Multilevel Monte Carlo for kinetic equation models

Thomas T. Sørensen

Supervisor: Kristian Debrabant, Giovanni Samaey

1. juni 2021

Resumé

Vedvarende energikilder er noget af det mest eftspurgte blandt verdens rigeste nationer og en af de kilder, som der næres stort håb til, er fusionskraft. At skabe energi ved hjælp af fusionskraft er ekstremt kompliceret og kræver store investeringer i forbindelse med fremstilling af prototyper og opstilling af sikkerhedsforanstaltninger. Det er derfor en helt essentiel del af processen at udarbejde matematiske modeller, som kan simulere forholdene i fusionsreaktorens plasma under forskellige omstændigheder og dermed teste forskellige scenarier inden der bruges penge på en dyr produktion. I denne opgave arbejdes der med to såkaldte Multelevel Monte Carlo metoder, som bygger ovenpå to metoder, der skal hjælpe med at simulere neutrale partikler i bestemte dele af plasmaen. Udfordringen er, at mængden af kollisioner er stor og variere meget, hvilket gør simuleringen uoverkommelig for de normale metoder. De metoder, som de to Multilevel Monte Carlo metoder beror på, forsøger at afhjælpe problemet ved på den ene side at have en køretid, som er uafhængig af mængden af kollisioner og samtidig giver en god løsning både når der er mange kollisioner og når der er få kollisioner. Den ene metode, den såkaldt Kinetic-Diffusion metode, er en hybrid mellem en standard metode og en metode, der fungerer som en god approximation, når antallet af kollisioner er højt. Her er den primære udfordring at opnå den rette sammenkobling af metoderne, hvilket opnås ved at bevise hvordan de forskellige stokastiske variabler fordeler sig med udgangspunkt i antagelsen om at antallet af kollisioner er poissonfordelt . Den anden metode omformulerer den oprindelige matematiske model og udleder en lineær approximation af denne. Den linære approximation simuleres så ved en opdeling, hvor det såkaldte transportskridt simuleres først og herefter simuleres kollisionsskridtet. Den lineære approximation, som præsenteres i denne opgave, afviger fra den der bruges som udganspunkt for den ene af de to Multilevel metoder. Forskellen på de to metoder er, at den ene metode bruger information fra individuelle bevægelseshastigheder, mens den, som er udgangspunktet i artiklen, hvor Multilevel Monte Carlo metoden bliver præsenteret, kun bruger den gennemsnitlige information. De bliver begge to brugt i sammenligningen med hybrid-metoden og det viser sig, at den, som bruger individuelle bevægelseshastigheder afviger fra den ønskede og forventet opførsel. Det er især tydeligt for store skridtstørrelser. For små skridtstørrelser ses det at den asymptotiske opførsel af omkostningerne for Kinetic-Diffusion metoden er bedst. Selvom den kræver flere beregninger og dermed er mere omkostningsfuld ved simulering af en enkelt partikel, så gør den asymptotiske køretid, at den er overlegen når præcisionkravet bliver tilpas strengt.

Contents

1 Introduction

Kinetic equations occur in the context of many modelling problems. In particular, they play a key role in plasma physics, [6] and thus in the development of nuclear fusion reactors. At the moment, the worlds largest fusion experiment called ITER, which is an acronym for "International Thermonuclear Experimental Reactor", is in the making and scheduled to power on for the first time in December 2025, [22]. It is a collaboration between 35 countries, which means that there is great interest in understanding and modelling the plasma in the reactor. Kinetic simulations are used to model parts of the plasma for which fluid models are not accurate enough, [6]. In particular, the fluid models fall short in the plasma edge where the temperature of the plasma is significantly lower than in the core and the neutral particles play a much greater role. The main challenge when trying to model the neutral particles kinetically, which essentially corresponds to simulating each particle collision, is how to deal with multiple scales that may differ by several orders of magnitude. In the case of the plasma edge the rate of collisions may vary greatly across the domain and in areas with a low collision rate tracing each particle collision works well, but as the collision rate increases, a kinetic treatment of the particle simulation becomes infeasible. Numerically, the kinetic model can be dealt with via both discrete methods or Monte Carlo methods, but to the authors knowledge most work has been done on developing asymptotic-preserving methods using discrete methods. Some examples are [29, 8, 27, 11, 30]. The motivation for using Monte Carlo methods, is that computational costs are reduced and they are more flexible, [40]. However, the cost advantage of Monte Carlo methods is only true if they are able to deal with the problem of having multiple scales. Historically two main approaches have been taken. The first, is the so-called domain decomposition methods that aim at dividing the domain into a fluid domain and a kinetic domain, [40]. Some examples of such methods are [4, 9]. The difficulty with these methods is how to find a proper partitioning of the domain, which could be avoided by designing schemes that work equally well across domain. This is exactly the goal of the second approach and such methods are referred to as asymptotic-preserving methods. Asymptotic-preserving methods were first introduced in [28] as a way of simulation neutron transport in diffusive regime and then more rigorously explained in [18]. A way to speed up Monte Carlo simulations is by designing Multilevel Monte Carlo schemes that reduce the cost by spreading out the simulation across multiple levels. In the case of asymptotic-preserving methods, which are naturally biased, Multilevel Monte Carlo methods are tool to reduce the bias in an efficient way. In this work, the goal is to elaborate on and compare two asymptotic-preserving Monte Carlo methods [11, 36] and their multi level

extensions [32, 38].

1.1 Organisation of thesis

The thesis is organised s.t. Section 2, works as a motivating section, which a basic introduction to fusion energy as a relevant field of application and from that kinetic model equation that is the basis of the thesis is presented. The section is concluded with the presentation of two standard methods and their limitations, which then motivates the presentation of two quite recent methods that overcome those limitations. These methods are then introduced in Section 3. In Section 4, two extensions of the previous methods in the form of Multilevel Monte Carlo methods are presented. A summary of all the methods and some thoughts on alternative approaches are presented in Section 5. The methods are then compared based on a numerical example in Section 6.2.2.

2 Modelling plasma in fusion reactor

This section serves as a motivation for the two main methods. One of the applications in which these methods are applicable is in the modelling of plasma in fusion reactor and therefore a brief overview of fusion energy, plasmas and the Tokamak reactor from the ITER (International Thermonuclear Experimental Reactor) project, [22], are presented in Sections 2.1, 2.2, and 2.3. Then in Section 2.4, the kinetic modelling of the plasma in which the two methods are relevant is presented together with two standard methods for simulating it. The two standard methods suffer from severer computational costs in some parts of the domain, which motivates the design of new methods to overcome this problem.

2.1 Fusion

It is probably the most famous equation in the world and it is also the equation that is at the core of fusion theory, namely,

$$E = mc^2, \tag{1}$$

where E is the energy of a particle in its rest frame, m is the mass of the particle and c is the speed of light. The equation was presented by Albert Einsetin in [12]. Essentially, (1) tells us that for any change in mass there must be a corresponding change in energy and that change is very large relative to the change in mass due to c. To see why this is so important in fusion, it is important to understand the concept of the binding energy of an atom. The binding energy of an atom is the energy related to its mass defect through (1) and the mass defect is the difference

between the mass of the atom and the mass of each of its components, [47]. The binding energy of all atoms is given in Figure 1.

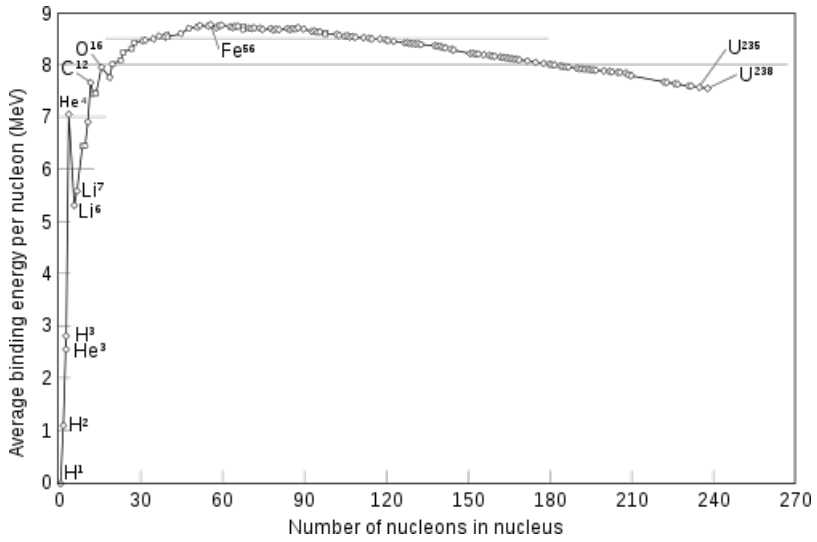


Figure 1: Average binding energy per nucleon, [46].

As the figure hints, the combined binding energy of atoms like deuterium(H^2) and tritium(H^3) is less than that of helium, which means that helium has a lower mass than deuterium and tritium combined. By (1), that decrease in mass must equate to a release in energy. To be specific, the fusion reaction looks as follows [6],

$$D + T \rightarrow He^4 + n + 17.6\text{MeV}, \tag{2}$$

where D is deuterium, T is tritium and n is a neutron. This process is what is attempted to replicate in a fusion reactor to produce energy. To replicate the process in (2), it is necessary to create extreme conditions. To put it into perspective, these sort of fusion reactions require conditions similar to the sun. In practice, the fusion happens when the ions of deuterium and tritium approach each other with high enough velocity for them to get so close that the nuclear force dominates the coulomb force. On earth, we do not experience much fusion at all because fusion takes place in plasma for which the atmosphere on earth creates poor conditions. Therefore, it is important to have a general understanding of plasma, which is the subject of the next section.

2.2 Plasma

Plasma is known as the fourth state of matter. For the purpose of this work a vague definition of plasma suffices to get an understanding of the concepts important to this work. For a more in-depth definition see [6]. When thinking of the three forms of matter excluding the plasma,

one basically thinks of a solid as something where the atoms are "pushed together" really close and then in a liquid they are more separate and in a gas even more so as the temperature increases. Just think of ice, water and steam. The plasma occurs when the temperature of a gas becomes extremely high. In fact, it has to be as high as 200 million degrees celsius for the gas to completely ionize, meaning that electrons separate from the nucleus and the gas becomes a medium of ions and electrons. At this stage, the probability of deuterium and tritium ions to diffuse becomes significant in terms of producing fusion energy. Since a gas is neutral and a plasma in principle is a heated gas, the plasma is also neutral. However, the plasma contains ions and electrons that are both charged and this means that the plasma may be influenced by magnetic fields. As a matter of fact the method of confinement in the ITER project is magnetic confinement.

2.3 The Tokamak reactor

Due to the high temperatures of the plasma, one of the main concerns when designing a fusion reactor is to ensure that the plasma is properly contained and does not come in close contact with the materials. One way to design a fusion reactor is to use the Tokamak design, which in this work is described in terms of the design in the ITER project, see [22]. For illustrative purposes, the ITER Tokamak reactor is shown below:

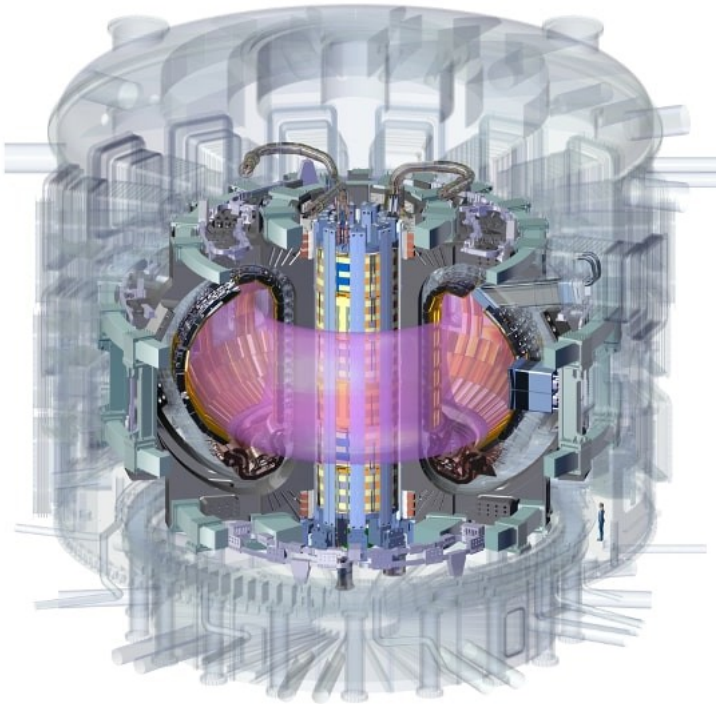


Figure 2: An illustration of the ITER Tokamak reactor, [22].

In the following, three main components of the reactor is described to get an understanding of how the plasma is confined, how energy is produced, and how ash and heat is extracted. A charged particle moving in a magnetic field is bound to the magnetic field lines in the sense that in a vacuum, a particle with a velocity that is not perpendicular to the magnetic field moves with a helical motion in the direction of the magnetic field, [39]. This is a nice feature, but measures need to be taken in order to avoid too much turbulence and drift, see [39, Figure 1]. In the case of the Tokamak reactor, the plasma is confined by combining two magnetic fields, a poloidal field and a toroidal field. Combined, these magnetic fields give a twisted magnetic field that helps keep the plasma confined and avoid plasma drift. This is illustrated in Figure 3.

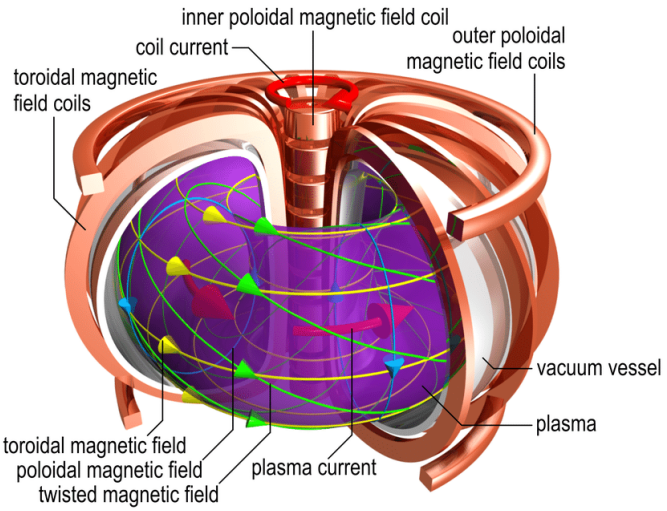


Figure 3: A schematic tokamak, figure courtesy of C. Brandt, [5]

Energy in the fusion reactor is produced in the walls inside the torus in Figure 2. The energy is produced by the walls absorbing the kinetic energy of the neutron from (2) when it hits the wall. The heat from the kinetic energy is collected by a water coolant and from that energy is produced in the power plant, [22]. As more fusion takes place waste products will build up and will need to be removed because the laws of physics limit the maximum plasma density that can be contained. Combined with the fact that the number of reactions is proportional to the density of the fuel, [20], it is clear that controlling the impurity of the plasma is essential. To minimize plasma contamination, so called divertors are placed in the bottom of the chamber to let some of the plasma leak through. This helps keep the plasma less contaminated and protects some of the other materials from extreme thermal loads. As a consequence, the divertor is the part of the reactor that must be able to sustain the greatest heat flux and as such it is crucial that the divertors function efficiently to protect other parts of the reactor from such heat fluxes.

As one can imagine it is quite expensive to build prototypes of the different components and it is therefore of great interest to simulate the plasma properly to be able to test new designs through simulation. One of the main issues is to ensure proper behaviour of the plasma at the plasma edge. One of the main roles of the plasma edge is to serve as a cushion or a barrier for the plasma, see [19]. If the temperature at the plasma edge is too high, the plasma-facing materials might be affected and start to emit dust particles that in turn will cool down the plasma and affect the fusion reactions negatively. Additionally, the so-called "ash" is transported to the plasma edge, [26], and the removal of the ash is crucial to the functioning of the reactor. In the setting of (2), the helium is the ash. Therefore, future prospects of fusion as an energy source is completely reliant upon the understanding of the plasma edge. Modelling the plasma edge is very complex and the modelling is usually split up into several models to cope with the large disparity of the time and length scales. For instance, collisions occur on sub-microsecond time scales and the evolution of the state of the materials surrounding the plasma can take up to 1000 seconds until saturation is achieved, [26]. A common practice is to split the modelling up into a fluid model and a kinetic model where the fluid model describes the plasma without the neutral particles and the kinetic model then describes the neutral particle transport. An example of this approach is the B2-EIRENE package, see [42]. The models in this work aim to solve the kinetic part for a single species, Deuterium, by using the Boltzmann-BGK equation as the model.

2.4 The kinetic model

In this section, the Boltzmann-BGK equation is presented together with its diffusive scaling. In addition, two standard Monte Carlo methods for simulating the Boltzmann-BGK equation together with their limitations in the diffusive limit are presented. The limitations of these methods, serve as motivation for the main methods in this work. The Boltzmann-BGK equation is presented in Section 2.4.1 and the two Monte Carlo methods are presented in Section 2.4.2.

2.4.1 Boltzmann-BGK model

In the most general form, the dynamics of the particles in the plasma are modelled by the Boltzmann equations, [6],

$$\frac{\partial f_\alpha}{\partial t} + v \cdot \nabla_x f_\alpha + \frac{F}{m} \nabla_v f_\alpha = \left(\frac{\partial f_\alpha}{\partial t} \right)_c, \tag{3}$$

where α is the particle species, F is the force acting on the particle and the term on the right hand side is the change in the density due to collisions. The methods in this work are based on

a model for the dominating species in fusion reactors, namely deuterium and there are several types of interactions that could be included in the model. These are collisions with other particles and interactions with the wall. In this work, the models only account for one type of interaction, namely the so-called charge-exchange collision, which are collisions where the neutral particle collides with an ion and the electron on the neutral particle is transferred to the ion. Thereby, the neutral becomes an ion and the ion becomes a neutral particle. The force term vanishes because the modelling is restricted to neutral particle transport. Additionally, it is assumed that the post-collisional distribution is Maxwellian and independent of the post-collisional velocity. The collision rate is also assumed to be independent of the post-collisional velocity. With these simplifications the model of interest becomes

$$\begin{aligned} \frac{\partial f(x,v,t)}{\partial t} + v\frac{\partial f(x,v,t)}{\partial x} &= R(x)\left(M(v;x)\int f(x,v,t)dv - f(x,v,t)\right), \\ f(x,v,0) &= S(x,v), \end{aligned} \tag{4}$$

where $S(x,\nu)$ is the initial condition, $R(x)$ is the collision rate , x is the particle position, v is the velocity and the collision term is modelled via the BGK-operator, see [3]. In the most general case considered in this thesis

$$M(v;x) = \frac{1}{\sqrt{2\pi\sigma^2(x)}} e^{-\frac{(v-\mu(x))^2}{2\sigma^2(x)}}. \tag{5}$$

In this work, the equation of interest is (4) and for the fusion case, the relevant scaling is the diffusive scaling. The diffusive scaling is given by $t \to \frac{t}{\epsilon^2}$ and $x \to \frac{x}{\epsilon}$, where ϵ is the so called diffusive parameter, and letting $f^\epsilon(x,v,t) = f(\frac{x}{\epsilon}, v, \frac{t}{\epsilon^2})$ one obtains

$$\frac{\partial f^\epsilon}{\partial t} = \frac{\partial f}{\partial t}\frac{1}{\epsilon^2}, \tag{6}$$

$$\frac{\partial f^\epsilon}{\partial x} = \frac{\partial f}{\partial x}\frac{1}{\epsilon}. \tag{7}$$

Thus, omitting ϵ in the superscript, the equation for the scaled density becomes

$$\begin{aligned} \frac{\partial f(x,\nu,t)}{\partial t} + \frac{\nu}{\epsilon}\frac{\partial f(x,\nu,t)}{\partial x} &= \frac{r(x)}{\epsilon^2}\left(M(\nu;x)\rho(x,t) - f(x,\nu,t)\right), \\ f(x,\nu,0) &= S(x,\nu), \end{aligned} \tag{8}$$

where $\rho(x,t) = \int f(x,\nu,t)d\nu$, $S(x,\nu)$ is the initial condition, $R(x) = \frac{r(x)}{\epsilon^2}$ is the collision rate at x and $\frac{\nu}{\epsilon}$ is the velocity. Note that $\mu_\nu(x) = \epsilon\mu(x)$ and $\sigma_\nu(x) = \epsilon\sigma(x)$. The diffusive scaling parameter ϵ captures the behaviour of a plasma with high velocities and very high collision rates as $\epsilon \to 0$.

In [34] it is established that the BGK collision operator is a Poisson process and hence the time

to the next collision, which will be referred to as the collision time, is found by solving

$$\int_0^{\mathcal{T}_n} R(x_n + v_n t)dt = \mathcal{E}_n, \tag{9}$$

for $\mathcal{T}_n$, where $\mathcal{E}_n \sim Exp(1)$, and x_n and v_n are the current position and velocity respectively, see [43]. The connection between the BGK-operator and the poisson process is key in the derivations of the so-called Kinetic-Diffusion scheme in Section 3.3. For convenience (10) is rewritten using the substitution $x = x_n + v_n t$ to obtain

$$\int_{x_n}^{x_n + v_n \mathcal{T}_n} R(x)dx = \mathcal{E}_n v_n. \tag{10}$$

Throughout this work, it is assumed that $R(x)$ is a piece-wise linear or constant function, i.e.

$$R(x) = \sum_{i=1}^{N} R_i(x) \mathbb{1}_{x \in (b_{i-1}, b_i]}(x) = \sum_{i=1}^{N} (\alpha_i x + \beta_i) \mathbb{1}_{x \in (b_{i-1}, b_i]}(x), \tag{11}$$

where $b = (b_0, b_1, \ldots, b_N)^T$ is the vector of boundaries and for the first interval, the left boundary is included. This assumption is in line with many practical cases, see [31]. The following is an algorithm to sample collision times from (10) with such a collision rate:

Algorithm 1: Sample collision times

Function `SampleCollision`(x,v,ε)**:**

$\tau = 0$

$d = \text{sign}(v)$ `// Direction of velocity`

$i =$ smallest index s.t. $x \leq b_i$ `// determine region`

$\varepsilon_r = |\varepsilon * v|$ `// rhs. of` (10)

while $\varepsilon > 0$ **do**

  if $d < 0$ **then**

    $x_b = b_{i-1}$ `// position at boundary in direction of` v

  else

    $x_b = b_i$ `// position at boundary in direction of` v

  $h = R(x_\text{b})$ `// height at boundary in direction of` v

  $I = |x - x_\text{b}|h + \frac{|x-x_\text{b}||R(x)-h|}{2}$ `// solution of` (10) `in` $[\min(x, x_b), \max(x, x_b)]$

  if $I < \varepsilon_r$ **then**

    $\varepsilon_r = \varepsilon_r - I$ `// update rhs. of` (10)

    $\tau = \tau + \frac{x_b - x}{v}$ `// update collision time with time to the boundary`

    $x = x_b$

    $i = i + d$ `// update index for current region`

  else

    solve for t: $\frac{\alpha_i}{2} v^2 t^2 + (\alpha_i x + \beta_i) v t - \varepsilon_r = 0$

    $\tau = \tau + t$

return τ

2.4.2 Time step restrictions for standard methods

In this section two standard Monte Carlo methods for the simulation of (8) are presented to illustrate the time step restrictions that the main methods attempt to overcome. The methods are explained in the setting of the diffusive scaling to illustrate how they do not work uniformly across all values of the diffusive parameter. These standard methods will be referred to as the standard kinetic method and the standard splitting method, respectively. Each of the two methods are connected to one of the two main methods, the standard kinetic method serves as motivation for the Kinetic-Diffusion(KD) method explained in Section 3.3 and the standard splitting method serves as motivation for the asymptotic preserving splitting(APS) method in Section 3.4.

2.4.2.1 Standard kinetic method

The standard kinetic method is in the literature referred to as the analog simulation, see [31]. The idea is to simply simulate the particle dynamics that (8) models. Hence, a particle is initialised according to $S(x, \nu)$ and moves according to its velocity until the time of the next collision, which is determined by (10). At the time of the next collision a new velocity is drawn from $M(\nu; x)$. Specifically, the particle movement is simulated by

$$X_{n+1} = X_n + \frac{\mathcal{V}_n}{\epsilon}\mathcal{T}_n, \tag{12}$$

where X_n is the position at time t_n, and $\mathcal{V}_n$ and $\mathcal{T}_n$ are respectively the random variables corresponding to the velocity and the n'th collision time. Note that $\mathcal{V}_n \sim M(\nu, x)$ and $\mathcal{T}_n$ is given by (10) for all n. As ϵ approaches zero, the collision rate increases to infinity causing the collision time to become very small. Consequently, the number of steps it takes to cover a fixed time interval is unbounded and this is the main issue that the Kinetic-Diffusion schemes attempts to remedy. The algorithm for this method is given below.

Algorithm 2: Kinetic simulation

```
// One-step function for kinetic scheme
Function ψ_k(x,v,τ):
    x = x + vτ
    return x
// Kinetic simulation in [0, T]
Function Kinetic(Δt, T):
    t = 0
    x, v ← S(x, v)                                   // from (4)
    ε ← Exp(1)
    τ = SampleCollision(x, v, ε)                     // see Algorithm 1
    while t + τ < T do
        x = ψ_k(x,v,τ)
        t = t + τ
        ε ← Exp(1)
        τ = SampleCollision(x, v, ε)
        v ← M(v; x)
    if t < T then
        x = x + v(T − t)                             // particle is moved to the end
    return x
```

2.4.2.2 Standard splitting method

This method is briefly introduced in [32]. It involves doing a first order splitting, see [21], on (8) to obtain a transport step and a collision step. The splitting is considered in an interval $[t_n, t_{n+1}]$ and look as follows:

$$\begin{cases} \partial_t f^{1/2} + \frac{\nu}{\epsilon}\partial_x f^{1/2} = 0, \quad t \in [t_n, t_{n+1}] \quad f_1(x,\nu,t_n) = f_{sp}(x,\nu,t_n) \\ \partial_t f^{2/2} = \frac{r}{\epsilon^2}\left(M\rho^{2/2} - f^{2/2}\right), \quad t \in [t_n, t_{n+1}] \quad f^{2/2}(x,\nu,t_n) = f_1(x,\nu,t_{n+1}), \end{cases} \tag{13}$$

where $f_{sp}(x,\nu,t_n) = f_2(x,\nu,t_{n-1})$ for $n \geq 1$ and $f_{sp}(x,\nu,t_0) = f(x,\nu,t_0)$ with $f(x,\nu,t_0) = S(x,\nu)$. The first equation in (13) is referred to as the transport step and the second equation is referred to as the collision step. In line with the approaches later in the thesis, the transport step is compared to the Fokker-Planck equation and the well-known simulation scheme for such

an equation, see [41]. The density function, $f(x,t)$, for the solution to the itó process,

$$dX_t = a(X_t,t)dt + \sqrt{2b(X_t,t)}dW_t, \tag{14}$$

is given by the solution of the Fokker-Planck equation

$$\frac{\partial f(x,t)}{\partial t} = -\frac{\partial (f(x,t)a(x,t))}{\partial x} + \frac{\partial^2 (b(x,t)f(x,t))}{\partial x^2}. \tag{15}$$

There is one notable difference between the transport equation, i.e. the first equation of (13) and the Fokker-Planck equation, namely that in the transport equation, there is an additional independent variable, ν. However, for a fixed $\nu^* \in \mathbb{V}$ where $\mathbb{V}$ is the velocity space, it is the case that $f^{1/2}(x,\nu^*,t) = \tilde{f}(x,t)$ where $\tilde{f}$ solves

$$\partial_t \tilde{f} + \frac{\nu^*}{\epsilon}\partial_x \tilde{f} = 0, \quad t \in [t_n, t_{n+1}] \quad \tilde{f}(x,t_n) = f_{sp}(x,\nu^*,t_n). \tag{16}$$

Thus, for each fixed velocity, the transport equation amounts to a Fokker-Planck-equation without a diffusion term and that can be translated to a Monte Carlo scheme by an Euler-Maruyama discretization of (14). Unless otherwise specified, the Euler-Maruyama method is the discretiztion of choice throughout the thesis. This argument also gives rise to the comparison with the Fokker-Planck equation in Section 3.4.2. Comparing the transport step to (15), it is clear that

$$a(x,t) = \frac{\nu}{\epsilon}, \tag{17}$$

$$b(x,t) = 0, \tag{18}$$

which means that the transport step may be simulated by

$$X_{n+1} = X_n + \frac{\mathcal{V}_n}{\epsilon}\Delta t, \tag{19}$$

where Δt is the chosen time increment in each step. Thus, in the transport step, the particle position is changed according to the velocity for at time Δt, which also motivates the term "transport step".

To deal with the collision step, first note that in this step ρ remains unchanged. To realise this, consider the interpretation of the differential equation giving the collision step. The change in the joint density over this time step is given by the difference in the two denisties $M\rho$ and f. Therefore, as time passes f approaches $M\rho$, but these two densities share the same marginal distribution in the particle position and only differ in the randomness of the velocity. Hence, it is really only the velocity distribution which changes in this step and the positional distribution remains unchanged. Therefore, the differential equation in the collision step may be solved

exactly as a first order linear differential equation in the collisional time step. Consider all quantities after the n'th transport step. By the above explanation $\rho(x,t) = \rho(x)$ in $[t_1, t_2]$ and assuming a uniform time discretization $\Delta t = t_2 - t_1$. Now the equation may be solve in $[t_n, t_{n+1}]$, using the integrating factor $u = e^{\frac{r}{\epsilon^2}(t-t_n)}$ in the following way:

$$
\begin{aligned}
&\partial_t f = \frac{r}{\epsilon^2}\left(M\rho - f\right), \\
\iff &\partial_t f + \frac{r}{\epsilon^2} f = \frac{r}{\epsilon^2} M\rho, \\
\iff &u\partial_t f + \frac{r}{\epsilon^2} uf = \frac{r}{\epsilon^2} M\rho u, \\
\iff &\partial_t (fu) = \frac{r}{\epsilon^2} M\rho u, \\
\iff &\int_{t_n}^{t_{n+1}} \partial_t (fu) dt = \frac{r}{\epsilon^2} M\rho \int_{t_n}^{t_{n+1}} u dt, \\
\iff &f(t_{n+1})u(t_{n+1}) - f(t_n)u(t_n) = \frac{r}{\epsilon^2} M\rho \left(\frac{\epsilon^2}{r} (e^{\frac{r}{\epsilon^2}(t_{n+1}-t_n)} - e^{\frac{r}{\epsilon^2}(t_n - t_n)}) \right), \\
\iff &f(t_{n+1}) e^{\frac{r}{\epsilon^2}(\Delta t)} - f(t_n) = M\rho (e^{\frac{r}{\epsilon^2}(\Delta t)} - 1), \\
\iff &f(t_{n+1}) = f(t_n) e^{-\frac{r}{\epsilon^2}(\Delta t)} + M\rho (1 - e^{-\frac{r}{\epsilon^2}(\Delta t)}).
\end{aligned} \tag{20}
$$

The density at $t = t_{n+1}$ in (20) corresponds to the density of the random vector, (X, V) s.t. with a probability of $e^{-\frac{r}{\epsilon^2}(\Delta t)}$ it is the case that $(X, V) \sim f(t_1)$ and with a probability of $(1 - e^{-\frac{r}{\epsilon^2}(\Delta t)})$ it is the case that $(X, V) \sim M\rho$. At a Monte Carlo level, the collision step then equates to either draw a new velocity from M with probability $(1 - e^{-\frac{r}{\epsilon^2}(\Delta t)})$ or keep the velocity unchanged. This also motivates the term "collision step". The problem with the approach sketched in this section is that the step size is restricted by ϵ both in the transport step and in the collision step. As ϵ approaches zero, the velocity in the transport step, $V_n = \frac{\mathcal{V}_n}{\epsilon}$, becomes infinitely large and to remedy this one must choose the time step size s.t. $\Delta t = \mathcal{O}(\epsilon)$. Due to the collision step, the overall restriction of Δt is even more severe because in order for the exponent, $-\frac{r}{\epsilon^2}(\Delta t)$ to remain finite it is necessary to choose $\Delta t = \mathcal{O}(\epsilon^2)$. In Section 3.4, a method to overcome this problem is presented. An algorithm for the standard splitting approach is given below:

Algorithm 3: Standard splitting simulation

```
// Transport step
Function ψ_t(x,v,Δt):
    x = x + vΔt
    return x
// Collision step
Function ψ_c(x,v,Δt):
    u ← U(0,1)                                          // draw uniform number
    if u > e^{-r/ε^2 Δt} then
        v ← M(v;x)                  // draw from post-collisional distribution
    return v
// Standard splitting step
Function φ_SS(x,v,Δt):
    x = ψ_t(x,v,Δt)
    v = ψ_c(x,v,Δt)
    return x
// Standard splitting simulation in [0,T]
Function SS(Δt, T):
    t = 0
    x, v ← S(x, v)                                      // from (4)
    while t < T do
        x = ψ_t(x,v,Δt)
        v = ψ_c(x,v,Δt)
        t = t + Δt
    return x, v
```

3 Asymptotic preserving Monte Carlo methods

The main difficulty when dealing with particles governed by Equation (8) is that in high collisional parts of the domain a kinetic description as given in Section 2.4 becomes computationally infeasible. To overcome this problem one may use several different strategies. One example would be methods based on domain decomposition, see [10, 2], but one of the main problems for those methods is to find a proper decomposition of the domain. As an alternative, the

two methods presented in this work, aim at being asymptotically preserving, i.e. preserving the asymptotic behaviour the limit from the microscopic to the macroscopic models in the discrete setting, [24]. The idea of an asymptotic-preserving method is illustrated in Figure 4. The asymptotic-preserving scheme with discretisation parameter Δt is represented by $f^{\epsilon}_{\Delta t}$ and in the limit $\Delta t \to 0$, $f^{\epsilon}_{\Delta t}$ approaches the original model f^{ϵ}. In the limit $\epsilon \to 0$, the original model is approximated by the macroscopic model f^0 and $f^{\epsilon}_{\Delta t}$ approaches $f^0_{\Delta t}$. Thus, the goal is to design schemes that in the limit $\Delta t \to 0$ approaches the original microscopic model and in the limit $\epsilon \to 0$ approaches the macroscopic approximation f^0 while having costs that are independent of ϵ. While no formal proofs are given in this work on the asymptotic-preserving property of the methods presented, informal arguments are provided and the properties are checked numerically in Section 6.1.

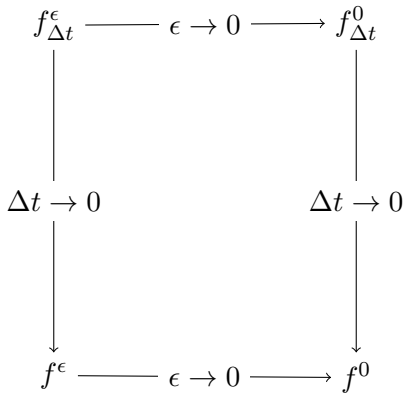


Figure 4: Illustration of how an asymptotic-preserving method should behave, [24].

In Section 3.1, some key assumptions for the solution of (8) and its partial derivatives are presented. Then in Section 3.2, the behaviour of (8) in the diffusive limit is determined to show how the methods should behave in the diffusive limit in order to be asymptotic-preserving. In Section 3.3, the so called Kinetic-Diffusion method is presented and then in Section 3.4 an asymptotic preserving version of the standard splitting method from Section 2.4.2, is presented. The structure of Sections 3.3 and 3.4 lead to the construction of a one-step function denoted by $\phi(\cdot)$ with a subscript indicating the specific method. Finally, in Section 3.5 a simple numerical example is presented to verify the implementation of the two methods.

3.1 Key assumptions

All derivations on the asymptotic preserving Monte Carlo methods are done in 1 dimension and it is assumed that

$$\int_{\mathbb{R}} f(x,v,t)^2 M(v;x)^{-1} dv < \infty, \tag{21}$$

where dv is the Lebesgue measure, $f(x,v,t)$ is the solution of (4) and $M(v;x)$ is the post-collisional velocity distribution. In other words, f belongs to the Hilbert space $L^2(\mathbb{R}, M^{-1}dv)$. Additionally, it is assumed that the partial derivatives of f w.r.t. either t or x are integrable w.r.t. the velocity. Working within the frame work of a Hilbert space provides the geometric properties needed in the next section. Let L be the collision operator known as the BGK-operator in (4), i.e.

$$Lf = M\rho - f, \tag{22}$$

where $\rho = \int_{\mathbb{V}} f dv$ with $\mathbb{V}$ being the velocity space. Then, the nullspace and range of L are given by

$$\mathcal{N}(L) = Span(M), \tag{23}$$

$$\mathcal{R}(L) = \{f :< f >= 0\}, \tag{24}$$

where $< f >= \int_{\mathbb{V}} f dv$. Note that $\mathcal{N}(L) = \mathcal{R}(L)^{\perp}$. To see this, let $f \in \mathcal{N}(L)$ and $g \in \mathcal{R}(L)$ then

$$< f, g >= \int_{\mathbb{V}} f g M^{-1} dv = \int_{\mathbb{V}} < f > M g M^{-1} dv =< f > \int_{\mathbb{V}} g dv = 0. \tag{25}$$

Therefore, the Projection Theorem, [33], gives that the solution f may be decomposed by

$$f(x,v,t) = M(v;x)\rho(x,t) + \epsilon g(x,v,t). \tag{26}$$

This decomposition is called the micro-macro decomposition and the first and second term are referred to as macroscopic and microscopic part of the model, respectively. This setup is based on [30].

3.2 Equation model in diffusive limit

An important property of an asymptotic-preserving scheme is that it approximates the macroscopic approximation of (8) well and therefore, an obvious first step is to analyse the behaviour of Equation (8) as $\epsilon \to 0$ based on the micro-macro decomposition in Section 3.1 as it is also done in [36]. The following is a more intuitive way of thinking about the decomposition. As ϵ becomes very small $R(x)$ will become very large, meaning that collisions will happen very

frequently. Consequently, more and more velocities are drawn from the post-collisional distribution and thus the distribution of the velocities will tend towards $M(\nu;x)$. Therefore, as $\epsilon \to 0$, the particle density, $f(x,\nu,t)$, will tend to $M(\nu;x)\rho(x,t)$, i.e. the first term of (26). The second term of (26) may then be thought of as an error representing the deviation of the particle distribution from the macroscopic representation.

Substituting (26) into Equation (8) gives

$$\frac{\partial(M(\nu;x)\rho(x,t)+\epsilon g(x,\nu,t))}{\partial t}+\frac{v}{\epsilon}\frac{\partial(M(\nu;x)\rho(x,t)+\epsilon g(x,\nu,t))}{\partial x}=\frac{r(x)}{\epsilon^2}\left(M(\nu;x)\rho(x,t)\right.$$
$$\left.-M(\nu;x)\rho(x,t)-\epsilon g(x,\nu,t)\right),$$

which is equivalent to

$$\frac{\partial(M(\nu;x)\rho(x,t)+\epsilon g(x,\nu,t))}{\partial t}+\frac{\nu}{\epsilon}\frac{\partial(M(\nu;x)\rho(x,t)+\epsilon g(x,\nu,t))}{\partial x}+\frac{r(x)}{\epsilon}g(x,\nu,t)=0. \quad (27)$$

To obtain the equation above as an equation in the marginal density of the particle position both sides are integrated w.r.t ν. The three terms are dealt with individually, but first it is noted that by Equation (26)

$$\int f(x,\nu,t)d\nu=\rho(x,t) \implies \int \epsilon g(x,\nu,t)d\nu=0. \quad (28)$$

Based on Equation (28) the following is derived for each term when integrating w.r.t. ν,

$$\int \frac{\partial(M(\nu;x)\rho(x,t)+\epsilon g(x,\nu,t))}{\partial t}d\nu=\frac{\partial\rho(x,t)}{\partial t}, \quad (29)$$

$$\begin{aligned}\int \frac{\nu}{\epsilon}\frac{\partial(M(\nu;x)\rho(x,t)+\epsilon g(x,\nu,t))}{\partial x}d\nu &= \frac{\partial}{\partial x}\int \frac{\nu}{\epsilon}M(\nu;x)\rho(x,t)+\nu g(x,\nu,t)d\nu \quad (30)\\ &= \frac{\partial}{\partial x}\left(\frac{\rho(x,t)}{\epsilon}E(\nu)+\int \nu g(x,\nu,t)d\nu\right)\\ &= \frac{\partial(\rho(x,t)\mu(x))}{\partial x}+\int \nu\frac{\partial g(x,\nu,t)}{\partial x}d\nu,\end{aligned}$$

$$\int \frac{r(x)}{\epsilon}g(x,\nu,t)d\nu=0. \quad (31)$$

Thus, we obtain

$$\frac{\partial\rho(x,t)}{\partial t}+\frac{\partial(\rho(x,t)\mu(x))}{\partial x}+\int \nu\frac{\partial g(x,\nu,t)}{\partial x}d\nu=0. \quad (32)$$

To calculate the last integral in (30) as $\epsilon \to 0$ note that by Equation (27), the dominant terms in the diffusive limit give that

$$\nu\frac{\partial(M(\nu;x)\rho(x,t))}{\partial x}=-r(x)g(x,\nu,t), \quad \epsilon \to 0, \quad (33)$$

which means that

$$\begin{aligned}\int \nu \frac{\partial g(x,\nu,t)}{\partial x} d\nu &= -\int \frac{\partial}{\partial x}\left(\frac{\nu^2}{r(x)}\frac{\partial(M(\nu;x)\rho(x,t))}{\partial x}\right) d\nu, \quad \epsilon \to 0 \\ &= -\frac{\partial}{\partial x}\left(\frac{1}{r(x)}\frac{\partial(\int \nu^2 M(\nu;x) d\nu \rho(x,t))}{\partial x}\right), \quad \epsilon \to 0 \\ &= -\frac{\partial}{\partial x}\left(\frac{1}{r(x)}\frac{\partial(E(\nu^2)\rho(x,t))}{\partial x}\right), \quad \epsilon \to 0 \\ &= -\frac{\partial}{\partial x}\left(\frac{1}{r(x)}\frac{\partial((V(\nu)+E(\nu)^2)\rho(x,t))}{\partial x}\right), \quad \epsilon \to 0 \\ &= -\frac{\partial}{\partial x}\left(\frac{1}{r(x)}\frac{\partial((\sigma_\nu(x)^2 + \overbrace{(\epsilon\mu(x))^2}^{\to 0})\rho(x,t))}{\partial x}\right), \quad \epsilon \to 0.\end{aligned} \tag{34}$$

As a result, we may rewrite Equation (32) as

$$\frac{\partial \rho(x,t)}{\partial t} + \frac{\partial(\rho(x,t)\mu(x))}{\partial x} - \frac{\partial}{\partial x}\left(\frac{1}{r(x)}\frac{\partial(\sigma_\nu(x)^2\rho(x,t))}{\partial x}\right) = 0, \quad \epsilon \to 0. \tag{35}$$

Thus, the methods must converge to Equation (35) in the diffusive limit and to the original equation as the step size approaches zero with the diffusive parameter being fixed and finite.

3.3 Kinetic-Diffusion Monte-Carlo scheme

The method presented in this section is one that combines the standard kinetic particle method from Section 2.4 with a simulation method for (35). In Section 3.3.1, a Monte Carlo method for simulating Equation (35) and the main challenges in coupling it with the standard kinetic particle method are described. In Section 3.3.2, the first two moments of the kinetic position increment are determined to achieve a proper coupling.

3.3.1 Simulation using the advection-diffusion equation model

The model in Equation (35) can be simulated in multiple ways, both deterministically using finite elements or stochastically via Monte Carlo methods. For this method a Monte Carlo approach is used and as in Section 2.4.2.2 a comparison with the Fokker-Planck equation in (15) is used to obtain a simulation scheme. First, it is necessary to rewrite Equation (35) to obtain an Equation of the same form as (15). Below it is rewritten by using the product rule

of differentiation:

$$
\begin{aligned}
0 = \rho_t + (\rho\mu)_x - \left(\frac{1}{r}(\sigma_\nu^2\rho)_x\right)_x &= \rho_t + \rho_x\mu + \rho\mu_x - \left(\frac{1}{r}\right)_x(\sigma_\nu^2\rho)_x - \frac{1}{r}(\sigma_\nu^2\rho)_{xx} \\
&= \rho_t + \rho_x\mu + \rho\mu_x - 2\left(\frac{1}{r}\right)_x(\sigma_\nu^2\rho)_x - \frac{1}{r}(\sigma_\nu^2\rho)_{xx} + \left(\frac{1}{r}\right)_x(\sigma_\nu^2\rho)_x \\
&= \rho_t + \rho_x\mu + \rho\mu_x - 2\left(\frac{1}{r}\right)_x(\sigma_\nu^2\rho)_x - \frac{1}{r}(\sigma_\nu^2\rho)_{xx} + \left(\frac{1}{r}\right)_x(\sigma_\nu^2\rho)_x \\
&+ \left(\frac{1}{r}\right)_{xx}\sigma_\nu^2\rho - \left(\frac{1}{r}\right)_{xx}\sigma_\nu^2\rho \\
&= \rho_t + \rho\left(\mu_x + \left(\frac{1}{r}\right)_{xx}\sigma_\nu^2 + \left(\frac{1}{r}\right)_x(\sigma_\nu^2)_x\right) + \rho_x\left(\mu + \left(\frac{1}{r}\right)_x\sigma_\nu^2\right) \\
&- \left(\frac{1}{r}\right)_{xx}\sigma_\nu^2\rho - 2\left(\frac{1}{r}\right)_x(\sigma_\nu^2\rho)_x - \frac{1}{r}(\sigma_\nu^2\rho)_{xx} \\
&= \rho_t + \left(\rho\left(\mu + \left(\frac{1}{r}\right)_x\sigma_\nu^2\right)\right)_x - \left(\frac{1}{r}\sigma_\nu^2\rho\right)_{xx}. \qquad (36)
\end{aligned}
$$

For the sake of brevity all arguments have been omitted in Equation (36) and the notation $f_t := \frac{\partial f(x,t)}{\partial t}$ has been used. Comparing Equations (15) and (36), we see that the corresponding itö process for the particle position given by (36) is

$$
dX_t = \left(\left(\mu(X_t) + \frac{\partial}{\partial x}\left(\frac{1}{r(X_t)}\right)\sigma_\nu^2(X_t)\right)\rho\right)dt + \sigma_\nu(X_t)\sqrt{\frac{2}{r(X_t)}}dW. \qquad (37)
$$

Using the Euler-Maruyama method, Equation (37) is simulated in the following manner

$$
X_{n+1} = X_n + A_n^{\epsilon\to 0}\Delta t + \sqrt{2D_n^{\epsilon\to 0}\Delta t}Z_n, \qquad (38)
$$

where $A_n^{\epsilon\to 0} = \left(\mu(X_n) + \frac{\partial}{\partial x}\left(\frac{1}{r(X_n)}\right)\sigma_\nu^2(X_n)\right)$, $D_n^{\epsilon\to 0} = \frac{\sigma_\nu^2(X_n)}{r(X_n)}$ and $Z_n \sim N(0,1)$. The above will be referred to as the random walk simulation. It is important to realise that the simulation is only valid in the diffusive limit and hence yields a bias for finite values of ϵ. To remedy this, a two-fold strategy is incorporated. First of all, the random walk simulation is combined with the standard kinetic simulation in such a way that in low collisional regimes the kinetic part of the simulation is dominant and high collisional regimes the random walk part is dominant. This property is achieved by splitting each step up s.t. the particle is moved kinetically until the first collision and then it is moved using a random walk for the remainder of the step. The idea is that the random walk captures many collisions in single simulation step. Thus, from the design of the method it is intuitively clear that it converges to the standard kintec scheme in the limit $\Delta t \to 0$ and to the approximate macroscopic scheme in the limit $\epsilon \to 0$. The second part of the strategy is to use different coefficients in the drift and diffusion terms of (38) to match the first two moments of the kinetic position increment. These new terms are referred to as A_n and D_n, respectively and the kinetic-diffusion method will then be given by the composite of

$$
\psi_k(X_n, V_n, \mathcal{E}_n) = X_n + \mathcal{T}_n V_n, \qquad (39)
$$

and

$$\psi_d(X_n, \theta_n, Z_n) = X_n + A_n^{\epsilon \to 0}\theta_n + \sqrt{D_n^{\epsilon \to 0}} Z_n, \tag{40}$$

where $\mathcal{T}_n$ is the time to the first collision in the n'th step given by $\mathcal{E}_n \sim exp(1)$ via (10), V_n is the velocity, X_n is the estimated position at time t_n, $Z_n \sim N(0,1)$ and $\theta_n = \Delta t - (\mathcal{T}_n \mod \Delta t)$. Thus, the one-step method is given by

$$\phi_{KD}(X_n, \Delta t, \xi_n) = (\psi_d \circ \psi_k)(X_n, \Delta t, \xi_n), \tag{41}$$

where $\xi_n = (V_n, \mathcal{E}_n, Z_n)^T$ and $\psi_k(\cdot)$ and $\psi_d(\cdot)$ are referred to as the kinetic part and the diffusive part of the step, respectively. In Section 3.3.2, the first two moments of the kinetic motion are determined in order to define A_n and D_n. Figure 5 gives a visual representation of how the diffusive part of the kinetic-diffusion scheme can replace several collisions in each step when the collision rate becomes high.

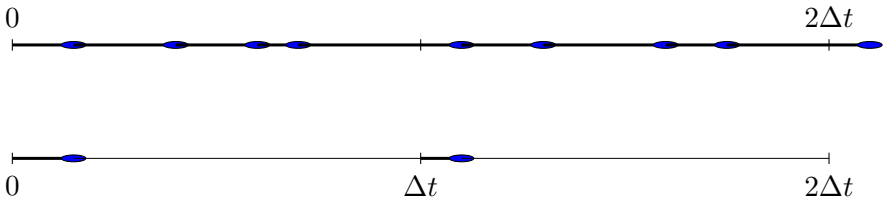


Figure 5: A visual comparison of the kinetic and kinetic-diffusion scheme with the kinetic scheme at the top and the kinetic-diffusion scheme at the bottom. The non-bold part of the line in the kinetic-diffusion scheme represent the diffusive part of the step.

3.3.2 The mean and variance of the kinetic motion

In this Section, the goal is to determine the drift and diffusion coefficient in (38) by setting them equal to respectively the mean and the variance of a kinetic positional increment in a time step Δt. First, in Section 3.3.2.1 the derivations are done for a single time step, disregarding the influence of other time steps. Then in Section 3.3.2.2, the dependence between time steps is included. This is the main result of [36]. The derivations in this section are based on the same assumptions as in 3.2 with the addition of the background being homogenous, i.e. the collision rate is constant and the velocity distribution is independent of the position.

3.3.2.1 Mean and variance without conditioning

In this section, the mean and variance of a kinetic motion in a time increment of size Δt is determined. First, recall that the collision times follow an exponential distribution with mean

$\frac{\epsilon^2}{r}$ and that the number of collisions follows a Poisson distribution with rate $\frac{r}{\epsilon^2}\Delta t$. A positional increment in the kinetic scheme during a time Δt is then given by

$$\Delta X = \sum_{n=0}^{N(\Delta t)-1} \frac{\mathcal{V}_n}{\epsilon}\mathcal{T}_n + \frac{\mathcal{V}_{N(\Delta t)}}{\epsilon}\mathcal{T}^*_{N(\Delta t)}, \tag{42}$$

where $N(\Delta t) \sim Pois(\frac{r}{\epsilon^2}\Delta t)$, $\mathcal{T}_n \sim Exp(\frac{r}{\epsilon^2})$ for $n = 0, 1, 2, \ldots, N(\Delta t) - 1$, $\mathcal{V}_n \sim M(\nu)$ for $n = 0, 1, 2, \ldots, N(\Delta t)$ and

$$\mathcal{T}^*_{N(\Delta t)} = \begin{cases} \Delta t - \sum_{n=0}^{N(\Delta t)-1} \mathcal{T}_n, & N(\Delta t) > 0, \\ \Delta t, & N(\Delta t) = 0. \end{cases} \tag{43}$$

Equation (42) is true for any starting point, in particular $\mathcal{T}_0 \sim Exp(\frac{r}{\epsilon^2})$, regardless of the starting point due to the memory-less property of the exponential distribution. Consider a starting time t where the last collision happened at $t - h$. Then, the next collision happens at time $t - h + \mathcal{T}$ with $\mathcal{T} \sim Exp(\frac{r}{\epsilon^2})$. The distribution of the length, $\mathcal{T}_0$, between t and the next collision, is then given by the distribution of $\mathcal{T}$ given that it is greater than h. Hence,

$$P(\mathcal{T}_0 > \tau_0) = P(\mathcal{T} > \tau_0 + h | \mathcal{T} > h) = P(\mathcal{T} > \tau_0),$$

where the last equality is due to the memory-less property. Consequently $\mathcal{T}_0 \overset{D}{=} \mathcal{T}$. Before actually calculating the expectation and the variance it is necessary to determine the conditional distributions of each $\mathcal{T}_n$ and $\mathcal{T}^*_{N(\Delta t)}$ given that $N(\Delta t) = N$. The approach follows along the lines of [13]. Consider the joint distribution of $Y^{(N)} = (\mathcal{T}_0, \mathcal{T}_1, \ldots, \mathcal{T}_N)^T$ given that $N(\Delta t) = N$, i.e.

$$P(Y^{(N)} \leq y | N(\Delta t) = N). \tag{44}$$

The distribution of Y given $N(\Delta t) = N$ is determined by first noting that by Baye's Rule

$$\begin{aligned} P(Y^{(N)} \leq y | N(\Delta t) = N) &= \frac{P(N(\Delta t) = N | Y^{(N)} \leq y) P(Y^{(N)} \leq y)}{P(N(\Delta t) = N)} \\ &= \begin{cases} \frac{\prod_{n=0}^{N} F_{\mathcal{T}_n}(\tau_n)}{P(N(\Delta t)=N)} & \text{if } \sum_{n=0}^{N-1} \tau_n \leq \Delta t \text{ and } \sum_{n=0}^{N} \tau_n > \Delta t, \\ 0, \text{else.} \end{cases} \end{aligned}$$

Hence, the density of $Y^{(N)}$ is given by

$$\begin{aligned} f_{Y^{(N)}|N(\Delta t)}(y|N) &= \frac{\prod_{n=0}^{N} f_{\mathcal{T}_n}(\tau_n)}{P(N(\Delta t)=N)}, \quad \sum_{n=0}^{N-1}\tau_n \le \Delta t, \quad \sum_{n=0}^{N}\tau_n > \Delta t \\ &= \frac{(\frac{r}{\epsilon^2})^{N+1} e^{-\frac{r}{\epsilon^2}\sum_{n=0}^{N}\tau_n}}{\frac{(\frac{r}{\epsilon^2}\Delta t)^N}{N!}e^{-\frac{r}{\epsilon^2}\Delta t}}, \quad \sum_{n=0}^{N-1}\tau_n \le \Delta t, \quad \sum_{n=0}^{N}\tau_n > \Delta t \\ &= \frac{r}{\epsilon^2}\frac{N!}{\Delta t^N} e^{\frac{r}{\epsilon^2}(\Delta t - \sum_{n=0}^{N}\tau_n)}, \quad \sum_{n=0}^{N-1}\tau_n \le \Delta t, \quad \sum_{n=0}^{N}\tau_n > \Delta t. \end{aligned} \tag{45}$$

As is clear from (42), $\mathcal{T}_N$ is not part of the positional increment. To obtain a joint density consisting only of random variables relevant in (42) $Y^{(N)}$ is marginalised to $Y^{(N-1)}$,

$$\begin{aligned} f_{Y^{(N-1)}|N(\Delta t)}(y|N) &= \int_{\Delta t - \sum_{n=0}^{N-1}\tau_n}^{\infty} \frac{r}{\epsilon^2}\frac{N!}{\Delta t^N} e^{\frac{r}{\epsilon^2}(\Delta t - \sum_{n=0}^{N}\tau_n)} d\tau_N, \quad \sum_{n=0}^{N-1}\tau_n \le \Delta t \\ &= \frac{N!}{\Delta t^N} e^{\frac{r}{\epsilon^2}(\Delta t - \sum_{n=0}^{N-1}\tau_n)} \int_{\Delta t - \sum_{n=0}^{N-1}\tau_n}^{\infty} \frac{r}{\epsilon^2} e^{-\frac{r}{\epsilon^2}\tau_N} d\tau_N, \quad \sum_{n=0}^{N-1}\tau_n \le \Delta t \\ &= \frac{N!}{\Delta t^N} e^{\frac{r}{\epsilon^2}(\Delta t - \sum_{n=0}^{N-1}\tau_n)} \left[-e^{-\frac{r}{\epsilon^2}\tau_N}\right]_{\Delta t - \sum_{n=0}^{N-1}\tau_n}^{\infty}, \quad \sum_{n=0}^{N-1}\tau_n \le \Delta t \\ &= \frac{N!}{\Delta t^N} e^{\frac{r}{\epsilon^2}(\Delta t - \sum_{n=0}^{N-1}\tau_n)} e^{-\frac{r}{\epsilon^2}(\Delta t - \sum_{n=0}^{N-1}\tau_n)}, \quad \sum_{n=0}^{N-1}\tau_n \le \Delta t \\ &= \frac{N!}{\Delta t^N}, \quad \sum_{n=0}^{N-1}\tau_n \le \Delta t. \end{aligned} \tag{46}$$

Since the distribution of $Y^{(N-1)}$ is the same for all τ_n, it must be the case that the conditional distribution of $\mathcal{T}_n$ given $N(\Delta t) = N$ is the same for all $n = 0, 1, \ldots, N-1$. It remains to determine how the distribution of $\mathcal{T}_N^*$ is related to the rest of the random variables. The relationship is determined by considering the transformation based on

$$\omega^*(Y^{(N-1)}) = \Delta t - \sum_{n=0}^{N-1}\mathcal{T}_n, \tag{47}$$

$$\omega_n(Y^{(N-1)}) = \mathcal{T}_n, \quad n = 0, 1, \ldots, N-1. \tag{48}$$

Let W be the random vector s.t. all but the j'th component are defined by (48) and the j'th component is defined by (47). Denote this transformation by $W = \omega(Y^{(N-1)})$. Then by Jacobi's Transformation Formula, [23], the density function of W is

$$f_{W|N(\Delta t)}(w|N) = f_{Y^{(N-1)}|N(\Delta t)}(\omega(y)|N)|det(J)|, \tag{49}$$

where J is the Jacobian matrix. The Jacobian matrix for the transformation of interest is

$$\begin{array}{c} \begin{matrix} 1 & \dots & j-1 & j & j+1 & \dots & N \end{matrix} \\ \begin{pmatrix} 1 & \dots & 0 & 0 & 0 & \dots & 0 \\ 0 & \ddots & 0 & 0 & 0 & \dots & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ -1 & \dots & -1 & -1 & \dots & -1 & -1 \\ 0 & \dots & 0 & 0 & 1 & 0 & 0 \\ 0 & \ddots & 0 & 0 & 0 & \ddots & 0 \\ 0 & \dots & 0 & 0 & 0 & \dots & 1 \end{pmatrix} \end{array} \begin{matrix} 1 \\ \vdots \\ j-1 \\ j \\ j+1 \\ \vdots \\ N. \end{matrix}$$

Thus, it only differs from an identity matrix in the j'th row where all elements are -1. Therefore, the determinant of the Jacobian matrix may by calculated to have an absolute value of 1 by use of cofactor exapnsion. Hence $f_{W|N(\Delta t)}(w|N) = f_{Y^{(N-1)}|N(\Delta t)}(\omega(y)|N)$, which shows that the conditional distribution of $\mathcal{T}_N^*$ given $N(\Delta t) = N$ is distributed in the same way as the conditional distributions of the collision times $\mathcal{T}_n$ with $n = 0, 1, \dots N-1$. So it suffices to find the distribution of one of them. Given that N collisions occur in a time step, the time at which the n'th collision happens is distributed according to the n'th order statistic of a random sample from a uniform distribution of size N, [43]. This fact, makes it straight forward to determine $f_{\mathcal{T}_0|N(\Delta t)=N}(\tau_0)$, since the stationarity of the Poisson process means that $\mathcal{T}_0$ may be seen as the time of the first collision and thus as the minimum of a sequence of uniform random variables. Hence,

$$f_{\mathcal{T}_n|N(\Delta t)}(\tau_n|N) = N\frac{(\Delta t - \tau_n)^{N-1}}{\Delta t^N}, \quad 0 \leq \tau_n \leq \Delta t, \quad N > 0, \quad n = 0, 1, \dots, N, \tag{50}$$

where $\mathcal{T}_N = \mathcal{T}_N^*$. In the case $N = 0$, the conditional density only exists $\mathcal{T}_N^*$ and is

$$f_{\mathcal{T}_N^*|N(\Delta t)}(\tau_n^*|0) = \delta(\Delta t - \tau_n^*), \quad 0 \leq \tau_n^* \leq \Delta t. \tag{51}$$

In the remaining part of this section and in the next section the positional increment is represented by

$$\Delta X_{\Delta t} = \sum_{n=0}^{N(\Delta t)} \frac{\mathcal{V}_n}{\epsilon}\mathcal{T}_n, \quad \sum_{n=0}^{N(\Delta t)} \mathcal{T}_n = \Delta t, \tag{52}$$

where $\mathcal{T}_{N(\Delta t)} = \mathcal{T}_{N(\Delta t)}^*$, which is justified by the fact the $\mathcal{T}_{N(\Delta t)}^*$ is identically distributed with the other collision times in the step when conditioning on $N(\Delta t)$. Before finding the mean and

variance of $\Delta X_{\Delta t}$, it is necessary to determine the first two moments of (50):

$$\begin{aligned}
E(\mathcal{T}_n|N(\Delta t)=N) &= \int_0^{\Delta t} \tau_n N \frac{(\Delta t-\tau_n)^{N-1}}{\Delta t^N} d\tau_n = \left[-\tau_n \frac{(\Delta t-\tau_n)^N}{\Delta t^N}\right]_0^{\Delta t} + \int_0^{\Delta t} \frac{(\Delta t-\tau_n)^N}{\Delta t^N} d\tau_n, \\
&= \left[-\frac{(\Delta t-\tau_n)^{N+1}}{\Delta t^N(N+1)}\right]_0^{\Delta t} = \frac{\Delta t}{N+1}, \qquad (53) \\
E(\mathcal{T}_n^2|N(\Delta t)=N) &= \int_0^{\Delta t} \tau_n^2 N \frac{(\Delta t-\tau_n)^{N-1}}{\Delta t^N} d\tau_n = \left[-\tau_n^2 \frac{(\Delta t-\tau_n)^N}{\Delta t^N}\right]_0^{\Delta t} + \int_0^{\Delta t} 2\tau_n \frac{(\Delta t-\tau_n)^N}{\Delta t^N} d\tau_n \\
&= \left[-2\tau_n \frac{(\Delta t-\tau_n)^{N+1}}{\Delta t^N(N+1)}\right]_0^{\Delta t} + \int_0^{\Delta t} 2\frac{(\Delta t-\tau_n)^{N+1}}{\Delta t^N(N+1)} d\tau_n \\
&= \frac{2\Delta t^2}{(N+1)(N+2)}, \qquad (54) \\
V(\mathcal{T}_n|N(\Delta t)=N) &= \frac{2\Delta t^2}{(N+1)(N+2)} - \frac{\Delta t^2}{(N+1)^2} = \frac{2(N+1)\Delta t^2 - (N+2)\Delta t^2}{(N+1)^2(N+2)} \\
&= \frac{N\Delta t^2}{(N+1)^2(N+2)}. \qquad (55)
\end{aligned}$$

Note that the case with $N = 0$, i.e. (51) is also correctly represented by the above since for $N = 0$ the mean becomes Δt and the variance becomes 0.

Now everything is set up to calculate the mean and variance of $\Delta X_{\Delta t}$. Starting with the mean, the rule of repeated expectation gives that

$$E(\Delta X_{\Delta t}) = E[E(\Delta X_{\Delta t}|N(\Delta t))]. \qquad (56)$$

By definition of the conditional expectation when conditioning on a countable r.v., [23], it can be found by first calculating it for a fixed value of the Poisson variable. Noting that $\mathcal{V}_n$ is independent of all other random variables and denoting the common distribution of respectively $\mathcal{V}_n$ and $\mathcal{T}_n$ for all n as $\mathcal{T}$ and $\mathcal{V}$ when conditioning in $N(\Delta t)$

$$\begin{aligned}
E(\Delta X_{\Delta t}|N(\Delta t)=N) &= E\left(\sum_{n=0}^{N} \frac{\mathcal{V}_n}{\epsilon}\mathcal{T}_n|N(\Delta t)=N\right) = (N+1)E\left(\frac{\mathcal{V}}{\epsilon}\mathcal{T}|N(\Delta t)=N\right) \\
&= (N+1)E\left(\frac{\mathcal{V}}{\epsilon}\right)E(\mathcal{T}|N(\Delta t)=N) \\
&= \mu\Delta t. \qquad (57)
\end{aligned}$$

Since the conditional expectation above is constant in N,

$$E(\Delta X_{\Delta t}) = \mu\Delta t. \qquad (58)$$

A similar strategy is used for the variance of the positional increment by using the law of the total variance, conditioning on $N(\Delta t)$:

$$V(\Delta X_{\Delta t}) = E[V(\Delta X_{\Delta t}|N(\Delta t))] + V[E(\Delta X_{\Delta t}|N(\Delta t))]. \qquad (59)$$

By (57), the second term in (59) is zero. Since the expectation in the first term is w.r.t. to the Poisson distribution

$$E[V(\Delta X_{\Delta t}|N(\Delta t))] = \sum_{N=0}^{\infty} \frac{\left(\frac{r}{\epsilon^2}\Delta t\right)^N}{N!} e^{-\frac{r}{\epsilon^2}\Delta t} V(\Delta X_{\Delta t}|N(\Delta t) = N). \tag{60}$$

Again it is utilised that all terms in (52) are identically distributed and the common distribution is denoted by $\frac{\mathcal{V}}{\epsilon}\mathcal{T}$, which means that

$$\begin{aligned} V(\Delta X_{\Delta t}|N(\Delta t) = N) &= \sum_{n=0}^{N} V\left(\frac{\mathcal{V}_n}{\epsilon}\mathcal{T}_n|N(\Delta t) = N\right) + 2\sum_{i=0}^{N}\sum_{j=i+1}^{N} Cov\left(\frac{\mathcal{V}_i}{\epsilon}\mathcal{T}_i, \frac{\mathcal{V}_j}{\epsilon}\mathcal{T}_j|N(\Delta t) = N\right) \\ &= (N+1)V\left(\frac{\mathcal{V}}{\epsilon}\mathcal{T}|N(\Delta t) = N\right) + 2\sum_{i=0}^{N}\sum_{j=i+1}^{N} Cov\left(\frac{\mathcal{V}_i}{\epsilon}\mathcal{T}_i, \frac{\mathcal{V}_j}{\epsilon}\mathcal{T}_j|N(\Delta t) = N\right) \\ &= (N+1)V\left(\frac{\mathcal{V}}{\epsilon}\mathcal{T}|N(\Delta t) = N\right) + (N+1)NCov\left(\frac{\mathcal{V}_0}{\epsilon}\mathcal{T}_0, \frac{\mathcal{V}_1}{\epsilon}\mathcal{T}_1|N(\Delta t) = N\right), \end{aligned}$$

where it is utilized that the covariance is the same between all terms. The two terms above are dealt with separately. The variance may be calculated by using the definition,

$$\begin{aligned} V\left(\frac{\mathcal{V}}{\epsilon}\mathcal{T}|N(\Delta t) = N\right) &= E\left(\frac{\mathcal{V}^2}{\epsilon^2}\mathcal{T}^2|N(\Delta t) = N\right) - E\left(\frac{\mathcal{V}}{\epsilon}\mathcal{T}|N(\Delta t) = N\right)^2 \\ &= E\left(\frac{\mathcal{V}^2}{\epsilon^2}\right) E\left(\mathcal{T}^2|N(\Delta t) = N\right) - E\left(\frac{\mathcal{V}}{\epsilon}\right)^2 E\left(\mathcal{T}|N(\Delta t) = N\right)^2 \\ &= (\sigma^2 + \mu^2)\frac{2\Delta t^2}{(N+1)(N+2)} - \mu^2\frac{\Delta t^2}{(N+1)^2} \\ &= \sigma^2\frac{2\Delta t^2}{(N+1)(N+2)} - \mu^2 N\frac{\Delta t^2}{(N+1)^2(N+2)}. \end{aligned} \tag{61}$$

Using the definition of the covariance yields,

$$\begin{aligned} Cov\left(\frac{\mathcal{V}_0}{\epsilon}\mathcal{T}_0, \frac{\mathcal{V}_1}{\epsilon}\mathcal{T}_1|N(\Delta t) = N\right) &= E\left(\frac{\mathcal{V}_0}{\epsilon}\mathcal{T}_0\frac{\mathcal{V}_1}{\epsilon}\mathcal{T}_1|N(\Delta t) = N\right) \\ &\quad - E\left(\frac{\mathcal{V}_0}{\epsilon}\mathcal{T}_0|N(\Delta t) = N\right) E\left(\frac{\mathcal{V}_1}{\epsilon}\mathcal{T}_1|N(\Delta t) = N\right) \\ &= E\left(\frac{\mathcal{V}_0}{\epsilon}\right) E\left(\frac{\mathcal{V}_1}{\epsilon}\right) (E(\mathcal{T}_0\mathcal{T}_1|N(\Delta t) = N) \\ &\quad - E(\mathcal{T}_0|N(\Delta t) = N)E(\mathcal{T}_1|N(\Delta t) = N) \\ &= E\left(\frac{\mathcal{V}_0}{\epsilon}\right) E\left(\frac{\mathcal{V}_1}{\epsilon}\right) Cov(\mathcal{T}_0, \mathcal{T}_1|N(\Delta t) = N) \\ &= \mu^2 Cov(\mathcal{T}_0, \mathcal{T}_1|N(\Delta t) = N). \end{aligned}$$

The covariance between two collision times may be obtained by considering the variance of the sum of them

$$\begin{aligned} &V\left(\sum_{n=0}^{N}\mathcal{T}_n|N(\Delta t) = N\right) = (N+1)V(\mathcal{T}|N(\Delta t) = N) + (N+1)NCov(\mathcal{T}_0, \mathcal{T}_1|N(\Delta t) = N) = 0 \\ &\iff Cov(\mathcal{T}_0, \mathcal{T}_1|N(\Delta t) = N) = -\frac{V(\mathcal{T}|N(\Delta t) = N)}{N} = -\frac{\Delta t^2}{(N+1)^2(N+2)}. \end{aligned} \tag{62}$$

Now putting things together

$$V(\Delta X_{\Delta t}|N(\Delta t)=N)=\sigma^2\frac{2\Delta t^2}{(N+2)}. \tag{63}$$

Plugging (63) into (60) gives

$$\begin{aligned}
V(\Delta X_{\Delta t}) &= \sum_{N=0}^{\infty}\frac{\left(\frac{r}{\epsilon^2}\Delta t\right)^N}{N!}e^{-\frac{r}{\epsilon^2}\Delta t}\sigma^2\frac{2\Delta t^2}{(N+2)}\\
&= 2\Delta t^2\sigma^2 e^{-\frac{r}{\epsilon^2}\Delta t}\sum_{N=0}^{\infty}\frac{\left(\frac{r}{\epsilon^2}\Delta t\right)^N}{N!\,(N+2)}\\
&= 2\Delta t^2\sigma^2 e^{-\frac{r}{\epsilon^2}\Delta t}\sum_{N=0}^{\infty}\frac{\left(\frac{r}{\epsilon^2}\Delta t\right)^N}{(N+2)!}(N+1)\\
&= 2\Delta t^2\sigma^2 e^{-\frac{r}{\epsilon^2}\Delta t}\sum_{N=0}^{\infty}\left(\frac{\left(\frac{r}{\epsilon^2}\Delta t\right)^N}{(N+2)!}(N+2)-\frac{\left(\frac{r}{\epsilon^2}\Delta t\right)^N}{(N+2)!}\right).
\end{aligned} \tag{64}$$

Now letting $K=N+2$

$$\begin{aligned}
V(\Delta X_{\Delta t}) &= 2\Delta t^2\sigma^2 e^{-\frac{r}{\epsilon^2}\Delta t}\sum_{K=2}^{\infty}\left(\frac{\left(\frac{r}{\epsilon^2}\Delta t\right)^{K-2}}{K!}K-\frac{\left(\frac{r}{\epsilon^2}\Delta t\right)^{K-2}}{K!}\right)\\
&= 2\Delta t^2\sigma^2 e^{-\frac{r}{\epsilon^2}\Delta t}\sum_{K=2}^{\infty}\left(\frac{\epsilon^2}{r\Delta t}\frac{\left(\frac{r}{\epsilon^2}\Delta t\right)^{K-1}}{(K-1)!}-\left(\frac{\epsilon^2}{r\Delta t}\right)^2\frac{\left(\frac{r}{\epsilon^2}\Delta t\right)^{K}}{K!}\right)\\
&= 2\Delta t^2\sigma^2 e^{-\frac{r}{\epsilon^2}\Delta t}\left(\frac{\epsilon^2}{r\Delta t}\sum_{K=2}^{\infty}\frac{\left(\frac{r}{\epsilon^2}\Delta t\right)^{K-1}}{(K-1)!}-\left(\frac{\epsilon^2}{r\Delta t}\right)^2\sum_{K=2}^{\infty}\frac{\left(\frac{r}{\epsilon^2}\Delta t\right)^{K}}{K!}\right).
\end{aligned}$$

Recall that

$$\begin{aligned}
e^x=\sum_{n=0}^{\infty}\frac{x^n}{n!} &\iff e^x-1-x=\sum_{n=2}^{\infty}\frac{x^n}{n!},\\
\frac{d}{dx}(e^x)=\sum_{n=1}^{\infty}\frac{x^{n-1}}{(n-1)!} &\iff e^x-1=\sum_{n=2}^{\infty}\frac{x^{n-1}}{(n-1)!},
\end{aligned}$$

which means that

$$\begin{aligned}
V[\Delta X_{\Delta t}] &= 2\Delta t^2\sigma^2 e^{-\frac{r}{\epsilon^2}\Delta t}\left(\frac{\epsilon^2}{r\Delta t}\left(e^{\frac{r}{\epsilon^2}\Delta t}-1\right)-\left(\frac{\epsilon^2}{r\Delta t}\right)^2\left(e^{\frac{r}{\epsilon^2}\Delta t}-1-\frac{r}{\epsilon^2}\Delta t\right)\right)\\
&= 2\Delta t^2\sigma^2 e^{-\frac{r}{\epsilon^2}\Delta t}\left(e^{\frac{r}{\epsilon^2}\Delta t}\left(\frac{\epsilon^2}{r\Delta t}-\left(\frac{\epsilon^2}{r\Delta t}\right)^2\right)+\left(\frac{\epsilon^2}{r\Delta t}\right)^2\right)\\
&= 2\sigma^2\left(\frac{\epsilon^2}{r}\Delta t-\frac{\epsilon^4}{r^2}+e^{\frac{-r}{\epsilon^2}\Delta t}\frac{\epsilon^4}{r^2}\right)\\
&= 2\sigma^2\frac{\epsilon^4}{r^2}\left(\frac{r}{\epsilon^2}\Delta t-1+e^{\frac{-r}{\epsilon^2}\Delta t}\right)\\
&= 2\sigma_\nu^2\frac{\epsilon^2}{r^2}\left(\frac{r}{\epsilon^2}\Delta t-1+e^{\frac{-r}{\epsilon^2}\Delta t}\right).
\end{aligned} \tag{65}$$

It should be verified that asymptotic behaviour of the mean in (56) and the variance in (65) is the same as for (8). In the so-called kinetic limit, i.e. for fixed epsilon and $\Delta t\to 0$, the

particle motion governed by (8) is determined by the standard kinetic scheme in Section 2.4.2.1 and hence we compare with kinetic limit of that scheme. As $\Delta t \to 0$ the probability of a collision happening in a step of size Δt becomes zero and thus the particle motion in a time Δt corresponds to

$$Xn + 1 = X_n + \frac{\mathcal{V}}{\epsilon}\Delta t, \tag{66}$$

where $\mathcal{V} \sim M(\nu)$. So the mean and variance of an increment in the standard kinetic scheme are given by

$$E\left[\frac{\mathcal{V}}{\epsilon}\Delta t\right] = \frac{\mu_\nu}{\epsilon}\Delta t = \mu\Delta t, \tag{67}$$

$$V\left[\frac{\mathcal{V}}{\epsilon}\Delta t\right] = \frac{\Delta t^2}{\epsilon^2}V(\mathcal{V}) = \sigma^2\Delta t^2. \tag{68}$$

The mean in (56) clearly fits. Now consider (65) with fixed ϵ and $\Delta t \to 0$. By expanding the exponential around $\Delta t = 0$

$$\begin{aligned} V[\Delta X_{\Delta t}] &= 2\sigma_\nu^2\frac{\epsilon^2}{r^2}\left(\frac{r}{\epsilon^2}\Delta t - 1 + e^{\frac{-r}{\epsilon^2}\Delta t}\right) \\ &= 2\sigma_\nu^2\frac{\epsilon^2}{r^2}\left(\frac{r}{\epsilon^2}\Delta t - 1 + \left(1 - \frac{r}{\epsilon^2}\Delta t - \frac{r^2}{\epsilon^4}\frac{\Delta t^2}{2} + \mathcal{O}(\Delta t^3)\right)\right), \quad \Delta t \to 0 \\ &= \sigma_\nu^2\frac{\Delta t^2}{\epsilon^2} + \mathcal{O}(\Delta t^3), \quad \Delta t \to 0 \\ &\approx \sigma^2\Delta t^2, \quad \Delta t \to 0, \end{aligned} \tag{69}$$

which is coincides with the variance in the kinetic limit. In the diffusive limit, i.e. for $\epsilon \to 0$ with fixed Δt, the particle motion governed by (8) is determined by (35) and hence the mean in (56) and variance in (65) should correspond to the mean and variance of the particle motion given by (38). Again, the mean fits and the variance is checked by letting ϵ approach zero:

$$V[\Delta X_{\Delta t}] = 2\frac{\sigma_\nu^2}{r}\Delta t, \quad \epsilon \to 0. \tag{70}$$

Recall that the real contribution of this method is to alter the derivations of the mean and variance of $\Delta X_{\Delta t}$ by incorporating the dependence between steps. The reason for this is that regardless of the step size, the standard kinetic scheme moves the particle with a constant velocity until the next collision. To see this, recall the definition of $\mathcal{T}_{N(\Delta t)}$, which is the time from the last collision in a step until the end of the step. This also means that $\mathcal{T}_0$ in the next step is the time from the beginning of the step to the first collision. Therefore, the particle moves with a constant velocity for a time $\mathcal{T}_{N(\Delta t)} + \mathcal{T}_0$. Hence, the velocity at the very end of a step will be the same as the velocity at the beginning of the next step. This relation is taken into account in the next section.

3.3.2.2 Mean and variance with conditioning

In this section, the goal is to obtain the mean and expectation of $\Delta X_{\Delta t}$ given that the last velocity is know. The reason for this approach is found in the design of the scheme in Section 3.3.1. In that scheme a step starts with moving the particle kinetically until the first collision. Then, for the remainder of the step, the particle is moved diffusively. However, as mentioned previously the last velocity of the current step and the first velocity of the next step would be the same in a kinetic scheme. Therefore, the velocity for the next step is drawn before performing the diffusive part of the step and then used to determine the moments of the diffusive part of the step s.t. on average it corresponds to a sequence of kinetic steps where the last step has the specified velocity. Starting with the expectation, the same strategy as in the previous section is used:

$$E\left[\Delta X_{\Delta t}|\mathcal{V}_{N(\Delta t)}\right] = E\left[E[\Delta X_{\Delta t}|N(\Delta t), \mathcal{V}_{N(\Delta t)}]\middle|\mathcal{V}_{N(\Delta t)}\right].$$

Again, since $N(\Delta t)$ is countable, the conditional expectation is first found for a fixed value:

$$\begin{aligned} E\left(\Delta X_{\Delta t}|\mathcal{V}_{N(\Delta t)}\right) &= E\left[E\left[\sum_{n=0}^{N-1}\frac{\mathcal{V}_n}{\epsilon}\mathcal{T}_n + \frac{\mathcal{V}_N}{\epsilon}T_N\middle|N(\Delta t)=N, \mathcal{V}_N\right]\middle|\mathcal{V}_{N(\Delta t)}\right] \\ &= E\left[E\left[\sum_{n=0}^{N-1}\frac{\mathcal{V}_n}{\epsilon}\mathcal{T}_n\middle|N(\Delta t)=N, \mathcal{V}_N\right] + E\left[\frac{\mathcal{V}_N}{\epsilon}\mathcal{T}_N\middle|N(\Delta t)=N, \mathcal{V}_N\right]\middle|\mathcal{V}_{N(\Delta t)}\right]. \end{aligned}$$

Recall that $\mathcal{V}_N$ is independent of all other variables and therefore

$$\begin{aligned} E\left(\Delta X_{\Delta t}|\mathcal{V}_{N(\Delta t)}\right) &= E\left[E\left[\sum_{n=0}^{N-1}\frac{\mathcal{V}_n}{\epsilon}\mathcal{T}_n\middle|N(\Delta t)=N\right] + \frac{\mathcal{V}_{N(\Delta t)}}{\epsilon}E\left[\mathcal{T}_N|N(\Delta t)=N\right]\middle|\mathcal{V}_{N(\Delta t)}\right] \\ &= E\left[E\left[\sum_{n=0}^{N-1}\frac{\mathcal{V}_n}{\epsilon}\mathcal{T}_n\middle|N(\Delta t)=N\right] + \frac{\mathcal{V}_{N(\Delta t)}}{\epsilon}\frac{\Delta t}{N+1}\middle|\mathcal{V}_{N(\Delta t)}\right] \\ &= E\left[E\left[\sum_{n=0}^{N-1}\frac{\mathcal{V}_n}{\epsilon}\mathcal{T}_n\middle|N(\Delta t)=N\right]\right] + \frac{\mathcal{V}_{N(\Delta t)}}{\epsilon}E\left[\frac{\Delta t}{N+1}\right]. \end{aligned} \tag{71}$$

The two terms in (71) are calculated separately. In the first term

$$\begin{aligned} E\left[\sum_{n=0}^{N-1}\frac{\mathcal{V}_n}{\epsilon}\mathcal{T}_n\middle|N(\Delta t)=N\right] &= NE\left[\frac{\mathcal{V}}{\epsilon}\mathcal{T}\middle|N(\Delta t)=N\right] \\ &= NE\left[\frac{\mathcal{V}}{\epsilon}\right]E\left[\mathcal{T}|N(\Delta t)=N\right] \\ &= \frac{N}{N+1}\mu\Delta t, \end{aligned}$$

which makes sense, since the times at which collisions occur are distributed uniformly across the interval. Now calculating the expectation for all values of N

$$\begin{aligned} E\left[E\left[\sum_{n=0}^{N-1}\frac{\mathcal{V}_n}{\epsilon}\mathcal{T}_n\middle|N(\Delta t)\right]\right] &= E\left[\frac{N}{N+1}\mu\Delta t\right] \\ &= \mu\Delta t e^{\frac{-r}{\epsilon^2}\Delta t}\sum_{N=0}^{\infty}\frac{N}{N+1}\frac{\left(\frac{r}{\epsilon^2}\Delta t\right)^N}{N!} \\ &= \mu\Delta t e^{\frac{-r}{\epsilon^2}\Delta t}\sum_{N=0}^{\infty}N\frac{\left(\frac{r}{\epsilon^2}\Delta t\right)^N}{(N+1)!}, \end{aligned} \tag{72}$$

and letting $K = N + 1$

$$\begin{aligned} \mu\Delta t e^{\frac{-r}{\epsilon^2}\Delta t}\sum_{N=0}^{\infty}N\frac{\left(\frac{r}{\epsilon^2}\Delta t\right)^N}{(N+1)!} &= \mu\Delta t e^{\frac{-r}{\epsilon^2}\Delta t}\sum_{N=1}^{\infty}(K-1)\frac{\left(\frac{r}{\epsilon^2}\Delta t\right)^{K-1}}{K!} \\ &= \mu\Delta t e^{\frac{-r}{\epsilon^2}\Delta t}\left(\sum_{N=1}^{\infty}K\frac{\left(\frac{r}{\epsilon^2}\Delta t\right)^{K-1}}{K!} - \sum_{N=1}^{\infty}\frac{\left(\frac{r}{\epsilon^2}\Delta t\right)^{K-1}}{K!}\right) \\ &= \mu\Delta t e^{\frac{-r}{\epsilon^2}\Delta t}\left(\sum_{N=1}^{\infty}\frac{\left(\frac{r}{\epsilon^2}\Delta t\right)^{K-1}}{(K-1)!} - \frac{\epsilon^2}{r\Delta t}\sum_{N=1}^{\infty}\frac{\left(\frac{r}{\epsilon^2}\Delta t\right)^{K}}{K!}\right) \\ &= \mu\Delta t e^{\frac{-r}{\epsilon^2}\Delta t}\left(e^{\frac{r}{\epsilon^2}} - \frac{\epsilon^2}{r\Delta t}\left(e^{\frac{r}{\epsilon^2}}-1\right)\right) \\ &= \mu\Delta t + \mu\frac{\epsilon^2}{r}(e^{\frac{-r}{\epsilon^2}\Delta t}-1). \end{aligned} \tag{73}$$

Similarly for the expectation in the second term of (71)

$$\begin{aligned} E\left[\frac{\Delta t}{N+1}\right] &= \sum_{N=0}^{\infty}\frac{\Delta t}{N+1}\frac{\left(\frac{r}{\epsilon}\Delta t\right)^N}{N!}e^{\frac{-r}{\epsilon}\Delta t} \\ &= \frac{\epsilon^2}{r}\sum_{N=0}^{\infty}\frac{\left(\frac{r}{\epsilon}\Delta t\right)^{N+1}}{(N+1)!}e^{\frac{-r}{\epsilon}\Delta t}. \end{aligned}$$

Letting $K = N + 1$

$$\begin{aligned} E\left[\frac{\Delta t}{N+1}\right] &= \frac{\epsilon^2}{r}e^{\frac{-r}{\epsilon}\Delta t}\sum_{K=1}^{\infty}\frac{\left(\frac{r}{\epsilon}\Delta t\right)^K}{K!} \\ &= \frac{\epsilon^2}{r}e^{\frac{-r}{\epsilon}\Delta t}\left(e^{\frac{r}{\epsilon^2}\Delta t}-1\right) \\ &= \frac{\epsilon^2}{r}\left(1-e^{\frac{-r}{\epsilon^2}\Delta t}\right). \end{aligned} \tag{74}$$

Putting things together

$$E\left[\Delta X_{\Delta t}|\mathcal{V}_{N(\Delta t)}\right] = \mu\Delta t + \frac{\epsilon^2}{r}\left(e^{\frac{-r}{\epsilon^2}\Delta t}-1\right)\left(\mu - \frac{\mathcal{V}_{N(\Delta t)}}{\epsilon}\right). \tag{75}$$

Again it should be checked, that the result has the desired kinetic and diffusive limit. In the diffusive limit the expectation becomes $\mu\Delta t$, which fits with (38). The kinetic limit is obtained by expanding the exponential:

$$\begin{aligned}\mu\Delta t + \frac{\epsilon^2}{r}\left(e^{\frac{-r}{\epsilon^2}\Delta t} - 1\right)\left(\mu - \frac{\mathcal{V}_{N(\Delta t)}}{\epsilon}\right) &= \mu\Delta t + \frac{\epsilon^2}{r}\left(\frac{-r}{\epsilon^2}\Delta t\right)\left(\mu - \frac{\mathcal{V}_{N(\Delta t)}}{\epsilon}\right) + \mathcal{O}(\Delta t^2) \\ &= \frac{\mathcal{V}_{N(\Delta t)}}{\epsilon}\Delta t + \mathcal{O}(\Delta t^2), \quad \Delta t \to 0. \end{aligned} \tag{76}$$

This makes sense because in the kinetic limit the rate of collisions remains the same, but the time increment becomes infinitely small. Therefore, the probability of a collision in a time interval Δt becomes zero and hence the velocity remains unchanged. It is important to note that the quantities in (75) and (76) are r.v. variables, but by the rule of repeated expectations their expectation match with the results in Section 3.3.2.1. As in Section 3.3.2.1 the strategy of calculating the variance when the last velocity is given, is based on the law of total variance, but instead of conditioning on $N(\Delta t)$ the conditioning is done on $\mathcal{T}_{N(\Delta t)}$:

$$V\left[\Delta X_{\Delta t}|\mathcal{V}_{N(\Delta t)}\right] = E\left[V\left[\Delta X_{\Delta t}|\mathcal{V}_{N(\Delta t)}, \mathcal{T}_{N(\Delta t)}\right]\middle|\mathcal{V}_{N(\Delta t)}\right] + V\left[E\left[\Delta X_{\Delta t}|\mathcal{V}_{N(\Delta t)}, \mathcal{T}_{N(\Delta t)}\right]\middle|\mathcal{V}_{N(\Delta t)}\right]. \tag{77}$$

Starting with the first term,

$$
\begin{aligned}
V\left[\Delta X_{\Delta t}|\mathcal{V}_{N(\Delta t)},\mathcal{T}_{N(\Delta t)}\right] &= V\left[\sum_{N=0}^{N(\Delta t)-1}\frac{\mathcal{V}_N}{\epsilon}\mathcal{T}_N+\frac{\mathcal{V}_{N(\Delta t)}}{\epsilon}\mathcal{T}_{N(\Delta t)}|\mathcal{V}_{N(\Delta t)},\mathcal{T}_{N(\Delta t)}\right]\\
&= E\left[\left(\sum_{N=0}^{N(\Delta t)-1}\frac{\mathcal{V}_N}{\epsilon}\mathcal{T}_N+\frac{\mathcal{V}_{N(\Delta t)}}{\epsilon}\mathcal{T}_{N(\Delta t)}\right)^2\middle|\mathcal{V}_{N(\Delta t)},\mathcal{T}_{N(\Delta t)}\right]\\
&- E\left[\sum_{N=0}^{N(\Delta t)-1}\frac{\mathcal{V}_N}{\epsilon}\mathcal{T}_N+\frac{\mathcal{V}_{N(\Delta t)}}{\epsilon}\mathcal{T}_{N(\Delta t)}\middle|\mathcal{V}_{N(\Delta t)},\mathcal{T}_{N(\Delta t)}\right]^2\\
&= E\left[\left(\sum_{N=0}^{N(\Delta t)-1}\frac{\mathcal{V}_N}{\epsilon}\mathcal{T}_N\right)^2\middle|\mathcal{V}_{N(\Delta t)},\mathcal{T}_{N(\Delta t)}\right]+E\left[\frac{\mathcal{V}^2_{N(\Delta t)}}{\epsilon^2}\mathcal{T}^2_{N(\Delta t)}\middle|\mathcal{V}_{N(\Delta t)},\mathcal{T}_{N(\Delta t)}\right]\\
&+ 2E\left[\frac{\mathcal{V}_{N(\Delta t)}}{\epsilon}\mathcal{T}_{N(\Delta t)}\sum_{N=0}^{N(\Delta t)-1}\frac{\mathcal{V}_N}{\epsilon}\mathcal{T}_N\middle|\mathcal{V}_{N(\Delta t)},\mathcal{T}_{N(\Delta t)}\right]\\
&- \left(E\left[\sum_{N=0}^{N(\Delta t)-1}\frac{\mathcal{V}_N}{\epsilon}\mathcal{T}_N\middle|\mathcal{T}_{N(\Delta t)}\right]+\frac{\mathcal{V}_{N(\Delta t)}}{\epsilon}\mathcal{T}_{N(\Delta t)}\right)^2\\
&= E\left[\left(\sum_{N=0}^{N(\Delta t)-1}\frac{\mathcal{V}_N}{\epsilon}\mathcal{T}_N\right)^2\middle|\mathcal{T}_{N(\Delta t)}\right]+\frac{\mathcal{V}^2_{N(\Delta t)}}{\epsilon^2}\mathcal{T}^2_{N(\Delta t)}\\
&+ 2\frac{\mathcal{V}_{N(\Delta t)}}{\epsilon}\mathcal{T}_{N(\Delta t)}E\left[\sum_{N=0}^{N(\Delta t)-1}\frac{\mathcal{V}_N}{\epsilon}\mathcal{T}_N\middle|\mathcal{T}_{N(\Delta t)}\right]-\left(E\left[\sum_{N=0}^{N(\Delta t)-1}\frac{\mathcal{V}_N}{\epsilon}\mathcal{T}_N\middle|\mathcal{T}_{N(\Delta t)}\right]\right)^2\\
&- \frac{\mathcal{V}^2_{N(\Delta t)}}{\epsilon^2}\mathcal{T}^2_{N(\Delta t)}-2\frac{\mathcal{V}_{N(\Delta t)}}{\epsilon}\mathcal{T}_{N(\Delta t)}E\left[\sum_{N=0}^{N(\Delta t)-1}\frac{\mathcal{V}_N}{\epsilon}\mathcal{T}_N\middle|\mathcal{T}_{N(\Delta t)}\right]\\
&= V\left[\sum_{N=0}^{N(\Delta t)-1}\frac{\mathcal{V}_N}{\epsilon}\mathcal{T}_N\middle|\mathcal{T}_{N(\Delta t)}\right],
\end{aligned}
\tag{78}
$$

which is the same as the variance of a positional increment during the random time $\Delta t-\mathcal{T}_{N(\Delta t)}$. In other words, the result follows from Equation (65) by replacing Δt with $\Delta t-\mathcal{T}_{N(\Delta t)}$. To justify this statement some conditions need to be proven, but before that a Lemma is required.

Lemma 1. *Given the assumptions of Section 3.3.2.1 the distribution of $\mathcal{T}^*_{N(\Delta t)}=\mathcal{T}_{N(\Delta t)}$ is given by*

$$
f_{\mathcal{T}_{N(\Delta t)}}(\tau)=\frac{r}{\epsilon^2}e^{\frac{-r}{\epsilon^2}\tau}+\delta(\Delta t-\tau)e^{\frac{-r}{\epsilon^2}\tau},\quad 0\leq\tau\leq\Delta t. \tag{79}
$$

Proof. Recall the definition of $\mathcal{T}_{N(\Delta t)}$:

$$
\mathcal{T}_{N(\Delta t)}=\begin{cases}\Delta t-\sum_{n=0}^{N(\Delta t)-1}\mathcal{T}_n, & N(\Delta t)>0,\\ \Delta t, & N(\Delta t)=0.\end{cases} \tag{80}
$$

The density, $f_{\mathcal{T}_{N(\Delta t)}}(\tau)$ may be found by considering the joint distribution of $N(\Delta t)$ and $\mathcal{T}_{N(\Delta t)}$ and then finding the marginal distribution w.r.t $\mathcal{T}_{N(\Delta t)}$. The joint density is found by multiplying the conditional density $f_{\mathcal{T}_{N(\Delta t)}|N(\Delta t)}(\tau)$ with the probability mass function of $N(\Delta t)$, i.e

$$f_{\mathcal{T}_{N(\Delta t)},N(\Delta t)}(\tau, N) = f_{\mathcal{T}_{N(\Delta t)}|N(\Delta t)}(\tau|N)P(N(\Delta t) = N), \quad 0 \leq \tau \leq \Delta t$$

where $f_{\mathcal{T}_{N(\Delta t)}|N(\Delta t)}(\tau)$ is given by (50) for $N(\Delta t) > 0$. However, for $N(\Delta t) = 0$ there are no collisions meaning that $\mathcal{T}_{N(\Delta t)} = \Delta t$ and thus $f_{\mathcal{T}_{N(\Delta t)}|N(\Delta t)}(\tau|0)$ is the degenerate distribution given by the density

$$f_{\mathcal{T}_{N(\Delta t)}|N(\Delta t)}(\tau|0) = \delta(\Delta t - \tau).$$

Hence,

$$\begin{aligned}
f_{\mathcal{T}_{N(\Delta t)}}(\tau) &= \sum_{N=0}^{\infty} f_{\mathcal{T}_{N(\Delta t)}|N(\Delta t)}(\tau, N) \frac{\left(\frac{r}{\epsilon^2}\Delta t\right)^N}{N!} e^{\frac{-r}{\epsilon^2}\Delta t} \\
&= e^{\frac{-r}{\epsilon^2}\Delta t}\delta(\Delta t - \tau) + \sum_{N=1}^{\infty} N \frac{(\Delta t - \tau)^{N-1}}{\Delta t^N} \frac{\left(\frac{r}{\epsilon^2}\Delta t\right)^N}{N!} e^{\frac{-r}{\epsilon^2}\Delta t} \\
&= e^{\frac{-r}{\epsilon^2}\Delta t}\delta(\Delta t - \tau) + \frac{r}{\epsilon^2} e^{-\frac{r}{\epsilon^2}\Delta t} \sum_{N=1}^{\infty} \frac{\left((\Delta t - \tau)\,\frac{r}{\epsilon^2}\right)^{N-1}}{(N-1)!} \\
&= e^{\frac{-r}{\epsilon^2}\Delta t}\delta(\Delta t - \tau) + \frac{r}{\epsilon^2} e^{\frac{-r}{\epsilon^2}\tau}, \quad 0 \leq \tau \leq \Delta t.
\end{aligned} \tag{81}$$

□

The following Theorem gives the sufficient conditions for the statement that (78) conforms to Equation (65) with a time step of $\Delta t - \mathcal{T}_{N(\Delta t)}$.

Theorem 1. *Assume the setting of Section 3.3.2.1. The conditional distribution of the number of collisions given the time length from the last collision to the end of the time interval is given by*

$$f_{N(\Delta t)|\mathcal{T}_{N(\Delta t)}}(N|\tau) = \frac{((\Delta t - \tau)\frac{r}{\epsilon^2})^{N-1}}{(N-1)!} e^{\frac{-r}{\epsilon^2}(\Delta t - \tau)}, \quad N \geq 1, 0 \leq \tau < \Delta t, \tag{82}$$

where $f_{N(\Delta t)|\mathcal{T}_{N(\Delta t)}}(N|\tau)$ is the probability mass function of $N(\Delta t)$ given $\mathcal{T}_{N(\Delta t)}$. In addition, the conditional distribution of $\mathcal{T}_n$ given the number of collisions in the time increment and the length from the last collision to the end of the time step is given by

$$f_{\mathcal{T}_n|N(\Delta t)=N,\mathcal{T}_N}(\tau|N, \tau_N) = (N-1)\frac{(\Delta t - \tau_N - \tau)^{N-2}}{(\Delta t - \tau_N)^{N-1}}, \quad \tau \leq \Delta t - \mathcal{T}_N, \mathcal{T}_N < \Delta t \tag{83}$$

for $n = 0, 1, \ldots, N-1$ and $N > 0$.

Proof. We start by deriving (82). First, note that by (80) the probability of having zero collisions given the length of $\mathcal{T}_{N(\Delta t)}$ is zero unless $\mathcal{T}_{N(\Delta t)} = \Delta t$ and since $P(\mathcal{T}_{N(\Delta t)} = \Delta t) = 0$, the case $N = 0$ and $\tau = \Delta t$ is excluded. To find Equation (82), Baye's rule for density functions is applied, using the density from Lemma 1:

$$f_{N(\Delta t)|\mathcal{T}_{N(\Delta t)}}(N) = \frac{f_{\mathcal{T}_{N(\Delta t)}|N(\Delta t)}(\tau)}{f_{\mathcal{T}_{N(\Delta t)}}(\tau)} P(N(\Delta t) = N) = \frac{N\frac{(\Delta t-\tau)^{N-1}}{\Delta t^N}}{\frac{r}{\epsilon^2}e^{\frac{-r}{\epsilon^2}\tau}} \frac{\left(\frac{r}{\epsilon^2}\Delta t\right)^N}{N!} e^{\frac{-r}{\epsilon^2}\Delta t}$$
$$= \frac{\left((\Delta t-\tau)\frac{r}{\epsilon^2}\right)^{N-1}}{(N-1)!} e^{\frac{-r}{\epsilon^2}(\Delta t-\tau)}, \quad N > 0, 0 \leq \tau < \Delta t. \tag{84}$$

For Equation (83), the situation is similar to Section 3.3.2.1 where the goal was to derive (50). By definition of the conditional density

$$f_{T_n|N(\Delta t),T_{N(\Delta t)}}(\tau|N,\tau_N) = \frac{f_{T_n,N(\Delta t),T_{N(\Delta t)}}(\tau,N,\tau_N)}{f_{N(\Delta t),T_{N(\Delta t)}}(N,\tau_N)} = \frac{f_{T_n,T_{N(\Delta t)}|N(\Delta t)}(\tau,\tau_N|N)P(N(\Delta t)=N)}{f_{T_{N(\Delta t)}|N(\Delta t)}(\tau_N|N)P(N(\Delta t)=N)}$$
$$= \frac{f_{T_n,T_{N(\Delta t)}|N(\Delta t)}(\tau,\tau_N|N)}{f_{T_{N(\Delta t)}|N(\Delta t)}(\tau_N|N)}, \tag{85}$$

for $n = 0, 1, \ldots N-1$. The denominator is given by (50) and the nominator is the marginal density of (49) w.r.t. $(T_n, T_{N(\Delta t)})^T$. Since the density remained unchanged in (49), the marginal density is found for $f_{Y^{(N-1)}|N(\Delta t)}(y|N)$. Also, note that $f_{Y^{(N-1)}|N(\Delta t)}(y|N)$ is completely symmetric in the collision times and therefore $\mathcal{T}_n$ and $\mathcal{T}_{N(\Delta t)}$ may be labelled $\mathcal{T}_0$ and $\mathcal{T}_1$ respectively. Hence, the goal is to find $f_{Y^{(1)}|N(\Delta t)}(\tau_0,\tau_1|N)$. To do this, note that

$$f_{Y^{(N-2)}|N(\Delta t)}(y|N) = \int_0^{\Delta t-\sum_{n=0}^{N-2}\tau_n} \frac{N!}{\Delta t^N} d\tau_{N-1} = \frac{N!}{1!\,\Delta t^N}\left(\Delta t - \sum_{n=0}^{N-2}\tau_n\right), \tag{86}$$

$$f_{Y^{(N-3)}|N(\Delta t)}(y|N) = \int_0^{\Delta t-\sum_{n=0}^{N-3}\tau_n} \frac{N!}{1!\,\Delta t^N}\left(\Delta t - \sum_{n=0}^{N-2}\tau_n\right) d\tau_{N-2} = \frac{N!}{2!\,\Delta t^N}\left(\Delta t - \sum_{n=0}^{N-3}\tau_n\right)^2, \tag{87}$$

which means that

$$f_{Y^{(1)}|N(\Delta t)}(\tau_0,\tau_1|N) = \frac{N!}{(N-2)!\,\Delta t^N}\left(\Delta t - \tau_0 - \tau_1\right)^{N-2}. \tag{88}$$

Using the original labelling

$$f_{T_n,T_{N(\Delta t)}|N(\Delta t)}(\tau,\tau_N|N) = \frac{N!}{(N-2)!\,\Delta t^N}\left(\Delta t - \tau - \tau_N\right)^{N-2}, \quad \tau \leq \Delta t - \mathcal{T}_N, \mathcal{T}_N < \Delta t \tag{89}$$

for $n = 0, 1, \ldots, N-1$ and $N > 0$. Hence,

$$f_{T_n|N(\Delta t),T_{N(\Delta t)}}(\tau|N,\tau_N) = \frac{\frac{N!}{(N-2)!\Delta t^N}\left(\Delta t - \tau - \tau_N\right)^{N-2}}{N\frac{(\Delta t-\tau_N)^{N-1}}{\Delta t^N}}$$
$$= (N-1)\frac{\left(\Delta t - \tau - \tau_N\right)^{N-2}}{\left(\Delta t - \tau_N\right)^{N-1}}, \quad \tau \leq \Delta t - \mathcal{T}_N, \mathcal{T}_N < \Delta t. \tag{90}$$

□

Calculating the variance in (78) may then be done by using the law of total variance, conditioning on $N(\Delta t)$, i.e.

$$V\left[\sum_{N=0}^{N(\Delta t)-1}\frac{\mathcal{V}_N}{\epsilon}\mathcal{T}_N\middle|\mathcal{T}_{N(\Delta t)}\right] = E\left[V\left[\Delta X_{\Delta t-\mathcal{T}_N}|N(\Delta t)=N,\mathcal{T}_N\right]\middle|\mathcal{T}_{N(\Delta t)}\right] + V\left[E\left[\Delta X_{\Delta t-\mathcal{T}_N}|N(\Delta t)=N,\mathcal{T}_N\right]\middle|\mathcal{T}_{N(\Delta t)}\right]. \tag{91}$$

Based on (83) the moments given in (53) and (54) only change by replacing the time increment, Δt with $\Delta t - \mathcal{T}_N$ and replacing N by $N-1$. Thus, the second term is again zero and by (63)

$$V\left[\Delta X_{\Delta t-\mathcal{T}_N}|N(\Delta t)=N,\mathcal{T}_N\right] = \sigma^2\frac{2(\Delta t-\mathcal{T}_N)^2}{(N+1)}. \tag{92}$$

By Equation (82)

$$\begin{aligned}
E\left[\sigma^2\frac{2(\Delta t-\mathcal{T}_N)^2}{(N+1)}\middle|\mathcal{T}_{N(\Delta t)}\right] &= \sum_{N=1}^{\infty}\sigma^2\frac{2(\Delta t-\mathcal{T}_{N(\Delta t)})^2}{(N+1)}\frac{((\Delta t-\mathcal{T}_{N(\Delta t)})\frac{r}{\epsilon^2})^{N-1}}{(N-1)!}e^{\frac{-r}{\epsilon^2}(\Delta t-\mathcal{T}_{N(\Delta t)})}\\
&= \sum_{K=0}^{\infty}\sigma^2\frac{2(\Delta t-\mathcal{T}_{N(\Delta t)})^2}{(K+2)}\frac{((\Delta t-\mathcal{T}_{N(\Delta t)})\frac{r}{\epsilon^2})^{K}}{K!}e^{\frac{-r}{\epsilon^2}(\Delta t-\mathcal{T}_{N(\Delta t)})}\\
&= 2\sigma^2(\Delta t-\mathcal{T}_{N(\Delta t)})^2e^{\frac{-r}{\epsilon^2}(\Delta t-\mathcal{T}_{N(\Delta t)})}\sum_{K=0}^{\infty}(N+1)\frac{((\Delta t-\mathcal{T}_{N(\Delta t)})\frac{r}{\epsilon^2})^{K}}{(K+2)!},
\end{aligned}$$

which is the same as (64) with $\Delta t = \Delta t - \mathcal{T}_{N(\Delta t)}$. So,

$$V\left[\Delta X_{\Delta t}\middle|\mathcal{V}_{N(\Delta t)},\mathcal{T}_{N(\Delta t)}\right] = 2\sigma_\nu^2\frac{\epsilon^2}{r^2}\left(\frac{r}{\epsilon^2}(\Delta t-\mathcal{T}_{N(\Delta t)})-1+e^{\frac{-r}{\epsilon^2}(\Delta t-\mathcal{T}_{N(\Delta t)})}\right). \tag{93}$$

Therefore,

$$\begin{aligned}
E\left[V\left[\Delta X_{\Delta t}|\mathcal{V}_{N(\Delta t)},\mathcal{T}_{N(\Delta t)}\right]\middle|\mathcal{V}_{N(\Delta t)}\right] &= \int_0^{\Delta t}2\sigma_\nu^2\frac{\epsilon^2}{r^2}\left(\frac{r}{\epsilon^2}(\Delta t-\tau)-1+e^{\frac{-r}{\epsilon^2}(\Delta t-\tau)}\right)f_{\mathcal{T}_{N(\Delta t)}}(\tau)d\tau,\\
&= \int_0^{\Delta t}2\sigma_\nu^2\frac{\epsilon^2}{r^2}\left(\frac{r}{\epsilon^2}(\Delta t-\tau)-1+e^{\frac{-r}{\epsilon^2}(\Delta t-\tau)}\right)\frac{r}{\epsilon^2}e^{\frac{-r}{\epsilon^2}\tau}d\tau\\
&+\left(2\sigma_\nu^2\frac{\epsilon^2}{r^2}\left(\frac{r}{\epsilon^2}(\Delta t-\tau)-1+e^{\frac{-r}{\epsilon^2}(\Delta t-\tau)}\right)e^{\frac{-r}{\epsilon^2}\Delta t}\right)|_{\tau=\Delta t}\\
&= 2\sigma_\nu^2\frac{\epsilon^2}{r^2}\left(\int_0^{\Delta t}\frac{r^2}{\epsilon^4}(\Delta t-\tau)e^{\frac{-r}{\epsilon^2}\tau}d\tau-\int_0^{\Delta t}\frac{r}{\epsilon^2}e^{\frac{-r}{\epsilon^2}\tau}d\tau\right.\\
&\left.+\frac{r}{\epsilon^2}e^{\frac{-r}{\epsilon^2}\Delta t}\int_0^{\Delta t}d\tau\right)\\
&= 2\sigma_\nu^2\frac{\epsilon^2}{r^2}\left(\left[-\frac{r}{\epsilon^2}(\Delta t-\tau)e^{\frac{-r}{\epsilon^2}\tau}\right]_0^{\Delta t}-\int_0^{\Delta t}\frac{r}{\epsilon^2}e^{\frac{-r}{\epsilon^2}\tau}d\tau\right.\\
&\left.-1+e^{\frac{-r}{\epsilon^2}\Delta t}+\frac{r}{\epsilon^2}\Delta te^{\frac{-r}{\epsilon^2}\Delta t}\right)\\
&= 2\sigma_\nu^2\frac{\epsilon^2}{r^2}\left(\frac{r}{\epsilon^2}\Delta t-1+e^{\frac{-r}{\epsilon^2}\Delta t}-1+e^{\frac{-r}{\epsilon^2}\Delta t}+\frac{r}{\epsilon^2}\Delta te^{\frac{-r}{\epsilon^2}\Delta t}\right)\\
&= 2\sigma_\nu^2\frac{\epsilon^2}{r^2}\left(\frac{r}{\epsilon^2}\Delta t+2e^{\frac{-r}{\epsilon^2}\Delta t}-2+\frac{r}{\epsilon^2}\Delta te^{\frac{-r}{\epsilon^2}\Delta t}\right),
\end{aligned} \tag{94}$$

where $f_{\mathcal{T}_{N(\Delta t)}}(\tau)$ is given by (79). For the second term in Equation (77)

$$\begin{aligned}
E\left[\Delta X_{\Delta t}|\mathcal{V}_{N(\Delta t)},\mathcal{T}_{N(\Delta t)}\right] &= \frac{V_{N(\Delta t)}}{\epsilon}\mathcal{T}_{N(\Delta t)} + E\left[\sum_{n=0}^{N(\Delta t)-1}\frac{\mathcal{V}_n}{\epsilon}\mathcal{T}_n\middle|\mathcal{T}_{N(\Delta t)}\right] \\
&= \frac{V_{N(\Delta t)}}{\epsilon}\mathcal{T}_{N(\Delta t)} + E\left[E\left[\sum_{n=0}^{N(\Delta t)-1}\frac{\mathcal{V}_n}{\epsilon}\mathcal{T}_n\middle|N(\Delta t)=N,\mathcal{T}_N\right]\middle|\mathcal{T}_{N(\Delta t)}\right] \\
&= \frac{V_{N(\Delta t)}}{\epsilon}\mathcal{T}_{N(\Delta t)} + \mu E\left[NE\left[\mathcal{T}|\mathcal{T}_N,N(\Delta t)=N\right]\middle|\mathcal{T}_{N(\Delta t)}\right] \\
&= \frac{V_{N(\Delta t)}}{\epsilon}\mathcal{T}_{N(\Delta t)} + \mu E\left[\Delta t-\mathcal{T}_{N(\Delta t)}\middle|\mathcal{T}_{N(\Delta t)}\right] \\
&= \mu\Delta t + \mathcal{T}_{N(\Delta t)}\left(\frac{\mathcal{V}_{N(\Delta t)}}{\epsilon}-\mu\right),
\end{aligned}$$

which means that Equation (78) amounts to

$$\begin{aligned}
V\left[E\left[\Delta X_{\Delta t}|\mathcal{V}_{N(\Delta t)},\mathcal{T}_{N(\Delta t)}\right]\middle|\mathcal{V}_{N(\Delta t)}\right] &= V\left[\mathcal{T}_{N(\Delta t)}\left(\frac{\mathcal{V}_{N(\Delta t)}}{\epsilon}-\mu\right)\middle|\mathcal{V}_{N(\Delta t)}\right] \\
&= \left(\frac{\mathcal{V}_{N(\Delta t)}}{\epsilon}-\mu\right)^2 V\left[\mathcal{T}_{N(\Delta t)}\right].
\end{aligned}$$

Thus, it only remains to determine $V\left[\mathcal{T}_{N(\Delta t)}\right]$, which is done by calculating the first two moments separately, starting with the first moment:

$$\begin{aligned}
E\left[\mathcal{T}_{N(\Delta t)}\right] &= \int_0^{\Delta t}\tau\frac{r}{\epsilon^2}e^{\frac{-r}{\epsilon^2}\tau}d\tau + \int_0^{\Delta t}\delta(\Delta t-\tau)\tau e^{\frac{-r}{\epsilon^2}\tau}d\tau \\
&= \left[-\tau e^{\frac{-r}{\epsilon^2}\tau}\right]_0^{\Delta t} + \int_0^{\Delta t}e^{\frac{-r}{\epsilon^2}\tau}d\tau + \Delta t e^{\frac{-r}{\epsilon^2}\Delta t} \\
&= -\Delta t e^{\frac{-r}{\epsilon^2}\Delta t} + \frac{\epsilon^2}{r}(1-e^{\frac{-r}{\epsilon^2}\Delta t}) + \Delta t e^{\frac{-r}{\epsilon^2}\Delta t} \\
&= \frac{\epsilon^2}{r}(1-e^{\frac{-r}{\epsilon^2}\Delta t}).
\end{aligned} \tag{95}$$

The second moment is calculated in a similar matter:

$$\begin{aligned}
E\left[\mathcal{T}_{N(\Delta t)}^2\right] &= \int_0^{\Delta t}\tau^2\frac{r}{\epsilon^2}e^{\frac{-r}{\epsilon^2}\tau}d\tau + \int_0^{\Delta t}\delta(\Delta t-\tau)\tau^2 e^{\frac{-r}{\epsilon^2}\tau}d\tau \\
&= \left[-\tau^2 e^{\frac{-r}{\epsilon^2}\tau}\right]_0^{\Delta t} + 2\int_0^{\Delta t}\tau e^{\frac{-r}{\epsilon^2}\tau}d\tau + \Delta t^2 e^{\frac{-r}{\epsilon^2}\Delta t} \\
&= 2\left(\left[-\frac{\epsilon^2}{r}\tau e^{\frac{-r}{\epsilon^2}\tau}\right]_0^{\Delta t} + \int_0^{\Delta t}\frac{\epsilon^2}{r}e^{\frac{-r}{\epsilon^2}\tau}d\tau\right) \\
&= 2\left(-\frac{\epsilon^2}{r}\Delta t e^{\frac{-r}{\epsilon^2}\Delta t} + \frac{\epsilon^4}{r^2}(1-e^{\frac{-r}{\epsilon^2}\Delta t})\right).
\end{aligned} \tag{96}$$

Thus,

$$\begin{aligned}
V\left[\mathcal{T}_{N(\Delta t)}\right] &= 2\left(-\frac{\epsilon^2}{r}\Delta t e^{\frac{-r}{\epsilon^2}\Delta t} + \frac{\epsilon^4}{r^2}(1-e^{\frac{-r}{\epsilon^2}\Delta t})\right) - \frac{\epsilon^4}{r^2}(1-e^{\frac{-r}{\epsilon^2}\Delta t})^2 \\
&= 2\left(-\frac{\epsilon^2}{r}\Delta t e^{\frac{-r}{\epsilon^2}\Delta t} + \frac{\epsilon^4}{r^2}(1-e^{\frac{-r}{\epsilon^2}\Delta t})\right) - \frac{\epsilon^4}{r^2} - \frac{\epsilon^4}{r^2}e^{\frac{-r}{\epsilon^2}2\Delta t} + 2\frac{\epsilon^4}{r^2}e^{\frac{-r}{\epsilon^2}\Delta t} \\
&= \frac{\epsilon^4}{r^2}\left(1-e^{\frac{-r}{\epsilon^2}2\Delta t} - 2\frac{r}{\epsilon^2}\Delta t e^{\frac{-r}{\epsilon^2}\Delta t}\right),
\end{aligned} \tag{97}$$

which means that

$$V\left[\Delta X_{\Delta t}|\mathcal{V}_{N(\Delta t)}\right] = 2\sigma_\nu^2\frac{\epsilon^2}{r^2}\left(\frac{r}{\epsilon^2}\Delta t + 2e^{\frac{-r}{\epsilon^2}\Delta t} - 2 + \frac{r}{\epsilon^2}\Delta t e^{\frac{-r}{\epsilon^2}\Delta t}\right) + \left(\frac{\mathcal{V}_{N(\Delta t)}}{\epsilon} - \mu\right)^2\frac{\epsilon^4}{r^2}\left(1 - e^{\frac{-r}{\epsilon^2}2\Delta t} - 2\frac{r}{\epsilon^2}\Delta t e^{\frac{-r}{\epsilon^2}\Delta t}\right). \tag{98}$$

In the diffusive limit, the variance becomes $2\frac{\sigma_\nu^2}{r}\Delta t$. The kinetic limit is obtained by using the Maclaurin series of each exponential. Starting with the first term

$$\begin{aligned}\frac{r}{\epsilon^2}\Delta t + 2e^{\frac{-r}{\epsilon^2}\Delta t} - 2 + \frac{r}{\epsilon^2}\Delta t e^{\frac{-r}{\epsilon^2}\Delta t} &= \frac{r}{\epsilon^2}\Delta t + \left(2 - 2\frac{r}{\epsilon^2}\Delta t + \frac{r^2}{\epsilon^4}\Delta t^2 - \frac{r^3}{\epsilon^6 3}\Delta t^3 + \mathcal{O}(\Delta t^4)\right)\\ &\quad - 2 + \left(\frac{r}{\epsilon^2}\Delta t - \frac{r^2}{\epsilon^4}\Delta t^2 + \frac{r^3}{\epsilon^6 2}\Delta t^3 + \mathcal{O}(\Delta t^4)\right)\\ &= \frac{r^3}{\epsilon^6 2}\Delta t^3 - \frac{r^3}{\epsilon^6 3}\Delta t^3 + \mathcal{O}(\Delta t^4)\\ &\approx \frac{r^3}{\epsilon^6 6}\Delta t^3, \quad \Delta t \to 0.\end{aligned} \tag{99}$$

Similarly for the second term

$$\begin{aligned}1 - e^{\frac{-r}{\epsilon^2}2\Delta t} - 2\frac{r}{\epsilon^2}\Delta t e^{\frac{-r}{\epsilon^2}\Delta t} &= 1 - \left(1 - \frac{r}{\epsilon^2}2\Delta t + \frac{r^2}{\epsilon^4}2\Delta t^2 - \frac{r^3}{\epsilon^6}\frac{4}{3}\Delta t^3 + \mathcal{O}(\Delta t^4)\right)\\ &\quad - 2\left(\frac{r}{\epsilon^2}\Delta t - \frac{r^2}{\epsilon^4}\Delta t^2 + \frac{r^3}{\epsilon^6 2}\Delta t^3 + \mathcal{O}(\Delta t^4)\right)\\ &= \frac{r^3}{\epsilon^6}\frac{4}{3}\Delta t^3 - \frac{r^3}{\epsilon^6}\Delta t^3 + \mathcal{O}(\Delta t^4)\\ &\approx \frac{r^3}{\epsilon^6 3}\Delta t^3, \quad \Delta t \to 0.\end{aligned} \tag{100}$$

Combining the results

$$\begin{aligned}V\left[\Delta X_{\Delta t}|\mathcal{V}_{N(\Delta t)}\right] &= \sigma_\nu^2\frac{r}{\epsilon^4 3}\Delta t^3 + \left(\frac{\mathcal{V}_{N(\Delta t)}}{\epsilon} - \mu\right)^2\frac{r}{\epsilon^2 3}\Delta t^3\\ &= \frac{r}{\epsilon^2 3}\Delta t^3\left(\sigma_\nu^2 + \left(\frac{\mathcal{V}_{N(\Delta t)}}{\epsilon} - \mu\right)^2\right), \quad \Delta t \to 0,\end{aligned} \tag{101}$$

which is one order higher than the variance obtained in Section 3.3.2.1, with the additional term that in expectation corresponds to the variance of the velocity.

3.3.3 Dealing with heterogeneity

The derivations in Sections 3.3.2.1 and 3.3.2.2 were based on a homogeneous background, i.e. a constant collision rate and a fixed Maxwellian distribution at all particle positions. To avoid redoing the derivations under the more general assumption of heterogeneity it is instead suggested to simply adapt (75) and (98), see [37]. The adaptation is two-fold. First of all, σ_ν^2, μ and r become dependent on the position and instead of evaluating them at the position after a

kinetic movement they are evaluated at the intermediate position

$$x_n'' = x_n' + \mu(x_n')\frac{\theta_n}{2}, \tag{102}$$

where x_n' is the position after moving kinetically to the first collision in the n'th step and $\theta_n = \Delta t - (\tau_n \mod \Delta t)$, i.e. θ_n is the remaining time to the next step. The second adaptation is based on the advection and diffusion coefficients in Section 3.3.1. Recall that in the n'th step

$$A_n^{\epsilon\to 0} = \left(\mu(x_n) + \frac{\partial}{\partial x}\left(\frac{1}{r(x_n)}\right)\sigma_\nu^2(x_n)\right), \tag{103}$$

$$D_n^{\epsilon\to 0} = \frac{\sigma_\nu^2(x_n)}{r(x_n)}. \tag{104}$$

Hence, when the collision rate depends on the position of the particle a term is added to the advection coefficient. Note that

$$\frac{\partial D_n^{\epsilon\to 0}}{\partial r}\frac{dr}{dx_n} = -\frac{\sigma_\nu^2(x_n)}{r^2(x_n)}r'(x_n) = \sigma_\nu^2(x_n)\frac{d}{dx_n}\left(\frac{1}{r(x_n)}\right), \tag{105}$$

and that for a step of time length θ given by (38)

$$E\left[\Delta X_\theta\right] = A_n^{\epsilon\to 0}\theta, \tag{106}$$

$$V\left[\Delta X_\theta\right] = 2D_n^{\epsilon\to 0}\theta. \tag{107}$$

This leads to the final terms for the advection and diffusion coefficients:

$$\begin{aligned} D_n &= \frac{1}{2\theta}V[\Delta X_\theta|\mathcal{V}_{N(\theta)}] \\ &= \sigma_\nu^2(x_n'')\frac{\epsilon^2}{\theta r(x_n'')^2}\left(\frac{r(x_n'')}{\epsilon^2}\theta + 2e^{\frac{-r(x_n'')}{\epsilon^2}\theta} - 2 + \frac{r(x_n'')}{\epsilon^2}\theta e^{\frac{-r(x_n'')}{\epsilon^2}\theta}\right) \\ &+ \left(\frac{\nu_{n+1}}{\epsilon} - \mu(x_n'')\right)^2\frac{\epsilon^4}{2\theta r(x_n'')^2}\left(1 - e^{\frac{-r(x_n'')}{\epsilon^2}2\theta} - 2\frac{r(x_n'')}{\epsilon^2}\theta e^{\frac{-r(x_n'')}{\epsilon^2}\theta}\right), \end{aligned} \tag{108}$$

$$\begin{aligned} A_n &= \frac{1}{\theta}E[\Delta X_\theta|\mathcal{V}_{N(\theta)}] + \frac{\partial D_n}{\partial r}\frac{dr}{dx}(x_n'') \\ &= \mu(x_n'') + \frac{\epsilon^2}{r(x_n'')\theta}\left(e^{\frac{-r(x_n'')}{\epsilon^2}\theta} - 1\right)\left(\mu(x_n'') - \frac{\nu_{n+1}}{\epsilon}\right) + \frac{\partial D_n}{\partial r}\frac{dr}{dx}(x_n''). \end{aligned} \tag{109}$$

Now everything is set up for the kinetic-diffusion scheme.

3.3.4 Kinetic-diffusion algorithm

Algorithm 4: Kinetic-diffusion simulation

Function $\psi_d(x,v,\theta,z)$**:**
- $x'' = x + \mu(x)\frac{\theta}{2}$
- $D \leftarrow (108)$ `// determine diffusion coefficient`
- $A \leftarrow (109)$ `// determine advection coefficient`
- $x = x + A\theta + \sqrt{2D\theta}z$ `// move particle diffusively untill next time point`
- **return** x, z

`// One-step function for kinetic-diffusion scheme`

Function $\phi_{KD}(\Delta t,x,v,t,\tau,y,z)$**:**
- $x' = \psi_k(x,v,\tau)$ `// see Algorithm 2`
- $t = t + \tau$
- $\theta = \Delta t - (\tau \mod \Delta t)$
- $v = \mu(x') + \sigma(x')y$ `// sample from post-collisional distrbution`
- $t = t + \theta$
- $x, z = \psi_d(x',v,\theta,z)$
- **return** x, v, t, θ

`// Kinetic-diffusion simulation in` $[0, T]$

Function `KDMC(`Δt, T`)`**:**
- $t = 0$
- $x, v \leftarrow S(x, v)$ `// from` (4)
- $\varepsilon \leftarrow Exp(1)$
- $\tau =$ `SampleCollision(`x, v, ε`)`
- **while** $t + \tau < T$ **do**
  - $z \leftarrow N(0, 1)$, $y \leftarrow N(0, 1)$
  - $x, v, t, \_ = \phi_{KD}(\Delta t,x,v,t,\tau,y,z)$
  - $\varepsilon \leftarrow Exp(1)$
  - $\tau =$ `SampleCollision(`x, v, ε`)` `// see Algorithm 1`
- **if** $t < T$ **then**
  - $x = x + v(T - t)$ `// particle is moved kinetically to the end`
- **return** x, v

3.4 Asymptotic preserving splitting method

In this section, the second main Monte Carlo method is presented, [11]. Throughout the section, a homogeneous background, i.e. $r(x) = r$, is assumed and it also assumed that the velocity distribution is symmetric around 0. The section is split into two parts. In Section 3.4.1, a reformulation of Equation (8) is presented. Based on this reformulation a Monte Carlo scheme is set up in Section 3.4.2 and finally, the algorithm is presented in Section 3.4.3.

3.4.1 Reformulation

We start by summarising the different steps of the method. The method is constructed for the Goldstein-Taylor equations, [17], under the diffusive scaling. Therefore, the first step is to reformulate Equation (8) in two ways and use an operator splitting technique, see [21], s.t. it has a form similar to a version of the Goldstein-Taylor equations. The next step is to use a fully implicit discretiszation of the reformulated system, and by rewriting the discretized system, obtaining an advection-diffusion equation which is equivalent to the original system up to $\mathcal{O}(\Delta t)$. The third step is to use a first order operator splitting technique on the new system to obtain the Fokker-Planck equation and an ODE. These are then estimated numerically using a Monte Carlo method. First, the two versions of the Goldstein-Taylor equations are presented. The first version is given by

$$\begin{cases} \frac{\partial f_+}{\partial t} + \frac{1}{\epsilon}\frac{\partial f_+}{\partial x} = \frac{1}{\epsilon^2}\left(f_- - f_+\right), \\ \frac{\partial f_-}{\partial t} - \frac{1}{\epsilon}\frac{\partial f_-}{\partial x} = \frac{1}{\epsilon^2}\left(f_+ - f_-\right), \end{cases} \tag{110}$$

where $f_\pm = f_\pm(x,t)$ are the densities of the position of particles with velocity given by, respectively, $\frac{1}{\epsilon}$ and $-\frac{1}{\epsilon}$, and ϵ is the diffusion parameter. It motivates an even-odd parity transformation of more general equations such as the one in Equation (8), see [25]. This transformation helps to extend the method constructed on the Goldstein-Taylor equations to Equation (8). The second version is obtained by defining

$$h(x,t) = \frac{f_+(x,t) + f_-(x,t)}{2}, \tag{111}$$

$$j(x,t) = \frac{f_+(x,t) - f_-(x,t)}{2\epsilon}, \tag{112}$$

and setting up the equations for h and j:

$$\begin{cases} \frac{\partial h}{\partial t} + \frac{\partial j}{\partial x} = 0, \\ \frac{\partial j}{\partial t} + \frac{1}{\epsilon^2}\frac{\partial h}{\partial x} = -\frac{1}{\epsilon^2} j. \end{cases} \tag{113}$$

This version, is the one that is discretised in time and reformulated to obtain an advection-diffusion equation on which a standard operator splitting method is used to obtain a system with a Fokker-Planck equation and an ODE.

To bring Equation (8) on the same form as (113), it is rewritten using an even-odd parity formulation as in [25], by first defining

$$f_+(x,\nu,t) = f(x,\nu,t), \tag{114}$$

$$f_-(x,\nu,t) = f(x,-\nu,t), \tag{115}$$

for $\nu > 0$ and obtaining the equivalent equation system,

$$\begin{cases} \frac{\partial f_+}{\partial t} + \frac{\nu}{\epsilon}\frac{\partial f_+}{\partial x} = \frac{r}{\epsilon^2}\left(M\rho - f_+\right), \\ \frac{\partial f_-}{\partial t} - \frac{\nu}{\epsilon}\frac{\partial f_-}{\partial x} = \frac{r}{\epsilon^2}\left(M\rho - f_-\right), \end{cases} \tag{116}$$

where arguments have been omitted for concision. Similar to the reformulation of the Goldstein-Taylor model, the above system is reformulated by defining parities as in Equations (111) and (112). The equation system for h and j is obtained as follows, starting with the equation for h:

$$\begin{aligned} &\frac{1}{2}\left(\partial_t f_+ + \frac{\nu}{\epsilon}\partial_x f_+\right) + \frac{1}{2}\left(\partial_t f_- - \frac{\nu}{\epsilon}\partial_x f_-\right) = \frac{r}{2\epsilon^2}\left(M\rho - f_+\right) + \frac{r}{2\epsilon^2}\left(M\rho - f_-\right) \\ \iff &\frac{1}{2}\left(\partial_t (f_+ + f_-) + \frac{\nu}{\epsilon}\partial_x (f_+ - f_-)\right) = \frac{r}{2\epsilon^2}\left(2M\rho - (f_+ + f_-)\right) \\ \iff &\partial_t h + \nu \partial_x j = \frac{r}{\epsilon^2}(M\rho - h). \end{aligned}$$

The equation for j is derived in a similar manner:

$$\begin{aligned} &\frac{1}{2\epsilon}\left(\partial_t f_+ + \frac{\nu}{\epsilon}\partial_x f_+\right) - \frac{1}{2\epsilon}\left(\partial f_- - \frac{\nu}{\epsilon}\partial_x f_-\right) = \frac{r}{2\epsilon^3}\left(M\rho - f_+\right) - \frac{r}{2\epsilon^3}\left(M\rho - f_-\right) \\ \iff &\frac{1}{2\epsilon}\left(\partial_t (f_+ - f_-) + \frac{\nu}{\epsilon}\partial_x (f_+ + f_-)\right) = \frac{r}{2\epsilon^3}(f_- - f_+) \\ \iff &\partial_t j + \frac{\nu}{\epsilon^2}\partial_x h = -\frac{r}{\epsilon^2} j. \end{aligned}$$

Putting things together the system for h and j is given by

$$\begin{cases} \partial_t h + \nu \partial_x j = \frac{r}{\epsilon^2}(M\rho - h), \\ \partial_t j + \frac{\nu}{\epsilon^2}\partial_x h = -\frac{r}{\epsilon^2} j. \end{cases} \tag{117}$$

In a time step $[t_n, t_{n+1}]$, the first order splitting technique amounts to solving

$$(I) \begin{cases} \partial_t h^{1/2} + \nu \partial_x j^{1/2} = 0, \quad t \in [t_n, t_{n+1}], \quad h^{1/2}(t_n) = h_{sp}(t_n), \\ \partial_t j^{1/2} + \frac{\nu}{\epsilon^2}\partial_x h^{1/2} = -\frac{r}{\epsilon^2} j^{1/2}, \quad t \in [t_n, t_{n+1}], \quad j^{1/2}(t_n) = j_{sp}(t_n) \end{cases}, \tag{118}$$

$$(II) \begin{cases} \partial_t h^{2/2} = \frac{r}{\epsilon^2}(M\rho - h^{2/2}), \quad t \in [t_n, t_{n+1}], \quad h^{2/2}(t_n) = h_{1/2}(t_{n+1}), \\ \partial_t j^{2/2} = 0, \quad t \in [t_n, t_{n+1}], \quad j^{2/2}(t_n) = j_{1/2}(t_{n+1}), \end{cases} \tag{119}$$

where $h_{sp}(x,\nu,t_n) = h^{2/2}(x,\nu,t_{n-1})$ and $j_{sp}(x,\nu,t_n) = j^{2/2}(x,\nu,t_{n-1})$ for $n \geq 1$. For $n = 0$ h_{sp} and j_{sp} are given by the initial conditions. For more on operator splitting see Appendix A. At the end of each step $h(x,\nu,t_{n+1})$ and $j(x,\nu,t_{n+1})$ are approximated by $h^{2/2}(x,\nu,t_{n+1})$ and $j^{2/2}(x,\nu,t_{n+1})$, respectively. For concision, the emphasis on the time increment in which the different parts of the splitting are solved, is omitted moving forward. The system in (I) is of the same form as the Goldstein-Taylor equations in (113). Now the reformulation through an implicit discretization, presented in [11] for the Goldstein-Taylor equations and for the radiative transport equation, may be performed on (118). Both equations in (118) are discretized in a fully implicit way:

$$\begin{cases} \frac{h_{n+1}^{1/2} - h_n^{1/2}}{\Delta t} + \nu \partial_x j_{n+1}^{1/2} = 0, \\ \frac{j_{n+1}^{1/2} - j_n^{1/2}}{\Delta t} + \frac{\nu}{\epsilon^2} \partial_x h_{n+1}^{1/2} = -\frac{r}{\epsilon^2} j_{n+1}^{1/2}, \end{cases} \tag{120}$$

where $h_n^{1/2}$ is the estimate for $h(x,v,t_n)$ at step n and the same for $j_n^{1/2}$. Isolating for $h_{n+1}^{1/2}$ gives

$$h_{n+1}^{1/2} = h_n^{1/2} - \Delta t(\nu \partial_x j_{n+1}^{1/2}), \tag{121}$$

and isolating for $j_{n+1}^{1/2}$ in the second equation of (120) is done as follows:

$$\begin{aligned} & \epsilon^2(j_{n+1}^{1/2} - j_n^{1/2}) = -\Delta t(\nu \partial_x h_{n+1}^{1/2} + r j_{n+1}^{1/2}), \\ \iff & \epsilon^2 j_{n+1}^{1/2} + \Delta t r j_{n+1}^{1/2} = \epsilon^2 j_n^{1/2} - \Delta t \nu \partial_x h_{n+1}^{1/2}, \\ \iff & j_{n+1}^{1/2} = \frac{\epsilon^2}{\epsilon^2 + \Delta t r} j_n^{1/2} - \frac{\Delta t}{\epsilon^2 + \Delta t r} \nu \partial_x h_{n+1}^{1/2}, \qquad (122) \\ \iff & \frac{j_{n+1}^{1/2} - j_n^{1/2}}{\Delta t} = \frac{-r}{\epsilon^2 + \Delta t r} j_n^{1/2} - \frac{1}{\epsilon^2 + \Delta t r} \nu \partial_x h_{n+1}^{1/2}. \qquad (123) \end{aligned}$$

Plugging (122) into the first equation of (120) gives

$$\begin{aligned} & \frac{h_{n+1}^{1/2} - h_n^{1/2}}{\Delta t} + \nu \partial_x \left(\frac{\epsilon^2}{\epsilon^2 + \Delta t r} j_n^{1/2} - \frac{\Delta t}{\epsilon^2 + \Delta t r} \nu \partial_x h_{n+1}^{1/2} \right) = 0, \\ \iff & \frac{h_{n+1}^{1/2} - h_n^{1/2}}{\Delta t} + \nu \partial_x \left(\frac{\epsilon^2}{\epsilon^2 + \Delta t r} j_n^{1/2} \right) - \nu^2 \partial_x \left(\frac{\Delta t}{\epsilon^2 + \Delta t r} \partial_x h_{n+1}^{1/2} \right) = 0. \end{aligned} \tag{124}$$

Plugging (121) in (123) yields

$$\begin{aligned} & \frac{j_{n+1}^{1/2} - j_n^{1/2}}{\Delta t} = \frac{-r}{\epsilon^2 + \Delta t r} j_n^{1/2} - \frac{1}{\epsilon^2 + \Delta t r} \nu \partial_x \left(h_n^{1/2} - \Delta t(\nu \partial_x j_{n+1}^{1/2}) \right), \\ \iff & \frac{j_{n+1}^{1/2} - j_n^{1/2}}{\Delta t} + \frac{1}{\epsilon^2 + \Delta t r} \nu \partial_x h_n^{1/2} = \frac{-r}{\epsilon^2 + \Delta t r} j_n^{1/2} + \frac{\Delta t}{\epsilon^2 + \Delta t r} \nu^2 \partial_{xx} j_{n+1}^{1/2}. \end{aligned} \tag{125}$$

Equations (124) and (125) are first order approximations of the system

$$\begin{cases} \partial_t h^{1/2} + \nu \partial_x \left(\frac{\epsilon^2}{\epsilon^2 + \Delta t r} j^{1/2} \right) - \nu^2 \partial_x \left(\frac{\Delta t}{\epsilon^2 + \Delta t r} \partial_x h^{1/2} \right) = 0, \\ \partial_t j^{1/2} + \frac{1}{\epsilon^2 + \Delta t r} \nu \partial_x h^{1/2} = \frac{-r}{\epsilon^2 + \Delta t r} j^{1/2} + \frac{\Delta t}{\epsilon^2 + \Delta t r} \nu^2 \partial_{xx} j^{1/2}. \end{cases} \tag{126}$$

The same approach may be used for (119) to obtain a modified system that is an order $\mathcal{O}(\Delta t)$ approximation of (117). The system is discretized in a fully implicit manner,

$$\begin{cases} \frac{h_{n+1}^{2/2} - h_{n+1}^{1/2}}{\Delta t} = \frac{r}{\epsilon^2}(M\rho_{n+1}^{2/2} - h_{n+1}^{2/2}), \\ \frac{j_{n+1}^{2/2} - j_{n+1}^{1/2}}{\Delta t} = 0. \end{cases} \tag{127}$$

Solving for $h_{n+1}^{2/2}$ in the first equation gives

$$\begin{aligned} &\epsilon^2(h_{n+1}^{2/2} - h_{n+1}^{1/2}) = \Delta t r(M\rho_{n+1}^{2/2} - h_{n+1}^{2/2}), \\ \iff &(\epsilon^2 + \Delta t r)h_{n+1}^{2/2} = \epsilon^2 h_{n+1}^{1/2} + \Delta t r M\rho_{n+1}^{2/2}, \\ \iff &h_{n+1}^{2/2} = \frac{\epsilon^2}{\epsilon^2 + \Delta t r}h_{n+1}^{1/2} + \frac{\Delta t r}{\epsilon^2 + \Delta t r}M\rho_{n+1}^{2/2}, \\ \iff &\frac{h_{n+1}^{2/2} - h_{n+1}^{1/2}}{\Delta t} = -\frac{r}{\epsilon^2 + \Delta t r}h_{n+1}^{1/2} + \frac{r}{\epsilon^2 + \Delta t r}M\rho_{n+1}^{2/2} = \frac{r}{\epsilon^2 + \Delta t r}(M\rho_{n+1}^{2/2} - h_{n+1}^{1/2}). \end{aligned} \tag{128}$$

Thus, up to an error of $\mathcal{O}(\Delta t)$, (127) is equivalent to the system

$$\begin{cases} \partial_t h^{2/2} = \frac{r}{\epsilon^2 + \Delta t r}(M\rho^{2/2} - h^{2/2}), \\ \partial_t j^{2/2} = 0, \end{cases} \tag{129}$$

and combined with (126), it constitutes a first order splitting of the system

$$\begin{cases} \partial_t h + \nu\partial_x\left(\frac{\epsilon^2}{\epsilon^2 + \Delta t r}j\right) = \nu^2\partial_x\left(\frac{\Delta t}{\epsilon^2 + \Delta t r}\partial_x h\right) + \frac{r}{\epsilon^2 + \Delta t r}(M\rho - h), \\ \partial_t j + \frac{1}{\epsilon^2 + \Delta t r}\nu\partial_x h = \frac{-r}{\epsilon^2 + \Delta t r}j + \frac{\Delta t}{\epsilon^2 + \Delta t r}\nu^2\partial_{xx} j. \end{cases} \tag{130}$$

Given the back transformations of (111) and (112),

$$f_+(x, \nu, t) = h(x, \nu, t) + \epsilon j(x, v, t), \tag{131}$$

$$f_-(x, \nu, t) = h(x, \nu, t) - \epsilon j(x, \nu, t), \tag{132}$$

the equation for $f_+(x, \nu, t)$ is found by first considering the left hand side (LHS), which becomes

$$\begin{aligned} \text{LHS} &= \partial_t h + \nu\partial_x\left(\frac{\epsilon^2}{\epsilon^2 + \Delta t r}j\right) + \epsilon\left(\partial_t j + \frac{1}{\epsilon^2 + \Delta t r}\nu\partial_x h\right) \\ &= \partial_t f_+ + \nu\frac{\epsilon}{\epsilon^2 + \Delta t r}\partial_x \epsilon j + \frac{\epsilon}{\epsilon^2 + \Delta t r}\nu\partial_x h \\ &= \partial_t f_+ + \nu\frac{\epsilon}{\epsilon^2 + \Delta t r}\partial_x f_+, \end{aligned} \tag{133}$$

and then the right hand side (RHS)

$$\begin{aligned} \text{RHS} &= \nu^2\partial_x\left(\frac{\Delta t}{\epsilon^2 + \Delta t r}\partial_x h\right) + \frac{r}{\epsilon^2 + \Delta t r}(M\rho - h) + \epsilon\left(\frac{-r}{\epsilon^2 + \Delta t r}j + \frac{\Delta t}{\epsilon^2 + \Delta t r}\nu^2\partial_{xx} j\right) \\ &= \frac{r}{\epsilon^2 + \Delta t r}(M\rho - f_+) + \nu^2\frac{\Delta t}{\epsilon^2 + \Delta t r}\partial_{xx} f_+. \end{aligned} \tag{134}$$

Putting it together one obtains

$$\partial_t f_+ + \nu \frac{\epsilon}{\epsilon^2 + \Delta t r} \partial_x f_+ = \frac{r}{\epsilon^2 + \Delta t r}(M\rho - f_+) + \nu^2 \frac{\Delta t}{\epsilon^2 + \Delta t r} \partial_{xx} f_+. \tag{135}$$

The equation for f_- is obtained in the same manner and thus the resulting system is

$$\begin{cases} \partial_t f_+ + \nu \frac{\epsilon}{\epsilon^2 + \Delta t r} \partial_x f_+ = \frac{r}{\epsilon^2 + \Delta t r}(M\rho - f_+) + \nu^2 \frac{\Delta t}{\epsilon^2 + \Delta t r} \partial_{xx} f_+, \\ \partial_t f_- - \nu \frac{\epsilon}{\epsilon^2 + \Delta t r} \partial_x f_- = \frac{r}{\epsilon^2 + \Delta t r}(M\rho - f_-) + \nu^2 \frac{\Delta t}{\epsilon^2 + \Delta t r} \partial_{xx} f_-, \end{cases} \tag{136}$$

for $\nu > 0$, which is equivalent to

$$\partial_t f + \nu \frac{\epsilon}{\epsilon^2 + \Delta t r} \partial_x f = \frac{r}{\epsilon^2 + \Delta t r}(M\rho - f) + \nu^2 \frac{\Delta t}{\epsilon^2 + \Delta t r} \partial_{xx} f, \quad \nu \in \mathbb{R}. \tag{137}$$

Before proceeding to the simulation of (137), we make an argument for why this model should be appropriate based on 4. Recall that for fixed ϵ, the equation should tend towards (8) as the time step decreases and this is clearly the case. Secondly, (137) should approach (35) as $\epsilon \to 0$ for fixed Δt. The second property requires a bit more work to show, but it is essentially done by following along the lines of Section 3.2. Before deriving the asymptotic result, it is important to note that in this section more restrictive assumptions have been made compared to Section 3.2. Namely, the assumptions of a constant collision rate and a symmetric velocity distribution, which is independent of the particle position. Thus, the second term in (35) vanishes and the last terms becomes the second order derivative of the particle density position multiplied by the variance of the velocity distribution divided by the constant collision rate. As for the assumptions of the differentiability and integrability of the functions involved, the assumptions are the same as in Section 3.2. Based on the aforementioned modifications, the limit of (137) is shown to be (35), by first rewriting the joint density function as

$$f(x, \nu, t) = M(\nu)\rho(x, t) + \epsilon g(x, \nu, t), \tag{138}$$

and substituting it into (137)

$$\partial_t(M\rho + \epsilon g) + \nu \frac{\epsilon}{\epsilon^2 + \Delta t r} \partial_x(M\rho + \epsilon g) = \frac{r}{\epsilon^2 + \Delta t r}(\epsilon g) + \nu^2 \frac{\Delta t}{\epsilon^2 + \Delta t r} \partial_{xx}(M\rho + \epsilon g),$$

where the arguments have been omitted. Next, the equivalent equation system in ρ and g is derived. To obtain an equation in terms of the particle density function, ρ, ν is integrated out

$$< \partial_t M\rho > + < \nu \frac{\epsilon}{\epsilon^2 + \Delta t r} \partial_x(M\rho + \epsilon g) >= \frac{\Delta t}{\epsilon^2 + \Delta t r} < \nu^2 \partial_{xx}(M\rho + \epsilon g) >, \tag{139}$$

where it has been used that $< \epsilon g >= 0$. Further, integration and differentiation is interchanged and the knowledge of the velocity distribution is applied,

$$\partial_t \rho + \frac{\epsilon}{\epsilon^2 + \Delta t r} \partial_x < \nu \epsilon g >= \frac{\Delta t}{\epsilon^2 + \Delta t r} \left(\sigma_\nu^2 \partial_{xx} \rho + \partial_{xx} < \nu^2 \epsilon g > \right). \tag{140}$$

In the limit of $\epsilon \to 0$ one obtains

$$\partial_t \rho = \frac{\sigma_\nu^2}{r} \partial_{xx} \rho. \tag{141}$$

The equation for g is found by applying the operator $(I - \Pi)$, where $\Pi f = M < f >$, to (139), which results in

$$\partial_t g + \frac{1}{\epsilon^2 + \Delta t r} \nu M \partial_x \rho + \frac{\epsilon}{\epsilon^2 + \Delta t r} (I - \Pi) \nu \partial_x g - \frac{\Delta t}{\epsilon(\epsilon^2 + \Delta t r)} (I - \Pi) \nu^2 M \partial_{xx} \rho$$
$$- \frac{\Delta t}{\epsilon^2 + \Delta t r} (I - \Pi) \nu^2 \partial_{xx} g = - \frac{r}{\epsilon^2 + \Delta t r} g, \tag{142}$$

which when considering the dominant terms gives

$$\nu^2 M \partial_{xx} \rho = M \partial_{xx} \rho < \nu^2 M > . \tag{143}$$

This shows that the velocity information is captured by the mean square velocity in the diffusive limit, thereby indicating that the model used in [32] has equivalent properties since the diffusive term vanishes in the limit $\Delta t \to 0$. However, at the Monte Carlo level, one would expect to see a difference between the two models due to the extra variance introduced by accounting for individual velocities in the diffusion term. Therefore, attempts have been made to formally derive the model where the mean square velocity replaces the individual velocity by using the micro-macro decomposition instead of the even-odd decomposition, but without any success. However, the model with the mean square velocity is still compared numerically to the other two methods in Section 6.1 and will be referred to as the asymptotic-preserving splitting method with an altered diffusion coefficient (APSD).

3.4.2 Monte Carlo scheme

In this section, a simulation method for Equation (137) is given, see [11]. The idea is to use a first order splitting, which gives

$$\begin{cases} \partial_t f^{1/2} + \nu \frac{\epsilon}{\epsilon^2 + \Delta t r} \partial_x f^{1/2} - \nu^2 \frac{\Delta t}{\epsilon^2 + \Delta t r} \partial_{xx} f^{1/2} = 0, \\ \partial_t f^{2/2} = \frac{r}{\epsilon^2 + \Delta t r} (M \rho^{2/2} - f^{2/2}), \end{cases} \tag{144}$$

where simulation of the first equation is referred to as the transport step and the simulation of the second equation is referred to as the collision step. The first equation in (144), is simulated as the Fokker-Planck equation in (15) by the same approach as in Section 2.4.2.2. It is clear that in the case of the first equation in (144),

$$a(x,t) = \nu \frac{\epsilon}{\epsilon^2 + \Delta t r}, \tag{145}$$

$$b(x,t) = \nu^2 \frac{\Delta t}{\epsilon^2 + \Delta t r}, \tag{146}$$

and thus the transport step is simulated in the following way:

$$\psi_t(X_n, \mathcal{V}_n, Z_n) = X_{n+1} = X_n + A_n\Delta t + \sqrt{2D_n\Delta t}Z_n, \tag{147}$$

for $n = 1, 2, \ldots, N$, where $A_n = \mathcal{V}_n\frac{\epsilon}{\epsilon^2+\Delta tr}$, $D_n = (\mathcal{V}_n)^2\frac{\Delta t}{\epsilon^2+\Delta tr}$, N is the number of steps, $Z_n \sim N(0,1)$ and $\mathcal{V}_n \sim M(\nu)$. Since no benefit is obtained by solving the collision exactly it is solved via the forward Euler method:

$$\begin{aligned}
&\frac{f_{n+1}^{2/2} - f_{n+1}^{1/2}}{\Delta t} = \frac{r}{\epsilon^2 + \Delta tr}(M\rho_{n+1}^{2/2} - f_{n+1}^{1/2}),\\
\iff &f_{n+1}^{2/2} = f_{n+1}^{1/2} + \frac{\Delta tr}{\epsilon^2 + \Delta tr}(M\rho_{n+1}^{1/2} - f_{n+1}^{1/2}),\\
\iff &f_{n+1}^{2/2} = \frac{\Delta tr}{\epsilon^2 + \Delta tr}M\rho_{n+1}^{1/2} + \frac{\epsilon^2}{\epsilon^2 + \Delta tr}f_{n+1}^{1/2},
\end{aligned} \tag{148}$$

where as explained in Section 2.4.2.2, $\rho_{n+1}^{2/2} = \rho_{n+1}^{1/2}$. Equation (148) may be interpreted in the same way as (20), which means that the collision step corresponds to drawing a new velocity from $M(\nu)$ with a probability of $\frac{\Delta tr}{\epsilon^2+\Delta tr}$. More precisely,

$$\psi_c(\mathcal{V}_n) = \mathcal{V}_{n+1} = \begin{cases} \mathcal{V}^* \sim M(\nu), & \text{with probability } \frac{\Delta tr}{\epsilon^2+\Delta tr},\\ \mathcal{V}_n, & \text{with probability } \frac{\epsilon^2}{\epsilon^2+\Delta tr}, \end{cases} \tag{149}$$

for $n = 0, 1, \ldots, N$. Hence, beyond the first step, the one step method is given by

$$\phi_{APS}(X_n, \Delta t, \xi_n) = X_n + \mathcal{V}_n\frac{\epsilon}{\epsilon^2 + \Delta tr}\Delta t + \sqrt{2(\mathcal{V}_n)^2\frac{\Delta t}{\epsilon^2 + \Delta tr}}\sqrt{\Delta t}Z_n, \tag{150}$$

where $\xi_n = (\mathcal{V}_n, Z_n)^T = (\psi_c(\mathcal{V}_{n-1}), Z_n)^T$ for $n = 1, \ldots, N$. In the very first transport step, no collision step has been performed and the velocity is thus given by the initial conditions, i.e. $\xi_0 = (\mathcal{V}_0, Z_0)^T$. The only difference compared to the model with the mean square velocity is that $\mathcal{V}_n^2$ is replaced by $< \mathcal{V}^2 >$.

3.4.3 Asymptotic-preserving splitting algorithm

Algorithm 5: Asymptotic-preserving splitting simulation

Function $\psi_t(x,\nu,z,\Delta t)$**:**
  $x = x + A\Delta t + \sqrt{2D\Delta t}z$ `// from` (147)
  return x

Function $\psi_c(x,\nu,\Delta t)$**:**
  $u \leftarrow U(0,1)$
  if $u \leq \frac{\Delta tr}{\epsilon^2+\Delta tr}$ **then**
    $\nu \leftarrow M(\nu)$
  return ν

`// One-step function for Asymptotic-preserving splitting scheme`

Function $\phi_{APS}(\Delta t,x,v,t)$**:**
  $x = \psi_t(x,v,z,\Delta t)$
  $\nu = \psi_c(x,\nu,\Delta t)$
  $t = t + \Delta t$
  return x, v, t

`// Asymptotic-preserving splitting simulation in` $[0,T]$

Function `APS`$(\Delta t,\ T)$**:**
  $t = 0$
  $x, v \leftarrow S(x,v)$ `// from` (4)
  while $t + \tau < T$ **do**
    $x, \nu = \phi_{APS}(\Delta t,x,v,t)$
  return x, v

3.5 Numerical example

In this section, the implementation of the two methods is checked against [8, 29]. In the example, Equation (8) with $r(x) = 1$ is solved for different values of ϵ with the following initial conditions:

$$f(x,\nu,t=0) = \frac{1}{2}\left(1 + \cos(2\pi(x+\frac{1}{2}))\right), \quad x \in [0,1], \quad \nu \in [-1,1]. \tag{151}$$

Thus, the velocity is uniformly distributed over $[-1,1]$, i.e. $M(\nu) = \frac{1}{2}, -1 \leq \nu \leq 1$ and $\rho(x,0) = 1 + \cos(2\pi(x+\frac{1}{2}))$. The problem is considered with periodic boundary conditions and solved in the time $[0,0.1]$. For both methods, the chosen time step is $\Delta t = 0.005$. At the Monte

Carlo level, the initial conditions represent distributions from which positions and velocities are drawn at time 0. It is straight forward to draw velocities from $M(\nu)$ and to draw initial positions from $\rho(x, 0)$ the accept-reject method has been used. Further elaborations on the accept-reject method in general and in relation to this example are given in Appendix B. As for dealing with the boundary conditions, the position is simply modified after each change by taking it modulo 1, i.e. the position is always given by $X = X \mod 1$. The results in Figure 6 coincide well with the ones in [8, 29], which indicates a consistent implementation of the asymptotic-preserving Monte Carlo methods.

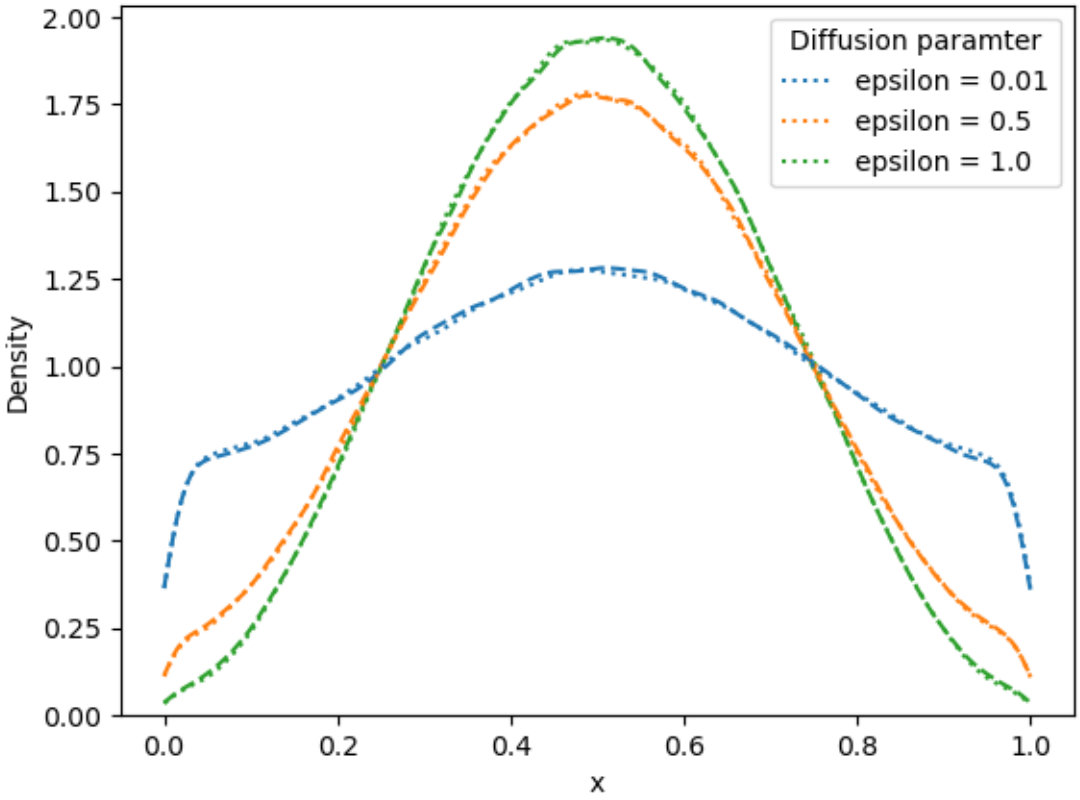


Figure 6: The estimated $\rho(x, 0.1)$ for different values of ϵ using 400,000 paths. The kinetic-diffusion scheme is represented by (·····) and the splitting scheme is represented by (- - - -).

4 Multilevel approaches

This section presents the two main methods of this work, namely the multilevel versions of the methods presented in Sections 3.3 and 3.4. In Section 4.1, the general Multilevel Monte Carlo approach is outlined and then in Sections 4.3 and 4.2 the two main methods are presented.

4.1 A general outline of the Multilevel Monte Carlo method

In the standard Monte Carlo method, one is usually interested in some quantity, which is given as some function of the stochastic process $(X_t)_{t\in\mathbb{R}^+}$, at time T, where

$$dX = a(X,t)dt + b(X,t)dW_t, \tag{152}$$

and W_t denotes the Wiener process. Let $I^{\Delta t} = \{t_0, t_1, \ldots, t_K\}$ with $t_0 < t_1 < \cdots < t_K = T$ and $t_{k+1} - t_k = \Delta t$ be a discretisation of the time interval $I = [t_0, T]$ and denote the approximation of $X(t_k)$ by $Y_k = Y(t_k)$. The sequence $(Y(t))_{t\in I^{\Delta t}}$ is then given by

$$\begin{aligned} Y_0 &= X_0, \\ Y_{k+1} &= \phi(Y_k, \Delta t, \xi_k), \end{aligned}$$

where $\phi(\cdot)$ is a one-step method for solving (152) and ξ_k for $k = 0, 1, 2, \ldots, K$ are random vectors s.t. ξ_0 is independent of Y_0 and for $k \geq 1$, ξ_k is independent of $Y_0, Y_1, \ldots, Y_n$ and $\xi_0, \xi_1, \ldots, \xi_{k-1}$. The quantity of interest, $E(F(X))$, is then estimated by

$$E(F(X)) \approx E(F(Y^{\Delta t})) \approx \frac{1}{N}\sum_{i=1}^{N} F(Y^{\Delta t,i}),$$

where $X = X(T)$, $Y^{\Delta t}$ is the approximation of $X(T)$ using step size Δt and $Y^{\Delta t}(\omega_i) = Y^{\Delta t,i}$ for the i'th path.

The goal of the multilevel approach presented in [15], is then to estimate the same quantity, but with lower computational costs. This is done by setting the simulation process up in a hierarchical manner based on step sizes. To understand the hierarchy of simulations consider the geometric series

$$E(F(Y^{\Delta t_J})) = E(F(Y^{\Delta t_0})) + \sum_{j=1}^{J} E(F(Y^{\Delta t_j}) - F(Y^{\Delta t_{j-1}})), \tag{153}$$

which is approximated by

$$E(F(Y^{\Delta t_J})) \approx \hat{Q} = \frac{1}{N_0}\sum_{i=1}^{N_0} F(Y^{\Delta t_0,i}) + \sum_{j=1}^{J}\sum_{i=1}^{N_j} \frac{F(Y^{\Delta t_j,i}) - F(Y^{\Delta t_{j-1},i})}{N_j}, \tag{154}$$

where Δt_j and Δt_{j-1} are the step sizes of the l_j'th and l_{j-1}'th level, respectively, and $l_J = L$. Usually the step sizes are defined in general as $\Delta t_j = \frac{T-t_0}{M^l}$, where $M \geq 2$ and $l = 0, 1, \ldots, L$. Thus, at the first level the quantity is estimated by only taking 1 step and then the error is corrected by adding the estimated bias from the finer levels. It is important to note that the overlapping estimates between levels are simulated independently, i.e. the first sum in Equation (154) is independent of the second term and the estimated bias is also independent across levels.

However, the estimates used to calculate a single bias estimate are calculated using the same paths and thus only differ due to the difference in step size. This is the main challenge when setting up multilevel approaches for the methods in Sections 3.3 and 3.4 is to correlate the paths appropriately. For later reference, the main Theorem in the theory of Multilevel Monte Carlo is stated below, see [14].

Theorem 2. *Let P denote $F(X(T))$, where $X(T)$ is the solution of (152) and $\hat{P}_l$ denote the approximation of P using a time step $\Delta t_l = \frac{T-t_0}{M^l}$, i.e. $\hat{P}_l = F(Y^{\Delta t_l})$. If there exists independent estimators $\hat{S}_l$ based on N_l Monte Carlo samples and positive constants $\alpha \geq \frac{1}{2}, \beta, \gamma, c_1, c_2, c_3$ s.t.*

$$
\begin{aligned}
&(i) \quad E\left[\hat{P}_l - P\right] \leq c_1 \Delta t_l^{\alpha}, \\
&(ii) \quad E[\hat{S}_l] = \begin{cases} E[\hat{P}_0], & l = 0, \\ E[\hat{P}_l - \hat{P}_{l-1}], & l > 0, \end{cases} \\
&(iii) \quad V_l \leq c_2 \Delta t_l^{\beta}, \quad V_l = V\left[\hat{P}_l - \hat{P}_{l-1}\right], \quad \hat{P}_{-1} = 0 \\
&(iv) \quad C_l \leq c_3 \Delta t_l^{-\gamma}, \quad \textit{where } C_l \textit{ is the computational complexity of } \hat{S}_l
\end{aligned}
$$

Then there exists a positive constant c_4 s.t. for any $E < e^{-1}$ there are values L and N_l for which $\hat{Q}$ has a mean square error with bound

$$MSE(\hat{Q}) = E((\hat{Q} - E(F(X)))^2) < E^2,$$

with a complexity, C, with bound

$$C \leq \begin{cases} c_4 E^{-2}, & \beta > \gamma, \\ c_4 \frac{log_e(E)^2}{E^2}, & \beta = \gamma, \\ c_4 E^{-2-(\gamma-\beta)/\alpha}, & 0 < \beta < \gamma. \end{cases}$$

A relevant takeaway point from the Theorem in relation to the methods presented in this work is that on a logarithmic scale both the bias and the variance must decrease when the step size decreases. For the methods in this work, the behaviour of the variance and the bias turns out to be non-linear on a logarithmic scale and therefore, the choice of levels require some attention to obtain results that resemble those assumed by the Theorem.

To better understand how the MSE is dealt with in practice note that

$$
\begin{aligned}
MSE(\hat{Q}) &= E((\hat{Q} - E(F(X)) + E(\hat{Q}) - E(\hat{Q}))^2) \qquad (155) \\
&= E((\hat{Q} - E(\hat{Q}))^2 + (E(\hat{Q}) - E(F(X)))^2 + 2(\hat{Q} - E(\hat{Q}))(E(\hat{Q}) - E(F(X)))) \\
&= \underbrace{E((\hat{Q} - E(\hat{Q}))^2)}_{variance} + \left(\underbrace{E(\hat{Q} - F(X))}_{bias}\right)^2.
\end{aligned}
$$

Thus, the error is bounded by E^2 if the following conditions are satisfied,

$$V[\hat{Q}] = \frac{V_{l_0}}{N_0} + \sum_{j=1}^{J} \frac{V_{l_j,l_{j-1}}}{N_j} \leq \frac{E^2}{2}, \tag{156}$$

$$E(\hat{Q} - F(X)) \leq \frac{E}{\sqrt{2}}, \tag{157}$$

where $V_{l_0} = V[\hat{P}_0]$ and $V_{l_j,l_{j-1}} = V[F(Y^{\Delta t_j}) - F(Y^{\Delta t_{j-1}}]$. The goal is now to minimise the cost of the computational costs while still satisfying Equations (156) and (157). The total computation cost for the multilevel estimator given in Equation (154) is given by

$$C = N_0 C_0 + \sum_{j=1}^{J} N_j C_{j,j-1}, \tag{158}$$

where C_0 and $C_{j,j-1}$ are the cost of one simulation at levels l_0 and l_j, respectively. Based on Equation (158), it is evident that in general, there are two components that can be manipulated in order minimise the cost. These are the set of levels, i.e. $\{l_0, l_1, \dots L\}$ and the number of paths at each level. The set of levels are chosen based on the specific method in order to adhere with Theorem 2 and the final level, L, is chosen s.t. (157) is fulfilled. In this work, the bias criteria is tested by checking if

$$\max_{j \in \{J-2, J-1, J\}} \left(M^{-\alpha(J-j)} |\hat{P}_j - \hat{P}_{j-1}| \frac{1}{M^\alpha - 1} \right) \leq \frac{E}{\sqrt{2}}(M-1), \tag{159}$$

which is suggested in [14]. The optimal number of paths, given a set of levels, may be found in general for both methods by minimising Equation (158) given the condition in Equation (156). Considering the number of paths as continuous random variables, the problem is solved by use of Lagrange Multipliers, see [1, §13.4], and is given by

$$\underset{N_0, N_1, \dots, N_J}{\operatorname{argmin}} \underbrace{N_0 C_0 + \sum_{j=1}^{J} N_j C_{j,j-1} + \mu \left(\frac{V_{l_0}}{N_0} + \sum_{j=1}^{J} \frac{V_{l_j,l_{j-1}}}{N_j} - \frac{E^2}{2} \right)}_{= L(N_0, N_1, \dots, N_J, \mu)}, \tag{160}$$

where $L(N_0, N_1, \dots, N_J, \mu)$ is the Lagrangian and μ is the Lagrange multiplier. To find the optimal number of paths one must compute the zero-gradient of the Lagrangian, but first it is necessary to confirm that the zero-gradient corresponds to a minimum. This is done by first calculating the Hessian of the Lagrangian w.r.t. the number of samples at each level and then evaluating it in the critical point to see that it is positive definite. Calculating the Hessian gives

$$HL(N_0, N_1, \dots, N_J) = \begin{pmatrix} \frac{2\mu V_{l_0}}{N_0^3} & 0 & 0 & \cdots & 0 \\ 0 & \frac{2\mu V_{l_1,l_0}}{N_1^3} & 0 & \cdots & 0 \\ \vdots & 0 & \ddots & \cdots & 0 \\ 0 & \cdots & 0 & \frac{2\mu V_{l_{J-1},l_{J-2}}}{N_{J-1}^3} & 0 \\ 0 & \cdots & \cdots & 0 & \frac{2\mu V_{l_J,l_J}}{N_J^3} \end{pmatrix}. \tag{161}$$

It is clear that since the Hessian is diagonal and the variance and number of paths are always positive, it suffices to show that $\mu \geq 0$ which is done by actually solving the minimisation problem (160).

The gradient of the Lagrangian is given by

$$\nabla L = \begin{pmatrix} C_0 - \mu \frac{V_{l_0}}{N_0^2} \\ C_{1,0} - \mu \frac{V_{l_1,l_0}}{N_1^2} \\ \vdots \\ C_{L,L-1} - \mu \frac{V_{l_J,l_{J-1}}}{N_J^2} \\ \frac{V_{l_0}}{N_0} + \sum_{j=1}^{J} \frac{V_{l_j,l_{j-1}}}{N_j} - \frac{E^2}{2} \end{pmatrix} \tag{162}$$

and setting it equal to zero yields the system

$$C_0 = \mu \frac{V_{l_0}}{N_0^2}, \tag{163}$$

$$C_{j,j-1} = \mu \frac{V_{l_j,l_{j-1}}}{N_j^2} \quad \textit{for } j = 1, 2, \ldots, J \tag{164}$$

$$0 = \frac{V_{l_0}}{N_0} + \sum_{j=1}^{J} \frac{V_{l_j,l_{j-1}}}{N_j} - \frac{E^2}{2}. \tag{165}$$

Solving for N_0 and N_l in Equations (163) and (164) then yields

$$N_0 = \sqrt{\mu \frac{V_{l_0}}{C_0}}, \tag{166}$$

$$N_j = \sqrt{\mu \frac{V_{l_j,l_{j-1}}}{C_{j,j-1}}}, \quad \textit{for } j = 1, 2, \ldots, J, \tag{167}$$

which is substituted into Equation (165) gives

$$0 = \frac{V_{l_0}\sqrt{C_0}}{\sqrt{\mu V_{l_0}}} + \sum_{j=1}^{J} \frac{V_{l_j,l_{j-1}}\sqrt{C_{j,j-1}}}{\sqrt{\mu V_{l_j,l_{j-1}}}} - \frac{E^2}{2} \tag{168}$$

$$\iff \frac{E^2}{2}\sqrt{\mu} = \sqrt{V_{l_0}C_0} + \sum_{j=1}^{J} \sqrt{V_{l_j,l_{j-1}}C_{j,j-1}}$$

$$\iff \sqrt{\mu} = \frac{2}{E^2}\left(\sqrt{V_{l_0}C_0} + \sum_{j=1}^{J} \sqrt{V_{l_j,l_{j-1}}C_{j,j-1}}\right). \tag{169}$$

Since $\mu \geq 0$ the optimal number of paths produce a minimum and they are given by

$$N_0 = \frac{2}{E^2}\sqrt{\frac{V_{l_0}}{C_0}}\left(\sqrt{V_{l_0}C_0} + \sum_{j=1}^{J} \sqrt{V_{l_j,l_{j-1}}C_{j,j-1}}\right) \tag{170}$$

$$N_j = \frac{2}{E^2}\sqrt{\frac{V_{l_j,l_{j-1}}}{C_{j,j-1}}}\left(\sqrt{V_{l_0}C_0} + \sum_{j=1}^{J} \sqrt{V_{l_j,l_{j-1}}C_{j,j-1}}\right), \quad \textit{for } j = 1, 2, \ldots, J. \tag{171}$$

In practice the optimal values are rounded up to the nearest integer. Recalling Theorem 2, Equations (170) and (171) show that in the case $\beta > \gamma$, the number of paths will be lower for finer levels and higher for more coarse levels. In the case $\beta = \gamma$, the work is distributed evenly across the levels while the worst case scenario is $\beta < \gamma$ because then most of the work is done on the finer paths. The methods in this work need to deal with two challenges. The first is to determine the optimal set of levels that satisfy Equations (156) and (157). The second is to deal with the problem of correlating the paths used in the estimation of the bias at each level s.t. the estimates only differ due to the difference in step sizes. In [14], it is suggested to test the consistency of the correlation, i.e. that $E(F(Y^{\Delta t_j})) = E(F(Y^{\Delta t_{j-1}}))$ for correlated paths, using the fact that by the Central Limit Theorem, [23] the sample means are approximately normally distributed. Thus, if the simulations are consistent then it should be the case that $a - b + c \approx 0$ where a is the sample mean of $E(F(Y^{\Delta t_j}) - F(Y^{\Delta t_{j-1}}))$ for correlated paths, b is the sample mean $E(F(Y^{\Delta t_j}))$ and c is the sample mean of $E(F(Y^{\Delta t_{j-1}}))$ with the paths from b c being independent from each other and the paths used in a. Due to the approximate normality of the sample means

$$\frac{a - b + c}{\sqrt{V[a - b + c]}} \sim N(0, 1), \tag{172}$$

and since $\sqrt{V[a - b + c]} \leq \sqrt{V[a]} + \sqrt{V[b]} + \sqrt{V[c]}$, the probability that

$$\frac{|a - b + c|}{3(\sqrt{V[a]} + \sqrt{V[b]} + \sqrt{V[c]})} \geq 1 \tag{173}$$

is less than 0.3%. Therefore, part of testing the multilevel procedure is to ensure that values above unity are not observed. If high values are observed, it is either because too few samples or it is a strong indication that either the mathematical formulation of the correlation method or its implementation is faulty.

4.2 Multilevel Kinetic-Diffusion method

This section covers the multilevel version, presented in [38], of the method in Section 3.3. It will be referred to as the KDML method. The section consists of three main parts. In Section 4.2.1 an algorithm to simulate correlated paths is derived and then in Section 4.2.2 some of the numerical results from [38] are replicated to motivate the algorithm for the level-selection in Section 4.2.3.

4.2.1 Correlating paths

This section presents an algorithm for correlating paths between two levels, l_j and l_{j-1}, given by their step sizes Δt_j and Δt_{j-1}. The former will be referred to as the fine path and the

latter as the coarse path, moreover the variables belonging to the fine or coarse path will be superscripted with an f or c, respectively. The considerations in this section are based on one path and are simply reproduced independently for multiple paths.

In the standard multilevel case presented in [15], the correlation of the paths is straight forward. It simply involves the correlation of two Brownian motions with different step sizes. However, by the one-step method given in Equation (41) it is evident that each step involves the generation of three types of random variables. Thus, when correlating the paths, the task is to correlate the set of r.v. $\pi^f = \{\xi_k^f\}_{k=0}^{K^f}$ and $\pi^c = \{\xi_k^c\}_{k=0}^{K^c}$ used at the fine and coarse level, respectively. This is done by first generating π^f and then map those random numbers to π^c. The appropriate mapping is determined by the combination of two phases, the mapping phase and the aggregation phase. In the mapping phase it is determined which random numbers from the fine path should be used in the calculation of each random number in the coarse path. This is described in Section 4.2.1.1. Due to the difference in step size, it is expected that the number of random numbers for the fine path is greater than that for the coarse level, i.e. $K^f > K^c$. So, each random number in the coarse path is a function of one or more random numbers from the fine path. The function for each type of random variable is determined in the aggregation phase described in Section 4.2.1.2. In this section it is assumed that the velocity, v_n in Equation (41), follows a normal distribution and may thus be written as

$$v_n = \mu(x_n) + \sigma_v(x_n)\nu_n, \tag{174}$$

where $\nu_n \sim N(0,1)$. This is somewhat abuse of notation, since in Section 3.3, ν is referred to as the diffusively scaled velocity. However, in this section the diffusive scaling parameter is hidden in the properties of the velocity distribution and the collision rate as it is already dealt with at the Monte Carlo level and does therefore not affect the approach in this section.

4.2.1.1 Mapping phase

The main idea of the mapping phase is that the random numbers of the fine path that correspond to the time span of a coarse step should be mapped to the random numbers of that step. In addition, the random numbers should be mapped s.t. they affect the same part of the time span. In other words, if a random number had an effect on the initial kinetic part of the fine step, then it should also affect the initial kinetic part of the coarse step. The ideal case is illustrated in the scenario (a) in Figure 7. The mapping phase is best understood by focusing on the kinetic part of the coarse movement in each step. Let κ_n be the index of the fine step corresponding to the n'th step of the coarse path. See for instance scenario (a) in Figure 7. Here, $\kappa_1 = 2$ because the kinetic phase of step 2 of the fine path is mapped to the kinetic phase

of step 1 of the coarse path. The rest of the variables in the time span $[0, \Delta t^c]$ are mapped to the diffusive phase and are given by the following sets $\{z_m^f\}_{m=\kappa_0}^{\kappa_1-1}$ and $\{(\nu_m^f, e_m^f)\}_{m=\kappa_0+1}^{\kappa_1-1}$. In general it is to be expected that some steps will not contain any collisions and hence make the mapping a bit more difficult. In scenario (b) of Figure 7, the first fine step does not contain any collisions and the particle is therefore moved beyond the time of the first coarse step. The mapping is then chosen to be s.t. the kinetic movement of the second coarse step makes use of the same random numbers as the kinetic movement in the corresponding time span of the fine path. Thus, $\kappa_1 = 1$ because step 1 of the fine path moves beyond step 0 of the coarse path. It might also be the case that the coarse path is in a lower collisional regime compared to the fine path. Then some of the coarse steps might not contain collisions even though collisions have occurred for the fine path in the same time interval. This scenario is illustrated in part (c) of Figure 7. As is evident from the figure, the first coarse step actually covers two coarse step sizes because the first does not contain a collision. Due to this, it takes 4 fine steps to reach a time point where the next step moves beyond the first coarse step, i.e. moves beyond $2\Delta t^c$. Hence, $\kappa_0 = 0$ and the rest of the random numbers are mapped to the diffusive phase of the first coarse step. This scenario shows that the aggregation phase must allow the inclusion of random numbers from multiple kinetic phases in the calculation of z^c in any given coarse step. Lastly, a case of strong decorrelation is shown in scenario (d) of Figure 7. In this case we see that if one only uses random numbers from fine steps that do not move beyond the time of the coarse step of interest, then the first coarse step would not have any numbers mapped to it. Thus, it is necessary to use the random numbers from minimum 1 fine step even though that step moves beyond the coarse step. Therefore, the random numbers of the next fine step are always used in the next coarse step and then the usage of random numbers from further fine steps depends on whether those move beyond the next coarse step. Scenario (d) also shows that it might be necessary to move the fine particle further than the end time to obtain the necessary random numbers for the coarse step. To see this, consider $4\Delta t^f$ as the end time. The fine path must move until $5\Delta t^f$ to obtain enough random numbers for the second coarse step.

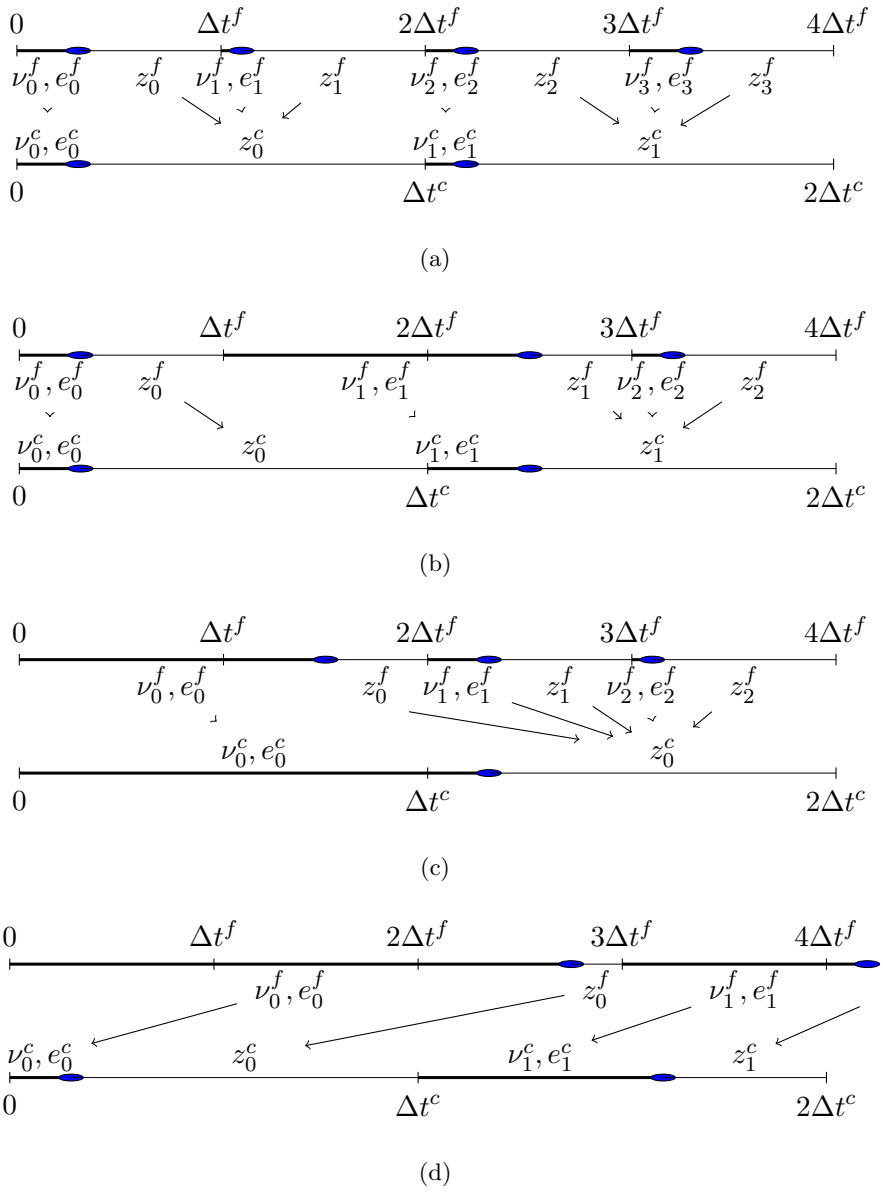


Figure 7: Scenarios for mapping random numbers from fine path to coarse path. Taken from [38].

The mapping phase may be summarised as follows: Determine the values $\{\kappa_n\}_{n=0}^{K^f}$ by

$$\kappa_n = \begin{cases} 0, & n = 0, \\ \max(\kappa_{n-1} + 1, \min(k | t_{k+1}^f > t_n^c)), & else, \end{cases} \tag{175}$$

where t_{k+1}^f is the time after $k+1$ steps for the fine path. In practice, κ_{n+1} is determined after the n'th coarse step and $\kappa_{n+1} - \kappa_n$ fine steps are performed to generate the random numbers needed for the $(n+1)$'th coarse step. The random numbers are then mapped in the following

way:

$$\nu^f_{\kappa_n} \to \nu^c_n, \tag{176}$$

$$e^f_{\kappa_n} \to e^c_n, \tag{177}$$

$$\{(\nu^f_m, e^f_m)\}_{m=\kappa_n+1}^{\kappa_{n+1}-1} \cup \{z^f_m\}_{m=\kappa_n}^{\kappa_{n+1}-1} \to z^c_n. \tag{178}$$

4.2.1.2 Aggregation phase

In Section 4.2.1.1, the mapping phase is outlined and based on the mapping given in Equations (176), (177) and (178), it is necessary to determine transformations for each mapping that preserve the correct distribution. Based on the mapping it makes sense to simply set

$$\nu^f_{\kappa_n} = \nu^c_n, \tag{179}$$

as the velocity of the kinetic phase in the fine step should correspond to the velocity of the kinetic phase in the coarse step. For the mapping in Equation (177), the case is a bit different. We have that

$$\varepsilon^f_{\kappa_n} = \int_0^{\Delta\tau^f_n} R(x^f_n + v^f_n t)dt, \tag{180}$$

and in scenario (b) of Figure 7 one may think of $\Delta\tau^f_1$ as the length of time s.t. the number of collisions in $[\Delta t^f, \Delta t^f + \Delta\tau^f_1]$ equals ε^f_1. However, the coarse path starts at a later point in time so it makes sense to transform ε^f_1 by subtracting the number of collisions in $[\Delta t^f, \Delta t^c]$. Hence, the transformed coarse variable becomes

$$\varepsilon^c_n = \varepsilon^f_{\kappa_n} - \underbrace{\int_0^{t^c_n - t^f_{\kappa_n}} R(x^f_n + v^f_n t)dt}_{a:=}, \tag{181}$$

which is still exponentially distributed because of the memory-less property of the exponential distribution. To see this, note that by the definition of a it is given that $\varepsilon^f_{\kappa_n} > a$ and therefore the following holds

$$P(\varepsilon^c_n > b) = P(\varepsilon^f_{\kappa_n} > a + b | \varepsilon^f_{\kappa_n} > a) = P(\varepsilon^f_{\kappa_n} > b), \tag{182}$$

where the last equality follows from the memory-less property of the exponential distribution. Lastly, it is necessary to define an aggregation of the mapping in Equation (178). This is done by defining the aggregated variable, z^c_n, as a weighted sum of the independent and normally distributed random variables $\{(\nu^f_m)\}_{m=\kappa_n+1}^{\kappa_{n+1}-1}$ and $\{z^f_m\}_{m=\kappa_n}^{\kappa_{n+1}}$. It is well known that this produces a new normal random variable and since z^c_n should be standard normally distributed, a normalising

constant is added. Thus, the aggregated variable for the diffusive phase of the coarse step is defined to be:

$$z_n^c = \frac{\beta_{\kappa_n} z_{\kappa_n}^f + \sum_{m=\kappa_n+1}^{\kappa_{n+1}-1} (\alpha_m \nu_m^f + \beta_m z_m^f)}{\sqrt{\beta_{\kappa_n}^2 + \sum_{m=\kappa_n+1}^{\kappa_{n+1}-1} (\alpha_m^2 + \beta_m^2)}}. \tag{183}$$

The expectation and variance are easily verified to be respectively 0 and 1:

$$E(z_n^c) = \frac{1}{\sqrt{\beta_{\kappa_n}^2 + \sum_{m=\kappa_n+1}^{\kappa_{n+1}-1} (\alpha_m^2 + \beta_m^2)}} \left(\beta_{\kappa_n} \underbrace{E(z_{\kappa_n}^f)}_{=0} + \sum_{m=\kappa_n+1}^{\kappa_{n+1}-1} (\alpha_m \underbrace{E(\nu_m^f)}_{=0} + \beta_m \underbrace{E(z_m^f)}_{=0}) \right) = 0,$$

$$Var(z_n^c) = \frac{1}{\beta_{\kappa_n}^2 + \sum_{m=\kappa_n+1}^{\kappa_{n+1}-1} (\alpha_m^2 + \beta_m^2)} \left(\beta_{\kappa_n}^2 \underbrace{Var(z_{\kappa_n}^f)}_{=1} + \sum_{m=\kappa_n+1}^{\kappa_{n+1}-1} (\alpha_m^2 \underbrace{Var(\nu_m^f)}_{=1} + \beta_m^2 \underbrace{Var(z_m^f)}_{=1}) \right) = 1.$$

The remaining variables, $\{(e_m^f)\}_{m=\kappa_n+1}^{\kappa_{n+1}-1}$, from the mapping in Equation (178), are used implicitly in the definition of the weights, through their effect on the collision times. The weights are chosen s.t. that they intuitively correspond to the impact that ν_m^f and z_m^f have on the fine positional increment corresponding to the coarse diffusive motion given by Equation (183). The impact of the random variables is the amount of randomness in the positional increment caused by those variables. To understand this, consider the extreme case where the positional increment corresponding to a certain random variable is almost constant then the randomness provided by that variable has almost no impact on the positional increment. On the other hand, if the positional increment corresponding to that same variable instead varies a lot then the randomness of that variable must have a greater impact. With this in mind, the α_m weights are chosen to be the standard deviation of the positional increment when moving the particle kinetically during the time that ν_m^f has an impact. From Section 2.4.2.1 such an increment is given by

$$\Delta x_m^f = v_m^f \tilde{\tau}_m^f, \tag{184}$$

where $\tilde{\tau}_m^f$ is the time during which ν_m^f has an impact. The standard deviation of (184) is

$$\alpha_m = \sqrt{V\left[\Delta X_m^f | \tilde{\mathcal{T}}_m^f = \tilde{\tau}_m^f\right]} = \sigma \tilde{\tau}_m^f. \tag{185}$$

The time during which ν_m^f has an impact is best found by considering ν_1^f in scenario (a) of Figure 7. It clearly has an impact during the time τ_1^f where the fine particle moves kinetically, but it also has an impact on the diffusive movement given by z_0^f. This is due to the fact that the

advection and diffusion coefficients in (109) and (108), respectively, are found by conditioning on the last velocity being known from the kinetic part of the next step. By (52) one diffusive motion is a collection of kinetic motions and therefore ν_1^f has an effect in the time from the last collision to the end of the time step. Thus, the time, $\tilde{\tau}_m^f$, that ν_{m+1}^f is expected to have an impact on is

$$\begin{aligned}\tilde{\tau}_{m+1}^f &= E\left[\mathcal{T}_{N(\Delta t)}\right] + \tau_{m+1}^f \\ &= \frac{\epsilon^2}{r(x_m''^f)}(1 - e^{\frac{-r(x_m''^f)}{\epsilon^2}\theta_m^f}) + \tau_{m+1}^f. \end{aligned} \tag{186}$$

The time added in the impact of ν_{m+1}^f then also needs to be subtracted from the time during which z_m^f has an impact. Let

$$\tilde{\theta}_m^f = \theta_m^f - \frac{\epsilon^2}{r(x_m''^f)}(1 - e^{\frac{-r(x_m''^f)}{\epsilon^2}\theta_m^f}), \tag{187}$$

where $\tilde{\theta}_m^f$ is the time during which z_m^f has an impact and θ_m^f is the remaining time after the kinetic part of step m. In accordance with the α_m weights, the β_m weights are defined to be the standard deviation of the increment of a kinetically moved particle during the time $\tilde{\theta}_m^f$. The only difference is that z_m^f corresponds to a diffusive motion which in essence is a collection of kinetic steps and thus, by Equation (65),

$$\beta_m = \sqrt{V\left[\Delta X_{\tilde{\theta}_m^f}\right]} = \sqrt{2\sigma^2 \frac{1}{R^2}\left(R\tilde{\theta}_m^f - 1 + e^{-R\tilde{\theta}_m^f}\right)}. \tag{188}$$

4.2.1.3 Algorithm for correlating paths

Algorithm 6: Simulation of correlated paths in $[0, T]$

Function `correlated(`Δt^f,Δt^c, T`):`

- $k^c = 0$, $k^f = 0$
- $t^f_{k^f} = 0$, $t^c_{k^c} = 0$
- $x^f_{k^f}, v^f_{k^f} \leftarrow S(x, v)$ `// from` (4)
- $x^c_{k^c} = x^f_{k^f}$, $v^c_{k^c} = v^f_{k^f}$
- $\nu^f_{k^f} = \frac{v^f_{k^f} - \mu(x^f_{k^f})}{\sigma(x^f_{k^f})}$, $\nu^c_{k^c} = \nu^f_{k^f}$
- $\varepsilon^f_{k^f} \leftarrow Exp(1)$
- $\varepsilon^c_{k^c} = \varepsilon^f_{k^f}$
- $\tau^f_{k^f} = \tau^c_{k^c} =$ `SampleCollision(`$x^f_{k^f}, v^f_{k^f}, \varepsilon^f_{k^f}$`)` `// see Algorithm 1`
- **while** $t^f_{k^f} + \tau^f_{k^f} < T$ *or* $t^c_{k^c} + \tau^c_{k^c} < T$ **do**
  - `// repeat until the fine path covers the next coarse step`
  - **repeat**
    - $z^f_{k^f} \leftarrow N(0,1)$, $\nu^f_{k^f+1} \leftarrow N(0,1)$
    - $x^f_{k^f+1}, v^f_{k^f+1}, t^f_{k^f+1}, \theta^f_{k^f} =$`\phi_KD(`$\Delta t^f$,$x^f_{k^f}$,$v^f_{k^f}$,$t^f_{k^f}$,$\tau^f_{k^f}$, $\nu^f_{k^f+1}$, $z^f_{k^f}$`)`
    - $\varepsilon^f_{k^f+1} \leftarrow Exp(1)$
    - $\tau^f_{k^f+1} =$ `SampleCollision(`$x^f_{k^f+1}, v^f_{k^f+1}, \varepsilon^f_{k^f+1}$`)`
    - $k^f = k^f + 1$
  - **until** $\lceil (t^f_{k^f} + \tau^f_{k^f})/\Delta t^f \rceil \Delta t^f > \lceil (t^c_{k^c} + \tau^c_{k^c})/\Delta t^c \rceil \Delta t^c$
  - **if** $t^c_{k^c} + \tau^c_{k^c} < T$ **then**
    - $\nu^c_{k^c+1} = \nu^f_{k^f}$, $\varepsilon^c_{k^c+1} \leftarrow$(181), $z^c_{k^c} \leftarrow$ (183)
    - $x^c_{k^c+1}, v^f_{k^c+1}, t^c_{k^c+1}, \theta^c_{k^c} =$`\phi_KD(`$\Delta t^c$,$x^c_{k^c}$,$v^c_{k^c}$,$t^c_{k^c}$,$\tau^c_{k^c}$,, $\nu^c_{k^c+1}$, $z^c_{k^c}$`)`
    - $\tau^c_{k^c+1} =$ `SampleCollision(`$x^c_{k^c+1}, v^c_{k^c+1}, \varepsilon^c_{k^c+1}$`)`
    - $k^c = k^c + 1$
- $k^f = \min(k^f | t^f_{k^f} + \tau^f_{k^f} \geq T)$ `// remove steps beyond` T
- **if** $t^f_{k^f} < T$ **then**
  - $x^f_{k^f} = x^f_{k^f} + v^f_{k^f}(T - t^f_{k^f})$ `// particle is moved to the end`
- **if** $t^c_{k^c} < T$ **then**
  - $x^c_{k^c} = x^c_{k^c} + v^c_{k^c}(T - t^c_{k^c})$ `// particle is moved to the end`
- **return** $x^f_{k^f}, x^c_{k^c}$

4.2.2 Numerical results

In this section, the behaviour of the variance is presented and shown numerically to not be monotone as in the standard case. Instead, it is highly non-trivial and calls for a method to choose which levels to simulate. This is illustrated for the test-case where $\mu(x) = 0$, $\sigma^2(x) = 1$, $x(0) = 1$ with $v(x) \sim M(v)$ and

$$R(x) = \begin{cases} -b(a(x-1)-1), & x \leq 1, \\ b(a(x-1)+1), & x > 1. \end{cases} \tag{189}$$

The case where $a = 0$ is referred to as the homogeneous case and the case where $a >> 0$ is referred to as the heterogeneous case.

With the parameters above, the following results for the variance are obtained:

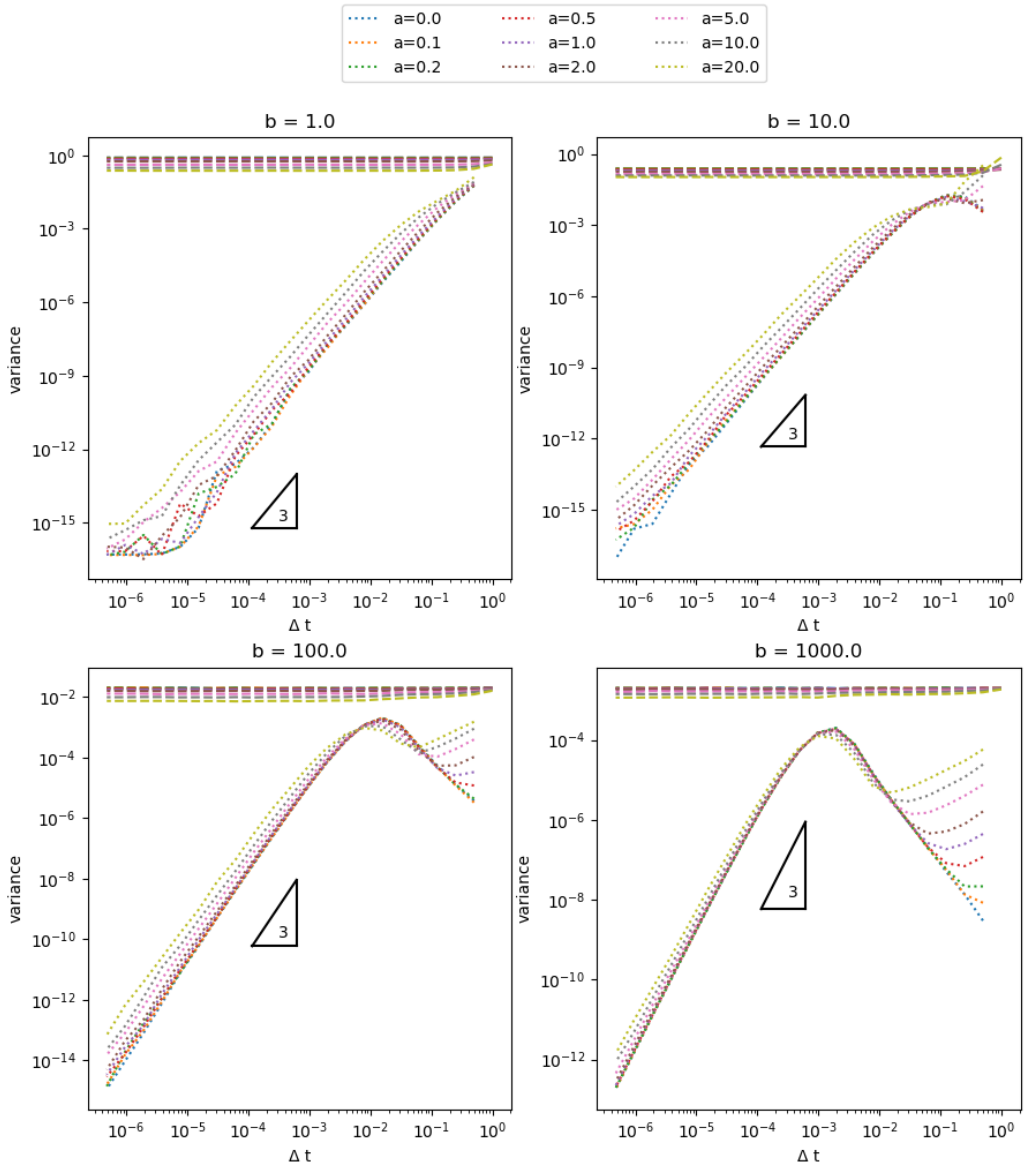


Figure 8: Behaviour of variance for standard KD method(- - - -) and multilevel KD method(·····).

4.2.3 Level-selection

In [38], it is noted that the level-selection may be done via linear programming that aims to minimize $\sqrt{V_{l_0} C_0} + \sum_{j=1}^{J} \sqrt{V_{l_j, l_{j-1}} C_{j,j-1}}$ from (170). However, it assumes the estimation of the variances of each possible level is known, which is a huge drawback. Instead it is suggested to use a heuristic method that chooses the levels, s.t. the variance of the chosen levels is monotone in the sense of Figure 8. For the homogeneous background this amounts to choosing the first level and then choose the next level if the decay in variance is sufficient. In this regard, sufficient

decay is a 50% decrease. In the heterogeneous case the first level is the level for which the standard method yields a higher variance than when simulating correlated paths. Then, the subsequent levels are again chosen based on a sufficient decay in variance. Based on the case $a = 10$, $b = 1000$, the heuristic method is represented visually in Figure 9.

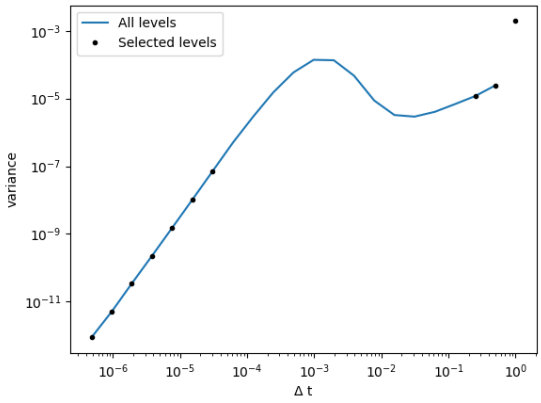


Figure 9: Visual representation of the level selection for $a = 10$ and $b = 1000$.

Like the exact optimisation, the heuristic method also assumes the knowledge of the variance of the level, but not all of them. It only assumes the knowledge of values for adjacent levels and as long as the monotonicity is properly approximated, the precision of the variances is not relevant. This leads to the suggestion of a warm-up scheme where only a few paths are used to approximate the values at adjacent levels and then the proper levels are chosen. The results from the warm-up scheme can then be used to "hot-start" the multilevel method. There are several ways to do the warm-up. The naive approach is to the simply simulate all levels both for correlated and single paths and use those results. Alternatively, it is suggested to use the asymptotic behaviour of the variance as the step size approaches zero. For all backgrounds shown in Figure 8 it is the case that after a certain point, approximately $\Delta t \leq \frac{1}{R(x(0))}$, the variance behaves as $\mathcal{O}(\Delta t^3)$. Hence, one could use this asymptotic approximation for some of the finer levels. In this work, only the naive approach is given as an explicit algorithm in Section 4.2.3.1. One important point still remains. Namely, the choice of L, i.e. the number of levels to include. The design of the heuristic method ensures that as long as one considers levels for which $\Delta t << \frac{1}{R(x(0))}$, the variance of the next level always decreases and thus by the heuristic level-selection should be included if the bias criteria calls for an additional level. This is important, because in order to ensure the bias (157), one usually uses the last three levels to test the behaviour of the bias, [14]. Thus, the initial choice of L is important. If L is too small and the bias constraint is not fulfilled then the next level that is added does not adhere to

the behaviour of the variance. If L is too large, the costs associated with the simulation might be too high compared to the accuracy. One way to get around this issue, is to choose at least 2 levels for which $\Delta t \leq \frac{1}{R(x(0))}$ in the warm-up and then add levels if more are needed in the multilevel algorithm. This way, the risk of too high simulation costs is much lower and due to the asymptotic behaviour of the variance, the adaptive approach adheres with the level-selection method.

4.2.3.1 Algorithm for level-selection

Algorithm 7: Heuristic level-selection for simulation in $[0, T]$

Function `warm_up`(L, T, N_{warm})**:**

  `// Initialze vectors to store results`

  $E_l = [\,]$; $E_d = [\,]$; $C_l = [\,]$; $C_d = [\,]$; $V_l = [\,]$; $V_d = [\,]$

  for $l := 0$ *to* $L - 1$ **do**

    if $l = 0$ **then**

      x = `KDMC`$(1/2^l, T)$ `// do the simulation for` $N_{\texttt{warm}}$ `paths`

      E_l.append(`mean`(x)), V_l.append(`var`(x))

      C_l.append(c); `// c is cost of run per path`

    else

      x^f, x^c =`correlated`$(1/2^l, 1/2^{l-1}, T)$ `// do the simulation for` $N_{\texttt{warm}}$ `paths`

      E_d.append(`mean`$(x^f - x^c)$), V_d.append(`var`$(x^f - x^c)$), C_d.append(c)

      x = `KDMC`$(1/2^l, T)$ `// do the simulation for` $N_{\texttt{warm}}$ `paths`

      E_l.append(`mean`(x))

      V_l.append(`var`(x))

      C_l.append(c);

  return $E_l, E_d, C_l, C_d, V_l, V_d$

Function `level_selection`(L, T, N_{warm})**:**

  $E_l, E_d, C_l, C_d, V_l, V_d$ =`warm_up`(L, T, N_{warm})

  $l = 1$

  for $i := 0$ *to* L **do**

    if $V_d[i] > V_l[i]$ **then**

      $l = l + 1$

  `levels` $= [l - 1, l]$ `// minimum two levels needed`

  $V_{\min} = V_d[l - 1]$

  for $j := l$ *to* $L - 1$ **do**

    if $V_d[j] < V_{min/2}$ **then**

      `levels`.append$(j + 1)$

      $V_{\min} = V_d[j]$

  $E, C, V, N =$ `relevant_results`$(E_l, E_d, C_l, C_d, V_l, V_d,$ *`levels`*$, N_{warm})$

  return $E, C, V, N,$ `levels`

In the above algorithm, the function `relevant_results`, identifies where levels have been skipped and sets the result at those indexes to zero. This is necessary because `warm_up` only calculates adjacent levels. To be precise consider a case where the chosen levels are $\{0, 1, 2, 6, 7\}$ then $E = [\texttt{mean}(x^{\Delta t_0}), \texttt{mean}(x^{\Delta t_1} - x^{\Delta t_0}), \texttt{mean}(x^{\Delta t_2} - x^{\Delta t_1}), 0, \texttt{mean}(x^{\Delta t_7} - x^{\Delta t_6})]$, where $x^{\Delta t_2}$ is the array of particle positions at the end of the simulation with the use of a step size Δt_2 and at indexes where differences are used, these have been simulated in a correlated fashion. Lastly, N is an array indicating how many particles have been simulated at each level, which is relevant when using the Multilevel Monte Carlo method with a "hot start". The section on the multilevel simulation using the Kinetic-Diffusion method is summarised by the algorithm below.

Algorithm 8: KDML simulation in $[0, T]$

Function `KDML`$(\epsilon^2, L, T, N_{warm})$**:**
- $E, C, V, N,$ `levels` = `level_selection`(L, T, N_{warm})
- **if** $N[l] = 0$ **then**
  - $N_{\text{diff}}[l] = N_{\text{warm}}$ `// number of paths needed at each level`
- **else**
  - $N_{\text{diff}}[l] \leftarrow$(171)
- **while** *error is above* ϵ^2 **do**
  - `// add paths to to satisfy` (156)
  - **while** *any level need mord paths* **do**
    - **for** *each level,* l*, that needs more paths* **do**
      - **if** l *is the first level* **then**
        - $x =$ `KDMC`$(\Delta t_l, T)$
      - **else**
        - $x^f, x^c =$ `correlated`$(\Delta t_l, \Delta t_{l-1}, T)$
      - `// update` $E[l], C[l], V[l], N[l]$
    - $N_{\text{diff}} \leftarrow$ (171) `// based on the updated levels recalculate the paths needed`
  - `// use final levels to determine if the bias satisfies` (157)
  - **if** *bias is too large* **then**
    - add another level and require a minimum of N_{warm} paths at that level
- **return** E, C, V, N

4.3 Multilevel version of asymptotic preserving splitting method

This section covers the multilevel version, presented in [32], of the method in Section 3.4. The method will be referred to as the APSML method. As mentioned in Section 3.4.1, the standard Monte Carlo method presented in Section 3.4.2 has a different diffusion coefficient than the scheme used as a starting point in [32], where the squared velocity is replaced by the second moment of the velocity distribution. Nevertheless, the explanations in this section remain in accordance with [32] because for the Goldstein-Taylor distribution $\mathcal{V}^2 =< \mathcal{V}^2 >$. The section consists of three main parts. In Section 4.3.1, an algorithm to simulate correlated paths is derived and then in Section 4.3.2 some of the numerical results from [32] are replicated to motivate the algorithm for the level-selection in Section 4.3.3 and to show consistency with the results in the paper.

4.3.1 Correlating paths

In this section, the goal is to correlate two paths with different time steps given by Δt_j and Δt_{j-1} and it is assumed that for each step in the coarse path, M steps are needed in the fine path to cover the same time span, i.e. $\Delta t_{j-1} = M\Delta t_j$. To be more precise, consider a particle movement for a time Δt_{j-1} given by (150), in terms of respectively the coarse path and the fine path:

$$X_{n+1} = X_n + \mathcal{V}_n \frac{\epsilon}{\epsilon^2 + \Delta t_{j-1} r} \Delta t_{j-1} + \sqrt{2(\mathcal{V}_n)^2 \frac{\Delta t_{j-1}}{\epsilon^2 + \Delta t_{j-1} r}} \sqrt{\Delta t_{j-1}} Z_n, \tag{190}$$

$$X_{n+1,0} = X_{n,0} + \sum_{m=0}^{M-1} \left(\mathcal{V}_{n,m} \frac{\epsilon}{\epsilon^2 + \Delta t_j r} \Delta t_j + \sqrt{2(\mathcal{V}_{n,m})^2 \frac{\Delta t_j}{\epsilon^2 + \Delta t_j r}} \sqrt{\Delta t_j} Z_{n,m} \right), \tag{191}$$

where the velocities in each step are given by (149) and an extra index m has been introduced to keep track of the extra steps in the fine path. The correlation of (190) and (191) is done by determining a mapping from $\{\xi_{n,m}\}_{m=0}^{M-1}$ to ξ_n. The random vector, ξ, consists of two sources of randomness, namely the velocity and the normal variable, Z, representing the Brownian increment. These two sources are kept separate in the mapping, meaning that $\{V_{n,m}\}_{m=0}^{M-1}$ is mapped to V_n and $\{Z_{n,m}\}_{m=0}^{M-1}$ is mapped to Z_n. Starting with the velocities, it is important to recall that they are influenced by two sources of randomness. The first source is due to the probability of a collision in the given time step and the second source is, of course $M(\nu)$. By definition, the velocity in the coarse step should either remain unchanged if no collision occurs or change to a velocity that is a function of the velocities drawn at collisions in the M fine steps. Therefore, it is necessary to ensure that a collision only occurs in a coarse step if at least 1 collision has occurred in the M fine steps. The occurrence of collisions in the fine steps is

determined by $\{u_{n,m}\}_{m=0}^{M-1}$ with $u_{n,m} \sim U(0,1)$. A collision occurs if

$$u_{n,m} \geq \frac{\epsilon^2}{\epsilon^2 + \Delta t^f r}. \tag{192}$$

Therefore, at least one collision takes place in the M fine steps if

$$u_{n,m}^{max} = \max_{m \in \{0,1,\ldots,M-1\}} u_{n,m} \geq \frac{\epsilon^2}{\epsilon^2 + \Delta t^f r}, \tag{193}$$

and it is natural to determine, the random number, u_n, for the coarse step based on $u_{n,m}^{max}$ with the requirement that u_n is still uniform. To obtain u_n The Inverse Transformation Method, see [43], is used. For any r.v. Y with distribution function $F_Y(y)$ that is invertible, the following can be derived about the distribution function $G(y)$ for the r.v. $F_Y(Y)$:

$$G(y) = P(F_Y(Y) \leq y) = P(Y \leq F_Y^{-1}(y)) = F_Y(F_Y^{-1}(y) = y, \quad y \in [0,1]. \tag{194}$$

Hence, the transformed r.v. $F_Y(Y)$ is uniformly distributed and thus, the goal is to define $U_n = F_{U_{n,m}^{max}}(U_{n,m}^{max})$. Since $U_{n,m}^{max}$ is the maximum order statistic of a sequence of M independent uniform random variables, it is well-known, see [45], that its distribution function is given by

$$F_{U_{n,m}^{max}}(u) = u^M, \quad u \in [0,1]. \tag{195}$$

Hence, $u_n = (u_{n,m}^{max})^M$ and a collision occurs in a coarse step if

$$u_n \geq \frac{\epsilon^2}{\epsilon^2 + \Delta t^c r}. \tag{196}$$

It is still necessary to prove that a collision indeed only occurs in the coarse step if at one collisions also occurs in the fine steps. The reason why this is not a trivial consequence of the definition of u_n, is because the probability of a collision happening depends on the step size. Therefore, it needs to be shown that

$$u_n = (u_{n,m}^{max})^M \geq \frac{\epsilon^2}{\epsilon^2 + \Delta t^c r} = \frac{\epsilon^2}{\epsilon^2 + M\Delta t^f r} \implies u_{n,m}^{max} \geq \frac{\epsilon^2}{\epsilon^2 + \Delta t^f r}, \tag{197}$$

or equivalently that

$$\begin{aligned} \sqrt[M]{\frac{\epsilon^2}{\epsilon^2 + M\Delta t^f r}} &\geq \frac{\epsilon^2}{\epsilon^2 + \Delta t^f r} \\ \iff \frac{\epsilon^2}{\epsilon^2 + M\Delta t^f r} &\geq \left(\frac{\epsilon^2}{\epsilon^2 + \Delta t^f r}\right)^M. \end{aligned} \tag{198}$$

Equation (198) can be rewritten as

$$\left(\epsilon^2 + \Delta t^f r\right)^M \geq \left(\epsilon^2 + M\Delta t^f r\right) \epsilon^{2M-2}, \tag{199}$$

where the left hand side can be expanded via the Binomial Theorem to give

$$\sum_{m=0}^{M} \binom{M}{m} \epsilon^{2(M-m)} (\Delta t^f r)^m \geq \epsilon^{2M} + M\epsilon^{2(M-1)} \Delta t^f r. \tag{200}$$

The inequality above clearly holds since the first two terms on the left hand side match the right hand side exactly and the rest of the terms on the left hand side are all positive. The mapping of the velocity is chosen to simply be s.t. the last velocity drawn in the fine step is reused in the coarse step, i.e. in the case of a collision happening

$$\mathcal{V}_n^* = \mathcal{V}_{n,M-1}. \tag{201}$$

Given that all velocities in the fine step are drawn independently, the mapping in (201) clearly ensures that the velocity in the coarse step follows the correct distribution. For the mapping $\{Z_{n,m}\}_{m=0}^{M-1} \to Z_n$, the choice is similar to the standard multilevel case, [15], i.e.

$$Z_n = \frac{1}{\sqrt{M}} \sum_{m=0}^{M-1} Z_{n,m}, \tag{202}$$

which clearly ensures that Z_n is standard normally distributed.

4.3.1.1 Algorithm for correlating paths

Algorithm 9: Simulation of correlated paths in $[0, T]$

Function `correlated`$(\Delta t^f, M, T)$`:`

- `// Initialze`
- $x^f, \nu^f \leftarrow S(x, \nu)$ `// from` (4)
- $t = 0$
- $\Delta t^c = M\Delta t^f$
- **while** $t < T$ **do**
  - **for** $m := 0$ *tp* $M - 1$ **do**
    - $u_m \sim U(0, 1)$
    - $z_m \sim N(0, 1)$
    - $x^f, \nu^f = \phi_{APS}$`(`Δt^f`,`x^f`,`v^f`,`$t + m\Delta t^f$`)`
  - $z = \frac{1}{\sqrt{M}} \sum_{m=0}^{M-1} z_m$ $x^c = \psi_t$`(`x^c`,`ν^c`,`z`,`Δt^c`)`
  - `// Collision step based r.v. from fine steps`
  - **if** $(u_m^{max})^M > \frac{\epsilon^2}{\epsilon^2 + \Delta t^c r}$ **then**
    - $\nu^c = \nu^f$
  - $t = t + \Delta t^c$
- **return** x^f, x^c

4.3.2 Numerical results

As in the case of the multilevel Kinetic-Diffusion method, the level selection for this method is also motivated by considering the variance structure for a numerical example, [32]. The quantity of interest, is given by $F(X) = X^2$, i.e. at the first level $E((X^{\Delta t_0})^2)$ is estimated and at subsequent levels $E((X^{\Delta t_j})^2 - (X^{\Delta t_{j-1}})^2)$ is estimated. The levels are chosen with $\Delta t_j = \frac{2.5}{2^j}$ and the time span is from 0 to 5. The particles are initialised at position 0 with velocity according to the Goldstein-Taylor distribution. Both the mean and the variance is calculated for different values of ϵ using 10^5 paths for each simulation. From Figures 10, 11 and 12 it is clear that the variance behaves differently in different domains of the time step. In Figure 11, the variance is somewhat flat for the large step sizes, which could indicate that for even larger step sizes the variance does not simply increase, but stays constant or starts to decrease. This behaviour occurs around $\Delta t = \epsilon^2$ so by decreasing ϵ in Figure 12 a more complete picture emerges for larger step sizes in comparison to ϵ^2, and it becomes evident that the variance increases as a function of the time step in the regime $\Delta t << \epsilon^2$ and it decreases in the regime $\Delta t >> \epsilon^2$. This behaviour is similar to the behaviour of the method in Section 4.2 in the case of a homogeneous background, which is also what is considered in this section. From the perspective of Equation (137) the behaviour can be understood by considering the limit $\Delta t \to \infty$ with fixed ϵ. In that limit one obtains

$$\partial_t f = \nu^2 \partial_{xx} f, \tag{203}$$

which after integrating over the velocity space becomes the same as (35) in the homogeneous case. As a consequence, the limiting models for (137) in the two limits of the time step are different. For small time steps, the model tends towards the kinetic equation in (8) and for large time steps, it tends towards (35).

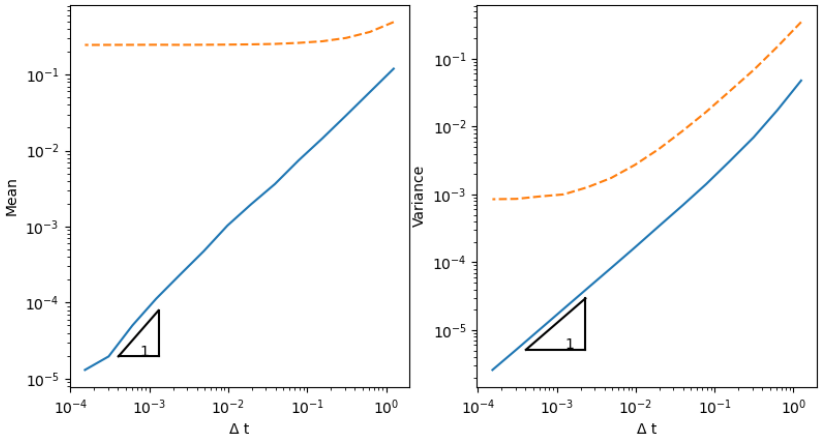


Figure 10: Variance and mean of $\hat{F}_j$(- - - -) and $\hat{F}_j - \hat{F}_{j-1}$(——) with $\epsilon = 10$.

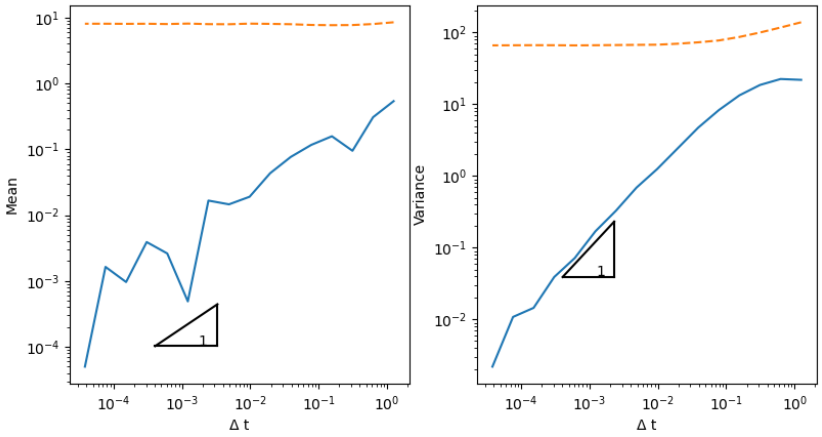


Figure 11: Variance and mean of $\hat{F}_j$(- - - -) and $\hat{F}_j - \hat{F}_{j-1}$(——) with $\epsilon = 1$.

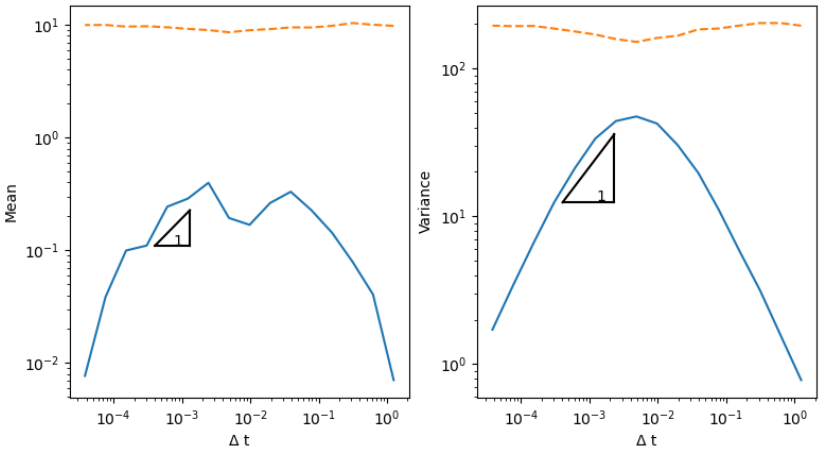


Figure 12: Variance and mean of $\hat{F}_j$(- - - -) and $\hat{F}_j - \hat{F}_{j-1}$(——) with $\epsilon = 0.1$.

4.3.3 level-selection

As in Section 4.2.3, the choice of levels is based on the goal of having a variance that behaves as in Theorem 2. In [32] two approaches are proposed. The first is to simply start the first level $\Delta t = \epsilon^2$, i.e.

$$\begin{aligned}\Delta t_0 &= \epsilon^2, \\ \Delta t_j &= \frac{\epsilon^2}{M^j}, \quad j = 1, 2, \ldots, J.\end{aligned}$$

The second strategy is very similar, but proposes to set the first level at some very coarse value of Δt and then make a jump such that for the next level $\Delta t = \epsilon^2$, i.e.

$$\begin{aligned}\Delta t_0 &= T, \\ \Delta t_j &= \frac{\epsilon^2}{M^j}, \quad j = 1, 2, \ldots, J,\end{aligned}$$

where T is the end time. In this second strategy the correlation between the paths given by Δt_0 and Δt_1 is different from all other levels because it is not the case that $\Delta t_0 = M\Delta t_1$. Additionally, a minimum of 3 levels are needed in order to check the criterion on the bias given in (159).

As opposed to Section 4.2 this section is not concluded with a multilevel algorithm because the only difference from the on at the end of Section 4.2 is that the correlation method, the level-selection and the standard Monte Carlo method is exchanged with the ones relevant for the splitting approach.

5 Summary of methods and ideas for alterations

This section contains a brief summary of the Monte Carlo methods and their multilevel counterparts in addition to some ideas from the author on how to potentially modify the splitting approach for better performance.

In Section 3.3 the Kinetic-Diffusion scheme from [36] is derived based on the assumption of a collision rate and a post-collisional distribution, which are independent of the pre-collisional velocity and the particle position. In 3.3.3 a proposed adaptation from [37] is described to expand the method to more general cases. In essence the Kinetic-Diffusion scheme is a combination of the standard kinetic scheme and the scheme in 3.3.1 which approximates (8) in the diffusive limit. When correlating paths with two different step sizes, the basic idea is to match the kinetic part of the coarse step with the kinetic part of the first fine step that covers approximately the same time frame. Then the rest of the variables from the fine steps are mapped in the form of

a linear combination to the random source in the diffusive part of the coarse step. In the linear combination each variable is weighted according to their expected impact on the step.

In Section 3.4.2, the asymptotic-preserving splitting method from [11] is presented for Equation (8). The method essentially splits each step up into two parts, a transport step and a collision step, which are performed in that order. The correlation proposed in [32] then ensures that if a collision happens in a coarse step there has also been a collision in the fine steps corresponding to the same time frame. Then, the velocity drawn from the last collision in the fine steps is the one used in the coarse step if a collision happens. The standard normal random numbers representing the Brownian motion are simply mapped as linear combination to the corresponding variable in the coarse step.

There are two things that could be possible weaknesses of the splitting approach. Firstly, when it is performed with a few number of steps or even just one step, a big challenge is to explain the background well since few or no collision steps are performed. In the limit $\epsilon \to 0$ this may not be a problem for the scheme with an altered diffusion coefficient, i.e. the scheme using the mean square velocity, because the diffusion term is independent of the velocity and the drift term becomes negligible for very small ϵ. However, for the splitting approach presented in this thesis, where the velocity plays a role in the diffusive term, it could be relevant, and for small but finite values of ϵ, it could also prove to have an impact for the method with the altered diffusion coefficient. A proposed alteration to accommodate this challenge is to reverse the splitting s.t. the collision step is performed before the transport step. This will be referred to as the reverse APS method. A consequence of the reverse APS method is that instead of using the velocity from the final collision in the fine steps one should use the velocity from the first collision. Secondly, only using information from one of the collisions in the fine steps may be a problem if other collisions have happened and have had a greater impact on the position of the particle from the fine path. This could happen in a couple of ways. The first is that the velocities, which are not used, are on average much greater in magnitude. The second is that if each coarse step corresponds to many fine steps then it may be the case that the velocity mapped from the fine steps only influenced one fine step before a new collision happened. Thus, only little information about the post-collision velocities that influenced the positional increments in the fine steps is actually carried over to the coarse step. A suggestion to remedy this in the case of the reversed splitting approach, is to map all the post-collision velocities drawn in the fine steps to the post-collision velocity in the coarse step. Let $\{\nu_m^*\}_{m=1}^{M_c}$ be the post-collision velocities from a sequence of M fine steps corresponding to one coarse step and M_c is the number of collisions in that sequence of steps. Then if a collision happens, the post-collision velocity, $\bar{\nu}^*$,

in the coarse step is given by

$$\bar{\nu}^* = \sum_{m=1}^{M_c} \sqrt{\alpha_m}\nu_m^*, \tag{204}$$

where α_m is the chosen weight for ν_m^* with $\sum_{m=1}^{M_c} \alpha_m = 1$. Under the assumption that the post-collision velocity is standard normal, it is clear that $\bar{\mathcal{V}}^*$ is still normally distributed and

$$\begin{aligned} E[\bar{\nu}^*] &= \sum_{m=1}^{M_c} \sqrt{\alpha_m} E[\nu_m^*] = \sum_{m=1}^{M_c} \sqrt{\alpha_m} 0 = 0 \\ V[\bar{\nu}^*] &= \sum_{m=1}^{M_c} \sqrt{\alpha_m} V[\nu_m^*] = \sum_{m=1}^{M_c} \alpha_m 1 = 1. \end{aligned} \tag{205}$$

In this work the weights are chosen based on the intuition that the more steps a velocity influences the greater the corresponding weight should be. A very straight forward way of doing of this is to let s_m be the number of steps that a certain velocity has been used. This means that, if following a collision no collisions happen in the next two steps then $s_m = 3$ because ν_m^* influences the step in which it is drawn and the next two steps in which no new velocities are drawn. Also let M_s be the total number of steps that velocities have influenced during the M steps, i.e. it is given by subtracting from M the number of steps it took until the first collision. The weights are then defined by

$$\alpha_m = \frac{s_m}{M_s}. \tag{206}$$

Note that when $M = 2$, there is no reason to calculate the weights, since they will be either 0, 1 or $\frac{1}{2}$ depending on whether zero, one or to collisions occur. In the next section the different methods are compared, and to avoid confusion the methods are identified according to

- KD is the Kinetic-Diffusion method,
- APS is the splitting method from Section 3.4.2,
- RAPS is the same as APS, but with the collision step preceding the transport step,
- APSD is the same as APS but with the squared velocity replaced by the mean square velocity in the diffusion coefficient,
- RAPSD is the same as RAPS, but with the same diffusion coefficient as APSD.

If any of the acronyms are appended with "MC" or "ML" it simply specifies whether it is respectively the standard Monte Carlo version or if it is the multilevel version. The suggested correlation method from this section is referred to as "the weighted correlation method". Moreover, the one where either the first or last post-collision velocity is chosen depending on the order of splitting is referred to as "the standard correlation method".

6 Comparison of methods

In this section, the two asymptotic-preserving methods and their multilevel extensions are compared. First, the Monte Carlo methods from Section 3 are compared in terms of their asymptotic properties meaning that the Wasserstein distance between the observations from the empirical distribution of the standard kinetic method and the observations from each of the methods of interest is calculated. Afterwards, a convergence test of the multilevel methods is performed by analysing the behaviour of the bias, variance and cost from Theorem 2 to determine the cost complexity of the different models. Clearly, the assumptions used when deriving the asymptotic splitting method that gives rise to the multilevel method in Section 4.3 are more restrictive. First of all, the velocity distribution is assumed to be symmetric around 0 and the collision rate is assumed to be constant. The assumption of a constant collision rate is violated for some parameters in the numerical example in Section 6.1 to check how significant this assumption is for a certain type of collision rate and to compare the two methods in a non-homogenous setting.

6.1 Numerical example

In this section, the two methods are tested on a numerical example, inspired by [7], where (8) is solved with

$$r(x) = \begin{cases} -a(x - \frac{1}{2}) + b), & x \leq \frac{1}{2}, \\ a(x - \frac{1}{2}) + b), & x > \frac{1}{2}, \end{cases} \tag{207}$$

$$S(x, \nu) = \frac{1}{\sqrt{2\pi}} e^{-\frac{\nu^2}{2}} \left(1 + \cos(2\pi(x + \frac{1}{2}))\right), \quad x \in [0, 1], \quad \nu \in \mathbb{R}, \tag{208}$$

$$M(\nu) = \frac{1}{\sqrt{2\pi}} e^{-\frac{\nu^2}{2}}, \tag{209}$$

where the case $a = 0$ represents the homogeneous case and $a = 1$ represents the heterogeneous case. No boundary conditions are applied.

The initial distribution is simulated, using the accept-reject method exactly as in 3.5 with the only exception being that the post-collisional distribution is standard normal.

In Section 6.1.1, the asymptotic properties of the methods are tested and then in Section 6.1.2 a convergence test of the different multilevel methods is conducted to determine how they compare and which case they fall under in Theorem 2. All the plots in the following sections are shown on a logarithmic scale.

6.1.1 Asymptotic properties

In this section the asymptotic properties of the methods are tested numerically and it is also tested that they do in fact work uniformly in ϵ.

Figure 13 shows the Wasserstein distance, [44], between each asymptotic-preserving method and a reference solution as a function of ϵ for fixed Δt with a homogenoues background. The Wasserstein distance is calculated ensemble of particles obtained via simulation. As reference solution, the standard kinetic method is used. The figure illustrates that the methods, KD, RAPSD and APSD all seem to work as expected in the limit $\epsilon \to 0$. Both RAPSD and APSD show a behaviour where the error is greatest around $\Delta t = \epsilon^2$, which is congruent with the observation that the equation model in (137) approaches the diffusion equation in the limit $\epsilon \to 0$ for fixed Δt and in the limit Δt if approaches the original kinetic equation. Take the left figure in 13 where $\Delta t = 0.015625$. For large values of ϵ, i.e. $\epsilon^2 >> \Delta t$, the models are closer to the kinetic solution, but as ϵ decreases the models move away from the kinetic limit and become less precise until they transition to approaching the diffusive limit and then become more precise for decreasing ϵ. The same kind of behaviour would be expected for the for the KD method since it naturally moves the particle in a kinetic manner when the rate of collisions per step is low and when it is high the particle is moved predominantly using the diffusive part of the step. For a single step, the error increases between the largest and the next largest value of ϵ, which could indicate that the transition between the two limits does not happen at exactly the same location in the domain. This difference from the other methods is investigated further in Section 6.1.2. For $\Delta t = 0.015625$, it simply seems that the dominating error of the KD model is the statistical error and therefore it is difficult to draw any meaningfull conclusion about how it behaves in the transition between the two limits. However, the figure does show that in the limits, it works as intended when compared to the other methods. The method APS and RAPS are a different story. For $\Delta t = 0.015625$ they show the same kind of behaviour as RAPSD and APSD but with larger errors. It is not unexpected that the use individual velocities as opposed to the mean square velocity has an effect on the error, but this effect should be on the statistical error rather than the Wasserstein distance and since the approximated standard deviation is one order of magnitude lower than the measured Wasserstein distance this does not seem to be the case. For $\Delta t = 1$ it becomes apparent that for small step sizes, APS and RAPS do not behave as intended. The reason for this instability is not clear at the moment.

In Figure 14 the limit $\Delta t \to 0$ in investigated a bit further by plotting the Wasserstein distance as a function of Δt for fixed ϵ. All methods have the intended behaviour in the limit, but for there seems to be some instabilities for larger step sizes across all methods, but as Figure 13

illustrates, it is not because the methods do not have the correct asymptotic properties in the limit $\epsilon \to 0$. At least not the methods KD, RAPSD and APSD. For the other two methods, the situation is less clear. Figure 15 shows the behaviour of the Wasserstein distance in the limit $\Delta t \to 0$ for a heterogenoues background and despite it being a violation of assumptions for the derivation of the splitting approaches, they do converge. The KD method converges as well, which fits with the observations in [37]. However, for very coarse step sizes, the KD method shows significant instability, which could be a problem for the ML method since this initial bias becomes much higher. It would have been preferable to analyse the behaviour for more values of ϵ and also to see the error as a function of ϵ. However, due to the high computational costs involved in simulating the standard kinetic method with enough paths to avoid the statistical error dominating the results, no such analysis has been performed. Finally, Figure 16 shows the the methods do indeed work uniformly in ϵ. The KD method has larger computational costs than the APS method, which is to be expected since the KD method involves more complex computations as discussed in Section 6.2.2.

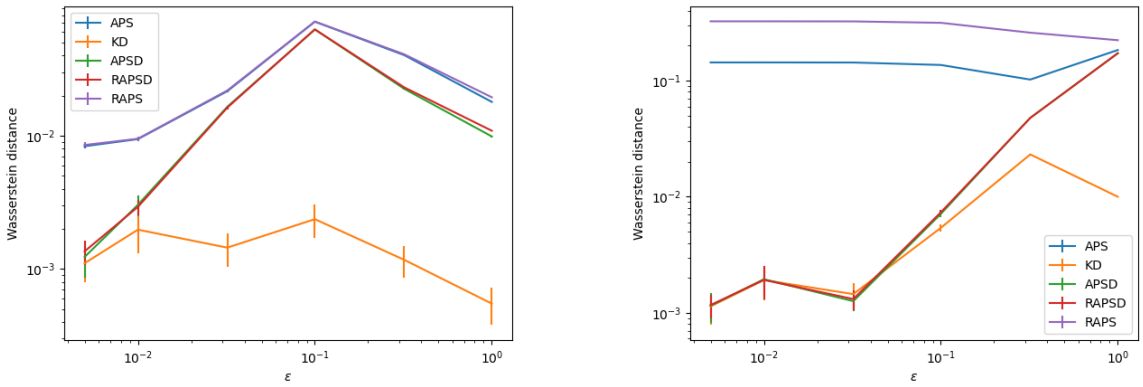


Figure 13: Wasserstein distance between each method and the standard kinetic scheme. At each value of ϵ the standard deviation of the Wasserstein distance is approximated using 20 runs and those standard deviation are represented by the vertical lines. 4 million paths have been used for the background with parameters $a = 0$ and $b = 1$. The methods tested against the standard kinetic method have been performed with $\Delta t = 0.015625$ for the left figure and $\Delta t = 1$ for the right figure.

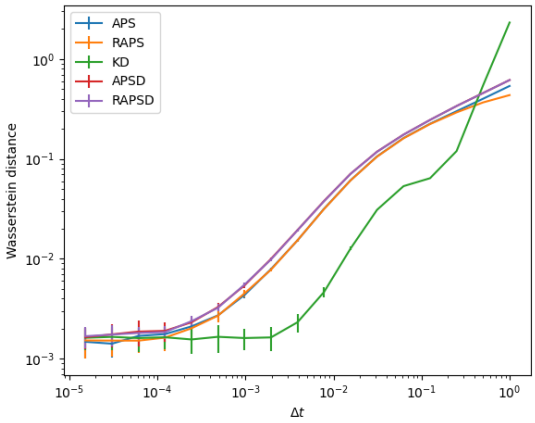


Figure 15: Wasserstein distance between each method and the standard kinetic scheme with $6*10^5$ paths. The simulations have been done with the parameters $a = 10$, $b = 0.1$ and $\epsilon = 0.32$.

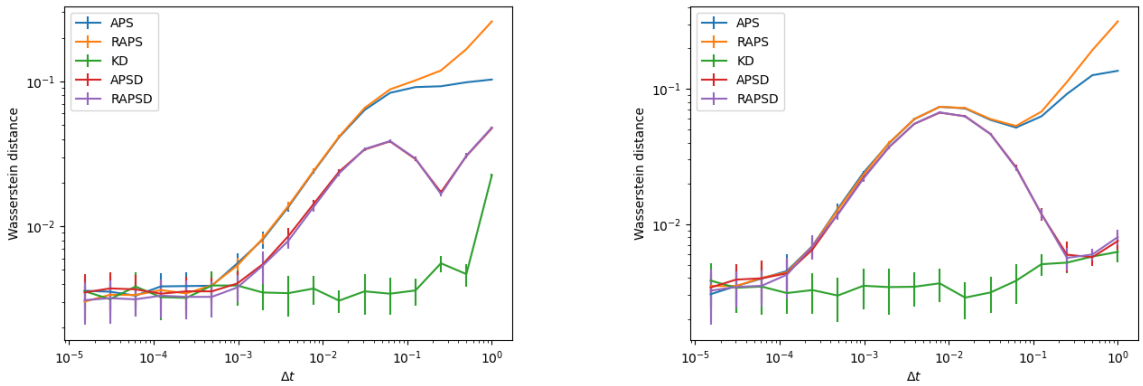


Figure 14: Wasserstein distance between each method and the standard kinetic scheme with $6 * 10^5$ paths. The simulations have been done with the parameters $a = 0$, $b = 1$ and then $\epsilon = 0.32$ on the right figure and $\epsilon = 0.1$ on the left figure.

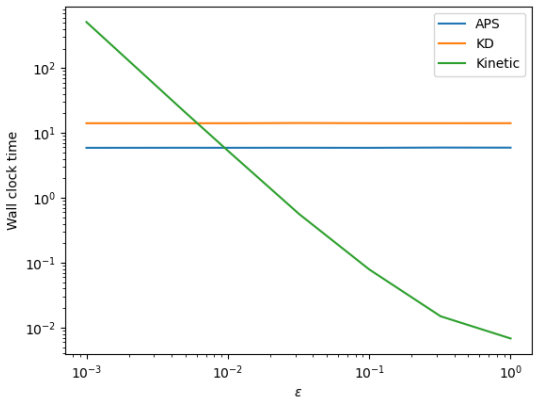


Figure 16: Wall clock time as a function of ϵ. Simulations are run with parameters $a = 0$, $b = 1$ using 10^4 paths with $\Delta t = 0.5$.

6.1.2 Convergence test

In this section, the multilevel versions of the methods are compared based on the behaviour of the bias and the variance of the difference estimate at different levels where the quantity of interest is simply given by $F(X) = X$. The tests that are the main focus in this section have been performed using $2.2 * 10^5$ paths for $a = 0$, $b = 1$ and $\epsilon \in \{0.32, 0.1, 0.032\}$ in one unit of time, which means that the number of expected collisions approximately range from 10 to 10^3, and the results are shown in Figures 17, 18 and 19. Results for a heterogeneous background are shown in Figure 20 and discussed in the end of the section. Note that the x-axis is flipped in this section s.t. that the results are displayed for increasing levels from left to right instead of increasing step size. To avoid too many test cases the weighted correlation method and reversing the splitting approach is only tested for the APSD method, which is also justified by the fact that the asymptotic tests seem to suggest a greater potential for this method compared to APS and RAPS.

As in the previous section the expected behaviour of the bias and the variance of the difference estimate at each level of the ML methods is that $\Delta t \approx \epsilon^2$ represents the transition domain where each method transitions to the kinetic limit model as $\Delta t \to 0$. The interesting part is to compare the location of the transition domain and the rate at which the bias and variance decrease once each method has transitioned to the kinetic limiting model. For the APS method, it is the case that for large step sizes, the behaviour is not as expected, but corresponds to the observations in Section 6.1.1. For $\Delta t >> \epsilon^2$, the bias and variance do not decrease with increasing step size like the other methods, but they do decrease for $\Delta t << \epsilon^2$. For the other methods the variance seems to behave as expected when comparing to [32] and [38]. As Table 1 shows it is the case

that for the splitting approaches the variances all decrease proportional to $\mathcal{O}(\Delta t)$ for $\Delta t << \epsilon^2$ and for the KD method it approximately decreases proportional to $\mathcal{O}(\Delta t^3)$. The bias estimates are in general quite erratic, which results in fairly poor estimates of the weak errors in Table 1, which is based on the results in Figure 18. Based on the fact that the stochastic discretisations are done via the Euler-Maruyama method, the expectation would be to see weak errors that asymptotically behave as $\mathcal{O}(\Delta t)$, at least for the splitting methods. The KD method is not as straight forward because, it is essentially a hybrid of two discretisations. Based on Figure 17 the KD-method has an approximated weak error of 1.57, while the splitting approaches range from 0.31 for the APS method to 0.6 for RAPSD with weighted correlation, which in both cases fall short of the expectation. However, it is worth noting that the behaviour of the bias estimates is much more erratic than for the KD approach and it is quite likely that simulating with more paths would yield approximate weak errors close to 1. Figures 17, 18 and 19 also show that the region where the methods transition between the two limiting models changes with ϵ^2 as expected, but there seems to be a difference between the splitting methods and the KD method. For the KD method, the transition happens a bit sooner, i.e. for higher step sizes and in the results of this example, it is consistently the case that the variance of the difference estimate is decreasing for $\Delta t \leq \epsilon^2$. In fact, the variance peaks at the step size which is the closest to ϵ^2. For those splitting approaches where the variance and bias increase with decreasing step size for large step sizes, it is not the case that the variance has started to decrease with the step size at $\Delta t = \epsilon^2$. This observation fits with Figure 13 where it was observed that the transition of the KD approach seems to happen for $\Delta t > \epsilon^2$. In terms of a level-selection procedure this would allow the choice of coarser levels than the case would be for the splitting approaches where the fine step size of the first level chosen after the transition would have to be at most of size ϵ^2, and even then, the rate of decay of the variance is a low lower for levels around ϵ^2. Amongst the splitting approaches that use the mean square velocity in the diffusion coefficient there is also a difference in the domain where they transition between limiting models. For both methods with reversed splitting the variance of the difference estimate peak at a higher value than in the case of the APSD approach. This effect is worse for RAPSD with standard correlation than for the same approach with weighted correlation. The effect might be caused by the fact that an extra collision step, which has an influence on the position, is performed. When the collision step is done after the transport step then the very last collision step becomes redundant in terms of the particle position. When reversing the splitting, an extra collision step is performed, which introduces more variance in to the simulation if collisions happen much more often for the fine path than for the coarse path. For large steps the probability of a collision happening in a coarse

step very much agrees with the probability that at least one collision happens in the fine steps taken in the same time frame. However, as $\Delta t \to \epsilon^2$ the probability of having a collision in a coarse step compared to having at least one collision in two fine steps becomes relatively small. So an extra collision step actually causes a greater decorrelation between the fine path and the coarse path due to having more steps where the fine path is affected by a collision without the coarse path experiencing a collision. For $\Delta t << \epsilon^2$ the effect diminishes once again. The weighted correlation approach makes somewhat up for the extra decorrelation when $\Delta t \approx \epsilon^2$ by ensuring that when a collision happens in the coarse path, information from all collisions in the fine path is included.

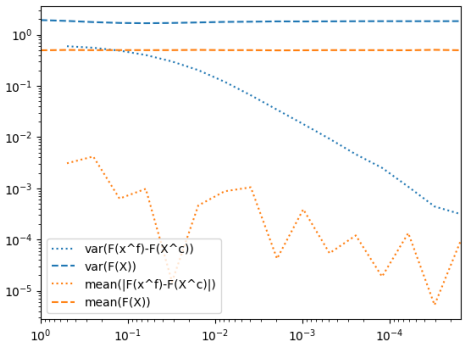
The domain before the transition, i.e. $\Delta t >> \epsilon^2$, is not of as much interest since it is the part of the domain where the requirements of Theorem 2 are not met. Therefore, it might not be of great importance that the variance for large step sizes is almost an entire order of magnitude lower for RAPSD with weighted correlation compared to APSD and RAPSD with standard correlation. The costs of each level as a function of Δt are shown in Figure 21 where it is evident that for large step sizes the cost of all methods behave proportional to $\mathcal{O}(\Delta t^{-1})$. For the KD method this behaviour changes around $\Delta t \approx \epsilon^2$ because by design it can "skip" steps, i.e. it can move kinetically across several steps when no collision occurs. Recall that the time between collisions is an exponential r.v. with rate $\frac{1}{R}$ so for $\Delta t < 1/R$ it will on average be s.t. the extra steps added will be skipped because the average time between collisions is large than the step size. In Table 1, the estimated rate of the cost in the limit $\Delta t \to 0$ comes out to around zero.

Finally, it also seems that the bias estimations are more stable for the KD approach than for the splitting approaches, which could play an important factor in the cost and accuracy of the multilevel approach. Especially APSD suffers from this problem and consequently may not be as consistent as the KDML scheme when it comes to adding levels for different values of the MSE.

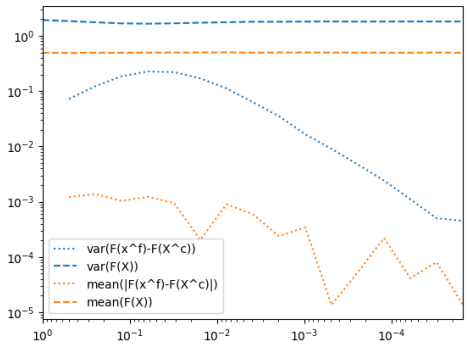
In conclusion, Theorem 2 says that under the assumption of the levels being chosen such that the variance and bias decrease with the step size, the splitting approaches are in the case $\beta = \gamma$ or maybe even $\beta < \gamma$, and the KD method falls under the case $\beta > \gamma$. Thus, the cost complexity of the splitting methods are at best proportional to $\mathcal{O}(\frac{\log(E)^2}{E^2})$ while the KD method is proportional to $\mathcal{O}(E^{-2})$ with E^2 being the MSE bound. Using the asymptotic behaviour to estimate the cost and the variance in the multilevel estimator the cost-complexity behaviour is shown in Figures 22 and 23. The cost-complexity behaviour based on the results in Figures 17 and 18 is shown in Figures 23 and 22, respectively. Figure 22 is based on the case where $a = 0$,

$b = 1$ and $\epsilon = 0.1$, and the cost is determined for lower values of E compared to Figure 23 because too fine step sizes are needed for more restrictive bounds on the MSE. The two figures illustrate quite well what is already seen in Figure 21, namely that for lower error bounds the KD method is more expensive due to a higher constant on its cost, but as the error bound is restricted the superior cost-complexity behaviour of the KD method means that the splitting methods become more expensive.

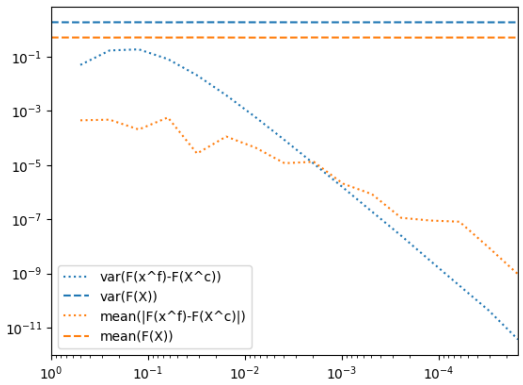
To end this section, some remarks are made on the behaviour of the methods for a heterogeneous background. Figure 20 shows that all methods become unstable resulting in a higher variance for coarse levels and the splitting methods seem to be affected quite a lot for small step sizes as well, which makes sense since a heterogeneous background not only conflicts with the derivation of the standard Monte Carlo version of the method, but it also conflicts with how the paths are correlated. In Figure 15 the splitting approaches converge with approximately the same rate as for the homogeneous case so the bade behaviour of the bias and variance in Figure 20 is most likely caused in the correlation process. Recall from Section 4.3.1 that it was argued that collision could only happen in the coarse step if at least one collision has happened in the fine step. However, that is not longer the case because the r appearing on both sides of the inequality (200) is no longer the same. On the right hand side, it depends on the position of the coarse particle and on the left hand side it depends on the position of the fine particle. Either this violation should cause an error in the code or mean that wrong numbers are mapped in the correlation method, which is the case for the splitting results in Figure 20. For the KD approach the instability in the variance is also present in the Wasserstein error in Figure 15 so the approach presented in 3.3 is surely a significant source of error. The Multilevel Monte Carlo method may remedy this extra bias on coarser levels, but it seems that at least it may be worth the effort to generalise the results shown in Sections 3.3.2.1 and 3.3.2.2 by assuming non-homogeneous poisson process for the generation of collisions.



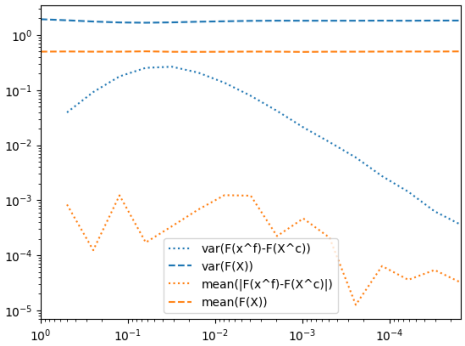
Results for APS.



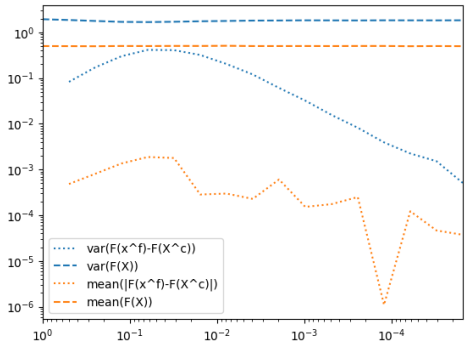
Results for APSD.



Results for KD.

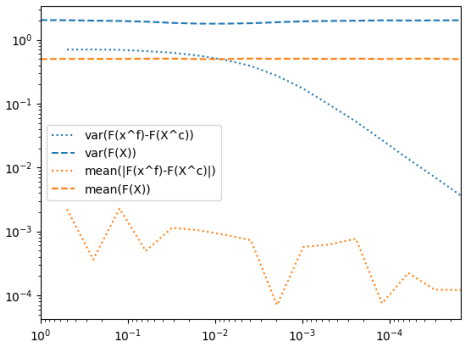


Results for RAPSD with weighted correlation method.

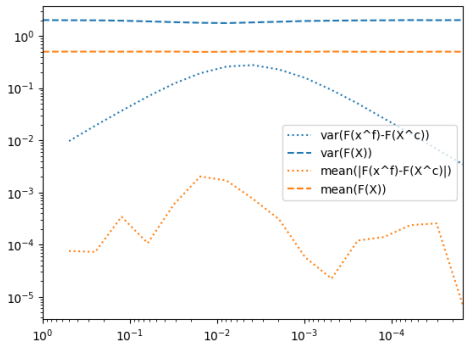


Results for RAPSD with standard correlation method.

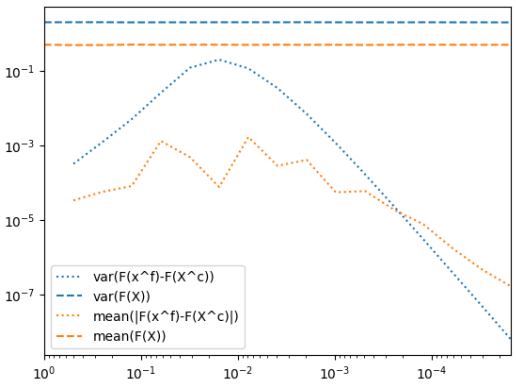
Figure 17: Simulation results on the variance and bias for both the MC and the ML methods. Simulations are done with $2.2 * 10^5$ paths for parameters $a = 0$, $b = 1$ and $\epsilon = 0.32$.



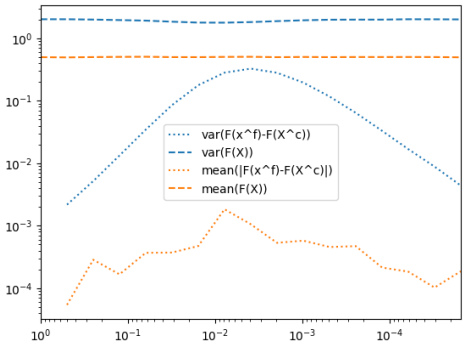
Results for APS.



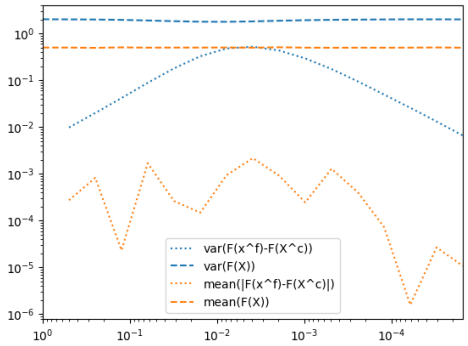
Results for APSD.



Results for KD.

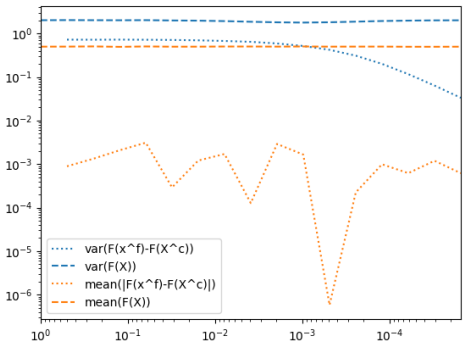


Results for RAPSD with weighted correlation method.

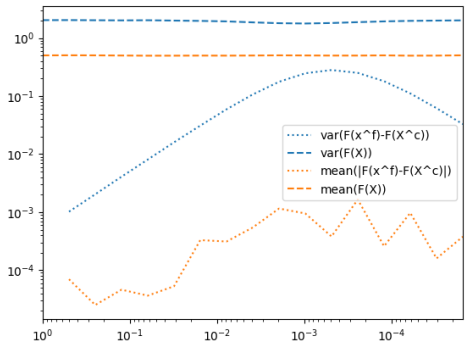


Results for RAPSD with standard correlation method.

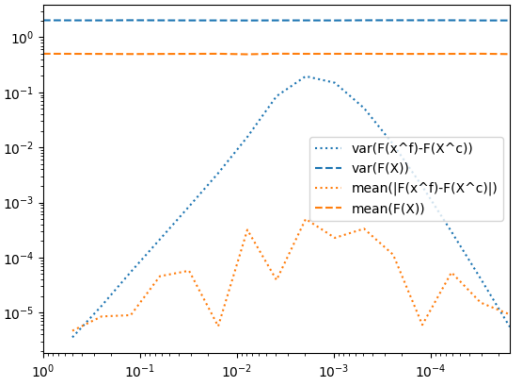
Figure 18: Simulation results on the variance and bias for both the MC and the ML methods. Simulations are done with $2.2 * 10^5$ paths for parameters $a = 0$, $b = 1$ and $\epsilon = 0.1$.



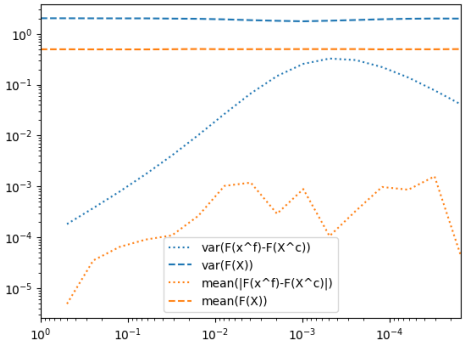
Results for APS.



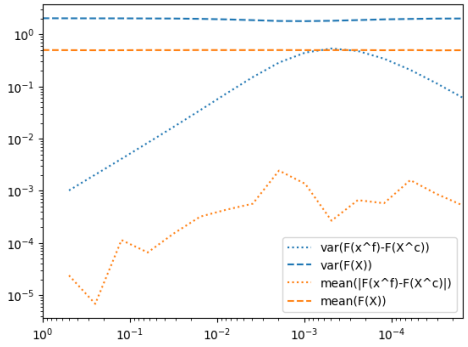
Results for APSD.



Results for KD.

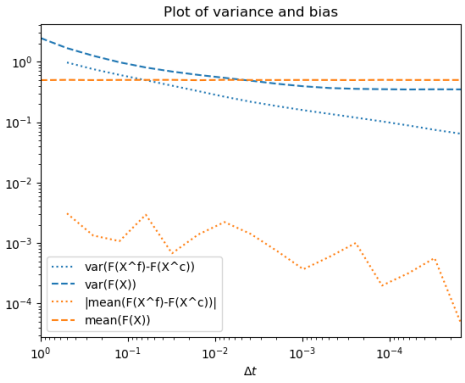


Results for RAPSD with weighted correlation method.

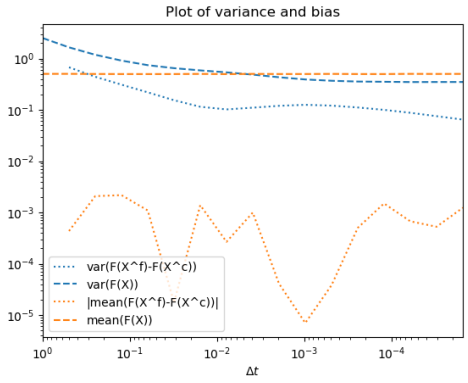


Results for RAPSD with standard correlation method.

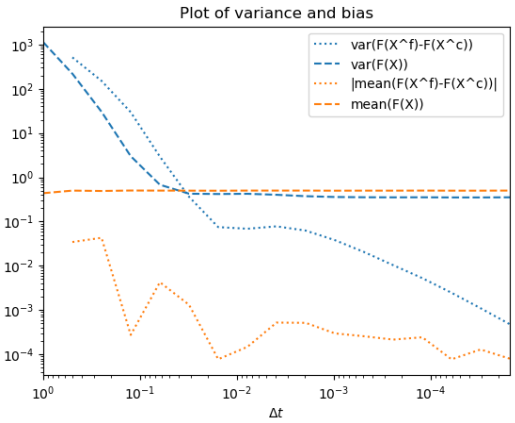
Figure 19: Simulation results on the variance and bias for both the MC and the ML methods. Simulations are done with $2.2 * 10^5$ paths for parameters $a = 0$, $b = 1$ and $\epsilon = 0.032$.



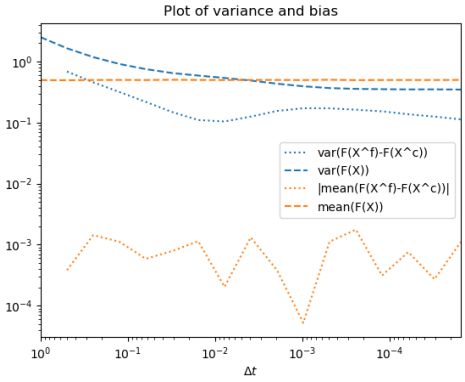
Results for APS.



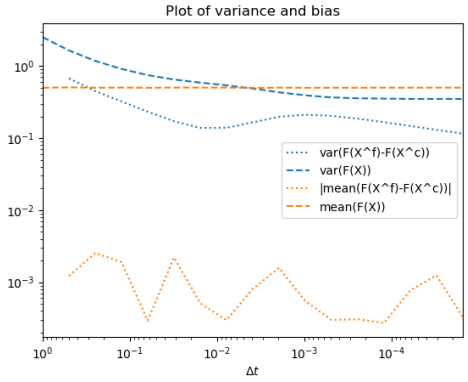
Results for APSD.



Results for KD.



Results for RAPSD with weighted correlation method.



Results for RAPSD with standard correlation method.

Figure 20: Simulation results on the variance and bias for both the MC and the ML methods. Simulations are done with $1.2 * 10^5$ paths for parameters $a = 10$, $b = 0.1$ and $\epsilon = 0.1$.

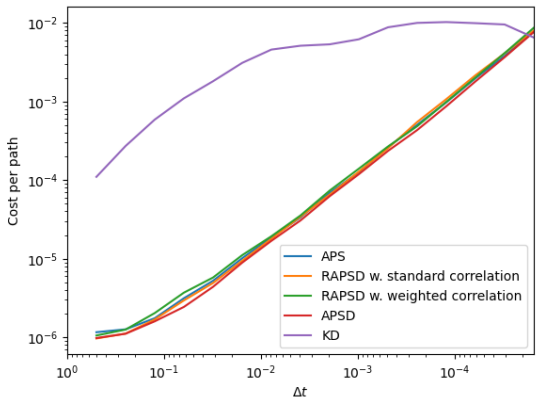


Figure 21: Cost per path as a function of Δt. Cost is measure is the wall clock time associated with estimating the difference at each level in Figure 18.

Method	α	β	γ
APS	0.31	0.91	0.97
APSD	0.53	0.0.94	1.04
KD	1.57	2.97	0.01
RAPSD w. weighted correlation	0.6	0.98	1.02
RAPSD w. standard correlation	0.46	0.94	0.96

Table 1: The estimated parameters from Theorem 2 based on results from Figure 17.

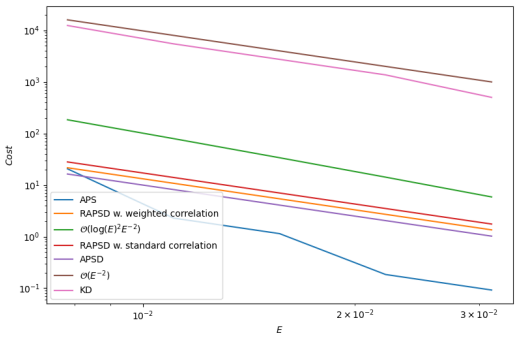


Figure 22: Total cost of ML methods as a function of the RMSE for parameters $a = 0$, $b = 1$ and $\epsilon = 0.1$.

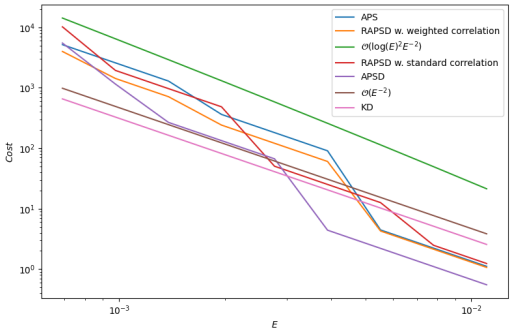


Figure 23: Total cost of ML methods as a function of the RMSE for parameters $a = 0$, $b = 1$ and $\epsilon = 0.32$.

6.2 Implementation

This section covers the hardware and software used in the project and it discusses the differences in implementation between the two Multilevel Monte Carlo methods. The specifications on hardware and software are presented in Section 6.2.1 and the differences in implementation is discussed in Section 6.2.2.

6.2.1 Hardware and software specifications

Most testing and simulation runs have been done on a local machine for which the relevant hardware specifications are the following:

- CPU: AMD Ryzen 7 3700X Eight Core CPU (3.6GHz-4.4GHz/36MB CACHE/AM4)
- RAM: 16GB Corsair VENGEANCE DDR4 3200MHz (2 x 8GB)
- Motherboard: ASUS® STRIX B550-F GAMING (DDR4, USB 3.2, 6Gb/s)
- Memory: 250GB SEAGATE BARRACUDA 120 2.5" SSD, (optil 560MB/sR — 540MB/sW)

For the software, every method has been implemented in Python with the use of `numpy` to speed up computations of several paths in a single array. Additionally, the library `numba`, which is a just-in-time compiler for Python, has been used to further optimise the program by allowing the computational effort to be spread out across the different CPU cores. In general, the idea throughout all the implemented methods is to divide the work up by the number of paths. For instance, whenever a set of levels need more paths in order to satisfy the variance constraint, (156), the number of paths at each level is divided by the number of cores in an attempt to distribute the work evenly across the cores. Thereby, each core updates all levels with a fraction of the number of paths actually needed, and the results are then collected to obtain the result for all paths needed. This has increased the efficiency of testing the algorithms significantly and an example of this is given in Figure 24. Figure 24 shows that the computation time of the multilevel convergence test for the Kinetic-diffusion method and for the asymptotic preserving splitting method is approximately sped up by 80 times and 10 times respectively. A much greater effect is seen for the Kinetic-Diffusion method, which is due to the greater complexity of the implementation.

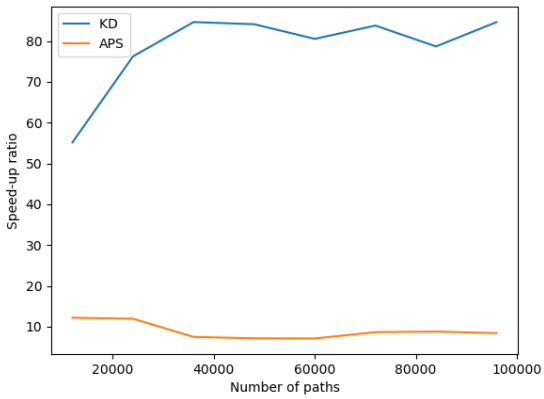


Figure 24: Shows the ratio of the computation of the multilevel convergence test without and with the use of `numba`.

Some of the downsides is that the use of `numba` means that only the `numpy` library can be used and even in that case some of the built in functions in `numpy` have not been implemented in `numba` and must therefore be implemented manually. In addition, spreading the work across all cores, causes less accurate estimations of the cost when doing an adaptive multilevel procedure. One reason for this is that all values of a level are not calculated in the same parallel loop but rather new information is added every time more paths are needed to satisfy variance criteria. Therefore, the available CPU capacity varies each time new information is added and the `numba` implementation is much more sensitive to how much CPU power is available than to how many paths the calculations are done for, which greatly affects the estimation of the cost at each level. The source code can be found at `https://github.com/Thue77/Simulators-for-Boltzmann_BGK-equation`.

6.2.2 Comparing implementations

The complexity in implementing the two methods differs a lot, which is also reflected in the amount of computation per pair of coupled particles at a fixed step size. There two parts of the implementation, the two methods differ the most. The first and most important part, is the fact that the KDML method is not restricted by the step size in its computational efficiency, meaning that if no collision happens for several steps, then all of these steps are simply dealt with by one kinetic movement of the particle. Computationally this is a big advantage as 21 also illustrates. For large step sizes the cost of the KD approach is higher, but where as the splitting approaches continue to increase with decreasing step size, the cost of the KD approach becomes constant. In the implementation this means that the mapping phase from Section

4.2.1.1 involves a different number of random variables for each path. Recall from Figure, 7 that the fine path has to move until it covers the next coarse step and the amount of steps that take might differ for different paths. Thus, when dealing with many paths in parallel one must keep track of the progress in each path. In addition to the challenge of keeping track of the progress of each individual path, the task of allocating the proper amount of memory for the number of random variables saved for each coarse step. Either, too much memory is allocated and one might end up with arrays and matrices that contain a lot of zeros or no space is allocated and extra space is only added if more variables are needed. Neither approach is optimal and it would be preferable to find a better alternative. Comparing with the APSML method, the number of fine steps for each coarse step is always the same and all particles have moved for the same amount of time at any point in the simulation. Thus, there is only one time to keep track off and the number of variables to use when correlating the coarse and fine path does not change. The second and less significant difference is the complexity of correlating the different random variables. First of all, the KDML method involves one more variable, namely the exponential variable for drawing collision times, and finding that involves solving an integral. Secondly, a method for drawing the collision times also needs to be implemented and it also involves solving an integral, which depends on the collision rate. Hence, a complex expression for the collision might complicate matters. However, in practice the main problem is probably dealing with a rapidly intermediary region in different parts of the domain.

7 Conclusion

In this thesis Multilevel Monte Carlo simulation of a kinetic equation model, namely the Boltzmann-BGK model, is investigated with the focus on how to deal with situations where the problem is governed by multiple scales, which lead to prohibitively high simulation costs. In practice, such a situation can be represented by the simulation of neutral particles in the plasma edge of a fusion reactor where drastic changes in the collision rate make standard kinetic methods infeasible. The two Multilevel Monte Carlo methods presented in this work are based on two asymptotic-preserving methods that overcome the problem of having computational costs that depend on the scaling parameter. The Kinetic-Diffusion scheme does this by using a Poisson Process to generate the time it takes for the next collision to happen and then it moves with the standard method until the collision after which each step is finished by a diffusive motion. The diffusive motion is obtained by using a scheme that approximates the equation model in the diffusive limit. The scheme is then altered by finding the mean and variance of a positional increment made by the standard kinetic method, but under the condition that the final velocity

used in that positional increment is known. The mean and variance of the diffusive motion is then made to match those of the standard kinetic motion. In that process some elaborations are made compared to the existing literature, [36, 35]. It is proven that in a specific time frame, the time from the last collision until the end of that time frame is identically distributed with the time between collisions when both are conditioned on the number of collisions that happen in the time frame. The knowledge of that distribution among other things lead to finding the mean and variance of the kinetic motion without conditioning on the final velocity. Formulating and proving Theorem 1 then allows to use the mean and variance without conditioning to obtain the corresponding quantities when conditioning on the last velocity in the kinetic motion. The asymptotic-preserving splitting scheme is obtained by reformulating the original equation model using an even-odd decomposition presented in [11]. The derivation yields a scheme that differs from the one that is the basis of one of the two multilevel methods that is of interest in this work. The one derived based on the even-odd splitting includes the information the individual particle velocities in the diffusive coefficient whereas the other exchanges the the individual velocities by their mean square value. Due to this difference and the intuition that model with the mean square velocity should perform better in the diffusive limit as a consequence of having one less source of randomness, the performance of the multilevel method using both schemes is tested. It turns out that the scheme using individual velocities in the diffusion coefficient performs worse in the diffusive limit than both the one with the mean square velocities and the Kinetic-Diffusion approach. It also shows in the behaviour of the variance of the multilevel estimator in the levels before the schemes transition to the kinetic limit.

Since the post-collisional velocity is assumed to be normal in the final experiment it is proposed to try and use the information of all post-collision velocities from the fine path when correlating it with the coarse path for the splitting approaches. However, it is suggested for a reversed splitting procedure, which based on the results actually makes the performance a bit worse in the domain where the schemes transition between the limiting models. The results for a homogeneous background also show that the decrease in performance is not caused by changing the correlation but by reversing the splitting, which suggests that the cause is the disagreement between the probability of having one collision in a coarse step and the probability of having at least one collision in the corresponding fine steps. Overall, the cost-complexity behaviour of the splitting schemes is worse than the Kinetic-Diffusion scheme based on the asymptotic behaviour of the variance and of the cost in terms of the levels. The only downside to the Kinetic-Diffusion scheme is found in the complexity of the implementation, but as the results show the benefits of its asymptotic behaviour outweighs this when accuracy requirements are restricted.

A Operator splitting

This section contains a short introduction to operator splitting of a system of two equations based on [16]. Consider the differential equation

$$\dot{u} = Au, \quad u(0) = u_0, \tag{210}$$

where A is some linear operator for which $A = A_1 + A_2$. In the context of this work, the operator A is a matrix obtained by discretising differential and collision operators. The exact solution is given by

$$u(t) = u_0 e^{At}, \tag{211}$$

where e^A is the matrix exponential of A. The main idea of operator splitting is to approximate the matrix exponential in (211) by

$$e^{At} \approx e^{A_1 t} e^{A_2 t}, \tag{212}$$

which is a first order approximation. $I^{\Delta t} = \{t_0, t_1, \ldots, t_N\}$ with $t_0 < t_1 < \cdots < t_N = T$ and $t_{n+1} - t_n = \Delta t$ be a discretization of the time interval $I = [t_0, T]$. Then (212), suggests the following solution scheme as an approximation to (210):

$$\dot{u}^{1/2} = A_1 u^{1/2}, \quad t \in [t_n, t_{n+1}], \quad u^{1/2}(t_n) = u_{sp}(t_n), \tag{213}$$

$$\dot{u}^{2/2} = A_2 u^{2/2}, \quad t \in [t_n, t_{n+1}], \quad u^{2/2}(t_n) = u^{1/2}(t_{n+1}), \tag{214}$$

where u_{sp} is the approximation of u and $u_{sp}(t_0) = u_0$, $u_{sp}(t_n) = u^{2/2}(t_n)$ for $n = 1, 2, \ldots, N$ and the superscript, $2/2$ indicates that $u^{2/2}$ is the solution of the second part of the splitting. For the specific problem:

$$\begin{cases} \partial_t h + \nu \partial_x j = \frac{r}{\epsilon^2}(M\rho - h), \\ \partial_t j + \frac{\nu}{\epsilon^2}\partial_x h = -\frac{r}{\epsilon^2} j. \end{cases} \tag{215}$$

In this case $u = (h, j)^T$ and

$$A = \begin{pmatrix} T & \nu\partial_x \\ \frac{\nu}{\epsilon^2}\partial_x & \frac{1}{\epsilon^2} \end{pmatrix} = \underbrace{\begin{pmatrix} 0 & \nu\partial_x \\ \frac{\nu}{\epsilon^2}\partial_x & \frac{1}{\epsilon^2} \end{pmatrix}}_{A_1 =} + \underbrace{\begin{pmatrix} T & 0 \\ 0 & 0 \end{pmatrix}}_{A_2 =}, \tag{216}$$

with $T : f \to \frac{r}{\epsilon^2}(M\rho - f)$. In this example the discretization of the operators is not explicit, i.e. $\nu\partial_x$ represents the advection matrix multiplied by ν. The reason for this is that the purpose is

not to actually solve the equation, instead the goal is to reformulate it. With this in mind the splitting in (213) and (214) amounts to

$$(I) = \begin{cases} \partial_t h^{1/2} + \nu \partial_x j^{1/2} = 0, \quad t \in [t_n, t_{n+1}], \quad h^{1/2}(t_n) = h_{sp}(t_n), \\ \partial_t j^{1/2} + \frac{\nu}{\epsilon^2} \partial_x h^{1/2} = -\frac{r}{\epsilon^2} j^{1/2}, \quad t \in [t_n, t_{n+1}], \quad j^{1/2}(t_n) = j_{sp}(t_n), \end{cases} \tag{217}$$

$$(II) = \begin{cases} \partial_t h^{2/2} = \frac{r}{\epsilon^2}(M\rho^{2/2} - h^{2/2}), \quad t \in [t_n, t_{n+1}], \quad h^{2/2}(t_n) = h^{1/2}(t_{n+1}), \\ \partial_t j^{2/2} = 0, \quad t \in [t_n, t_{n+1}], \quad j^{2/2}(t_n) = j^{1/2}(t_{n+1}), \end{cases} \tag{218}$$

and thus, after the final time step h is estimated by $h^{2/2}$ and j is estimated by $j^{2/2}$.

B Accept-reject method

The following brief introduction is based on [43]. The goal of the accept-reject is to draw random numbers from a complicated distribution, $p(x)$ using a well-known and easy to draw from distribution, $q(x)$ with the same support as $p(x)$. The method works as follows:

- Draw x^* from $q(x)$,
- Keep x^* with a probability of $\frac{p(x^*)}{cq(x^*)}$,

where c is some constant. It is clear that $cq(x)$ must be greater than $p(x)$ for all x, otherwise $\frac{p(x)}{cq(x)}$ cannot be considered a probability. Hence, $q(x)$ and c are chosen to ensure this. For completion, the method is explained in relation to the numerical example in Section 3.5. In the example, the goal is to draw random numbers from

$$p(x) = 1 + \cos(2\pi(x + 0.5)), \quad x \in [0, 1]. \tag{219}$$

Since the support of $p(x)$ is given by $[0, 1]$ and obvious choice is

$$q(x) = 1, x \in [0, 1]. \tag{220}$$

The constant, c is then chosen as $c = \max_{x \in [0,1]} p(x)$, since then $\frac{p(x)}{c} \leq 1$. The maximum of $p(x)$ is 2. Now, numbers are drawn from $q(x)$ and the higher the value of $p(x)$ is at that location, the more probable it is that we keep. Figure 25 shows that the estimation of $p(x)$ based on $q(x)$ turns out to be quite good.

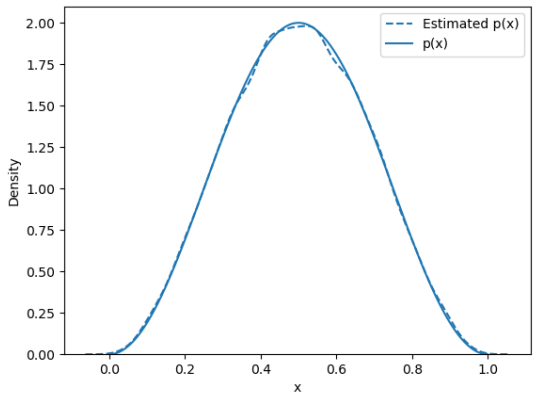


Figure 25: Estimation of (219).

References

[1] R. A. Adams and C. Essex. *Calculus*. Canada: Pearson, 2014.

[2] Guillaume BAL and Yvon MADAY. "Coupling of transport and diffusion models in linear transport theory". English. In: *Modélisation mathématique et analyse numérique (Print)* 36.1 (2002), pp. 69–86.

[3] P. L. Bhatnagar, E. P. Gross, and M. Krook. "A Model for Collision Processes in Gases. I. Small Amplitude Processes in Charged and Neutral One-Component Systems". English. In: *Physical review* 94.3 (1954), pp. 511–525.

[4] JF Bourgat et al. "Coupling Boltzmann and Euler equations without overlapping". In: *Contemporary Mathematics* 157 (1994), pp. 377–377.

[5] C. Brandt. *Trapped-particle instabilities in quasi-isodynamic stellarators - Scientific Figure on ResearchGate*. `https://www.researchgate.net/figure/A-schematic-tokamak-figure-courtesy-of-C-Brandt_fig1_262364216`. [Online; accessed 10-February-2021]. 2021.

[6] Francis F. Chen. *Introduction to Plasma Physics and Controlled Fusion*. English. Third. Cham: Springer International Publishing, 2016. ISBN: 3319223097.

[7] Anaïs Crestetto, Nicolas Crouseilles, and Mohammed Lemou. *A particle micro-macro decomposition based numerical scheme for collisional kinetic equations in the diffusion scaling*. 2017. arXiv: `1701.05069 [math.NA]`.

[8] Anaïs Crestetto, Nicolas Crouseilles, and Mohammed Lemou. "Asymptotic-Preserving Scheme Based on a Finite Volume/Particle-In-Cell Coupling for Boltzmann-BGK-Like Equations in the Diffusion Scaling". English. In: vol. 78. Cham: Springer International Publishing, 2014, pp. 827–835. ISBN: 2194-1009.

[9] Nicolas Crouseilles, Pierre Degond, and Mohammed Lemou. "A hybrid kinetic/fluid model for solving the gas dynamics Boltzmann–BGK equation". In: *Journal of Computational Physics* 199.2 (2004), pp. 776–808.

[10] Jeffery D. Densmore et al. "A hybrid transport-diffusion method for Monte Carlo radiative-transfer simulations". English. In: *Journal of computational physics* 222.2 (2007), pp. 485–503.

[11] G. Dimarco, L. Pareschi, and G. Samaey. "Asymptotic-Preserving Monte Carlo Methods for Transport Equations in the Diffusive Limit". English. In: *SIAM journal on scientific computing* 40.1 (2018), A504–A528.

[12] Albert Einstein. *Does the Inertia of a Body Depend Upon its Energy-Content?* 1905.

[13] Robert Gallager. *Chapter 2: Poisson Processes*. Massachusetts Institute of Technology: MIT OpenCourseWare. Spring 2011. URL: https://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-262-discrete-stochastic-processes-spring-2011/index.htm.

[14] Michael B. Giles. "Multilevel Monte Carlo methods". English. In: *Acta numerica* 24 (2015), pp. 259–328.

[15] Michael B. Giles. "Multilevel Monte Carlo Path Simulation". In: *Operations Research* 56.3 (2008), pp. 607–617. DOI: 10.1287/opre.1070.0496. eprint: https://doi.org/10.1287/opre.1070.0496. URL: https://doi.org/10.1287/opre.1070.0496.

[16] Roland Glowinski, Stanley J. Osher, and Wotao Yin. *Splitting Methods in Communication, Imaging, Science, and Engineering*. English. Cham: Springer International Publishing, 2016;2017; ISBN: 3319415891.

[17] S. Goldstein. "On diffusion by discontinuous movements, and on the telegraph equation". In: *The Quarterly Journal of Mechanics and Applied Mathematics* 4.2 (Jan. 1951), pp. 129–156. ISSN: 0033-5614. DOI: 10.1093/qjmam/4.2.129. eprint: https://academic.oup.com/qjmam/article-pdf/4/2/129/5301574/4-2-129.pdf. URL: https://doi.org/10.1093/qjmam/4.2.129.

[18] François Golse, Shi Jin, and C. D. Levermore. "The Convergence of Numerical Transfer Schemes in Diffusive Regimes I: Discrete-Ordinate Method". English. In: *SIAM journal on numerical analysis* 36.5 (1999), pp. 1333–1369.

[19] Michael Griener. "At the plasma edge". English. In: *EuroFuision* (Jan. 2020). URL: https://www.euro-fusion.org/news/2020/january/at-the-plasma-edge/.

[20] Kirsten Haupt. "Be clean, Be strong". English. In: *Plasma Physics* (Apr. 2018). URL: https://www.iter.org/newsline/-/2977.

[21] Willem Hundsdorfer, Jan Verwer, and SpringerLink (Online service). *Numerical Solution of Time-Dependent Advection-Diffusion-Reaction Equations*. English. Vol. 33. Berlin, Heidelberg: Springer Berlin Heidelberg, 2003. ISBN: 3662090171.

[22] ITER Organization. https://www.iter.org/. [Online; accessed 10-February-2021]. 2021.

[23] Jean Jacod, Philip Protter, and SpringerLink (Online service). *Probability essentials*. English. 2nd, correct 2nd printing;Second; Berlin: Springer, 2004;2002; ISBN: 3540438718;9783540438717.

[24] Shi Jin. "Asymptotic preserving (AP) schemes for multiscale kinetic and hyperbolic equations: a review". In: *Lecture notes for summer school on methods and models of kinetic theory (M&MKT), Porto Ercole (Grosseto, Italy)* (2010), pp. 177–216.

[25] Shi Jin, Lorenzo Pareschi, and Giuseppe Toscani. "Uniformly Accurate Diffusive Relaxation Schemes for Multiscale Transport Equations". English. In: *SIAM journal on numerical analysis* 38.3 (2000), pp. 913–936.

[26] Sergei Krasheninnikov, Andrei Kukushkin, and Andrei Smolyakov. *On the Edge of Magnetic Fusion Devices*. English. 1. 2020. Cham: Springer International Publishing, 2020. ISBN: 3030495949;9783030495947.

[27] Pauline Lafitte and Giovanni Samaey. "Asymptotic-preserving projective integration schemes for kinetic equations in the diffusion limit". In: *SIAM Journal on Scientific Computing* 34.2 (2012), A579–A602.

[28] Edward W. Larsen, J. E. Morel, and Warren F. Miller. "Asymptotic solutions of numerical transport problems in optically thick, diffusive regimes". English. In: *Journal of computational physics* 69.2 (1987), pp. 283–324.

[29] Mohammed Lemou. "Relaxed micro–macro schemes for kinetic equations". In: *Comptes Rendus Mathematique* 348.7 (2010), pp. 455–460. ISSN: 1631-073X. DOI: https://doi.org/10.1016/j.crma.2010.02.017. URL: https://www.sciencedirect.com/science/article/pii/S1631073X10000622.

[30] Mohammed Lemou and Luc Mieussens. “A New Asymptotic Preserving Scheme Based on Micro-Macro Formulation for Linear Kinetic Equations in the Diffusion Limit”. English. In: *SIAM journal on scientific computing* 31.1 (2008), pp. 334–368.

[31] Iván Lux. *Monte Carlo Particle Transport Methods: Neutron and Photon Calculations*. May 2018. ISBN: 9781351074834. DOI: 10.1201/9781351074834.

[32] Emil Løvbak, Giovanni Samaey, and Stefan Vandewalle. *A multilevel Monte Carlo method for asymptotic-preserving particle schemes*. 2020. arXiv: 1907.04610 [math.NA].

[33] Barbara MacCluer. *Elementary Functional Analysis*. English. 1. Aufl. Vol. 253. New York, NY: Springer Science+Business Media, LLC, 2009;2008; ISBN: 0387855297;9780387855295.

[34] Marco Milla and Erhan Kudeki. “Particle dynamics description of “BGK collisions” as a Poisson process”. English. In: *Journal of Geophysical Research - Space Physics* 114.A7 (2009), A07302–n/a.

[35] Bert Mortier. “Advanced Monte Carlo simulation and estimation for kinetic neutral particles in the plasma edge of fusion reactors”. PhD thesis. PhD thesis, KU Leuven, Department of Computer Science, 2020.

[36] Bert Mortier, Martine Baelmans, and Giovanni Samaey. *Kinetic-diffusion asymptotic-preserving Monte Carlo algorithm for Boltzmann-BGK in the diffusive scaling*. 2020. arXiv: 2004.04071 [math.NA].

[37] Bert Mortier, Martine Baelmans, and Giovanni Samaey. “Kinetic-diffusion asymptotic-preserving Monte Carlo algorithms for plasma edge neutral simulation”. English. In: *Contributions to plasma physics (1988)* 60.5-6 (2020), e201900134–n/a.

[38] Bert Mortier et al. *Multilevel Asymptotic-Preserving Monte Carlo for Particle Simulations*. 2020. arXiv: 2004.04071 [math.NA].

[39] J. Ongena et al. “Magnetic-confinement fusion”. English. In: *Nature Physics* 12.5 (May 2016). Copyright - Copyright Nature Publishing Group May 2016; Document feature - ; Last updated - 2016-06-29, pp. 398–410. URL: https://search-proquest-com.proxy3-bib.sdu.dk/scholarly-journals/magnetic-confinement-fusion/docview/1786644508/se-2?accountid=14211.

[40] Lorenzo Pareschi and Giovanni Russo. “An introduction to Monte Carlo method for the Boltzmann equation”. In: *ESAIM: Proceedings*. Vol. 10. EDP Sciences. 2001, pp. 35–75.

[41] Grigorios A. Pavliotis. *Stochastic Processes and Applications: Diffusion Processes, the Fokker-Planck and Langevin Equations*. English. Vol. 60. New York, NY: Springer New York, 2014. ISBN: 9781493913237.

[42] D. Reiter, M. Baelmans, and P. Börner. "The EIRENE and B2-EIRENE Codes". English. In: *Fusion science and technology* 47.2 (2005), pp. 172–186.

[43] Sheldon M. Ross. *Simulation*. English. 5. San Diego, CA: Academic Press, 2013;2012; ISBN: 9780124158252.

[44] Cédric Villani. *Optimal Transport: Old and New*. English. 1. Aufl. Vol. 338. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009;2008; ISBN: 3540710507;9783540710509.

[45] Dennis D. Wackerly, William Mendenhall, and Richard L. Scheaffer. *Mathematical statistics with applications*. English. 6. Pacific Grove, Calif: Duxbury, 2002. ISBN: 0534377416;9780534377410;

[46] Wikipedia contributors. *Nuclear binding energy — Wikipedia, The Free Encyclopedia*. `https://en.wikipedia.org/w/index.php?title=Nuclear_binding_energy&oldid=1000586727`. [Online; accessed 10-February-2021]. 2021.

[47] Samuel S. M. Wong. *Introductory nuclear physics*. English. Englewood Cliffs, N.J: Prentice Hall, 1990. ISBN: 0134911687;9780134911687;