

| Mean path | RMS speed | Mean time |
|---|---|---|
| $\lambda^* = \frac{\lambda}{\sigma}$ | $v^*_{RMS} = \sqrt{3T^*}$ | $\tau^* = \frac{\tau}{\sigma}\sqrt{\frac{\bar{\phi}_{gg}}{m}}$ |

Table 3.1: List of non-dimensional units.

## 3.1 Time splitting procedure

The numerical solution of (2.3) is obtained by a *time splitting* procedure, in which the distribution function, represented by the computational particles, is advanced from time $t$ to time $t + \Delta t$ in two main stages:

1. *free flight stage*, in which particles positions and velocities are propagated in the phase space following the transport part of the equation:

$$\frac{\partial f^{(1)}}{\partial t} + \boldsymbol{v}^{(0)} \cdot \nabla_{\boldsymbol{r}} f^{(1)} + \frac{\boldsymbol{F}^{(0)}}{m_g} \cdot \nabla_{\boldsymbol{v}} f^{(1)} = 0 \quad ; \tag{3.1}$$

2. *collisional stage*, in which the integral part of the equation is considered, that is [2]:

$$\frac{\partial f^{(2)}}{\partial t} = C(f^{(2)}, f^{(2)}) \quad . \tag{3.2}$$

Examining the details, phase (1) requires the self-consistent force field to be computed from the previous configuration, while phase (2) is in turn divided into (2.1) gas-gas collisions and (2.2) gas-walls collisions, so that the overall scheme reads:

1. the force field $\boldsymbol{F}^{(0)}(t)$ is computed from the initial ensemble;
2. the ensemble is streamed in the free-flight phase and the intermediate distribution $f^{(1)}(t + \Delta t)$ is obtained;
3. collisions between gas particles are simulated, giving the intermediate distribution $f^{(2)}(t + \Delta t)$ is obtained;
4. finally, collisions between gas and wall molecules are performed and the final distribution $f(t + \Delta t)$ is attained.

[2] With respect to what has been presented in section 2.5, the collisional stage does not describe a *locally homogeneous* relaxation, as it does in the case of the Boltzmann equation.

$$f(t) \overset{\text{streaming}}{\longrightarrow} f^{(1)}(t+\Delta t) \overset{\text{gas-gas coll.}}{\longrightarrow} f^{(2)}(t+\Delta t) \overset{\text{gas-wall coll.}}{\longrightarrow} f(t+\Delta t)\ .$$

It is crucial to observe that in the free-flight both positions and velocities are updated, while in the collision phase particles' positions are not changed, but their velocities are modified according to stochastic rules consistent with the mathematical structure of the collision integrals; in this sense the collisional operator may be interpreted as a stochastic source term for the velocities.

## 3.2 Computational domain

The numerical problem is set in a two-dimensional rectangular domain. The inner part (a rectangle of area $L_x L_y$) is where the fluid particles are placed, whereas the boundary is placed in a halo (of width $L_{x,ext}$ and $L_{y,ext}$): it would be populated by solid particle in case of a solid wall, while it will contain the density values of the diametrically opposite part of the domain, in case of periodic boundary conditions; in case of other boundary conditions (e.g. reflective walls) it will be left empty. Furthermore, in case of solid walls the area available for gas particles is reduced by an half-diameter of a solid particle, leaving a buffer between the 'logical' and the 'physical' domain of width $L_{x,lim}$ and $L_{y,lim}$.

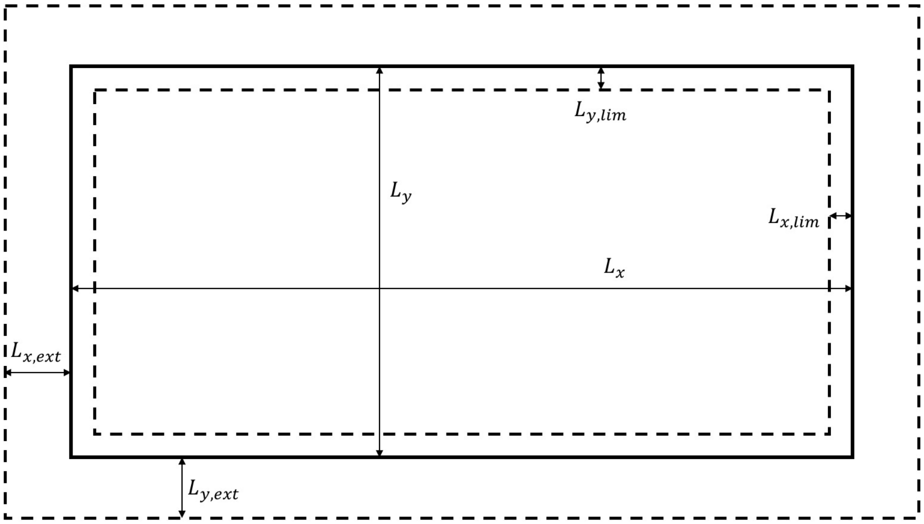


Figure 3.1: Layout of the computational domain.

The computational grid consists in a regular Cartesian mesh, with uniform spacing $\Delta x$ in the $x$ direction and $\Delta y$ in the $y$ direction. It is crucial to clarify the purpose of the computational grid: on one hand, it is instrumental for the collisional phase, as collision partners are searched in the cells neighbouring the one in which the selected particle is; on the other hand, all quantities on interests for the numerical procedure (e.g. number density and forces) are considered to be constant within each cell. A more advanced

DSMC procedure would employ different overlapping grids for collisions and for density or mean-field computations (in a 'chimera' fashion [62]): this is particularly important in case of rarefied gasses, as the collisional grid may be coarser; nonetheless, in our case of interest the shorter mean free path dictates a finer collisional grid, thus suitable to resolve flow field gradients.

## 3.3 Density binning and averaging

In the simulation of a two-dimensional flow-field the number of computational particles can always be made equal to the number of real molecules by a proper choice of the height of the computational domain along the homogeneity direction $z$ [17]; on other words, it is possible to tune $L_z$ (and therefore the volume per cell: $V_{ij} = V_{cell} = \Delta x \Delta y L_z$) and $N_p$ to achieve the desired initial number density for each cell:

$$n^0_{ij} = \frac{N_{p,ij}}{V_{cell}} \ , \quad i = 1, ..., N_x; \ j = 1, ..., N_y \ . \tag{3.3}$$

This expedient can be exploited in quasi one- or two-dimensional simulations, while in the three-dimensional case the ratio of real to simulation particles will be generally higher and an appropriate weight for the simulated particles has to be introduced; the weight can be interpreted as the inverse of the number of similar real systems needed to be simulated to obtain good statistics [19].

The number density is obtained at each stage via a binning procedure; the averaged density is computed by a discrete convolution of the number density and a mask matrix of weights, comprising the cells inscribed within a circumference of radius $\sigma$:

$$\overline{n}^t_{ij} = \sum_{k=i-N_{sx}}^{i+N_{sx}} \sum_{l=j-N_{sy}}^{j+N_{sy}} n^t_{kl} w_{kl} \ ; \quad w_{ij} = \frac{12}{\pi \sigma^3 \tilde{w}} \sqrt{\left(\frac{\sigma}{2}\right)^2 - s^2_{x,ij} - s^2_{y,ij}} \ ; \tag{3.4}$$

$$\begin{cases} s_{x,ij} = i\Delta x & i = -N_{sx}, ..., N_{sx} \\ s_{y,ij} = j\Delta y & j = -N_{sy}, ..., N_{sy} \end{cases} , \quad \begin{cases} N_{sx} = \left\lfloor \dfrac{\sigma}{2\sqrt{2}\Delta x} \right\rfloor \\ N_{sy} = \left\lfloor \dfrac{\sigma}{2\sqrt{2}\Delta y} \right\rfloor \end{cases} , \tag{3.5}$$

being $\tilde{w}$ a normalization constant.

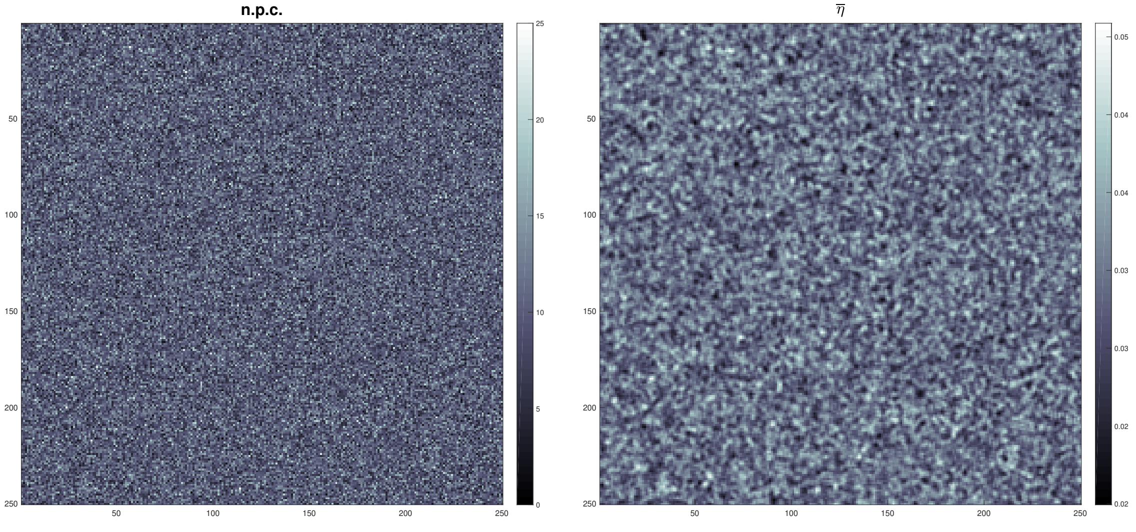


Figure 3.2: Effect of density averaging: on the left is the plot of the number of particles per cell, while on the right is the averaged reduced density $\overline{\eta}$. The system is composed by 625000 particles in a $250 \times 250$ grid; the reference density is $n = 6.551 \times 10^{-2}$.

## 3.4 Force field computation

A salient point of the numerical procedure is the computation of the mean-field. In the two-dimensional case only the $x$ and $y$ components of the force field are considered; hence, integrating over the homogeneity direction $z$ first, expression (2.10) reduces to:

$$\boldsymbol{F}(x,y|t) = \int_{||\boldsymbol{r}_1-\boldsymbol{r}||>\sigma} \big[(x_1-x)\boldsymbol{i} + (y_1-y)\boldsymbol{j}\big] n(x_1,y_1|t)\mathcal{K}(x_1-x,y_1-y)dx_1dy_1 \quad ; \tag{3.6}$$

where the *kernel*

$$\mathcal{K}(x_1-x,y_1-y) = \int_{\mathcal{S}_{z_1}} \frac{d\phi}{dr}\frac{1}{||\boldsymbol{r}_1-\boldsymbol{r}||}dz_1 \ , \tag{3.7}$$

$$\boldsymbol{r} = x\hat{\boldsymbol{u}}_x + y\hat{\boldsymbol{u}}_y + z\hat{\boldsymbol{u}}_z \ , \quad \mathcal{S}_{z_1} = \{z_1 \text{ s.t. } ||\boldsymbol{r}_1-\boldsymbol{r}|| > \sigma\} \ , \tag{3.8}$$

depends on the chosen potential and *not* on the actual value of the density field: this fact makes possible to compute the kernel $\mathcal{K}$ only once for a suitable range of the variables $x_1 - x$ and $y_1 - y$.

Expression (3.7) can be computed and stored by imposing $z = 0$ (which is a viable choice, since the flow is assumed to be homogeneous in this direction); being $\delta^2 = (x_1 - x)^2 + (y_1 - y)^2$, it becomes:

$$\mathcal{K}(\delta) = \int_{-\infty}^{+\infty} \frac{\overline{\phi}\gamma\sigma^\gamma}{\left(\delta^2+z_1^2\right)^{(\gamma+2)/2}} \mathbb{1}_{(\delta^2+z_1^2\geq\sigma^2)}dz_1 \quad , \tag{3.9}$$

in case of the Mie-type potential (2.43), while for the Morse-type potential (2.44):

$$\mathcal{K}(\delta) = \int_{-\infty}^{+\infty} \frac{\overline{\phi}\alpha e^{\alpha\sigma}}{\sqrt{\delta^2+z_1^2}\, e^{\alpha\sqrt{\delta^2+z_1^2}}} \mathbb{1}_{(\delta^2+z_1^2\geq\sigma^2)} dz_1 \quad . \tag{3.10}$$

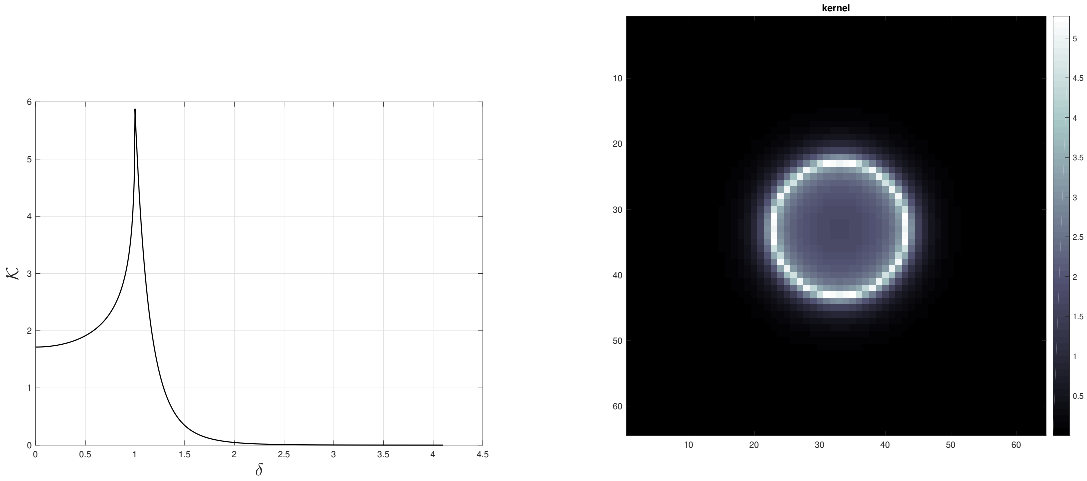


Figure 3.3: Left: profile of $\mathcal{K}(\delta)$ for a Mie-like potential with parameters $\gamma = 6.0$, $\overline{\phi} = 1.0$; right: values of $\mathcal{K}(\delta)$ stored in a mask-matrix.

Two storing strategies are can be applied:

- the values of the potential kernel may be evaluated for each inter-cell distance and stored in a *mask matrix*: the components of the force field would therefore result from a convolution of this matrix with the one storing the values for the number density; this strategy can be employed efficiently only in case of a uniform grid;
- the potential kernel may be stored in an array containing only a limited number of inter-cell distance evaluations, while other values are interpolated when needed; this strategy is more flexible, as it can be employed also in case of non-uniform grids.

In conclusion, it is common practice for particle methods and direct simulations to introduce a cut-off distance $\rho_c$, thus limiting the domain of integration for the mean-field:

$$\widetilde{\boldsymbol{F}}(x,y|t) = \int_{\rho_c>||\boldsymbol{r}_1-\boldsymbol{r}||>\sigma} \big[(x_1-x)\boldsymbol{i}+(y_1-y)\boldsymbol{j}\big] n(x_1,y_1|t)\mathcal{K}(x_1-x,y_1-y)dx_1dy_1 \quad ; \tag{3.11}$$

the introduction of a cut-off distance allows to reduce the memory required to store the potential kernel as well as the computational burden for the force field calculation.

## 3.5 Time marching

Let us now briefly discuss the time-marching technique adopted. In the streaming phase positions and velocities are updated via an explicit forward scheme:

$$\begin{cases} \boldsymbol{r}_i(t+\Delta t) = & \boldsymbol{r}_i(t) + \boldsymbol{v}_i(t)\Delta t + \dfrac{1}{2}\dfrac{\boldsymbol{F}\big(\boldsymbol{r}_i(t)|t\big)}{m}\Delta t^2 \\ \\ \boldsymbol{v}_i(t+\Delta t) = & \boldsymbol{v}_i(t) + \dfrac{\boldsymbol{F}\big(\boldsymbol{r}_i(t)|t\big)}{m}\Delta t \end{cases} ; \qquad (3.12)$$

in this case the overall scheme (streaming + collisions) is first-order accurate in time step $\Delta t$ and in linear cell dimension $\Delta r = \max(\Delta x, \Delta y)$. An acceptable level of accuracy is guaranteed *in all the phases of the time splitting procedure* if the following conditions are fulfilled [1]:

$$\begin{cases} \Delta t < \min\big(\tau_{gg}, \tau_{gw}\big) \\ \Delta r < \min\big(\lambda_{gg}, \lambda_{gw}\big) \end{cases} , \qquad (3.13)$$

being $\tau_{gg}$, $\tau_{gw}$ the mean free times between collisions dictated by fluid-fluid and fluid-wall short-range interactions and $\lambda_{gg}$, $\lambda_{gw}$ are the respective mean free paths, defines as (in non-dimensional units):

$$\lambda = \frac{1}{\sqrt{2}\pi n \chi(\eta)} , \quad \tau = \frac{\lambda}{v_{\mathrm{rms}}} , \qquad (3.14)$$

being $v_{\mathrm{rms}}$ the root-mean-square of a standard Maxwellian.

It may seem appropriate to employ more advanced time marching techniques in the advection phase. As an example, Verlet schemes are usually employed in molecular dynamics, in order to overcome the stability issues of (3.12) [6]; more sophisticated techniques would involve leapfrog or Runge-Kutta time integration. However, it is important to remind that the whole time step includes advection *and* collisions: even if more sophisticated techniques may in principle increase the order of accuracy of the streaming phase, it is not self-evident the same applies to the overall procedure.

Unless the fluid is dense everywhere in the computational domain, $\tau_{gw}$ can be much smaller than $\tau_{gg}$. In this case condition (3.13) would restrict the time step to an adequate value in a layer of thickness $\sigma_{gw}$ close to the wall, but unnecessarily small in the rest of the domain. A straightforward modification of the time splitting procedure is introduced to overcome such a limitation: fluid particles that are likely to collide with the wall are identified; for these particles the first and second stage are computed as a sequence of sub-cycles each with a time step of the order $\tau_{gw}$, the number of sub-cycles being $\mathcal{N}_s = \Delta t/\tau_{gw}$ [17]. This a convenient way to introduce an adaptive time-step in this numerical procedure; time-step adaptation is considered a crucial feature in state-of-the-art DSMC procedures [12].

A final remark regards the mesh-width. In case of dense vapour, it can be $\lambda > \sigma$; a choosing $\Delta r$ greater than the particles' cross-section should however be avoided, as it would prevent density averaging. As a rule of thumb, the mesh-with shouldn't be taken larger than 1/4 of the cross-section.

## 3.6 Simulating collisions

The final step in the DSMC iteration involves the simulation of particle collisions. The standard method devised by Bird is called *time-counter technique* (TC) [1] and consists in:

1. selecting a collisional pair with velocities $(\boldsymbol{v}, \boldsymbol{v}_1)$;
2. estimating a collisional time interval $\Delta t_c$;
3. updating the velocities from collision dynamics;
4. repeating 1., 2., 3. until $\sum \Delta t_c \geq \Delta t$.

Despite its simple nature, TC has a principle fault in the evaluation of the collision number, that is indeed overestimated.

Koura [40] proposed an alternative scheme, called *collision-frequency technique* (CF), which successfully estimates the expected number of collisions per unit time (i.e. the collisional frequency); the primary problem of this alternative technique is that it requires a significant amount of computation. In order to overcome this difficulty, Koura devised a *null-collision technique* (NC, also referred as *majorant frequency method*), in which the expected number of collisions is estimated via a stochastic process. In the case of the Enskog equation, the collision integral involves the distribution function at different spatial locations because of the finite extent of a molecule. Hence, the particles in a given cell interact with particles located in nearby cells, during the collisional step. A modification of the standard NC technique in case of dense gases is proposed by Frezzotti [20], and it will be discussed in the following.

The construction of the algorithm begins from the expression of the expected number of collisions per unit time in a subdomain $\mathbb{D}$:

$$\langle N_C \rangle = \frac{1}{2} \int_{\mathbb{D}} d\boldsymbol{r} \int_{\mathbb{R}^3} d\boldsymbol{v} \int_{\mathbb{R}^3} d\boldsymbol{v}_1 \int_{S(\boldsymbol{r})} W\big(\boldsymbol{r}, \hat{\boldsymbol{k}}, \boldsymbol{v}, \boldsymbol{v}_1 | t\big) d^2\hat{\boldsymbol{k}} \quad ; \tag{3.15}$$

$$W\big(\boldsymbol{r}, \hat{\boldsymbol{k}}, \boldsymbol{v}, \boldsymbol{v}_1 | t\big) = \sigma^2 \chi\big[\eta(\boldsymbol{r} + \sigma\hat{\boldsymbol{k}}/2)\big] f\big(\boldsymbol{r}, \boldsymbol{v}|t\big) f\big(\boldsymbol{r} - \sigma\hat{\boldsymbol{k}}, \boldsymbol{v}_1|t\big) (\boldsymbol{v}_r \cdot \hat{\boldsymbol{k}})^+ \ , \tag{3.16}$$

being $S(\boldsymbol{r})$ the set of admissible unite vectors $\hat{\boldsymbol{k}}$ applied at $\boldsymbol{r}$. This expression may be simplified by representing the ensemble density with a superposition of $N_p$ Dirac's delta functions:

$$f\big(\boldsymbol{r}, \boldsymbol{v}|t\big) = \sum_{i=1}^{N_p} \delta\big(\boldsymbol{r} - \boldsymbol{r}_i\big) \delta\big(\boldsymbol{v} - \boldsymbol{v}_i\big) \ . \tag{3.17}$$

Once the position of the first particle is fixed, the collision partner may lay in one of the $M_i$ cells containing a portion of the sphere of diameter $\sigma$ and centre $\boldsymbol{r}_i$. Furthermore, the number density is spatially regularized to be piecewise constant in each of the cells in which $\mathbb{D}$ has been subdivided; being $n_m$ and $N_m$ respectively the number density and the number of particles on the cell $\mathbb{C}_m$ (out of the possible $M_i$), the distribution takes the following form:

$$\widetilde{f}\big(\boldsymbol{r},\boldsymbol{v}|t\big)\Big|_{\boldsymbol{r}\in\mathbb{C}_m} = \frac{n_m}{N_m}\sum_{j=1}^{N_m}\delta\big(\boldsymbol{v}-\boldsymbol{v}_{(j,m)}\big)\ , \tag{3.18}$$

where $(j,m)$ indicates the $j$-th particle in the $m$-th cell. Inserting (3.17) and (3.18) in the expression for $\langle N_C\rangle$ one obtains:

$$\langle N_C\rangle \simeq \frac{1}{2}\int_{\mathbb{D}} d\boldsymbol{r}\int_{\mathbb{R}^3} d\boldsymbol{v}\int_{\mathbb{R}^3} d\boldsymbol{v}_1\int_{S(\boldsymbol{r})}\sum_{i=1}^{N_p}\sigma^2\chi\Big[\eta\Big(\boldsymbol{r}-\frac{\sigma}{2}\hat{\boldsymbol{k}}\Big)\Big]\delta\big(\boldsymbol{r}-\boldsymbol{r}_i\big)\delta\big(\boldsymbol{v}-\boldsymbol{v}_i\big)\cdot$$

$$\cdot\sum_{m}^{M_i}\frac{n_m}{N_m}\sum_{j}^{N_m}\delta\big(\boldsymbol{v}-\boldsymbol{v}_{(m,j)}\big)\big(\boldsymbol{v}_r\cdot\hat{\boldsymbol{k}}\big)^+d^2\hat{\boldsymbol{k}} = \frac{1}{2}\sum_{i}^{N_p}\sum_{m}^{M_i}\sum_{j}^{N_m}\nu_{(m,j),i}\ ; \tag{3.19}$$

$$\nu_{(m,j),i} = \sigma^2\frac{n_m}{N_m}\int_{S(m,j)}\chi\Big[\eta\Big(\boldsymbol{r}_i-\frac{\sigma}{2}\hat{\boldsymbol{k}}\Big)\Big]\big(\boldsymbol{v}_r\cdot\hat{\boldsymbol{k}}\big)^+d^2\hat{\boldsymbol{k}}\ . \tag{3.20}$$

The quantity $\nu_{(m,j),i}$ represent the contribution of the pair collision between $i$-th the $(m,j)$-th particles to the overall expected collisional frequency. The probability of collision between a selected $i$-th particle and a $(m,j)$-th neighbour particle can therefore be defined as:

$$\mathbb{P}\big[i,(m,j)\big] = \frac{1}{2}\frac{\nu_{(m,j),i}}{\langle N_C\rangle}\ ,\quad \text{so that: }\sum_{i,(m,j)}p_{i,(m,j)} = 1\ . \tag{3.21}$$

A direct evaluation of quantity (3.19) would require a computational effort proportional to $O(N_p^2)$; procedures with super-linear complexity are generally considered unacceptable in direct particles simulations. The null-collision technique allows to reduce the computational burden by employing a stochastic process that first produces an overestimate of the total number of collisions and then selects a subset composed of "true" collisions, the rest being "fake". This technique is also named *majorant frequency method* as it is based on the following inequality:

$$\nu_{(m,j),i} \le \overline{\nu}_{(m,j),i} = \sigma^2\frac{A_iC_i}{N_m}\int_{S(m,j)}d^2\boldsymbol{k}\ ; \tag{3.22}$$

$$A_i \ge n_m\chi\Big[n\Big(\boldsymbol{r}_i-\frac{\sigma}{2}\hat{\boldsymbol{k}}\Big)\Big]\quad \forall\hat{\boldsymbol{k}}\in S(m,i),\forall m\ ,\quad C_i \ge ||\boldsymbol{v}_{(m,j)}-\boldsymbol{v}_i||\quad \forall(m,j)\quad , \tag{3.23}$$

which leads to the following definition for the estimate number of collisions:

$$\overline{N}_C = \frac{1}{2}\sum_{i}^{N_p}\sum_{m}^{M_i}\sum_{j}^{N_m}\overline{\nu}_{(m,j),i} \le 2\pi\sigma^2\sum_{i}^{N_P}A_iC_i\ . \tag{3.24}$$

Once the above estimated is provided, the process to select and compute collisions reads as follows:

1. The $i$-th particle in $\mathbb{D}$ is selected for collision with probability:

$$p_i^{(1)} = \frac{A_i C_i}{\sum_{i=1}^{N_p} A_i C_i} \quad . \tag{3.25}$$

2. A random vector $\hat{\boldsymbol{k}}$ is selected on a unit sphere of radius $\sigma$; the $m$-th cell in the neighbourhood of the $i$-th particle is selected with probability:

$$p_{m,i}^{(2)} = \frac{\sum_{j=1}^{N_m} \overline{\nu}_{(m,j),i}}{\sum_{m=1}^{M_i} \sum_{j=1}^{N_m} \overline{\nu}_{(m,j),i}} = \frac{1}{4\pi} \int_{S(m,i)} d^2\hat{\boldsymbol{k}} \quad . \tag{3.26}$$

3. The $j$-th particle in the $m$-th cell is selected with probability:

$$p_{(m,j),i}^{(3)} = \frac{\overline{\nu}_{(m,j),i}}{\sum_{j=1}^{N_m} \overline{\nu}_{(m,j),i}} = \frac{1}{N_m} \quad . \tag{3.27}$$

4. The collision between $i$ and $(m,j)$ is either accepted (true collision) or rejected (false collision) accordingly, with probability:

$$p_{(m,j),i}^{(4)} = \frac{\int_{S(m,i)} \varphi_{(m,j),i}(\hat{\boldsymbol{k}}) d^2\hat{\boldsymbol{k}}}{\int_{S(m,i)} d^2\hat{\boldsymbol{k}}} \quad , \tag{3.28}$$

being:

$$\varphi_{(m,j),i}(\hat{\boldsymbol{k}}) = \frac{n_m \chi\big[n\big(\boldsymbol{r}_i - \sigma\hat{\boldsymbol{k}}/2\big)\big]\big(\boldsymbol{v}_r \cdot \hat{\boldsymbol{k}}\big)^+}{A_i C_i} \quad . \tag{3.29}$$

Once the collision has been selected as true, the new velocities for the $i$-th and $(m,j)$-th particles are computed, according to the following relations:

$$\begin{cases} \boldsymbol{v}_i^* = \boldsymbol{v}_i + (\boldsymbol{v}_r \cdot \hat{\boldsymbol{k}})\boldsymbol{v}_r \\ \boldsymbol{v}_{(m,j)}^* = \boldsymbol{v}_{(m,j)} - (\boldsymbol{v}_r \cdot \hat{\boldsymbol{k}})\boldsymbol{v}_r \end{cases} , \tag{3.30}$$

thus ensuring the exact global conservation of momentum and kinetic energy.

Lastly, let us discuss how the majorants involved in the computation of the total number of collisions are computed and updated. $A$ and $C$ are stored locally in each cell:

$$A_{ij} = \max_{k=0,\ldots,N_{p,ij}} A_k \ , \quad C_{ij} = \max_{k=0,\ldots,N_{p,ij}} C_k \ . \tag{3.31}$$

A first estimate is computed by sampling $N_{test}N_p$ particles, with $N_{test} \gg 1$, and selecting a possible collision partner for each one. The majorants are not updated at every time step, for efficiency reasons; this may cause the estimate of (3.28) to be grater than 1. If the number collisions for which $\hat{p}_{(m,j),i}^{(4)} > 1$ exceeds a pre-defined threshold, the majorants are re-computed. On the other hand, until the threshold is reached, the number of total collisions is reduced at each time steps, in order not to waste time simulating fake collisions.

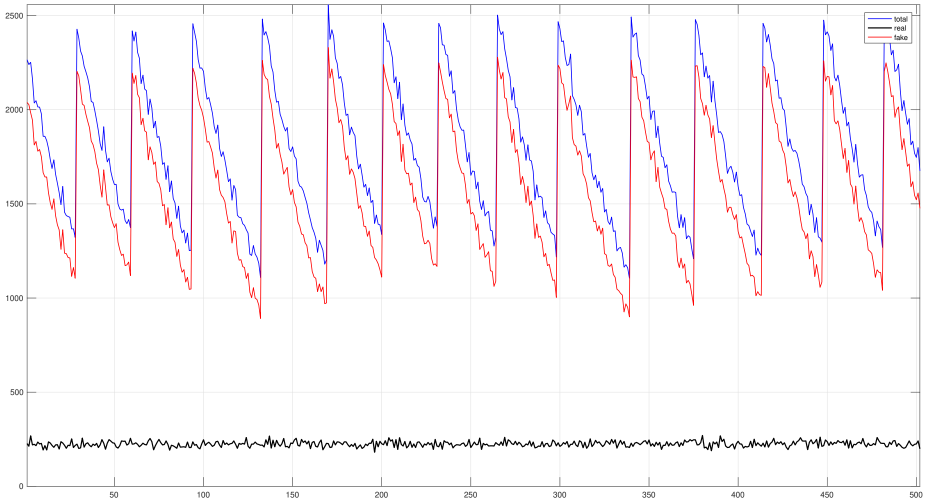


Figure 3.4: Time series of real (black), fake (red) and total (blue) collisions for 500 time-steps, displaying the effect of majorants updating. The simulated system is a homogeneous gas composed by 150000 particles, in a 120 × 120 grid ($T = 0.50$, $n = 1.2213 \times 10^{-2}$).

## 3.7 Thermostat and Sampling

Fluid temperature $T_0$ is prescribed in the initial configuration by sampling velocities from a suitable Maxwellian. At $t = 0$, once the ensemble has been populated, both kinetic and potential energy have been set; hence, total energy has been imposed as well:

$$U(0) = E_{kin}(0) + E_{pot}(0) = \frac{1}{2}\sum_{i=1}^{N_p} m_i ||\boldsymbol{v}_i(0)||^2 + \sum_{i=1}^{N_p}\sum_{j>i} \phi\big(\boldsymbol{r}_i(0), \boldsymbol{r}_j(0)\big) = U(t) \quad \forall t \ . \tag{3.32}$$

While the total energy is conserved in time, kinetic energy can transform into potential energy, and vice-versa. In other words, the conditions for a *microcanonical* statistical ensemble are implicitly imposing initially, while the desired macroscopic constraint on temperature defines a *canonical* ensemble. In order to preserve the desired temperature, a simple Gaussian thermostat procedure has to be employed: its purpose is to rescale velocities every $N_{ther}$ iterations of the DSMC algorithm, in order for them to match $T_0$ asymptotically; more advance thermostats can be found in literature (e.g. Berendsen [41]). The thermostat may be applied globally to the whole ensemble or locally, confining its action to a portion of the computational domain.

Macroscopic quantities are obtained via a sampling procedure. In statistical mechanics, *ensemble averages* of any given quantity $Q$ can be computed as *time averages* on a suitably long time window whenever the ergodic hypothesis holds; more practically, in particle simulation time averages can be employed if the sampling time is chosen sensibly shorter than the characteristic time of the phenomena to capture:

$$\langle Q \rangle \simeq \frac{1}{\Delta t_{sam}} \int_0^{\Delta t_{sam}} Q\big(\boldsymbol{r}(t), \boldsymbol{v}(t)\big) dt \ . \tag{3.33}$$

All physical properties of interest are evaluated in each cell. The sampling procedure involves capturing, averaging for each particle in the cell ($N_{p,ij}$) and averaging over $N_{sam}$ sampling steps; therefore, the total sample has size $N_{p,ij}N_{sam}$: increasing one of the two factors leads to less-noisy results. The relevant literature, however, suggests constraint on the minimum values of these parameters; Bird reports having at least 10 particles per cell, as rule of thumb [12]; physical arguments would suggest choosing the sampling time large enough that a signal moving at the speed of sound could traverse the computational domain at least more than one time.

## 3.8 Computational complexity

In order to assess the asymptotic complexity of the DSMC scheme each step is analysed separately.

**Density** : the binning procedure is $O(N_p)$, as cell coordinates need to be computed for each particle of the ensemble; the averaging procedure is instead $O(N_{cells}N_w)$, being $N_{cells} = N_xN_y$ and $N_w$ the number of values stored in the weights matrix (generally $N_w \ll N_{cells}$).

**Force-field** : the convolution between the kernel matrix and density is performed directly by point-wise multiplication and reduction, which entails a complexity of $O(N_{cut}N_{cells})$, being $N_{cut}$ the number of values stored in the kernel matrix; non-ionic pair potentials usually decay rapidly, so that also in this case $N_{cut} \ll N_{cells}$; however, this procedure is typically more computationally expensive than density averaging, as $N_w \ll N_{cut}$.

**Streaming** : time marching procedure is $O(N_p)$, since each particle is propagated independently using data already available.

**Collisions** : assessing the complexity of the collisional phase deserves more thought; the computation of the expected number of collisions is $O(N_p)$, provided the majorants are already available. Computing the majorants at each time step would take in principle $O(N_pM_iN_m)$ operations, which is infeasible, as it would essentially have the same complexity of the collision-frequency technique; therefore, majorants are initialized explicitly only initially and re-computed only when a collision takes place, therefore involving $O(\overline{N}_C)$ operations.

**Sampling** : the sampling procedure is $O(N_p)$. Every time-step, cell coordinates are extracted for each particle; velocities and local densities are aggregated in each cell. The actual computational cost of this procedure considerably depends on the number of physical properties being sampled.

## 3.9 General issues related to DSMC

In this last section some references regarding consistency and uncertainty quantification for DSMC are listed. The results obtained by the authors reported below concerns the standard DSMC algorithm (i.e. the one derived from Boltzmann's equation), while no result regarding the procedure for dense gas has been proposed in the relevant literature; we believe it is however opportune to report these papers, for the sake of completeness.

### 3.9.1 Consistency to kinetic equation

Given the physical basis of the DSMC method, the existence, uniqueness and convergence issues that are important in the traditional mathematical analysis of equations are often overlooked [1]. Furthermore, attempts to construct an analytical representation of all procedures have generally proved to be inimical, a common issue being the difficulty to incorporate many important practical details, especially regarding the probabilistic aspects of DSMC.

The main theoretical convergence result for the *original* DSMC algorithm was proposed by Wagner [42]. The proof associates a random empirical measure to the DSMC algorithm, whose limit for vanishing discretization and for an infinite number of particle is shown to be a deterministic measured-valued function satisfying an equation close to the Boltzmann equation. A Markov jump process is introduced, which is related to Bird's collision simulation procedure via a random time transformation; convergence is established for the Markov process and the random time transformation. Another convergence proof has been presented by Babovsky and Illner [43]; this one, however, refers to a variation of the original DSMC method introduced by Nambu [44], which does not posses the property of total energy conservation.

### 3.9.2 Convergence and error analysis

Facing the topic of error analysis and convergence in case of DSMC simulation may be a challenging task. Indeed, this method employs deterministic but also stochastic procedures: all physical macroscopic properties of interest are related to these either directly or indirectly and analysing these errors mathematically may be too involved. Moreover, the parameters influencing numerical errors and convergence are various and involve the time step, the cell dimensions, the number of particles and sampling time window.

Chen and Boyd studied the statistical error associated with DSMC [45]: the root-mean-square error related to the computed physical properties is expressed via a model equation involving a statistical scattering term and a bias term [46], with the intention of analysing and quantifying the effect of the number of particles per cell and the sample size. Alexander, Garcia et al. examined the problem of cell size dependence by comparing the calculated transport coefficients for a Couette flow with their reference values obtained from Green-Kubo theory [47]; it was shown that the truncation error in the transport coefficients is $O(\Delta x^2)$. In a similar spirit, Wagner and Garcia studied the time-truncation error, finding a $O(\Delta t^2)$ dependency with respect the reference transport coefficients [48].

# Chapter 4

# Code Implementation

In order to implement the DSMC algorithm described in the previous chapter, a C++ code with MPI parallelization routines has been produced; the main features as well as the design principles of the code will be presented in this chapter.

DSMC software has been developed in the past years both for research and industrial applications; notable examples are SPARTA (Stochastic PArallel Rarefied-gas Time-accurate Analyzer), by Sandia National Laboratories [66], MONACO, by I. D. Boyd (Department of Aerospace Engineering, University of Michigan) [52] and DS2V, by G. A. Bird [12]. For the purposes of this work one *ad hoc* code is developed, in order to study and analyze the algorithms needed to address the case described by Enskog-Vlasov equation; in fact, the need to compute a self-consistent mean field and to perform non-local collisions poses a challenge uncommon to standard DSMC codes.

C++ is a widely used programming language in the field of scientific computing; its value lies in the possibility to exploit general programming techniques, while retaining a satisfactory computational efficiency. DSMC is a inherently computationally intensive algorithm: the bulk of the code has therefore to be optimized in term of computational complexity, memory allocation and compilation; however, it is still possible to renounce to a certain degree of computational efficiency in the phases of pre- and post-processing, when general programming paradigms (e.g. run-time polymorphism) may be desirable.

Particles schemes are ordinarily easy to parallelize, as the communication between processes is usually very limited; indeed, in the standard DSMC procedure communication is employed only to move particles from a process to another. In our case, however, non-locality (due to interactions and collisions) poses an additional issue. The goal of this work is to provide thorough and scalable MPI procedures to manage particle exchange, mean-field computation and collisions.

While the code structure has been designed *de novo*, part of the algorithms defining the low-level procedures are based on a Fortran code developed by Professor Paolo Barbante [17].

## 4.1 Structure of the C++ code

While investing vast efforts into computational efficiency, the importance of software maintainability also needs to be addressed, especially in research projects, where codes are likely to be changed, for example, to test different physical models; modularity, therefore, represents an important aspect of software development for computational physics, and it is important to take this aspect into account in an early design phase [52]. With respect to the most used procedural languages (e.g. C, Fortran), C++ offers improvements in term of encapsulation, errors handling, unit testing and documentation; for these reasons much industrial and research software in the field of computational physics is moving toward C++ releases [55], [66].

The code for this project, although tailored to suit a specific DSMC procedure, has been developed taking into account the aforementioned design principles; the source code can be found in two separate public GitHub repositories (for the serial version and for the parallel one):

- `https://github.com/MichelePellegrino/enskog_vlasov_serial.git`
- `https://github.com/MichelePellegrino/enskog_vlasov_parallel.git`

### 4.1.1 DSMC modules

Modularity is facilitated by separating physical modelling, geometrical issues, and organizational tasks. All data related to one module should in principle kept local and can only be shared among subroutines in the same module: this encapsulated design ensures that source code changes cannot cause side effects outside of the module, so that searching for errors is also localized [52]. The design pattern implementing this paradigm has been inspired by the one utilized by SPARTA. The bulk of the source code is comprised by the classes representing the modules for each DSMC procedure. The main class, `DSMC`, encapsulate the whole algorithm; modules are implemented as classes, referred by `DSMC` via smart pointers. Each class inherits from a base (`Motherbase`), which contains references to smart pointers for all other classes: in this way it is possible to access public methods of the core classes. This expedient simplifies sharing data between modules. Moreover, code enhancement/maintenance is eased: adding an additional module would only require the addition of another smart pointer in the `DSMC` and the `Motherbase` classes; in case a module needs to be revised, changes in the code would be localized in the corresponding class.

The classes referenced by `DSMC` are briefly summarized below:

- `ParallelEnvironment` encapsulates information regarding the parallel environment (communicator, rank, size) and wraps some of the MPI functionalities used throughout the code.
- `ConfigurationReader` reads the input file, distributes data to other modules and set-up the initial configuration parameters; since most MPI implementations only

allow process 0 to access to standard input, reading is performed by the master, which then broadcast to slave processes.

- `Species` contains data regarding the physical properties of chemical species (in our case of interest: masses and cross-sections).
- `Times` contains numerical parameters for time marching, as well as the time constraints of the simulation.
- `Boundary` contains information regarding physical properties of boundaries (length, boundary conditions).
- `Grid` contains information regarding the computational grid (number of cells, cell volume, mesh width, centroids); it also encapsulates functions to map cells in lexicographic order.
- `Topology` defines the static domain decomposition and the partition of sub-domains in quarters (to perform parallel collisions, as will be explained in the following).
- `Ensemble` contains the data structure storing the particles and defines the functionalities to populate the ensemble and communicate particles between processes.
- `Thermostat` implements the functionality that rescales particles' velocities in order to match the prescribed temperature.
- `DensityKernel` stores density values and performs the operations of binning, density averaging and communication of boundary data regarding density between neighbouring processes; it also defines a map to locate particles referring to a given cell.
- `NondirectionalPairPotential` is a virtual class acting as base for the implementation of isotropic pair potentials (`SutherlandMie`, `SutherlandMorse`); classes in the hierarchy encapsulate the parameters defining the potential function, as well as the expression of its derivative and of the kernel function $\mathcal{K}$.

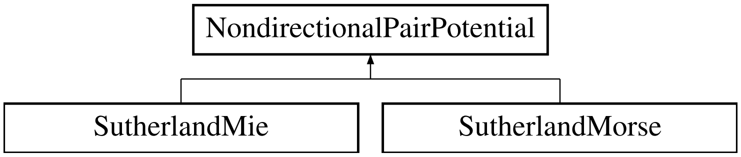


Figure 4.1: UML for the classes in the hierarchy of non-directional pair potentials.

- `CorrelationFun` is the class implementing a correlation function $\chi$ (by default: Carnahan-Starling expression).
- `ForceField` computes and stores the potential kernel matrix; it also encapsulates the functionalities to compute and store the force field.

- `TimeMarching` encapsulates the functionality handling particles' free-flight.
- `CollisionHandler` encapsulates all functionalities related to the collisional stage: computation and adjustment of majorants, simulation of collisions (local to the process and between processes) and gathering of collisional statistics.
- `Sampler` defines the functionalities to capture, average and gather samples from each sub-domain.
- `Output` deals with output files; since most MPI implementations don't provide any automatic scheduling of access to standard output, writing is performed by the master process only.

The code have been documented using Doxygen (*doxygen.nl*).

| DSMC |
|---|
| - DefaultPointer<ParallelEnvironment> |
| - DefaultPointer<ConfigurationReader> |
| - DefaultPointer<Species> |
| - DefaultPointer<Times> |
| - DefaultPointer<Boundary> |
| - DefaultPointer<Grid> |
| - DefaultPointer<Topology> |
| - DefaultPointer<Ensemble> |
| - DefaultPointer<Thermostat> |
| - DefaultPointer<DensityKernel> |
| - DefaultPointer<ForceField> |
| - DefaultPointer<TimeMarching> |
| - DefaultPointer<CollisionHandler> |
| - DefaultPointer<Sampler> |
| - DefaultPointer<Output> |
| - RandomEngine |
| - Stopwatch |
| - NondirectionalPairPotential |
| - CorrelationFun |
| + void initialize_simulation(void) |
| + void dsmc_iteration(void) |
| + void dsmc_loop(void) |
| + void output_all_samples(real_number) |

Table 4.1: Sketch of `DSMC` class layout.

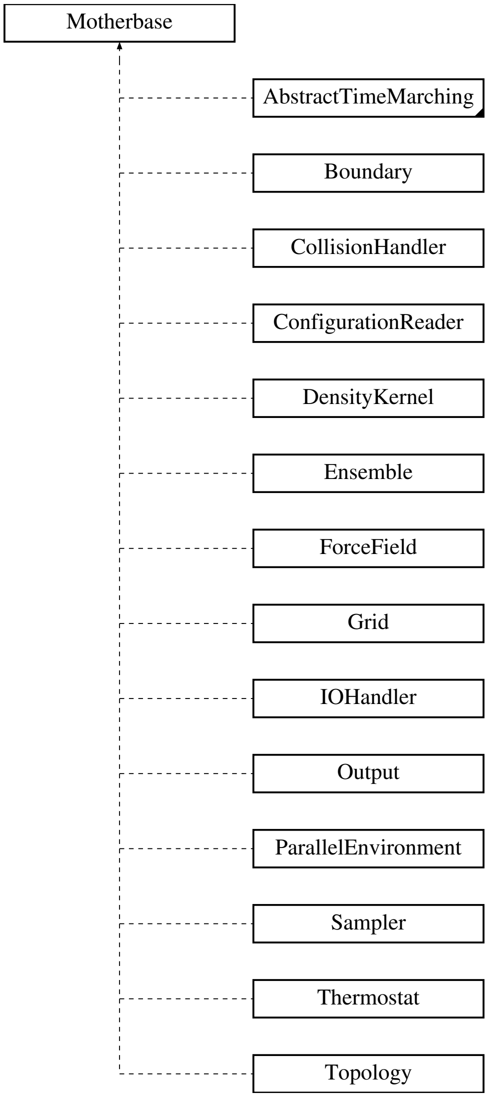


Figure 4.2: UML of the classes implementing DSMC modules; dashed arrows represent protected inheritance.

### 4.1.2 Libraries

Classes for numerical procedures and data structures are included in separate header files. Eigen library (*eigen.tuxfamily.org*) has been used to implement the arrays employed for storage. Dynamic-time polymorphism has been avoided in all stages that include intensive calculations, for efficiency reasons, in favour of static-time polymorphism; the latter has being implemented via the usage of template classes and template functions. The custom libraries are now briefly outlined:

- `integration.hpp` contains a class for numerical integration of 1D functions (with either bounded or unbounded support); it implements Romberg's quadrature scheme, which consists in $k$ refinements of the trapezoidal or the mid-point rule, in order to remove error terms less than order $O(N^{-2k})$ [7].

- `matrix.hpp` contains the classes implementing arrays for data storage; the base class of the hierarchy inherits from `Eigen::Array`, while MPI communication functionalities for exchanging blocks between processes have been added to the already available methods; moreover, these derived classes support Fortran-style negative indexing.

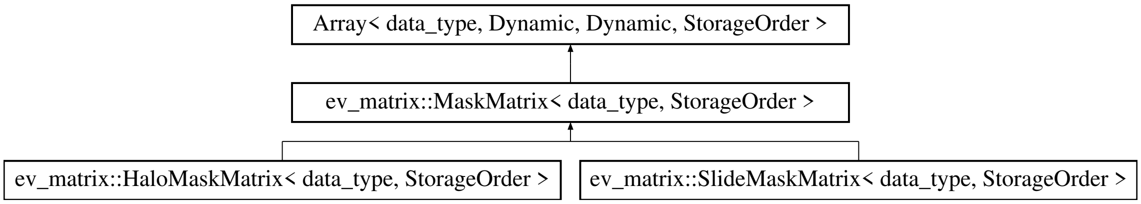

Array< data_type, Dynamic, Dynamic, StorageOrder >
ev_matrix::MaskMatrix< data_type, StorageOrder >
ev_matrix::HaloMaskMatrix< data_type, StorageOrder >
ev_matrix::SlideMaskMatrix< data_type, StorageOrder >

Figure 4.3: UML for the hierarchy of custom matrices for data storage.

- `random.hpp` consists in wrappers for STL RNG generators, as well as template specializations for custom ones; RNG represents a key element in DSMC, due to its stochastic nature, and investigating engines' suitability is paramount: a brief discussion is reported in the appendix.

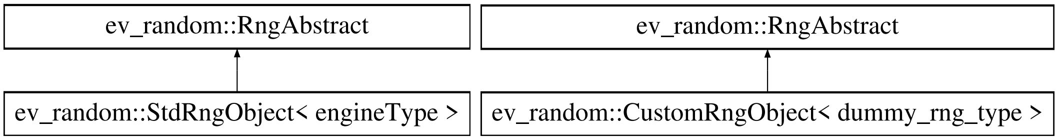

ev_random::RngAbstract
ev_random::StdRngObject< engineType >
ev_random::RngAbstract
ev_random::CustomRngObject< dummy_rng_type >

Figure 4.4: UML of the wrappers for (a) STL random number engines, (b) user-defined engines; in the latter case, template specialization defines a new kind of engine.

- `stopwatch.hpp` contains a class for timing numerical procedures, storing partial elapsed times.
- `types.hpp` contains numerical constants, macro for type names, and includes a functionality for converting C++ types to `MPI_Datatype`.
- `utility.hpp` includes all-purpose utility functions (i.e. exponentiation).

## 4.2 MPI parallelization

The world of parallel computing is, for the most part, divided into distributed-memory and shared-memory systems. A *distributed-memory* system (which represents our case of interest) consists of a collection of core-memory pairs connected by a network, and the memory associated with a core is directly accessible only to that core. These systems are usually programmed using the *message-passing* paradigm: the communication of signals and data between processes is performed via the exchange of messages. MPI (*Message-Passing Interface*) is a message-passing protocol which defines a library of functions that can be called from C, C++, and Fortran programs. MPI results suitable for SPMD (*Single Program Multiple Data*) paradigm: all process execute the same program, each one with its own local data [8], [9].

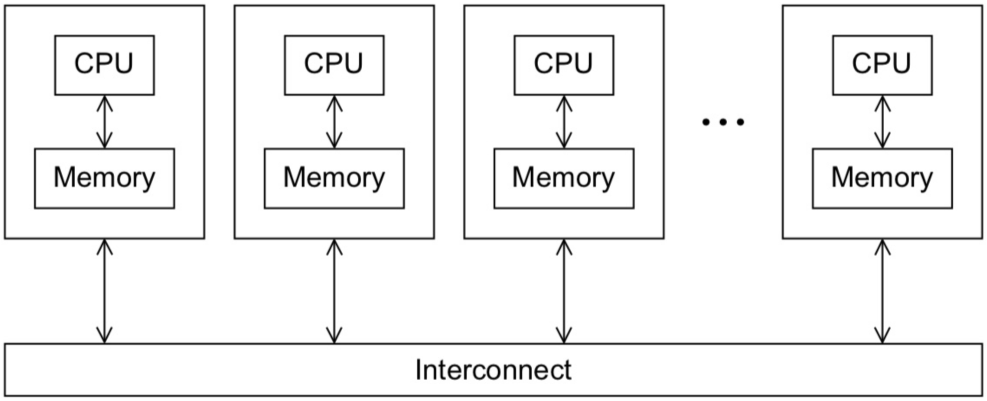


Figure 4.5: Layout of a distributed memory system.

A common parallelization strategy employed by DSMC codes is *domain decomposition* [51]: the computational domain is divided into a number of sub-domains equals to the number of processes that will be used; each process will therefore perform particles advection, collisions and compute the force field in within its sub-domain. The code currently supports *conforming* grids of rectangular sub-domains. Regarding data storage a *replicated grid* approach has been followed: each process stores all information regarding the computational grid, while particles and data concerning each sub-domain are distributed. Data within a sufficiently thick halo will be also collected in local memory, in order to be able to perform convolutions with the density and the force-field kernels.

One of the most significant obstacles to good parallel performance is the cost associated with inter-processor communication: from a software design perspective, serious consideration should be given to minimizing the amount of communication in an effort to delay the effects of this bottleneck as much as possible [51]. This can be accomplished by maximizing *data locality*: the data needed by each process should in principle be available locally, that is without involving communication.

Before delving into the details of each parallelization procedure, it is important to remark that the computational load for each process is characterized by the *number of particles*, the *number of computational cells* and also by the *amount of communication* needed at each step; these considerations are crucial in assessing parallel scaling and load balancing.

### 4.2.1 Particle streaming

Each process oversees the streaming of all particles that are within its sub-domain at the beginning of the time-step. Once all particles have been propagated, the ones that ended up outside the sub-domain are stored in a buffer and sent to the process they now belong to. Each process stores a incidence matrix and a counter for incoming and departing

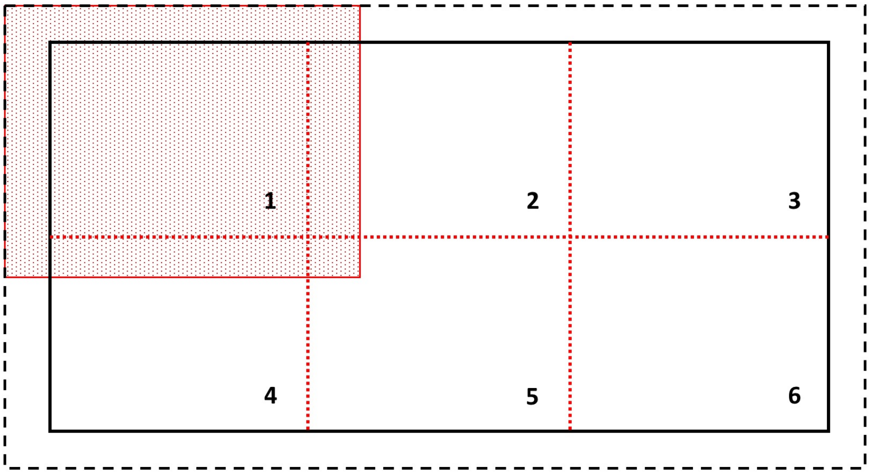
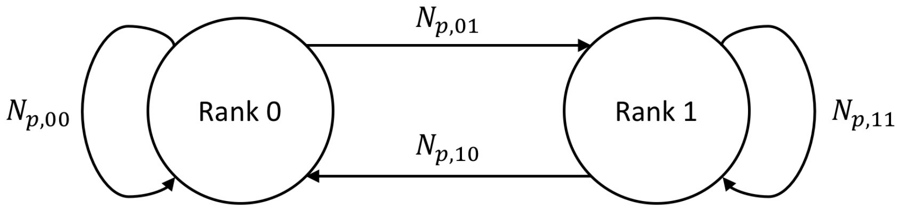


Figure 4.6: Layout of a feasible sub-domains grid; the shaded area represents the portion of data stored in local memory.

particles, as well as a map comprised by (rank, buffer) pairs; every time communication occurs buffers are resized, if needed, in order to contain enough memory, according to the incidence matrix and then particles are passed in blocks; the number of total particles per process is then updated accordingly.



| | rank 0 | rank 1 |
|---|---|---|
| rank 0 | 0 | $N_{p,01}$ |
| rank 1 | $N_{p,10}$ | 0 |

Figure 4.7: Scheme of particles communication between two processes; particles remaining inside their sub-domain are not counted in the incidence matrix, as they are not involved in the communication procedure.

A particle is represented by a struct containing data regarding its position, velocity, cell, domain rank and a tag; even if MPI does not support abstract data types, it is possible to construct aggregate types by using the `MPI_Type_create_struct` and `MPI_Type_commit` functions. A custom data type (`MPI_PARTICLE_TYPE`) is therefore committed, in order to be able to pass all quantities in blocks, without the need of multiple communication steps.

| MPI_PARTICLE_TYPE |
|---|
| double xp, yp |
| double vx, vy, vz |
| int cell_x, cell_x |
| int p_tag |
| int r_tag |

Table 4.2: Layout of the MPI data structure for particles.

### 4.2.2 Density kernel and mean-field computation

Regarding the computation of the force field, three steps have to be analysed: (a) the distribution of density values, (b) the actual computation of forces. Step (b) does not require any communication, provided each process already posses all density data within its halo. Step (a) instead requires matrix blocks to be communicated between processes.

In order to encapsulate block communication, as well as the convolution operation to compute the force field, three matrix classes are defined, augmenting the types already provided by Eigen:

- `MaskMatrix`, which stores force-field results and maps to the process sub-domain;
- `SlideMaskMatrix`, which stores the values of the potential kernel (inherits from `MaskMatrix` as specialization);
- `HaloMaskMatrix`, which stores density values and maps to the process sub-domain, plus its halo (inherits from `MaskMatrix` as it adds new features).

Class `MatrixConvolutioner` is used at each time-step to perform the convolution between density and potential kernel. `HaloMaskMatrix` also contains the subroutines to send/receive halo blocks. Communication of blacks is performed by the density kernel; no communication is employed in the forces kernel, making this procedure quite simple to parallelize.

Lastly, the same scheme is applied when computing the averaged density field, for the computation of correlation: the matrix of weights will be stored in a `SlideMaskMatrix`, while the resulting density values in a `MaskMatrix`.

### 4.2.3 Collisions

Message passing for collisions simulation is designed similarly as for streaming, since the information to be exchanged mostly concerns single particles instead of cells: it may

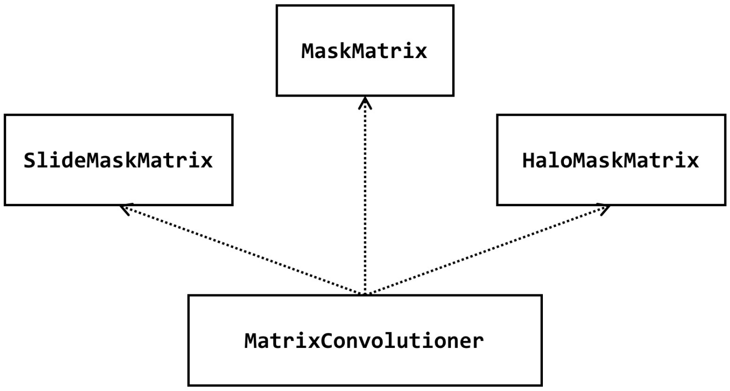


Figure 4.8: Class diagram for the types defined in `matrix.hpp` header; dotted arrows represent references.

happen that when process $i$ selects a collisional pair the second particle lies in the subdomain of process $j$; in this case the particle's velocity, as well as the local number density, has to be communicated to $i$; if the collision results to be true, the newly updated velocity vector has to be communicated back to $j$.

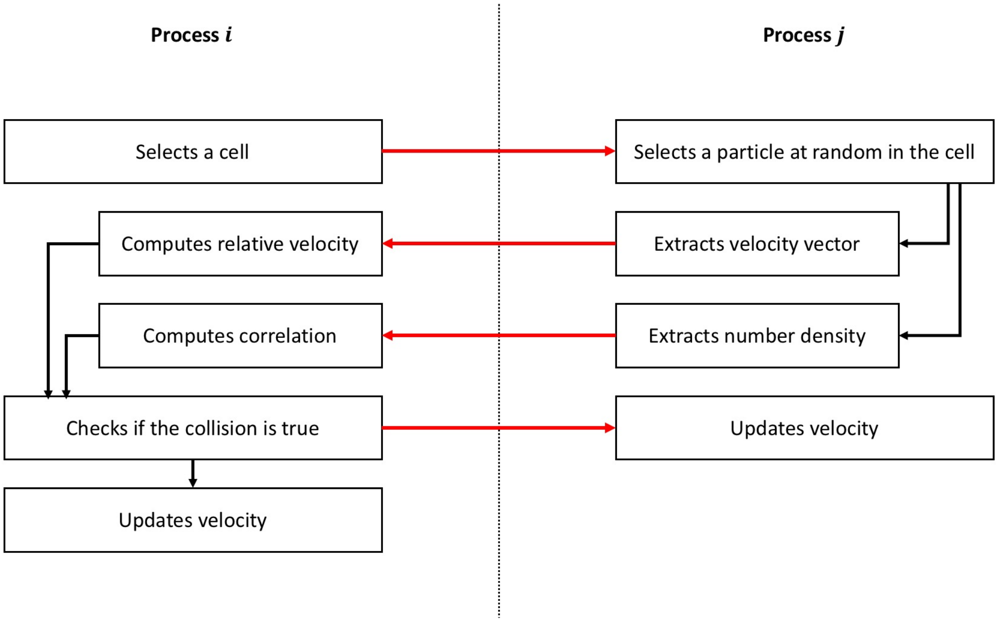


Figure 4.10: Flow chart for inter-processes communication in the collision kernel; red arrows represents message-passing.

A problem regarding collisional conflicts has to be tackled: incoherent velocity modifications between neighbouring processes needs to be prevented; indeed, it may happen that both processes $i$ and $j$ selects the same collisional pair; then, after the collision occurs, final velocities would almost surely differ for each process; since the algorithm does not

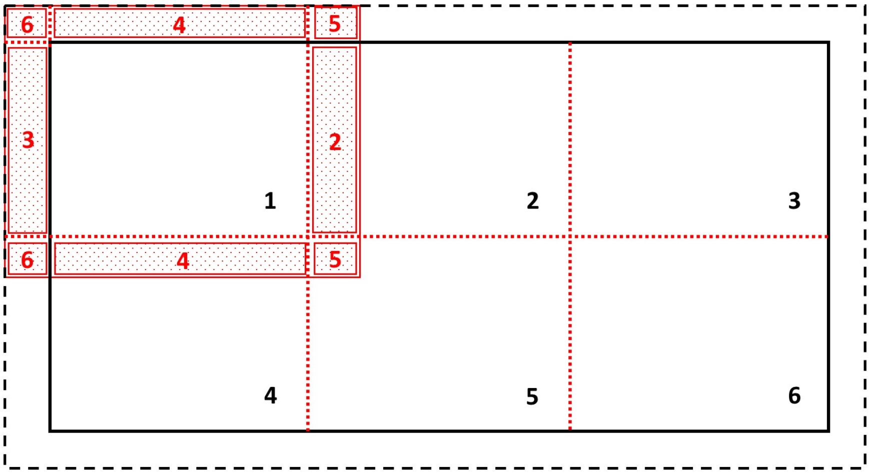


Figure 4.9: Layout example, showing the blocks that need to be sent to process 1 in case of periodic boundary conditions on every edge of the domain.

prescribe an order in which collisions should be performed, neither $i$ nor $j$ claims priority, and so a conflict in updating particles velocities arises. A way to circumvent this issue is to subdivide each sub-domain in quarters and to perform collisions concurrently in each process, but one quarter at the time; indeed, be the first particle in quarter $(i, k)$ and the second in $(j, l)$ (the first index referring to the process, the second to the quarter): if each sub-domain is sufficiently large, it will be $k \neq l$, so process $i$ would have priority in updating the particles' velocity (the collisional pair won't be selected by process $j$ until all collisions in $(j, k)$ are performed).

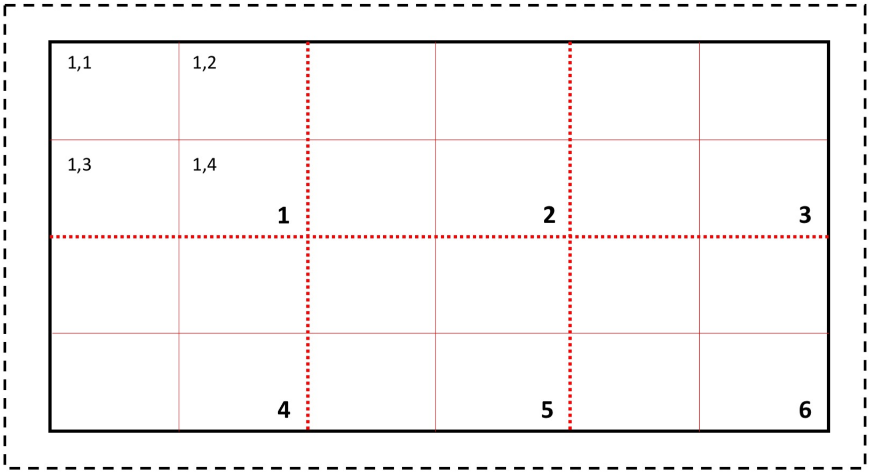


Figure 4.11: Layout of collisional quarters in each sub-domain.

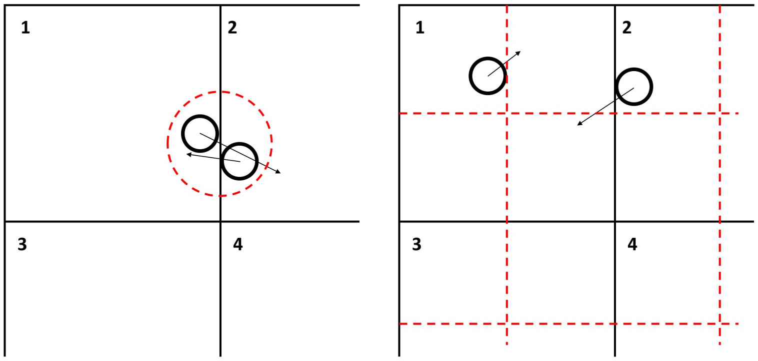


Figure 4.12: Left: two processes tries to compute the same collisions, leading to incoherence in the resulting post-collisional velocity. Right: the subdivision in collisional quarters prevents conflicts.

The procedure is devised so that collisions not involving communication are performed at first, while data for message-passing are stored in buffers; once all 'local' collisions have been performed the routine simulating collisions between processes is run. In order to prevent spurious correlations, cells are selected one at the time at random, instead of following a lexicographic order; the order of sub-domain quarters is also selected at random.

**Algorithm 1** DSMC parallel non-local collisions

```
procedure SimulateCollisions
    compute number of collisions per cell
    shuffle order of sub-domain quadrants
    for $q \in quadrants\_order\_array$ do
        $nc = n\_cells(q)$
        while $nc > 0$ do
            select a cell $(i_1, j_1)$ at random in $q$ with equiprobability
            select a particle belonging to the cell at random with equiprobability
            for $i = 1 : n\_collisions\_cell(i_1, j_1)$ do
                generate vector $\mathbf{k}$ s.t. $||\mathbf{k}|| = \sigma$ and the direction is random
                select the cell $(i_2, j_2)$ where vector $\mathbf{k}$ points
                if $(i_2, j_2)$ in within the sub-domain then
                    select at random with equiprobability a particle in the cell $(i_2, j_2)$
                    compute relative velocity and correlation
                    if collision is true then update velocities
                else
                    push cell and particle data in communication buffer
            $nc \leftarrow nc - 1$
        perform collisions involving communication
        update collisions statistics
```

# Chapter 5

# Results

In this chapter the DSMC code will be tested, both in terms of consistency with the physics of dense gas and in terms of parallel efficiency. Numerical simulations requiring either a considerable amount of computations or numerous cores (i.e. to asses speed-up) will be performed on the HPC system (cluster) of Politecnico di Milano; the available computational resources amount to 16 nodes DELL M630 with Intel(R) Xeon(R) CPU E5-2630 v3 @ 2.4GHz processors, for a total of 256 cores and 1024 Gb RAM. Smaller simulations, which do not require much computational resources, will be carried on a laptop (MacBook Pro 15" mid 2012, Intel Core i7 @ 2.6GHz, 8 Gb RAM).

## 5.1 Uniform liquid

The first test case for the DSMC code consists in checking the consistency of simulations with equation of state (2.42). In case of an algebraic potential (with parameters $\gamma$ and $\overline{\phi}$), EoS reads [19]:

$$p = CS(\eta) n k_b T - \frac{2\pi\sigma^3}{3}\frac{\gamma}{\gamma-3}\overline{\phi} n^2 \; ; \quad CS(\eta) = \frac{1+\eta+\eta^2-\eta^3}{(1-\eta)^3} \; . \tag{5.1}$$

Using the non-dimensional units defined at the beginning of chaper II we obtain:

$$p^* = CS(\eta) n^* T^* - \frac{2\pi}{3}\frac{\gamma}{\gamma-3}(n^*)^2 \; . \tag{5.2}$$

The equilibrium value for pressure can therefore be easily obtained; the results for some values of temperature and density have been tabulated; apices will be dropped in the following.

The kinetic contribution to the hydrostatic pressure can be computed from samples of the velocity field; according to its definition, in non-dimensional units [2]:

$$p_K = \frac{1}{3}\boldsymbol{P}_K : \boldsymbol{I} = \frac{1}{3} n^* \left\langle \boldsymbol{c}^2 \right\rangle = \frac{1}{3} n^* \left[ \left\langle c_x^2 \right\rangle + \left\langle c_y^2 \right\rangle + \left\langle c_z^2 \right\rangle \right] \, , \tag{5.3}$$

| $T$ | $T/T_C$ | $n_l$ | $n_v$ | $p$ | $\lambda_v$ |
|---|---|---|---|---|---|
| 0.40 | 0.5301 | $8.3952\times10^{-1}$ | $2.0776\times10^{-3}$ | $8.3338\times10^{-4}$ | 105.82 |
| 0.45 | 0.5964 | $7.7914\times10^{-1}$ | $5.6092\times10^{-3}$ | $2.4525\times10^{-3}$ | 39.315 |
| 0.50 | 0.6626 | $7.1866\times10^{-1}$ | $1.2213\times10^{-2}$ | $5.6822\times10^{-3}$ | 17.988 |
| 0.55 | 0.7288 | $6.5677\times10^{-1}$ | $2.3111\times10^{-2}$ | $1.1154\times10^{-2}$ | 9.4004 |
| 0.60 | 0.7951 | $5.9171\times10^{-1}$ | $3.9941\times10^{-2}$ | $1.9436\times10^{-2}$ | 5.3298 |
| 0.65 | 0.8613 | $5.2051\times10^{-1}$ | $6.5517\times10^{-2}$ | $3.1009\times10^{-2}$ | 3.1434 |

Table 5.1: Tabulated temperature, density, pressure and vapour mean-free path (in non-dimensional units) for a Lennard-Jones fluid ($\gamma$=6) [18]; $T_C$ indicated the critical temperature for phase coexistence.

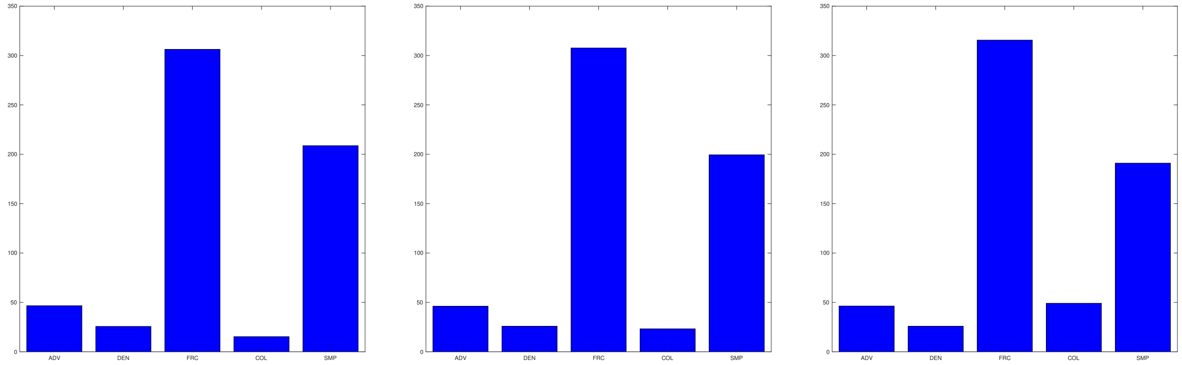


Figure 5.1: Averaged CPU time (in milliseconds) for advection, density kernel, forces computation, collisions simulation and sampling; the histograms correspond to the following cases: $T = 0.40$, $n = 2.0776 \times 10^{-3}$; $T = 0.50$, $n = 1.2213 \times 10^{-2}$; $T = 0.60$, $n = 3.9941 \times 10^{-2}$.

being $\boldsymbol{c} = \boldsymbol{v} - \boldsymbol{u}$ the *peculiar velocity* and $\boldsymbol{u}$ the local hydrodynamic velocity. We expect this quantity to diverge from the reference value (5.2) for dense configurations: as number density increases, the effect of collisional transfer and mean field interactions on the pressure tensor become preponderant.

The following results concerns simulations with a fixed parameters $\gamma = 6$. The system is composed by 150000 particles, while the domain dimensions are: $L_x = 35$, $L_y = 35$; the grid is composed by $120 \times 120$ cells, and the time-step have been set to $\Delta t = 0.5$; the simulations lasted 2000 time-steps: samples averages have been computed on windows of 200 steps and the thermostat has been applied every 20 iterations.

CPU time diagrams show that in the case of dense vapour the force-field stage dominates, in term of computations performed; the time expended for the collisional phase is negligible for rarefied gas, and it becomes more noteworthy as density increases. It can be noticed that the hydrostatic pressure obtained from simulations starts to diverge from its reference value as the vapour density increases (or, alternatively, as temperature approaches its critical value), as expected.

Simulations involving (a) an increased number of cells, (b) a larger sampling window and the number of particles are performed in order to asses the accuracy of the results. A new system of 625000 particles in a $250 \times 250$ grid is considered; the same behaviour

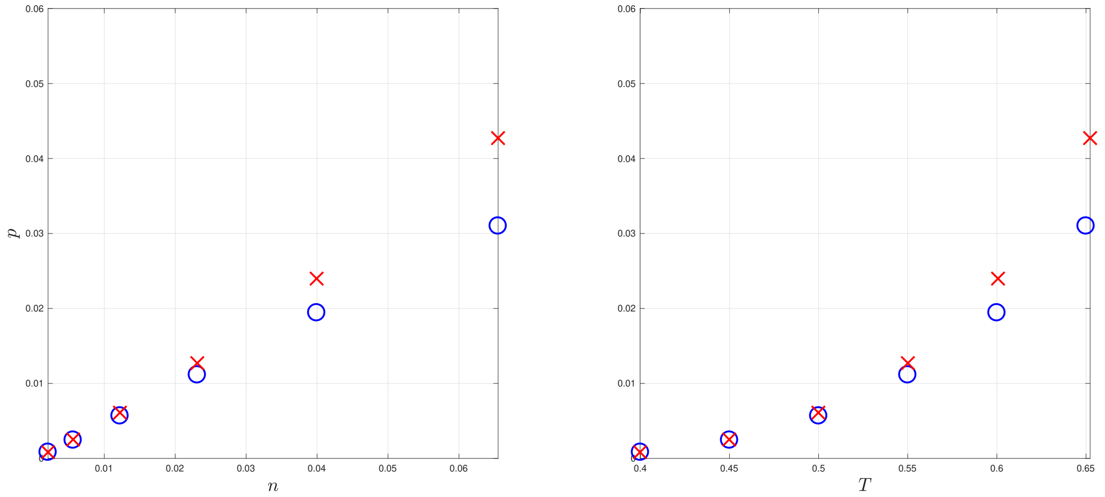


Figure 5.2: Results for the EoS calculations; blue circles (O) represent the reference values, while red crosses (×) the computed ones.

| $p_{ref}$ | $p_A$ | $p_B$ | $p_C$ | $p_D$ |
|---|---|---|---|---|
| 0.0194 | 0.0240 | 0.0240 | 0.0240 | 0.0240 |
| 0.0310 | 0.0427 | 0.0426 | 0.0426 | 0.0426 |

Table 5.2: Comparison with reference pressure for cases: (A) $120 \times 120$, 150000 particles; (B) $250 \times 250$, 625000 particles; (C) $250 \times 250$, 1250000 particles; (D) $350 \times 350$, 1250000 particles.

is observed. In fact, the values obtained from the new computation perfectly match the values from the old one. A further increase in the number of cells and particles produces similar results.

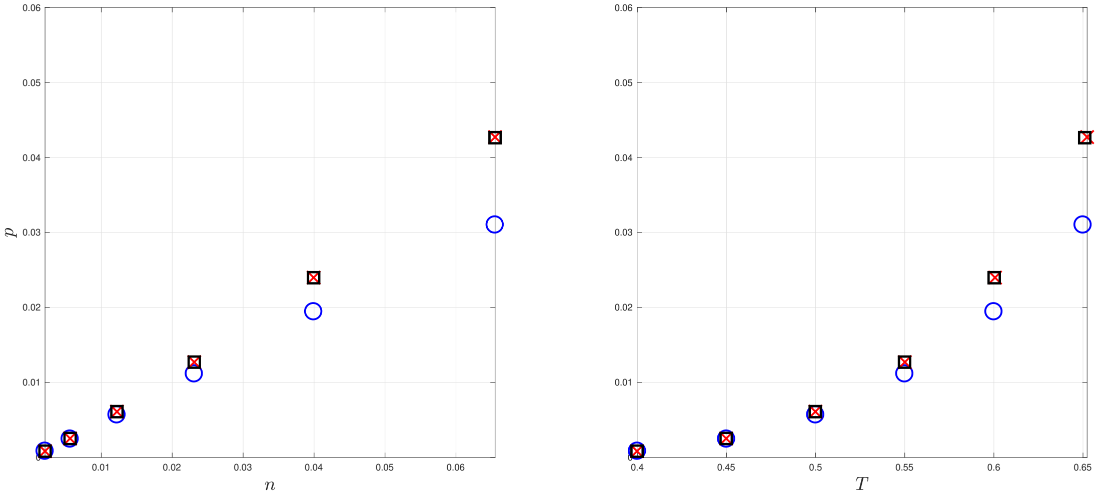


Figure 5.3: Comparison between results from 'coarser' (blue circles) and 'finer' (black squares) simulations.

## 5.2 Knudsen layer formation

The second test case consists in reproducing equilibrium density profile of a liquid-vapour interface. The simulation will be initialized by arranging a homogeneous liquid slab at temperature $T < T_c$ in the centre of the computational domain. The system evolution is then computed until the evaporation of part of the liquid brings the liquid and vapour in equilibrium [19].

A quantity characterizing the two-phase system is the *reciprocal interface thickness* (i.e. the thickness of the Knudsen layer); its characteristic length can be defined as [56]:

$$l^* = \frac{T}{\pi p} \; ; \tag{5.4}$$

a possible estimate of the reciprocal interface thickness can be obtained from the density profile along the non-uniform direction, as [19]:

$$\hat{l}^{-1} = \max_x \left| \frac{dn}{dx} \right| \frac{1}{n_l - n_v} \; . \tag{5.5}$$

The test for the Knudsen layer formation does not provide the results we hoped for: non-physical oscillations appear at the interface between liquid and vapour. A steep increase of the number density at the interface is typical of contact with hydrophilic surfaces, which naturally is not the configuration we are attempting to simulate. This observation leads to a thorough examination of the routines computing the force field and collisions.

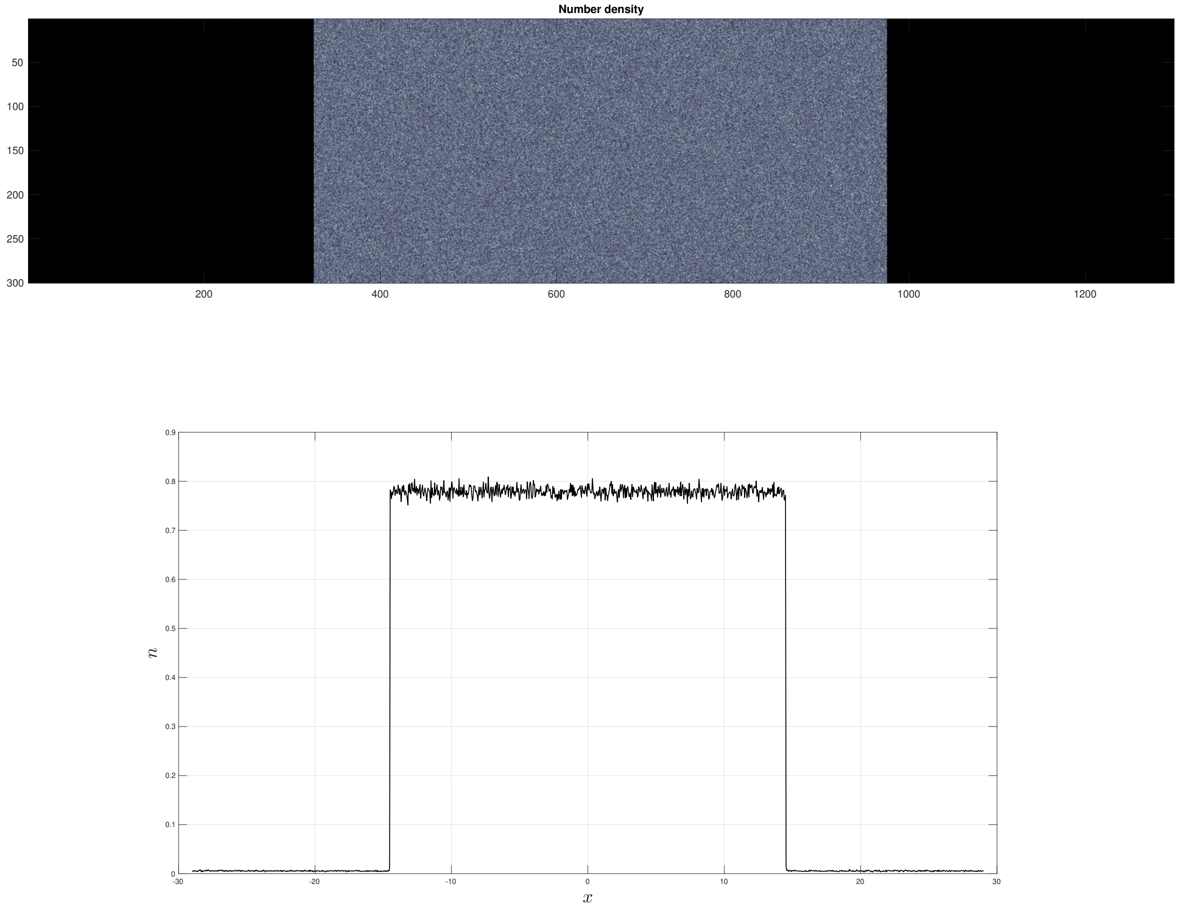


Figure 5.4: Initial configuration for a system composed by $5 \times 10^6$ particles in a $1300 \times 300$ grid (@ $T = 0.45$, $n_l = 0.7791$, $n_v = 5.619 \times 10^{-3}$); the density profile along $x$ is obtained by averaging along $y$.

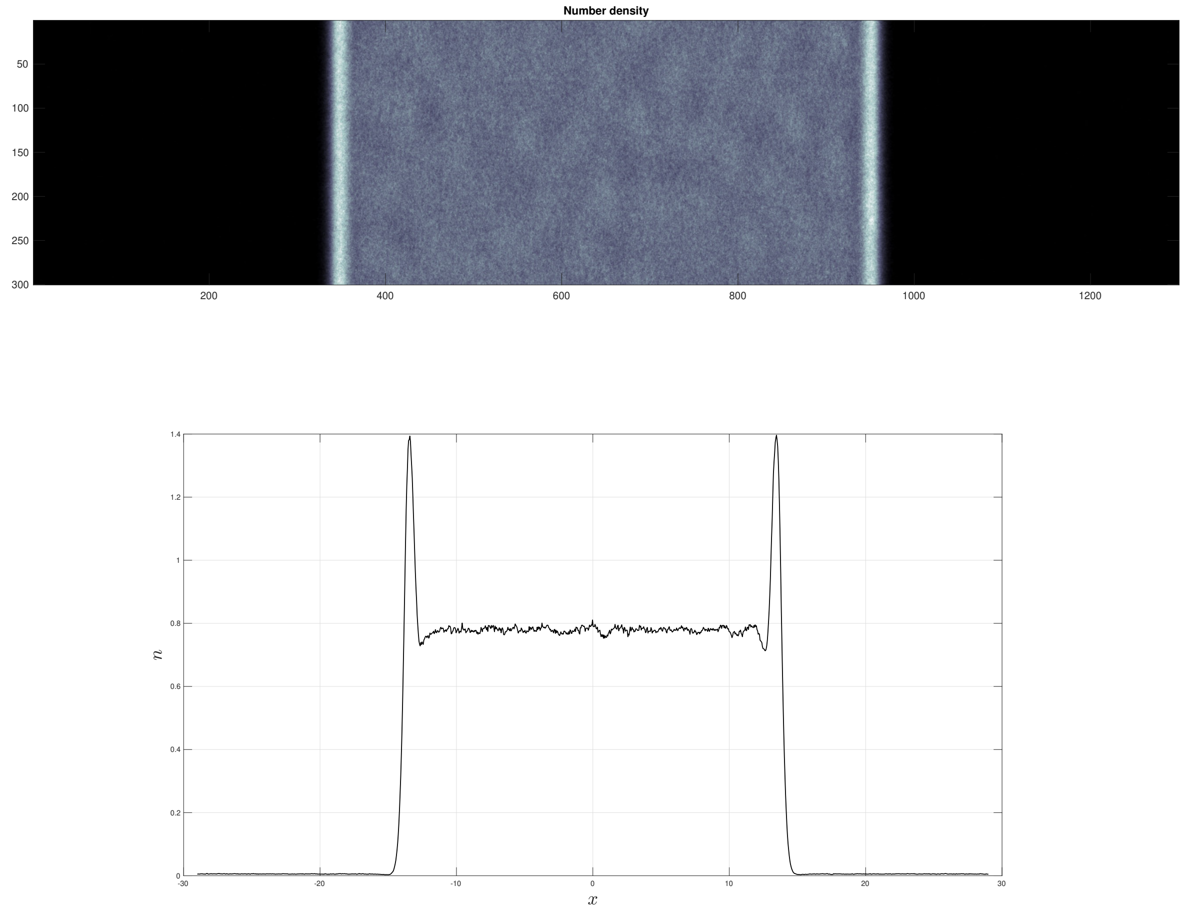


Figure 5.5: Density profile after 400 time-steps ($\Delta t = 0.25 \times 10^{-2}$).

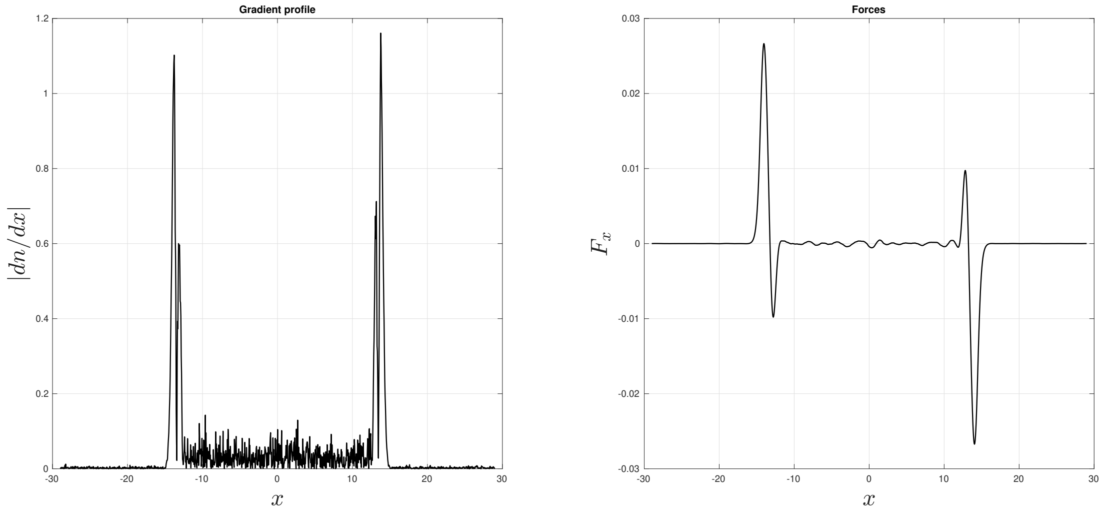


Figure 5.6: Profile of $|dn/dx|$ (left) and $F_x$ (right).

## 5.3 Parallel scaling

In order to evaluate the performance of the code in terms of parallelization, the following *speed-up* and *efficiency* parameters are introduced:

$$S = \frac{T_{serial}}{T_{parallel}} \; , \quad E = \frac{S}{N_{cores}} \; , \tag{5.6}$$

being $N_{cores}$ the number of processes. In case of ideal parallelization $S$ equals the number of cores (i.e. $E = 1$); however, since it is not possible to avoid the code having a serial fraction, ideal scaling is never achieved. More realistically, speed-up would follow *Amdahl's law*:

$$S = \frac{1}{f_s + \frac{f_p}{N_{cores}}} \; , \tag{5.7}$$

being $f_s$ the serial fraction of the code, $f_p$ the parallel fraction. Amdahl's law is valid for a small number of processors; in case of distributed memory architectures with many cores, communications would lead to a further degradation of performance [57].

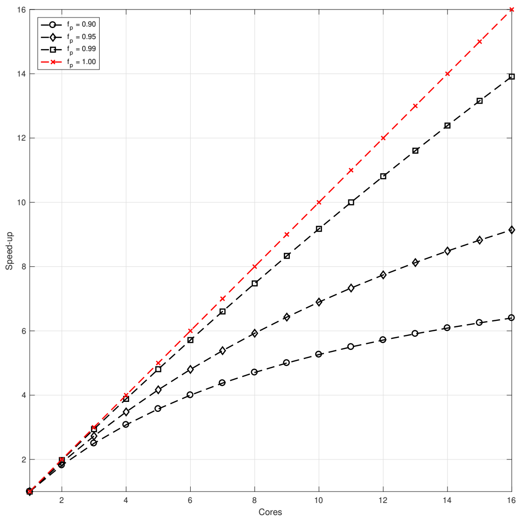


Figure 5.7: Amdahl's law in case of 16 cores and with $f_p = 1.00, 0.99, 0.95, 0.90$.

Furthermore, it is important to establish which kind of scaling to evaluate. *Strong scaling* is assessed when the problem size is kept fixed, while the number of processes is increased; otherwise, *weak scaling* is assessed when the size the problem is increased with the number of processes. In the following the strong scaling of the parallel procedure will be tested.

The case of a homogeneous dense gas represents an important benchmark for the parallel procedure; given the particles are equally distributed in space and the mesh-width is uniform, load should in principle be statically balanced, entailing a theoretically good parallel performance. Here the results for a simulation of 4500000 particles on a 700 × 580 grid are displayed; the computational domain has been evenly divided between processes.

The parallel scaling appears overall satisfactory. The force-field computation scales

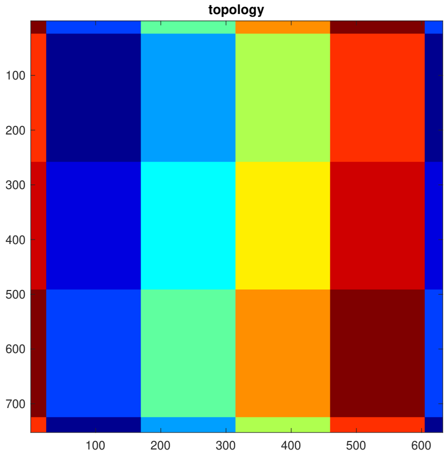


Figure 5.8: Layout of the parallel sub-domains; each colour corresponds to a different process. Currently the code support domain decomposition in quadrants and in stripes, along $x$ and $y$ direction.

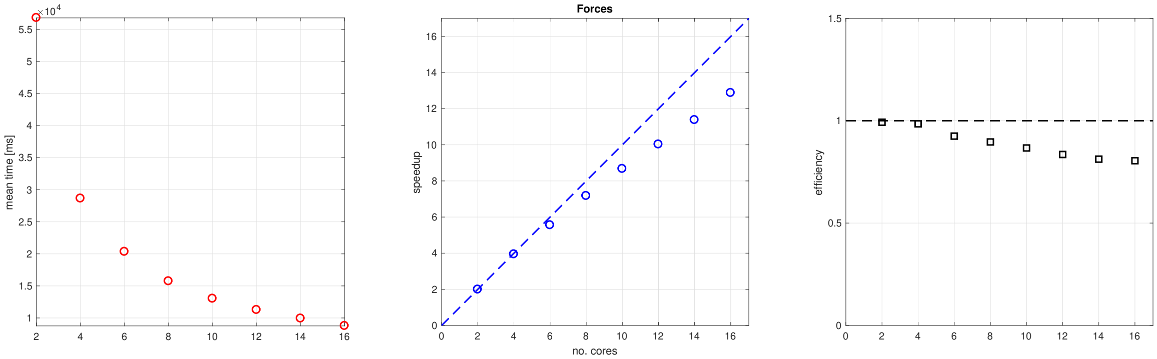


Figure 5.9: CPU time, speed-up and efficiency of the force field kernel in case of a homogeneous dense gas.

particularly well, due to the lack of communication; the deviation from ideal behaviour can be explained by considering that the data needed for computing the force field is proportional to $N_{cells}[k] + N_{cut}$, being $k$ the core label: as the number of cores increases, $N_{cells}[k]$ decreases quadratically, while $N_{cut}$ only linearly. The density kernel exhibits less-satisfactory scaling; this is probably due to large amount of communication needed for transferring boundary data from a process to another. Collision kernel achieves the worst scaling, most probably due to the fact that it requires three phases of point-to-point communication between pairs of processes, each time collisions are performed in a quadrant. The overall scaling, in case of a dense vapour, appears satisfactory. Testing and exception handling features should in general be augmented.

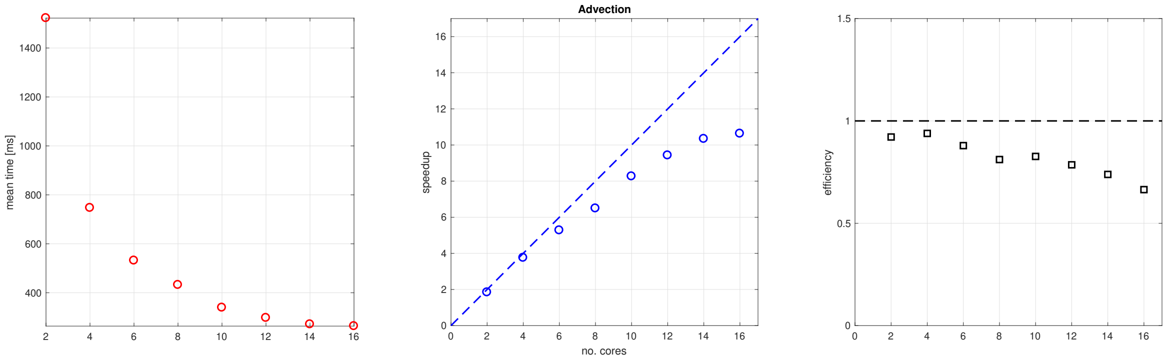


Figure 5.10: CPU time, speed-up and efficiency of the advection kernel in case of a homogeneous dense gas.

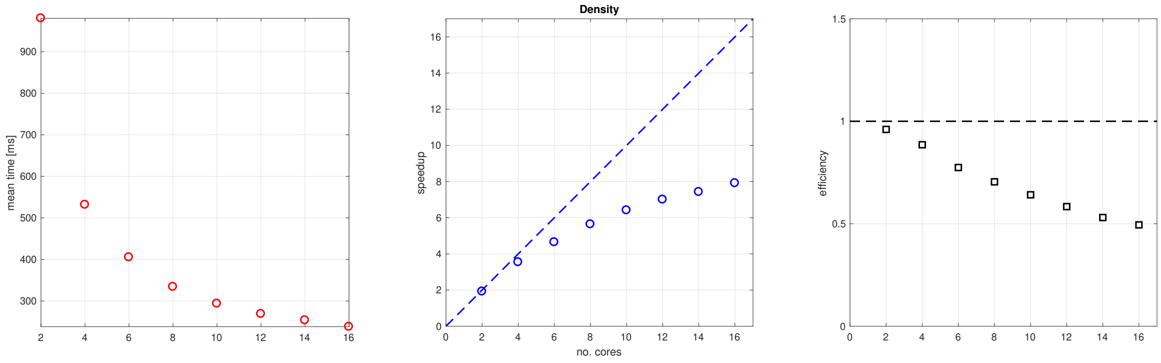


Figure 5.11: CPU time, speed-up and efficiency of the density kernel in case of a homogeneous dense gas.

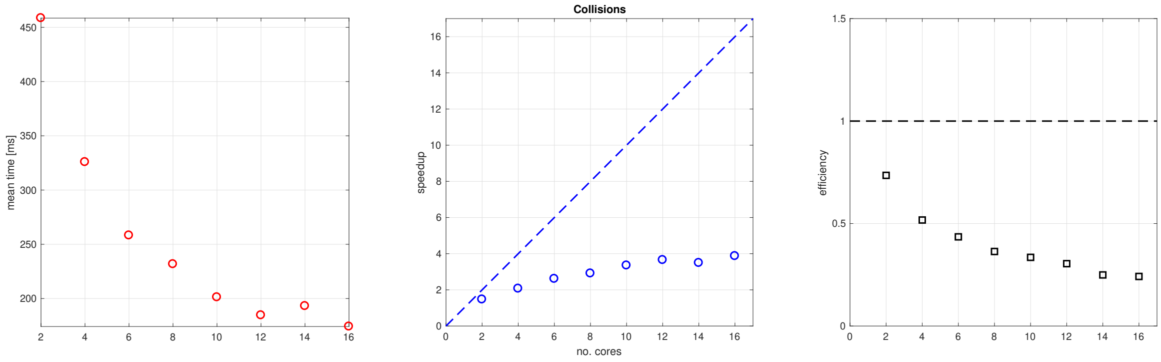


Figure 5.12: CPU time, speed-up and efficiency of the collision kernel in case of a homogeneous dense gas.

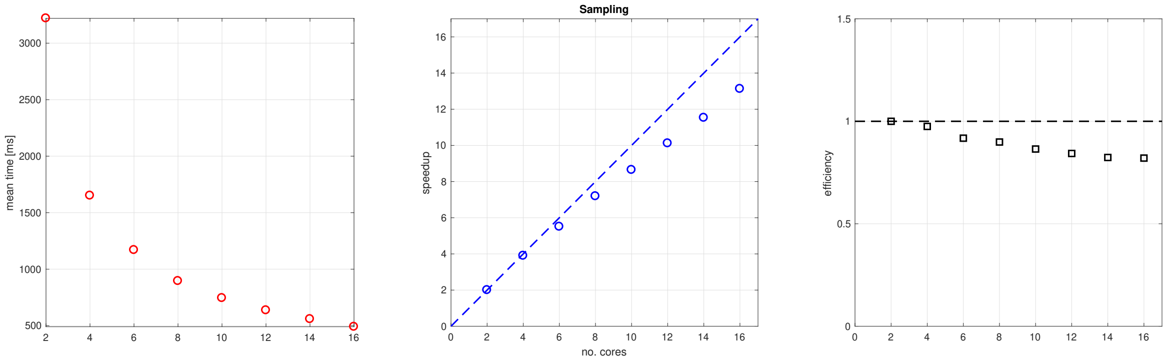


Figure 5.13: CPU time, speed-up and efficiency of the sampling kernel in case of a homogeneous dense gas.

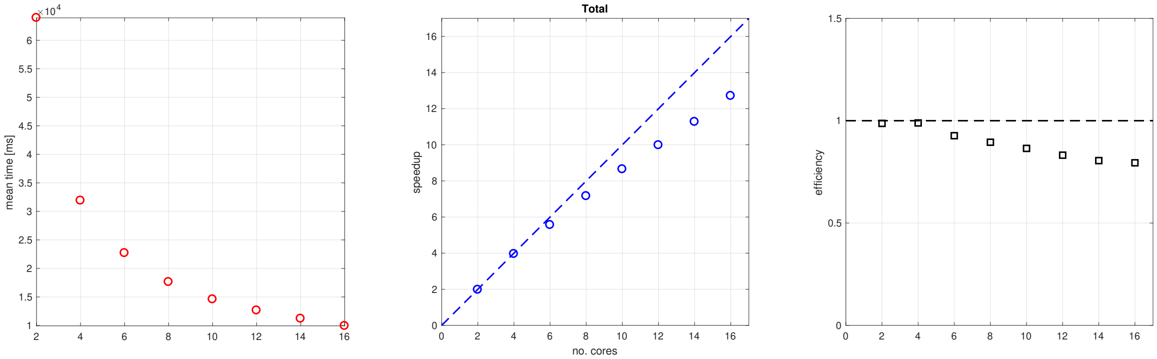


Figure 5.14: CPU time, speed-up and efficiency for the overall procedure in case of a homogeneous dense gas.

# Chapter 6

# Conclusions

Results for the tests conducted on physical consistency and parallel scaling are not completely satisfactory, both in terms of quality and completeness. The inability to reproduce the Knudsen layer entails the presence of a bias that may be explained solely by one or more implementation of numerical routines in the code being faulty. In order to be sure this is indeed the case, simulations may be repeated for a longer time, so to asses whether they actually attained convergence.

Checking the code correctness would involve inspecting each and every procedure in the density, advection, forces and collisions kernels; specific tests should be devised, and results might be possibly confronted with the ones outputted by the Fortran code.

Serial tests on dense fluid configurations should also be performed, so to study the equation of state and the CPU time employed by each kernel; parallel tests on dense configurations are impractical at the moment because of a bug regarding the evaluation of the pair correlation function: more debugging should be employed to solve this issue as soon as possible.

Parallel scaling for homogeneous configurations appears satisfactory. However, a thorough assessment in case of unfavourable configurations has to be performed. Such cases may concern:

- *high-speed collisionless flows*, where the advection procedure might be stressed, as the amount of data needed to be exchanged at each time-steps would be on the same order of magnitude of the total number of particles;

- *high-density configurations*, where the collisional procedure might be stressed, because of both the tapering of the grid and the increase in the number of collisions;

- *longer cut-off distances*, which would lead to the efficiency degradation of both the force field and the density kernels, due to an increase in the boundary data;

- *inhomogeneous configurations*, where the absence of dynamic load balancing and proper asynchronous communication would lead to an increase of idle time, possibly jeopardising the overall speed-up.

Lastly, the cost of parallel synchronous communication, in terms of CPU time, should be quantified, in particular in those situations where point-to-point communication may produce considerable idle times (in the collision kernel, for example). One possible estimate may be the difference between the maximum CPU time, over all processes, and the mean time.

All the issues mentioned above have to be addressed before properly assessing the quality of the current implementation of the parallel DSMC procedure. Once all these problems have been successfully tackled, it would be opportune to focus on enhancements. We report some possible suggestions.

## 6.1 Modelling enhancements

The hard-sphere model has proven its effectiveness in the computation of correlation functions and transport coefficients for simple fluids; its primary strength lies in the simplicity of collision mechanics calculations, due to isotropic scattering in the centre of mass frame reference. However, this model is unrealistic, as the cross-section results independent of the relative translational energy; this flaw leads to an inexact prediction of the viscosity dependence on temperature [1]. In order to overcome this issue, Bird (1981) introduced the *variable hard sphere* (VHS) molecular model; the effective cross section becomes a function of the relative velocity. This methodology, devised in the context of rarefied flows, may yield more accurate estimates of viscosity for relatively dense gas flows; for liquid phases the cross-section greatly influences the values of pressure: it should not be assumed a better estimate would be provided.

Another approach involving variable cross-section was proposed by Weeks, Chandler and Andersen, in order to study the scructure of dense fluids; the hard-sphere varies in function of the local density and temperature [59]. Both approaches present computational issues; first of all, the effective diameter is required *before* selecting the collision partner; secondly, the values of the potential kernel are now a function of the cross-section $\sigma[n(\boldsymbol{r})]$, other than just the inter-particles distance.

Another quite natural modelling enhancement would regard the presence of multiple species of particles. The resulting system of EV equations is reported in the appendix; while the generalization of the force field and the collision integral is straightforward, the pair correlation function presents some subtleties [21].

## 6.2 Numerical enhancements

The current implementation of the code does not support complex geometries. A possible extension may be aimed to tackle the problem of defining a grid compliant with complex surfaces. Unstructured grids result very flexible in this context; they are however not particularly advisable for DSMC, as they made particles' search rather cumbersome with respect to regular grids; moreover, the definition of boundary between sub-domains becomes more involved, as well as data communication. In order to combine the flexibility of unstructured grids with the simplicity of data access and manipulation of structured ones a *overlapping grids* or *embedded grid* approaches may be pursued; nevertheless, this route brings about the issue of sharing data, and possibly particles, between cells which may not be compliant [52] [63].

Inhomogeneous configurations are characterized by multiple mean free path values across the domain; therefore, it is reasonable to think about devising a mesh adaptation procedure. Unstructured grids are flexible also in this context, but they carry the aforementioned problems. A possible alternative would be to retain a regular grid and employ *octree* refinement; the resulting grid would be, however, non-conforming [51] [52].

Lastly, a possible enhancement of the numerical procedure would regard finding better estimate for the real collisions. Results show the number of fake collisions is at least one order of magnitude larger than the number of real ones, partly jeopardizing the efficiency of null-collision technique.

## 6.3 Code enhancements

One key feature the current version of code does not include is an automatic procedure for load balancing, i.e. the distribution of data and work to minimize processors idle time. Considering the currently supported grid (rectangular, Cartesian), a *orthogonal recursive coordinate bisection* method would be a simple, yet natural, choice [60]. There are however two subtleties regarding the implementation: (1) the work load of each process is determined by both the number of particles and the number of cells in the sub-domain, (2) RCB generates a non-conforming grid of sub-domains; the former can be tackled by defining the loss function associated to each process as a convex combination of the number of particles and the number of cells:

$$L[k] = \alpha N_{cells}[k] + (1-\alpha) N_p[k] \ , \quad k = 0, ..., N_{cores} \ , \quad \alpha \in [0,1] \ . \tag{6.1}$$

The latter issue is a bit more problematic: the code does not support non-conforming subdomains; forcing conformity would lead, on one hand, to a severe loss in terms of balancing optimality, while, on the other, allowing it would require a different implementation of the matrix classes. It is finally important to remark that applications to unsteady flows would require *dynamic* load balancing; a suitable disequilibrium index has to be defined, and a re-partitioning procedure has to be implemented [68].

Even if long-range interactions decay quickly, allowing a short cut-off distance, the force-field computation still results quite time demanding (especially when the meshwidth has to shrink, for reasons of physical consistency or for sampling). The procedure may be rendered more efficient by a proper re-arrangement of the degrees of freedom in the data structures used to perform convolution. Other possible solutions to increase efficiency can be borrowed by methods employed in molecular dynamics, such as *particle mesh methods*: instead of performing a direct convolution between the density field and the potential kernel, the operation may be performed in Fourier space, by employing a 2D FFT [61].

Lastly, another aspect limiting the efficiency of the current code is the lack of *asynchronous* message-passing [51], [52], [53]: each time communication between a pair of processes is involved data are exchanged one couple at the time, utilizing synchronous message passing and barriers. The parallel topology configuration may be exploited to reduce the total idle time.

## 6.4 New test cases

The study of fluid-wall interaction represents one of the fundamental goals of the Enskog-Vlasov kinetic formulation and the modified DSMC method; extensions of the model and the numerical procedure can be easily inferred from the discussions in chapters 2 and 3. Additions to the present code would concern the computation of mean-field interactions between fluid and walls particles, as well as the formulation of advection and collision procedures regarding the particles in the proximity of solid wall boundaries; parallelization should be straightforward, since the mean-field computation and the collisional stage would regard only local data. New test scenarios might involve the study of the three-phase contact angle or the formation of liquid menisci [17].

Finally, weak scaling should be addressed: since it is possible to increase the number of particles and taper the grid to obtain more accurate results, it is natural to envision an increment in the dimension of the numerical problem as the number of processes grows. Moreover, theoretical results assessing weak scaling are encouraging: *Gustafson's law* states that scaled speedup (i.e. the speedup gained with respect to the problem dimension) increases linearly with the number of processors [58]; in other words: enlarging the problem entails better parallel scaling.

# Appendix A

# Derivation of EV equation from general theory of dense gas

In this appendix the procedure for deriving EV proposed by Karkheck and Stell [29] is presented, focusing on the main concepts (while details in the computations will be omitted). Let us consider an ensemble of $N$ particles characterized by coordinates $\boldsymbol{x}_i = (\boldsymbol{r}_i, \boldsymbol{v}_i)$ in a phase-space of volume $V$; given a pair interaction function $\varphi$, the Hamiltonian for a set of $s$ particles reads:

$$H_s = \sum_{i=1}^{s} \frac{||\boldsymbol{v}_i||^2}{2m} + \sum_{i<j}^{s} \varphi_{ij} \ . \tag{A.1}$$

The $s$-particles distribution function $F_s$ satisfies the BBGKY-hierarchy of equations:

$$\frac{\partial F_s}{\partial t} + \big\{H_s, F_s\big\} = n \int_V \left[ \sum_{i=1}^{s} \frac{\partial \varphi_{i(s+1)}}{\partial \boldsymbol{r}_i} \cdot \frac{\partial F_{s+1}}{\partial \boldsymbol{v}_i} \right] dx_{s+1} \ , \tag{A.2}$$

being $\{\cdot, \cdot\}$ the Poisson's brackets and $n = N/V$; $F_s$ is normalized so that: $\int_V F_s d^s\boldsymbol{x} = |V|^s$. As it can be easily noticed, the hierarchy of equations is not closed, since the equation for the $s$-particles distribution depends on the $(s+1)$-particles' one. The goal of the derivation is to obtain a closed equation for $F_1$ in case of a dense gas where the intermolecular interactions consist in a hard-core repulsion plus smooth attractive tail.

**Closure principle**

Closure can be obtained by performing a constraint maximization of the statistical entropy:

$$S = -k_b \int_V W_N \ln\big(W_n\big) d^N\boldsymbol{x} \ . \tag{A.3}$$

$W_N$ is a probability density on the phase space $V$ (i.e. $\int_V W_N d^N\boldsymbol{x} = 1$); all known information regarding the *generic* $s$-particles distribution can be obtained from $W_N$:

$$f_s(\boldsymbol{x}^s|t) = n^s F_s(\boldsymbol{x}^s|t) = \frac{N!}{(N-s)!} \int_V W_N(\boldsymbol{x}^N|t) d^{N-s}\boldsymbol{x} \ . \tag{A.4}$$

The maximization problem can be expressed using the Lagrange multipliers formalism, that is multiplying the constraints expressed above by coefficients $\big\{\gamma, \lambda(\boldsymbol{x}^s|t)\big\}$ and maximizing the following functional:

$$I\big[W_N\big] = S\big[W_N\big] + k_b\gamma\left[1 - \int_V W_N d^N\boldsymbol{x}\right] + k_b\int_V \lambda f_s d^s\boldsymbol{x} - k_b\frac{N!}{(N-s)!}\int_V \lambda W_N d^N\boldsymbol{x} \ . \tag{A.5}$$

Imposing $\partial I/\partial\gamma = 0$ yields the normalization condition for $W_N$, while $\delta I/\delta\lambda = 0$ yields the consistency condition between $W_N$ and $f_s$; imposing the functional variation with respect to $W_N$ to be zero and using the symmetry properties of Lagrange multiplier $\lambda$ and functions $f_s$, the following is obtained:

$$\frac{\delta I}{\delta W_N} = 0 \quad \Longrightarrow \quad \ln\big(W_N(\boldsymbol{x}^N|t)\big) = -\gamma - 1 - \sideset{}{'}\sum_{i_1,\dots,i_s}^{N} \lambda(\boldsymbol{x}_{i_1}, \dots, \boldsymbol{x}_{i_s}|t) \ , \tag{A.6}$$

where the primed summation means that indices are not repeated.

**Assumptions**

One key assumption that has to be made in order to close the equations regards *time-smoothing*; let us consider the time-smoothed 1-particle distribution function:

$$\overline{F}_1(\boldsymbol{x}_1|t) = \frac{1}{\tau}\int_0^\tau F_1\big(S_\xi^{(1)}\boldsymbol{x}_1|t+\xi\big)d\xi \ , \tag{A.7}$$

with:

$$S_\tau^{(j)}[\,\cdot\,] = e^{-\tau\{H_j,\,\cdot\,\}} \ . \tag{A.8}$$

The role of time smoothing is to establish a time scale which captures a complete collision. We assume only binary collisions takes places, while higher order collisions are neglected; the range of validity for $\tau$ can be expressed as:

$$\tau_c \ll \tau \ll \tau_m \ , \tag{A.9}$$

being $\tau_c$ the mean collision time and $\tau_m$ the mean free time. Since:

$$\overline{F}_1(\boldsymbol{x}_1|t) = F_1(\boldsymbol{x}_1|t) + O\Big(\frac{\tau}{\tau_m}\Big) \ , \tag{A.10}$$

we can identify the smoothed distribution with the un-smoothed counterpart only in the limit $\tau \to 0^+$; to simplify matters the repulsive part of the potential potential is modelled by a hard-sphere repulsion: the duration of a collision in such a case is $\tau_c = 0$ so that above inequality can be maintained while taking the limit. This is a key assumption, as the leading order equation derived from the BBGKY hierarchy can be written in an advantageous form if the aforementioned identification is performed.

The entropy maximization principle and the assumption on hard-sphere collisions are the two crucial concepts leading to the Enskog-Vlasov equation; before delving into the explanation, let us define the two-particles correlation function as:

$$g_2(\boldsymbol{x}_1, \boldsymbol{x}_2|t) = \frac{F_2(\boldsymbol{x}_1, \boldsymbol{x}_2|t)}{F_1(\boldsymbol{x}_1|t)F_1(\boldsymbol{x}_2|t)} \ . \tag{A.11}$$

The starting point of the derivation is the solution of (A.2) in powers of $n$ (proposed by Lewis, 1961, [30]):

$$F_s(\boldsymbol{x}^s|t+\tau) = \sum_{k=0}^{\infty} n^k \int_V \left[ \sum_{j=0}^{k} \frac{(-1)^{k-j}}{j!(k-j)!} T_{-\tau}^{j+s} F_{s+k}(\boldsymbol{x}^{s+k}|t) \right] d^{k-1}\boldsymbol{x} \ , \tag{A.12}$$

being $T$ the multi-particle streaming operator:

$$T_{-\tau}^{(j+s)} F_{k+s} = F_{k+s}\big(S_{-\tau}^{(j+1)}\boldsymbol{x}^{j+s}, S_{-\tau}^{(1)}\boldsymbol{x}_{j+s+1}, ..., S_{-\tau}^{(1)}\boldsymbol{x}_{s+k}\big). \tag{A.13}$$

The leading order of (A.12) can be written in the following way by exploiting the symmetry properties of operators $S$ and $T$, and by considering (A.10):

$$\tau \left[ \frac{\partial}{\partial t} + \boldsymbol{v}_1 \cdot \nabla \right] F_1(\boldsymbol{x}_1|t) = n \int \big[ T_{-\tau}^{(2)} T_{\tau}^{(1)} F_2(\boldsymbol{x}_1, \boldsymbol{x}_2|t) - F_2(\boldsymbol{x}_1, \boldsymbol{x}_2|t) \big] d\boldsymbol{x}^2 \ . \tag{A.14}$$

It is now convenient to factorize the p.d.f. into $W_N = E_N(\boldsymbol{r}^N)D_N(\boldsymbol{x}^N)$; Enskog-Vlasov equation is obtained from (A.14) by assuming a prescribed form for $E_N$:

$$E_N(\boldsymbol{r}^N) = \Theta(\boldsymbol{r}^N) = \begin{cases} 1 & \text{if } ||\boldsymbol{r}_i - \boldsymbol{r}_j|| > \sigma \ \forall i \neq j \\ 0 & \text{if } ||\boldsymbol{r}_i - \boldsymbol{r}_j|| < \sigma \ \exists i, j \end{cases} , \tag{A.15}$$

and by splitting the inter-particles' interaction into a non-local hard sphere collisions and an attractive tail $\varphi_{ij}^{tail}$ (which depends on relative positions only). The following considerations are therefore made:

- taking $\tau \to 0^+$ and performing Hamiltonian dynamics calculations an exact equation is obtained from (A.14) in terms of generics distributions:

$$\begin{aligned} &\left[ \frac{\partial}{\partial t} + \boldsymbol{v}_1 \cdot \nabla \right] f_1(\boldsymbol{x}_1|t) = \\ &\quad = \int \left\{ \frac{\partial \varphi_{12}^{tail}}{\partial \boldsymbol{r}_1} \cdot \frac{\partial}{\partial \boldsymbol{v}_1} \big[ g_2(\boldsymbol{x}_1, \boldsymbol{x}_2|t) f_1(\boldsymbol{x}_1|t) \big] f_1(\boldsymbol{x}_2|t) \right\} d\boldsymbol{x}_2 + \\ &\quad + \sigma^2 \int\int \Big\{ \big[ g(\boldsymbol{r}_1, \boldsymbol{v}_1, \boldsymbol{r}_2 + \hat{\boldsymbol{k}}\sigma, \boldsymbol{v}_2'|t) f_1(\boldsymbol{r}_1, \boldsymbol{v}_1|t) f_1(\boldsymbol{r}_2 + \hat{\boldsymbol{k}}\sigma, \boldsymbol{v}_2'|t) + \\ &\quad - g(\boldsymbol{r}_1, \boldsymbol{v}_1, \boldsymbol{r}_2 - \hat{\boldsymbol{k}}\sigma, \boldsymbol{v}_2|t) f_1(\boldsymbol{r}_1, \boldsymbol{v}_1|t) f_1(\boldsymbol{r}_2 - \hat{\boldsymbol{k}}\sigma, \boldsymbol{v}_2|t) \big] \big( \hat{\boldsymbol{k}} \cdot \boldsymbol{v}_r \big)^+ d\hat{\boldsymbol{k}}^2 d\boldsymbol{v}_2 \Big\} \ ; \end{aligned} \tag{A.16}$$

- from the closure principle (A.6) the p.d.f. takes the form:

  $$W_N(\boldsymbol{x}^N) = \Theta_N(\boldsymbol{r}^N) D_N(\boldsymbol{x}^N) = e^{-\gamma-1} \Theta_N(\boldsymbol{r}^N) \prod_{i=1}^{N} e^{-\lambda(\boldsymbol{x}_i|t)} \; ; \tag{A.17}$$

  this expression hints to the correlation function having a RET expression:

  $$f_2(\boldsymbol{x}_1, \boldsymbol{x}_2|t) = g(\boldsymbol{r}_1, \boldsymbol{r}_2|n_1(t)) f_1(\boldsymbol{x}_1|t) f_1(\boldsymbol{x}_2|t) \; , \tag{A.18}$$

  which does not depend on particles' velocity.

The following kinetic-variational equation is obtained exploiting the two previous considerations:

$$\left[\frac{\partial}{\partial t} + \boldsymbol{v}_1 \cdot \nabla\right] f_1(\boldsymbol{x}_1|t) = \frac{\partial f_1}{\partial \boldsymbol{v}_1} \cdot \int \frac{\partial \varphi_{12}^{tail}}{\partial \boldsymbol{r}_1} n_1(\boldsymbol{r}_2|t) g_2\big(\boldsymbol{r}_1, \boldsymbol{r}_2|n(t)\big) d\boldsymbol{r}_2 + C_E\big(f_1, f_1\big) \; , \tag{A.19}$$

being $C_E$ Enskog's collision integral.

# Appendix B

# Generalization to a multi-species fluid

Let us now consider a *mixture* composed of spherical particles having masses $m_i$ and diameters $\sigma_i$, with $i = 1, ..., N_s$, being $N_s$ the number of components. This system, therefore, models a set with $N_s$ chemical species of monoatomic molecules, interacting through Sutherland potentials $\phi_{ij}$ and colliding according to the integrals $C_{ij}$, while no chemical reactions are taken into consideration; a system of EV equations is therefore obtained [21]:

$$\frac{\partial f_i}{\partial t} + \boldsymbol{v} \cdot \nabla_{\boldsymbol{r}} f_i + \frac{\sum_{j=1}^{N_s} \boldsymbol{F}_{ij}}{m_i} \cdot \nabla_{\boldsymbol{v}} f_i = \sum_{j=1}^{N_s} C_{ij}(f_i, f_j) \ , \quad i = 1, ..., N_s \ , \tag{B.1}$$

being:

$$\boldsymbol{F}_{ij}(\boldsymbol{r}|t) = \int_{\rho > \sigma_{ij}} \frac{d\phi_{ij}}{d\rho} \frac{\boldsymbol{r}_1 - \boldsymbol{r}}{||\boldsymbol{r}_1 - \boldsymbol{r}||} n_j(\boldsymbol{r}_1|t) d\boldsymbol{r}_1 \ , \quad \sigma_{ij} = \frac{\sigma_i + \sigma_j}{2} \ , \tag{B.2}$$

and:

$$\begin{aligned} C_{ij}(f_i, f_j) = \sigma_{ij}^2 \int_{\mathbb{R}^3} \int_{\mathcal{S}_+} \Big\{ & \chi_{ij}[n_i, n_j]\big(\boldsymbol{r}, \boldsymbol{r} + \sigma_{ij}\hat{\boldsymbol{k}}\big) f_j\big(\boldsymbol{r}, \boldsymbol{v}'|t\big) f_i\big(\boldsymbol{r} + \sigma_{ij}\hat{\boldsymbol{k}}, \boldsymbol{v}_1'|t\big) + \\ & - \chi_{ij}[n_i, n_j]\big(\boldsymbol{r}, \boldsymbol{r} - \sigma_{ij}\hat{\boldsymbol{k}}\big) f_j\big(\boldsymbol{r}, \boldsymbol{v}|t\big) f_i\big(\boldsymbol{r} - \sigma_{ij}\hat{\boldsymbol{k}}, \boldsymbol{v}_1|t\big) \Big\} \big(\boldsymbol{v}_r \cdot \hat{\boldsymbol{k}}\big)^+ d\boldsymbol{v}_1 d^2\hat{\boldsymbol{k}} \quad , \end{aligned} \tag{B.3}$$

being $n_i(\boldsymbol{r}|t)$ the number density of the $i$-th specie and $\chi_{ij}$ the short-range correlation between particles of the species $i$ and $j$. It can be noticed that the correlation has been expressed in the most general functional form; indeed, while the extension of the mean-field interactions to a multi-species system is straightforward, the collision integral deserves more thought.

If it is assumed that atoms of different species have the same diameter $\sigma_i = \sigma$ the pair correlation functions of different species can be taken equal to the one valid for the single-component case and subscripts can be omitted; here the assumption of section 2.3

regarding the form of $\chi$ can be employed again:

$$\chi_{ij}[n_i, n_j](\boldsymbol{r}, \boldsymbol{r} \pm \sigma_{ij} \hat{\boldsymbol{k}}) = \chi[n](\boldsymbol{r}, \boldsymbol{r} \pm \sigma \hat{\boldsymbol{k}}) = \chi\Big[n\Big(\boldsymbol{r} \pm \frac{\sigma}{2}\hat{\boldsymbol{k}}\Big)\Big] \; , \quad n(\boldsymbol{r}|t) = \sum_{i=1}^{N} n_i(\boldsymbol{r}|t) \; . \tag{B.4}$$

In case molecules of different species have different diameters $\sigma_i$ the correlation function has to be modified accordingly; a possible route is suggested by Santos, Yuste and de Haro [25], [26]. The extension of the equation of state (2.33) in case of a $N_s$-component hard-sphere mixture is given by:

$$\frac{p}{nk_bT} = 1 + \frac{2}{3}\pi n \sum_{i=1}^{N_s}\sum_{j=1}^{N_s} x_i x_j \sigma_{ij}^3 \chi_{ij} \; , \tag{B.5}$$

being $x_i = n_i/n$, the molar fraction of species $i$. In the same spirit as for the single component system, Mansoori et al. proposed the following equation of state (BMCSL) [27]:

$$\frac{p}{nk_bT} = \frac{1}{1-\eta} + \frac{\pi}{2}\frac{n\zeta_1\zeta_2}{(1-\eta)^2} + \frac{\pi^2}{36} n^2 \zeta_2^3 \frac{3-\eta}{(1-\eta)^3} \; , \tag{B.6}$$

being $\zeta_n = \sum_i x_i \sigma_i^n$ and $\eta = \pi n \zeta_3/6$. It is no more possible to obtain the expression for $\chi_{ij}$ by just comparing (B.5) and (B.6), similarly as what observed in section 2.4. Lee et al. proposed a possible closure by considering an expression for the correlation function (GHLL) which is consistent with BMCSL equation of state [28]:

$$\chi_{ij} = \frac{1}{2\pi}\Big(\lambda + \frac{\lambda'}{2}\frac{\sigma_i\sigma_j}{\sigma_{ij}} + \frac{1}{18}\frac{(\lambda')^2}{\lambda}\frac{\sigma_i^2\sigma_j^2}{\sigma_{ij}^2}\Big) \; , \tag{B.7}$$

being $\lambda = 2\pi/(1-\eta)$ and $\lambda' = \pi^2\zeta_2/(1-\eta)^2$.

# Appendix C

# RNG quality assessment

In this appendix we would like to briefly describe a set of strategies to test whether a random number generator is able to produce a sample that can be safely described as uniform.

## Histogram and scatterplot

The idea behind these simple visual methods is the following:

**a)** given any $b \in [0, 1]$, we should verify that:

$$\lim_{n \to \infty} \frac{\#\text{ samples s.t. } U_i \in [0, b]}{n} = b \quad , \tag{C.1}$$

that is $U_i$ is a uniform sample in $[0, 1]$;

**b)** the pairs $\{(U_1, U_2), ..., (U_{n-1}, U_n)\}$ should be uniformly distributed in $[0, 1]^2$;

**c)** $\forall\ 1 \leq k \leq n$, $\{(U_i, U_i + 1, ..., U_{k+i-1})\}_k$ should be uniformly distributed in $[0, 1]^k$.

Items (a) and (b) can be easily verified graphically by plotting the histogram of $U_i$ and the 2D scatterplot of $(U_i, U_{i+1})$.

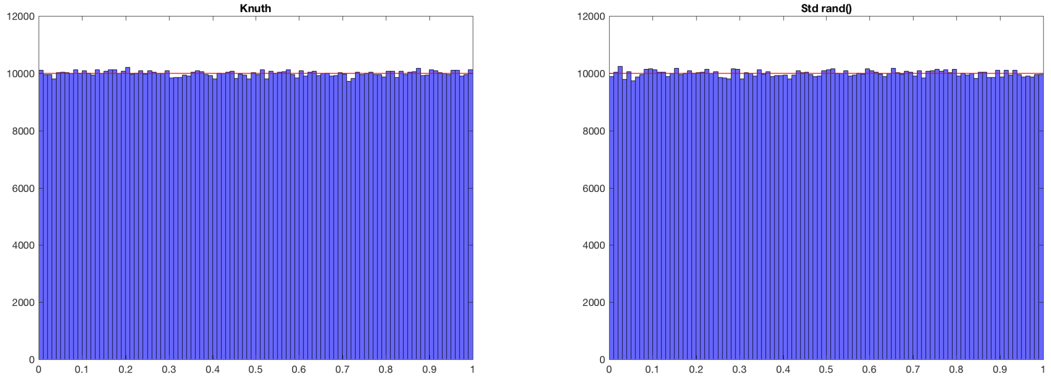


Figure C.1: Histograms representing the distribution of $10^6$ random numbers generated by the Knuth engine and by `std::rand()`.

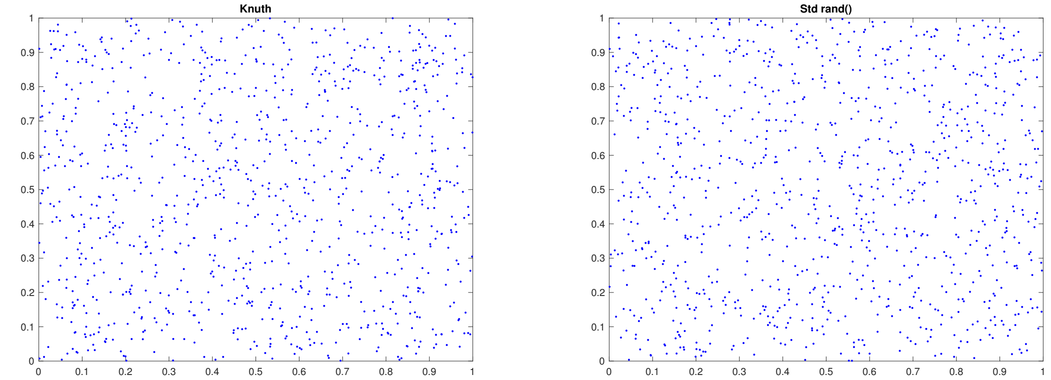


Figure C.2: Scatterplots of $10^3$ couples of random numbers generated by the Knuth engine and by `std::rand()`.

# Momenta

Another idea is to compare the $k$-th moment of the generated sequence with the $k$-th moment of a uniform distribution, that is:

$$m_k = \frac{1}{N}\sum_{i=1}^{N} U_i^k \simeq \int_0^1 x^k dx = \frac{1}{k+1} \quad ; \tag{C.2}$$

the tolerated deviation is of order $O(1/\sqrt{N})$.

## Correlations

It is possible to use the autocorrelation function of the generated sequence to asses whether the pseudo-random numbers are statistically not-independent:

$$C(k) = \mathbb{E}\big[U_i U_{i+k}\big] - \mathbb{E}\big[U_i\big]\mathbb{E}\big[U_{i+k}\big] \simeq \frac{1}{N-k}\sum_{i=1}^{N-k} U_i U_{i+k} - \frac{1}{4} \quad ; \tag{C.3}$$

also in this case, the tolerated deviation from the true correlation $C(k) = 0$ should be no larger than $O(1/\sqrt{N})$.

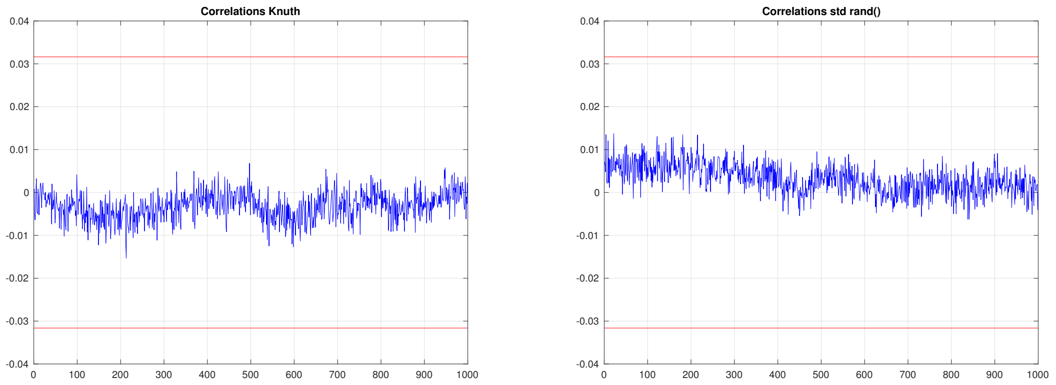


Figure C.3: Plots of $C(k)$ for $k = 0, ..., 1000$; the red lines represents the bounds $[-1/\sqrt{N}, 1/\sqrt{N}]$.

## $\chi^2$ test

This last method is the most consistent from a statistical point of view. The idea is to obtain a *discrete* uniform distribution via a *binning* procedure so that:

- $f_i = n_i/N$ is the observed relative frequency of a sample to be in the $i$-th bin;
- $p_i = P(X = i)$ is the P.D.F. of a discrete uniform in $\{1, ..., k\}$ ($k$ being the number of bins);

the following test is then considered:

$$\begin{cases} H_0 \; : \quad p_i = f_i \quad \forall\, i \\ H_1 \; : \quad p_i \neq f_i \quad \exists\, i \end{cases} \quad . \tag{C.4}$$

If $H_0$ is true, the expected number of outcomes that equal $i$ would be approximately $Np_i$ and the quantity $(n_i - Np_i)^2$ should not be too large; so the following statistic is considered:

$$V = \sum_{i=1}^{k} \frac{(n_i - Np_i)^2}{Np_i} \quad . \tag{C.5}$$

| Engine | $m_1$ | $m_2$ | $V$ |
|---|---|---|---|
| `Knuth` | 0.4998 | 0.3333 | 88.7380 |
| `Marsiglia` | 0.5004 | 0.3338 | 104.2392 |
| `ParkMiller` | 0.4998 | 0.3332 | 86.4082 |
| `Splitmix` | 0.5004 | 0.3336 | 118.5602 |
| `Xorshift64` | 0.4997 | 0.3330 | 100.3290 |
| `std::rand` | 0.4998 | 0.3331 | 111.8128 |

Table C.1: Table of first and second momenta, and $\chi^2$ statistics for the tested RNGs; the reference values are: $\chi^2_{99,0.95} = 123.2252$, $\chi^2_{99,0.99} = 134.6416$.

According to a result by Pearson (1900) for large values of $N$, when $H_0$ is true, $V$ will have an approximately *chi-squared* distribution with $k - 1$ degrees of freedom; therefore $H_0$ will be rejected ($H_1$ be accepted) in case

$$V > \chi^2_{k-1,1-\alpha} \quad , \tag{C.6}$$

being $\alpha$ typically 0.05 or 0.01.

# Final considerations

The examples reported above represent the simplest types of statistical tests; more advanced tests may be employed (i.e. Kolmogorov-Smirnov). RNG quality can be assessed also through *empirical* tests; large suites of such tests have been developed for research and for application purposes [67].

Lastly, it is important to remind that while RNG quality is paramount in stochastic simulations, efficiency has to be addressed too; usually, a trade-off between quality and efficiency occurs, as better RNG may require either more memory usage or more algorithmic steps.

| Engine | CPU time [ms] |
|---|---|
| `Knuth` | 17805 |
| `Marsiglia` | 25509 |
| `ParkMiller` | 14247 |
| `Splitmix` | 9811 |
| `Xorshift64` | 15013 |
| `std::rand` | 9625 |

Table C.2: CPU time occurred computing $10^6$ pseudo-random numbers.

# Bibliography

[1] **G. A. Bird**, *Molecular Gas Dynamics and the Direct Simulation of Gas Flows*, (Oxford University Press, Oxford Science Publications, 1994).

[2] **J. H. Ferziger, H. G. Kaper**, *Mathematical Theory of Transport Processes in Gases*, (North-Holland Publishing Company, 1972).

[3] **C. Cercignani**, *Slow Rarefied Flows: Theory and Application to Micro-Electro-Mechanical Systems*, (Birkhauser Verlag, Progress in Mathematical Physics, Volume 41, 2006).

[4] **S. Chapman, T. G. Cowling**, *The Mathematical Theory of Non-Uniform Gases*, (Cambridge University Press, 1990, III edition).

[5] **R. Balescu**, *Equilibrium and Nonequilibrium Statistical Mechanics*, (John Wiley & Sons, 1975).

[6] **R. LeSar**, *Introduction to Computational Materials Science: Fundamentals to Applications*, (Materials Research Society, Cambridge University Press, 2013).

[7] **W. H. Press, S. A. Teukolsky, W. T. Vetterling, B. P. Flannery**, *Numerical Recipes: third edition*, (Cambridge University Press, 2007).

[8] **P. S. Pacheco**, *An Introduction to Parallel Programming*, (Morgan Kaufmann - Elsevier, 2011).

[9] **Y. Aoyama, J. Nakano**, *RS/6000 SP - Practical MPI Programming*, (IBM, International Technical Support Organization, 1999).

[10] **G.A. Bird**, *Direct Simulation of Gas Flows at the Molecular Level*, (Communications in Applied Numerical Methods, Vol. 4, 165-172, 1988).

[11] **G.A. Bird**, *Direct Simulation and the Boltzmann Equation*, (The Physics of Fluids Vol. 13, No. 11, 1970).

[12] **G. A. Bird**, *Sophisticated DSMC*, (Notes prepared for a short course at the DSMC07 meeting Santa Fe, September 30, 2007, *gab.com.au*).

[13] **G. A. Bird**, *Forty Years of DSMC, and Now?*, (CP585 - Rarefied Gas Dynamics: 22nd International Symposium, American Institute of Physics, 2001).

[14] **M. A. Gallis, J. R. Torczynski, S. J. Plimpton, D. J. Rader, T. Koehler**, *Direct Simulation Monte Carlo: The Quest for Speed*, (AIP Conference Proceedings 1628, 27, 2014).

[15] **O. Aktas, N. R. Aluru, U. Ravaioli**, *Application of a Parallel DSMC Technique to Predict Flow Characteristics in Microfluidic Filters*, (Journal of Microelectromechanical Systems, vol. 10, no. 4, December 2001).

[16] **D. Raabe**, *Overview of the Lattice Boltzmann Method for Nano- and Micro-scale Fluid Dynamics in Materials Science and Engineering*, (Max-Planck-Institut fur Eisenforschung, Modelling Simul. Mater. Sci. Eng. 12, 2004, R13-R46).

[17] **P. Barbante, A. Frezzotti and L. Gibelli**, *A Kinetic Theory Description of Liquid Menisci at the Microscale*, (American Institute of Mathematical Sciences, Volume 8, Number 2, June 2015).

[18] **A. Frezzotti, P. Barbante and L. Gibelli**, *Direct simulation Monte Carlo applications to liquid-vapor flows*, (Physics of Fluids 31, 062103, 2019).

[19] **A. Frezzotti, L. Gibelli, S. Lorenzani**, *Mean field kinetic theory description of evaporation of a fluid into vacuum*, (Physics of Fluids 17, 012102, 2005).

[20] **A. Frezzotti**, *A Particle Scheme for the Numerical Solution of the Enskog equation*, (American Institute of Physics, Physics of Fluids 9, 1329, 1997).

[21] **A. Frezzotti, L. Gibelli, D. A. Lockerby, J. E. Sprittles**, *Mean-field kinetic theory approach to evaporation of a binary liquid into vacuum*, (Physical Review Fluids 3, 054001, 2018).

[22] **P. Barbante, A. Frezzotti, L. Gibelli, P. Legrenzi, A. Corigliano, A. Frangi**, *A Kinetic Model for Capillary Flows in MEMS* , (American Institute of Physics, AIP Conf. Proc. 1501, 713, 2012).

[23] **H. T. Davis**, *Kinetic theory of strongly inhomogeneous fluids*, (University of Minnesota Supercomputer Institute Research Report UMSI 92/78, April 1992).

[24] **J. Fischer, M. Methfessel**, *Born-Green-Yvon approach to the local densities of a fluid at interfaces*, (The American Physical Society, Physical review A, volume 22, number 6, December 1980).

[25] **S. B. Yuste, A. Santos, M. L. de Haro**, *Structure of multi-component hard-sphere mixtures* , (Journal of Chemical Physics Vol.108, No.9, 1998).

[26] **A. Santos, S. B. Yuste, M. L. de Haro**, *Equation of state of a multicomponent d-dimensional hard-sphere fluid*, (Molecular Physics, 1999, Vol.96, No.1, 1-5).

[27] **G. A. Mansoori, N. F. Carnahan, K. E. Starling, T. W. Leland**, *Equilibrium Thermodynamic Properties of Mixtures of Hard Spheres*, (The Journal of Chemical Physics, vol. 54 no. 5, 1971).

[28] **L. L. Lee, D. Levesque**, *Perturbation theory for mixtures of simple liquids*, (Molecular Physics, vol. 26, 1973).

[29] **J. Karkheck, G. Stell**, *Maximization of entropy, kinetic equations, and irreversible thermodynamics*, (The American Physical Society, vol. 25, no. 6, 1982).

[30] **R. M. Lewis**, *A Unifying Principle in Statistical Mechanics*, (Journal of Mathematical Physics, vol. 8 no. 3, July 1967).

[31] **M. Grmela**, *Kinetic Equation Approach to Phase Transitions*, (Journal of Statistical Physics, Vol. 3, No. 3, 1971).

[32] **J. G. Kirkwood, E. Monroe**, *Statistical Mechanics of Fusion*, (Journal of Chemical Physics, vol. 9, July 1941).

[33] **N. G. van Kampen**, *Condensation of a Classical Gas with Long-Range Attraction*, (Physical review, volume 135, number 2A, 20 July 1964).

[34] **H. van Beijeren, M. H. Ernst**, *The Non-linear Enskog-Boltzmann Equation* (Physics Letters, Vol. 43A, no. 4, March 1973).

[35] **H. van Beijeren, M. H. Ernst**, *The Modified Enskog Equation* (North-Holland Publishing Company, Physica Volume 68, Issue 3, September 1973).

[36] **J.K. Percus, G.J. Yevick**, *Analysis of Classical Statistical Mechanics by Means of Collective Coordinates*, (Phys. Rev. 110 1, 1958).

[37] **E. S. Benilov, M. S. Benilov**, *Energy conservation and H theorem for the Enskog-Vlasov equation*, (Physical Review E 97, 062115, 2018).

[38] **P. Resibois**, *H-Theorem for the (Modified) Nonlinear Enskog Equation*, (Journal of Statistical Physics, Vol. 19, No. 6, 1978).

[39] **N. F. Carnahan, K. E. Starling**, *Equation of state for nonattracting rigid spheres* (J. Chem. Phys. 51, 635, 1969).

[40] **K. Koura**, *Null-collision technique in the direct-simulation Monte Carlo method*, (Physics of Fluids 29, 3509, 1986).

[41] **H. J. C. Berendsen, J. P. M. Postma, W. F. van Gunsteren, A. DiNola, J. R. Haak**, *Molecular-Dynamics with Coupling to an External Bath*, (Journal of Chemical Physics. 81 (8): 36843690, 1984).

[42] **W. Wagner**, *A Convergence Proof for Bird's Direct Simulation Monte Carlo Method for the Boltzmann Equation*, (Journal of Statistical Physics, Vol. 66, Nos. 3/4, 1992).

[43] **H. Babovsky, R. Illner**, *A Convergence Proof for Nanbus Simulation Method for the Full Boltzmann Equation*, (SIAM J. Numerical Analysis, 26, 45-65, 1989).

[44] **K. Nambu**, *Direct simulation scheme derived from the Boltzmann equation. I. Multicomponent gases*, (J. Phys. Soc. of Japan 45, 2042-2049, 1980).

[45] **G. Chen, I. D. Boyd**, *Statistical Error Analysis for the Direct Simulation Monte Carlo Technique*, (Journal of Computational Physics 126, 434-448, 1996).

[46] **S. B. Pope**, *Particle Method for Turbulent Flows: Integration of Stochastic Model Equations*, (Journal of Computational Physics, vol. 117, issue 2, 1995).

[47] **F. J. Alexander, A. L. Garcia, B. J. Alder**, *Cell size dependence of transport coefficients in stochastic particle algorithms*, (Physics of Fluids, vol. 10, no. 6, 1998).

[48] **A. L. Garcia, W. Wagner**, *Time step truncation error in direct simulation Monte Carlo*, (Physics of Fluids, Vol. 12 No. 10, October 2000).

[49] **G. A. Bird, M. A. Gallis, J. R. Torczinsky, D. J. Rader**, *Accuracy and Efficiency of the Direct Simulation Monte Carlo Algorithm for Simulating Noncontinuum Gas Flows*, (American Institute of Physics, Physics of Fluids 21, 0171103, 2009).

[50] **T. E. Schwartzentruber, L. C. Scalabrin, I. D. Boyd**, *A modular particle-continuum numerical method for hypersonic non-equilibrium gas flows*, (Journal of Computational Physics 225, 1159-1174, 2007).

[51] **G. J. LeBeau**, *A parallel implementation of the direct simulation Monte Carlo method*, (Computer methods in applied mechanics and engineering, vol. 174, 1999).

[52] **S. Dietrich, I. D. Boyd**, *Scalar and Parallel Optimized Implementation of the Direct Simulation Monte Carlo Method*, (Journal of Computational Physics, vol. 126, 1996).

[53] **M. Ivanov, G. Markelov, S. Taylor, J. Watts**, *Parallel DSMC strategies for 3D computations*, (Parallel Computational Fluid Dynamics 1996, 485-492, 1997).

[54] **M. Rieffel, S. Taylor**, *The parallel scalability of the DSMC method* (Scalable Concurrent Programming Laboratory, Electrical Engineering and Computer Science, Syracuse University, 1998)

[55] **M. J. Abraham, T. Murtola, R. Schulz, S. Pall, J. C. Smith, B. Hess, E. Lindahl**, *GROMACS: High performance molecular simulations through multi-level parallelism from laptops to supercomputers*, (Elsevier, Science Direct, SoftwareX, vol. 1-2, June 2015).

[56] **A. V. Gusarov, I. Smurov**, *Gas-dynamic boundary conditions of evaporation and condensation: Numerical analysis of the Knudsen layer*, (Physic of Fluids, vol. 14, no. 12, December 2002).

[57] **G. M. Amdahl**, *Validity of the single processor approach to achieving large scale computing capabilities*, (1967, AFIPS Conference Proceedings. 30: 483-485).

[58] **J. L. Gustafson**, *Reevaluating Amdahl's law*, (1988, Communications of the ACM. 31 5: 532-533).

[59] **J. D. Weeks, D. Chandler, H. C. Andersen**, *Role of Repulsive Forces in Determining the Equilibrium Structure of Simple Liquids*, (J. Chem. Phys. 54, 5237, 1971).

[60] **G. C. Fox**, *A review of automatic load balancing and decomposition methods for the hypercube* (M. Schultz, editor, Numerical Algorithms for Modern Parallel Computer Architectures, pages 63-76. Springer-Verlag, New York, 1988. Caltech Report C3P-385b).

[61] **S. J. Plimpton, R. Pollock, M. Stevens**, *Particle-Mesh Ewald and rRESPA for Parallel Molecular Dynamics Simulations*, (in Proc of the Eighth SIAM Conference on Parallel Processing for Scientific Computing, Minneapolis, MN, March 1997).

[62] **W. D. Henshaw**, *Overture: An object-oriented framework for overlapping grid applications*, (Research Report UCRL-JC-147889, Lawrence Livermore National Laboratory, 2002).

[63] **K. C. Kannenberg, I. D. Boyd**, *Strategies for Efficient Particle Resolution in the Direct Simulation Monte Carlo Method*, (Journal of Computational Physics 157, 727-745, 2000).

[64] **A. Frezzotti, L. Gibelli**, *A Kinetic Model for Fluid-Wall Interaction*, (Proceedings of the Institution of Mechanical Engineers, part C, Journal of Mechanical Engineering Science, vol. 222, 2008).

[65] **P. Pickl**, *Derivation of the Vlasov equation*, (Mathematisches Institut LMU Munchen, 13 June 2016, *http://www.bcamath.org*).

[66] **M. A. Gallis, T. P. Koehler, S. J. Plimpton**, *SPARTA - Stochastic Particle Real Time Analyzer Validation and Verification Test Suite*, (Sandia National Laboratories, 2014, *sandia.gov*, *https://github.com/sparta/sparta.git*).

[67] **A. Rukhin et al.**, *A Statistical Test Suite for Random and Pseudorandom Number Generators for Cryptographic Applications*, (National Institute of Standards and Technology Special Publication, 800-22rev1a, April 2010).

[68] **F. Bariselli**, *Sviluppo di un software DSMC parallelo per la simulazione di flussi gassosi all'interno di pompe turbomolecolari*, (M.Sc. thesis, Industrial and Information Engineering: Aeronautical Engineering, academic year 2012-2013, Politecnico di Milano).